

Г. Н. ФЕДОРОВА

ДЕРЕКТЕР ҚОРЫН ӘЗІРЛЕУ ЖӘНЕ БАСҚАРУ

ОҚУЛЫҚ

*«Білім беруді дамытудың федералды институты» Федералды мемлекеттік қазынашылық институты «Компьютерлік жүйелердегі бағдарламалау» мамандығы бойынша орта кәсіптік білім беру бағдарламаларын іске асыратын білім беру мекемелерінің оқу үрдісінде қолдануға арналған оқулық ретінде
ҰСЫНҒАН*

Пікірдің тіркеу нөмірі 122 09 сәуір 2015 ж.

ФМММ «БДФИ»



Мәскеу
«Академия» баспа орталығы
2015

ӘОЖ 004(075.32)
КБЖ 32.973-018.2я723
Ф333

Бұл кітап Қазақстан Республикасының Білім және ғылым министрлігі және «Кәсіпкер» холдингі» КЕАҚ арасында жасалған шартқа сәйкес «ТЖКБ жүйесі үшін шетел әдебиетін сатып алуды және аударуды ұйымдастыру жөніндегі қызметтер» мемлекеттік тапсырмасын орындау аясында қазақ тіліне аударылды. Аталған кітаптың орыс тіліндегі нұсқасы Ресей Федерациясының білім беру үдерісіне қойылатын талаптардың ескерілуімен жасалды.

Қазақстан Республикасының техникалық және кәсіптік білім беру жүйесіндегі білім беру ұйымдарының осы жағдайды ескеруі және оқу үдерісінде мазмұнды бөлімді (технология, материалдар және қажетті ақпарат) қолдануы қажет.

Аударманы «Delta Consulting Group» ЖШС жүзеге асырды, заңды мекенжайы: Астана қ., Иманов көш., 19, «Алма-Ата» БО, 809С, телефоны: 8 (7172) 78 79 29, эл. поштасы: info@dcg.kz

Пікір беруші —
Уфа әуе техникумының оқытушысы ФГБОУ ВПО «УГАТУ»,
физика математика ғылымдарының кандидаты *Э.Р.Тимашева*

Федорова Г.Н.

Ф333 Деректер қорын әзірлеу және басқару: орта кәсіби білім беру мекемесінің студенттеріне арналған оқулық. — М. : «Академия», баспа орталығы 2015. — 320 бет.

ISBN 978-601-333-333-5 (каз.)

ISBN 978-5-4468-1584-5 (рус.)

Оқу құралы «Компьютерлік жүйелердегі бағдарламалау» мамандығы бойынша мамандандырылған орта кәсіптік білім берудің Федералдық мемлекеттік білім беру стандарты бойынша құрылды. ПМ. 02 «Деректер базаларын әзірлеу және басқару».

Деректерді логикалық ұйымдастырудың типтік үлгілері, деректер қорын жобалау механизмдері, деректерді қорғаудың негізгі әдістері мен құралдары, SQL тілінің дерекқорлармен жұмыс істеу мүмкіндіктері қарастырылады. Деректердің тұтастығын қамтамасыз ету, компьютерлік желілерде деректердің алмасу қағидаттары мен технологиялары қарастырылған.

Орта кәсіптік білім беру мекемелерінің студенттеріне арналған..

ӘОЖ 004(075.32)
КБЖ 32.973-018.2я723

© Федорова Г.Н., 2015
© «Академия» білім беру баспа орталығы, 2015
© Ресімдеу. «Академия» баспа орталығы, 2015

ISBN 978-601-333-333-5 (каз.)

ISBN978-5-4468-1584-5 (рус.)

Құрметті оқырман!

Бұл оқулық «Компьютерлік жүйелердегі бағдарламалау» мамандығы бойынша оқу-әдістемелік кешендердің бір бөлігі болып табылады.

Оқулық кәсіби «Деректер базасын дамыту мен басқару» модульін зерттеуге арналған.

Жаңа буын оқу-әдістемелік жинағы жалпы және кәсіби пәндер және кәсіби модульдерін зерттеуді қамтамасыз ету үшін дәстүрлі және инновациялық білім беру материалдарын қамтиды. Әрбір жинақта жұмыс берушінің талаптарын ескере отырып, жалпы және кәсіби құзыреттілікті меңгеруге қажетті оқулықтар мен оқу құралдары, оқыту және мониторинг құралдары бар.

Оқу басылымдары электронды білім беру ресурстарымен толықтырылады. Электрондық ресурстарда интерактивті жаттығулар мен тренажерлар, мультимедиялық нысандар, интернетте қосымша материалдар мен ресурстарға сілтемелері бар теориялық және практикалық модульдер бар. Оларға терминологиялық сөздік және оқу үрдісінің негізгі параметрлері белгіленетін электронды журнал кіреді: жұмыс уақыты, бақылау және практикалық тапсырмалардың нәтижесі. Электронды ресурстар оқу үдерісіне оңай енеді және әртүрлі оқу бағдарламаларына бейімделуі мүмкін.

Ұсынылған оқулықтың мақсаты - студенттердің деректер базасының модельдері мен құрылымы туралы жүйеленген білімдерін қалыптастыру, дереккорды дамытудың негізгі принциптері мен технологиялары, сондай-ақ компьютерлік желілерде деректермен алмасудың заманауи технологиялары.

6 тараудан тұратын «Инфокоммуникациялық жүйелер мен желілер» оқулықтарының *бірінші бөлімі* компьютерлік желілер архитектурасының негізгі аспектілерін, желілік өзара әрекеттесу технологиясын - сымды, сымсыз жергілікті және ғаламдық желілерді, олардың техникалық және бағдарламалық қамтамасыз етуін жүйелі түрде сипаттайды, желі қызметтері мен қызмет көрсету жүйесін сипаттайды.

Бірінші тарауда компьютерлік желілерді ұйымдастырудың жалпы мәселелерімен байланысты, негізгі түсініктер мен элементтерді, желілерді жіктеуді, желілік шешімдердің топологиясын көрсетеді. Кабельді байланыс каналдарының техникалық сипаттамалары келтірілген. Компьютерлік желілерге қойылатын негізгі талаптар ұсынылады.

Екінші тарауда компьютерлік желілерде деректерді беру және алмасу үшін технологиялар ашылады. Компьютерлік желілердің хаттамалары мен стандарттары сипатталады, желідегі түйіндерді шешу қағидаттары түсіндіріледі.

Үшінші тарау жергілікті желілердің негізгі технологияларын сипаттайды, барлық негізгі технологияларды - дәстүрлі Ethernet, TokenRing, FDDI және FastEthernet жоғары жылдамдықпен жұмыс жасайтын, GigabitEthernet көрсетеді.

Төртінші тарауда компьютерлік желілердің техникалық құралдары - құрылымдалған кабельдік жүйе, желілік адаптерлер, хабтар, трансиверлер, желілік көпірлер, желілік қосқыштар, маршрутизаторлар және желі шлюздері туралы сипатталады.

Бесінші тарау сымсыз технологияларға арналған, ол желілік жабдықтар мен стандарттарды сипаттайды, бұл кабель өткізгіштерін пайдаланбай компьютерлік желілерді құруға мүмкіндік береді.

Алтыншы тарауда желінің операциялық жүйелерінің құрылысы мен ерекшеліктері қарастырылады, желіде бөлінген сервердің тұжырымдамасын анықтайды. «Клиент-сервер» көп деңгейлі архитектурасының бағдарламасы мен техникалық іске асырылуы ашылады. «Желілік қызмет», «Желілік қызмет», «Желілік бағдарламалық қамтамасыз ету» түсініктері сипатталады.

Екінші бөлімде «Деректер қорларын әзірлеу және қорғау технологиясы, деректер үлгілері қарастырылады, дерекқорды жобалаудың теориялық негіздері ашылып, олардың тұтастығын қамтамасыз ету және деректермен жұмыс істеу үшін SQL тілін қолдану мәселелері талқыланады. Бөлім сегіз бөлімнен тұрады.

Жетінші тарауда дерекқор теориясының, деректер қоймасының, білім негіздерінің негізгі ережелері келтірілген. Мұнда дерекқорлар теориясы мен ақпараттық жүйелердің негізгі ұғымдарының анықтамалары берілген. Деректерді басқару жүйелерінің сипаттамалары берілген, деректер базасын басқару жүйесінің қызметі (СУБД) сипатталған. Деректерді өңдеу технологиясы дамуының негізгі кезеңдері және оларды одан әрі пайдалану перспективалары қарастырылады. Ақпараттың формалды ұсынылған мәселелері талқыланады, көп деңгейлі деректер үлгісінің тұжырымдамасы енгізіледі, деректердің физикалық және логикалық тәуелсіздігінің қағидалары қарастырылады.

Сегізінші тарауда деректер үлгісі мен деректер құрылымы туралы түсініктер келтірілген. Дерекқор жүйелерінде пайдаланылатын үлгілерді жіктеу берілген. Деректерді ұсыну үлгілерінің жалпы сипаттамалары берілген. СУБД-да қолда бар деректер үлгілерінің теориялық негіздері келтірілген. Деректер базасын басқару жүйелерінде пайдаланылатын теориялық-графикалық деректер үлгілері, сондай-ақ заманауи, перспективалық деректер үлгілері - көп өлшемді және нысанды бағдарланған деректер үлгілері сипатталған. Реляциялық үлгінің негізгі компоненттерінің анықтамалары берілген.

Тоғызыншы тарауда кеңінен таратылатын реляциялық деректер үлгісінің негізгі түсініктері мен компоненттері қарастырылады. Көптеген мысалдарды пайдалана отырып, реляциялық алгебраның негіздері түсіндіріледі. Кестелердің индекстелуінің, байланыстырудың механизмдері ашылып, анықтамалық тұтастық тұжырымдамасы сипатталған. Реляциялық дерекқорда тұтастығын қолдау принциптері түсіндіріледі. Реляциялық деректер үлгісінің артықшылығы мен кемшіліктері көрсетіледі.

Оныншы тарауда тұжырымдамалық, логикалық және физикалық деректер үлгісін құру принциптері түсіндіріледі. Жобалаудың негізгі сатыларының толық сипаттамасы - тақырыптық аумақты талдаудан деректер базасын нақты іске асыру болып табылады. Тұжырымдамалық және логикалық үлгілерді құру мысалдары келтірілген.

Нормализация процесі - кестенің ыдырау әдісімен деректер базасының құрылымын жасау. Нормализация принциптеріне негізделген дерекқорды жобалаудың мысалы келтірілген. Сондай-ақ автоматтандырылған дерекқорды жобалау құралдарын пайдалану қажеттілігі де бар. CASE-технологиясының негізгі ұғымдарының анықтамалары келтірілген, CASE-технологияларының әртүрлі сыныптарының сипаттамалары сипатталған.

Он бірінші тарауда әртүрлі дерекқор сәулетінің құндылықтары мен кемшіліктері қарастырылады, көп деңгейлі «клиент-сервер» сәулетінің үлгісін құру және пайдалану ерекшеліктері сипатталады. Анықтамалар дерекқор серверінің негізгі нысандарына беріледі.

Он екінші тарауда құрылымдық SQL сұрау тілі арқылы дерекқорды басқарудың негізгі әдістері талқыланады. Деректердің тұтастығын қамтамасыз ету мәселесіне көп көңіл бөлінеді - кесте деңгейінде шектеулерді жариялаудан бастап, дерекқорды тұтас күйде сақтау үшін триггерлерді дамытуға ерекше көңіл бөлінеді. Мәмілелер жасау сөз болады. SQL бүгінгі күні барлық негізгі деректер базасын сатушылар қолдайтын дерекқор тілінің жалғыз танылған стандарты болып табылады. SQL білімі кәсіби дерекқорлар мен қосымшаларды әзірлеушілер үшін міндетті болып табылады.

Он үшінші тарауда дерекқордың тұтастығын қамтамасыз ету мәселелері көрсетілген. Ол деректерге параллель қол жеткізумен туындайтын проблемаларды және осы мәселелерді шешу жолдарын айқындайды. Мәміле механизмі және құлыптауды басқару арқылы деректер тұтастығын қамтамасыз ету жолдары ашылады. Тіркеу өзгерістерінің тәртібі сипатталған.

Он төртінші тарауда заманауи дерекқорды басқару жүйелеріне қойылатын негізгі талаптар сипатталған және ең танымал СУБД-нің шағын шолуы ұсынылған.

Барлық материалда оның терең дамуына ықпал ететін көптеген мысалдар бар.

Оқулықтың әрбір тарауында студенттің зерттелген материалды біріктіруіне мүмкіндік беретін өздік жұмысқа арналған бақылау сұрақтары қамтылған.

Оқу құралы 230115 «Компьютерлік жүйелерде бағдарламалау» мамандығы бойынша «Деректер қорларын әзірлеу және басқару» кәсіби модулін қолданып, тәжірибелік қызмет барысында автор жинақтаған материалдарға негізделген Федералды білім беру стандарты бойынша толық дайындалды. Оқулықтың бөлімдері «Деректер қорларын әзірлеу және басқару» кәсіби модулінің МДО талаптарына сәйкес келеді.

Оқу құралын мұғалімдер мен аталған мамандық бойынша оқитын студенттерге тек осы кәсіби модульдің шеңберінде ғана емес, сондай-ақ компьютерлік желілерге қатысты басқа модульдер мен пәндерде, сондай-ақ деректер қорын жобалау және пайдалануда қолданыла алады.

Ұсынылған материалдар, сондай-ақ кәсіпорындарда деректер қорын құру және басқару, желілік басқару, компьютерлік желілерді техникалық және бағдарламалық қамтамасыз ету саласында мамандардың кең ауқымды қолданысы үшін пайдалы болуы мүмкін.

Оқулықтың мазмұны оқырманның компьютерлік ғылымдар мен ақпараттық технологиялар курстарын игергенін және операциялық жүйелер мен ақпараттандырудың техникалық құралдарының жалпы ұғымына ие екендігін болжайды.

I

БӨЛІМ

ИНФОКОММУНИКАЦИЯЛЫҚ ЖЕЛІ МЕН ЖҮЙЕЛЕР

- 1.Тарау. Негізгі ұғымдар, компьютерлік желі құрылымы мен элементтері.
- 2.Тарау. Деректерді беру мен компьютерлік желілермен алмасу технологиялары.
3. Тарау. Жергілікті желілердің негізгі технологиялары.
4. Тарау. Компьютерлік желінің техникалық құралдары.
5. Тарау. Сымсыз технологиялар.
6. Тарау. Желілік бағдарламалық қамтамасыз ету. Қызмет пен қызмет көрсету

НЕГІЗГІ ҰҒЫМДАР КОМПЬЮТЕРЛІК ЖЕЛІ ҚҰРЫЛЫМЫ МЕН ЭЛЕМЕНТТЕРІ

1.1 НЕГІЗГІ ҰҒЫМДАР МЕН АНЫҚТАМАЛАР. КОМПЬЮТЕРЛІК ЖЕЛІЛЕРДІҢ ЖІКТЕЛУІ

Сөздің кең мағынасында компьютер желісі деректер алмасу үшін бір-бірімен байланысқан компьютерлер жиынтығын білдіреді. Компьютерлер арасында ақпарат беру сізге жеке компьютерлердің бірлескен жұмысын ұйымдастыруға, бірнеше компьютерлердің көмегімен бір тапсырманы шешуге, ортақ ресурстарды бөлуге және көптеген басқа мәселелерді шешуге мүмкіндік береді. Компьютерлік желілер мен телекоммуникациялық технологиялар - бұл компьютерлік техниканы, бағдарламалық қамтамасыз етуді, коммуникацияларды және коммуникацияны дамыту болып табылады. Ақпаратты өндеудің компьютерлік желілері мен желілік технологиялары заманауи ақпараттық жүйелерді құру үшін негіз болды.

Компьютерлік желі (network) — бұл компьютерлер арасында ақпарат алмасуды қамтамасыз ететін телекоммуникациялық жабдықтардың жинағы.

Осылайша, компьютерлік желілердің негізгі мақсаты пайдаланушыларға ақпаратқа (деректер базасына, құжаттарға және т.б.) және аппараттық ресурстарға (қатты дискілер, принтерлер, CD-ROM дискілері, модемдер және т.б.) ортақ қолжетімділікті қамтамасыз ету болып табылады.

Желі түйіні (node) — компьютер желісінің бөлігі ретінде басқа құрылғыларға қосылған құрылғы. Түйіндер компьютерлерге, ұялы телефондарға, PDA-лерге және жол жоспарлағышы, коммутатор немесе хаб сияқты арнайы желілік құрылғылар болуы мүмкін

Коммутация — Бұл желіде деректерді беруді ұйымдастырудың технологиясы. Байланыс арнасындағы деректерді беру жылдамдығы уақыт бірлігіне жіберілетін ақпараттың бірліктерінің санымен өлшенеді. Өлшеу бірліктері - секундына биттер, секундына килобит және т.б.

Белгілі бір уақыт аралығында компьютер желісі арқылы берілетін ақпарат көлемі желі трафигі деп аталады. Трафиктің көлемі пакеттерде, биттерде, байттарда және олардың туындыларында өлшенеді: килобайт (Кб), мегабайт (Мб) және т.б..

Трафик жіктеледі:

- Кіріс ретінде (желіге түсетін ақпарат);
- Шығыс ретінде (желіден шығатын ақпарат);
- Ішкі (нақты бір желі шегінде, көбінесе жергілікті желі болып табылады);
- Сытқы (нақты бір желі шегінен тыс, көбінесе интернет трафик болып табылады).

Байланыс арнасы(channel, dataline) — ақпарат көзі мен қабылдағыштан мәліметтерді беру үшін техникалық құралдардың және сигнал тарату ортасының жүйесі.

Желілік коммутатор(switch) — Бір немесе бірнеше желі сегменттері ішінде компьютерлік желінің бірнеше түйінін жалғауға арналған құрылғы. Желілік қабаттағы, маршрутизаторларға негізделген бірнеше желілерді қосуға арналған.

Маршрутизатор, немесе *роутер*(router) — кем дегенде бір желілік интерфейске ие арнайы желілік компьютер және желі топологиясы және әкімші белгілеген белгілі бір ережелер туралы ақпаратты негізге ала отырып, әртүрлі сәулеттің әр түрлі желілерін қосатын әртүрлі желілік сегменттер арасында деректер пакеттерін жібереді.

Желілік концентратор, немесе *хаб*(hub) — кабельдік инфрақұрылымды пайдалана отырып, компьютерлерді желіге біріктіруге арналған құрылғы.

Қазіргі уақытта компьютерлік және коммуникациялық технологиялар, желілік архитектуралар қолданылатын әр түрлі мақсаттардан ерекшеленетін көптеген түрлі желілер бар.

Деректерді беру технологиясы бойынша желілерді екі түрге бөлуге болады: біреуден көп байланыс (хабар тарату) және нүктеден нүктеге қосылу. Бірінші жағдайда бір компьютермен жіберілген хабарды барлық желілік компьютерлер алады; нүкте-нүкте байланысында жеке ақпарат арнасы компьютерлермен ақпарат алмасу үшін пайдаланылады.

Абоненттер арасындағы деректермен алмасу қағидасына негізделе отырып, схемаларды ауыстырып-қосуға, хабарларды ауыстыруға және пакеттік коммутацияға негізделген желілер бар.

Коммутациялық арналары бар желі — олардың әрқайсысының ақпараттық өзара іс-қимыл сеансы кезінде жұбының физикалық қосылуын беретін желінің түрі.

Сонымен қатар, байланыс арнасы абоненттер арасындағы байланыс сеансы өткізілетін уақытта басқа абоненттер үшін қол жетімді болмайды. Схемадағы коммутацияланған желілерде деректерді беру уақытына абоненттерді қосатын байланыс желілерінің «тізбегін» сақтау арқылы қалыптасатын пайдаланушы сұрауы бойынша үздіксіз ақпараттық арна жасалады. Толық ұзақтығы бойы арна деректерді бірдей жылдамдықпен жібереді. Бұл кесте арқылы қосылатын желі арқылы кідірістерге (дауыс, бейне) сезімталдықты сапалы түрде беру мүмкін дегенді білдіреді. Дегенмен, физикалық арнаның өткізу қабілетін динамикалық түрде таратуға қабілетсіздігі схема-коммутацияланған желінің іргелі жетіспеушілігі болып табылады, бұл оны компьютерлік трафикті икемдеуге тиімсіз етеді.

Хабарларды ауыстырған кезде, ақпарат абоненттер арасында логикалық аяқталған деректер порты түрінде беріледі. Кестедегі коммутацияланған желі сияқты жұмыс істейді, бірақ байланыс арналары бүкіл байланыс кезеңінде физикалық түрде жұмыс істемейді, бірақ хабарларды беру кезеңі үшін ғана. Қазіргі уақытта хабар алмасу желілері іс жүзінде пайдаланылмайды. Олардың негізінде пакеттік коммутациялық желілер әзірленді және қолданылды.

Пакеттерді ауыстырған кезде берілетін деректер кіші өлшемдегі жеке блоктарға бөлінеді - пакеттердің коммутациялық тораптарында аралық сақтауы бар әртүрлі бағыттар бойынша желінің бір нүктесінен екіншісіне ауыстыруға болатын пакеттер болып табылады.

Пакет мәнің мекен-жайын анықтайтын үстіңгі деректемемен жабдықталады, сондықтан ол деректердің қалған бөліктеріне қарамастан коммутатор арқылы өңделуі мүмкін. Бумалар әр түрлі бағыттар бойынша өз мақсаттарына жетуі мүмкін. Пакет қабылдау түйініне келгеннен кейін бастапқы деректер хабарға жиналады. Пакеттер желісінде бір-біріне тәуелсіз түрде беріледі. Мұндай желілерде көптеген түйіндер деректерді бір физикалық сілтеме арқылы алмастыра алады. Пакеттер кейде датраммалар (datagram) деп аталады, ал жеке пакеттік коммутация режимі - деректер көзі режимі болып табылады.

Пакеттік коммутация пульсирленген трафикті беруде желінің өнімділігінен асып түседі, өйткені тәуелсіз ағындардың үлкен саны қызмет көрсетілсе, олардың қызмет мерзімі әрқашан сәйкес келмейді.

Пакеттер ресурстарға алдын-ала резервсіз, олардың көзі пайда болған мөлшерде желіге келеді. Желі толтырылған кезде, абоненттер желіден шығарылмайды, желі барлық абоненттер үшін тасымалдау жылдамдығын азайтады. Абонент өзінің арнасын толығымен пайдаланбайды, шын мәнінде, басқа абоненттерге желінің өткізу қабілетін береді. Дегенмен, коммутацияның бұл тәсілі де теріс жағына ие: тарату кешігуі кездейсоқ, сондықтан нақты уақыттағы трафикті беруде қиындықтар бар. Пакеттік коммутацияланған желі күрделі құрылғыға ие және өткізу жолағының бөлігі техникалық деректерге жұмсалады.

Аймақтық негізде, яғни желімен қамтылатын аумақтың көлемі бойынша, жергілікті, қалалық және ғаламдық желілер ерекшеленеді. Жергілікті және ғаламдық желілердің технологиялары арасындағы айырмашылықтар тұрақты конвергенцияға қарамастан өте маңызды болып табылады.

Жергілікті желі (LocalAreaNetworks— LAN) әдетте бір ғимаратта (ғимараттар кешенінде) орналасқан жеке компьютерлер мен перифериялық құрылғыларды қосады. Желілердің ең қарапайым түрі болып табылатын жергілікті желілер дербес компьютерлер тобын біріктіреді немесе оларды желілік сервер ретінде әрекет ететін қуатты компьютерге қосады. Жергілікті желілер жеке пайдаланушыларға бір-бірімен оңай және жылдам өзара әрекеттесуге мүмкіндік береді. Жергілікті желідегі барлық дербес компьютерлер желілік серверде сақталған арнайы қолданбаларды пайдалана алады және ортақ құрылғылармен жұмыс істей алады: принтерлер, факстар және басқа перифериялық құрылғылар. Жергілікті желідегі әрбір дербес компьютер *жұмыс станциясы* немесе *желі түйіні* деп аталады.

Ғаламдық желі (WideAreaNetworks— WAN) олар әртүрлі қалалар мен елдерде орналасқан жүздеген және мыңдаған шақырымдық қашықтыққа бөлінген компьютерлерді біріктіреді. Дискретті деректерді үнемі көшіру үшін, жергілікті желілерге қарағанда, күрделі әдістер мен жабдықтар қолданылады.

Аймақтық, немесе қалалық желі (MetropolitanAreaNetworksMAN) жергілікті және ғаламдық желілер арасындағы аралық орынды иеленеді. Компьютерлерді қаланың немесе қаланың аумағында толығымен біріктіреді. MAN компаниясы жергілікті және ғаламдық желілердің артықшылықтарына ие: олар компьютерлер арасындағы үлкен қашықтықты (бірнеше ондаған километрге дейін) қамтуға қабілетті және сол уақытта олар деректерді берудің жоғары жылдамдығымен ерекшеленеді, бұл желі тіпті жергілікті желілердегі жылдамдығынан да асып түседі.

Қазіргі уақытта желілерді нақты жіктеу мүмкін емес, атап айтқанда, көптеген жергілікті желілер жаһандық желілерге қол жеткізе алады. Дегенмен, деректерді айырбастау мен қол жеткізу қағидалары желілердің әртүрлі типтерінде айтарлықтай ерекшеленеді.

Есептеу желілері компьютерлер иерархиясын ұйымдастыру қағидасы бойынша түбегейлі әртүрлі екі сыныпқа бөлінеді: біртұтас (бір деңгейлі) желілер және иерархиялық (көп деңгейлі).

Біртұтас желі әрқайсысының бірегей аты (компьютер атауы) және жүктеу уақытында оған кіру үшін құпия сөзі бар тең дәрежелі желі. Логин аты мен құпия сөзі дербес компьютердің иесі операциялық жүйені пайдаланып тағайындайды. Тең-теңімен байланыс желілері операциялық жүйелердің әртүрлі түрлерін қолдана отырып ұйымдастырылуы мүмкін.

Иерархиялық желіде кейбір қызметтермен басқа нысандарды қамтамасыз ететін бір немесе бірнеше арнайы компьютерлер (серверлер) бар. Серверлер түрлі пайдаланушылармен ортақ пайдаланылатын ақпаратты сақтайды. Серверде ақпаратқа қол жеткізетін компьютерлер *жұмыс станциялары* немесе *клиенттер* деп аталады.

Сервер— бұл ақпаратты сақтау, айырбастау және өңдеу мақсатында тұтынушыларға қызметтері мен ресурстарына желі арқылы кіруге мүмкіндік беретін компьютер. Басқаша айтқанда, сервер тұрақты ресурстар болып табылады. Сервер өзі иерархияның жоғары деңгейлі серверінің клиенті бола алады. Сондықтан иерархиялық желілер кейде бөлінген сервермен желілер деп аталады. Серверлер әдетте жоғары өнімді компьютерлер болып табылады, бірнеше параллель процессорлар, жоғары қуатты қатты дискілер, жоғары жылдамдықты желілік карталарды қамтиды.

Серверді пайдаланудың екі архитектурасы бар: «сервер файл» және «сервер клиент». «Файл-сервер» архитектурасында деректер мен бағдарламалар серверде сақталады және тапсырмаларды өңдеу және орындау клиенттік компьютерлерде орындалады, сервер ақпарат пен файлды сақтау мүмкіндігін ғана қамтамасыз етеді. «Клиент-сервер» архитектурасында серверде деректерді сақтаудан басқа, сервер бағдарламалық жасақтамасы іске қосылып, қолданбалы бағдарламаларды жүзеге асырады, деректерді өңдейді және т.с.с.

Хост(host— қонақтарды қарсы алушы, үй иесі), немесе *мүйін*— «клиент-сервер» форматты қызметін ұсынатын кез келген құрылғы

серверлік режимде кез келген интерфейстер арқылы және осы интерфейстерде анықталған. Нақты жағдайда хост кез келген компьютер, жергілікті немесе кең аймақтық желіге қосылған сервер ретінде түсінікті болуы мүмкін. Кейде желіде белгілі бір құрылғы айтылғанда, «түйін» термині де пайдаланылады (компьютер желісінің прототипі ұқсас, өйткені нақты желі, мысалы, балық аулау торы көптеген тораптармен байланысқан ағындардан тұрады).

Деректер беру ортасының түрі бойынша желілер сымды және сымсыз (радиоарналар, спутниктік арналар) болып бөлінеді. Екі жағдайда да ақпарат беру қағидаттары мен технологиялары одан әрі түсіндіріледі.

1.2. КОМПЬЮТЕРЛІК ЖЕЛІ ТОПОЛОГИЯСЫ

Бірнеше компьютерлер арасында байланыс орнату кезінде физикалық байланыстар немесе топология конфигурациясын таңдау мәселесі пайда болады. Желінің **топологиясы** - шындары желінің соңғы түйіндеріне және байланыс жабдығына сәйкес келетін графиктің конфигурациясы және олардың арасында байланыс арналарының шеттері болып табылады. Басқаша айтқанда, компьютерлік желілердің топологиясы - бұл компьютерлер мен түрлі желілік құрылғылардың физикалық байланыстарының конфигурациясы болып табылады.

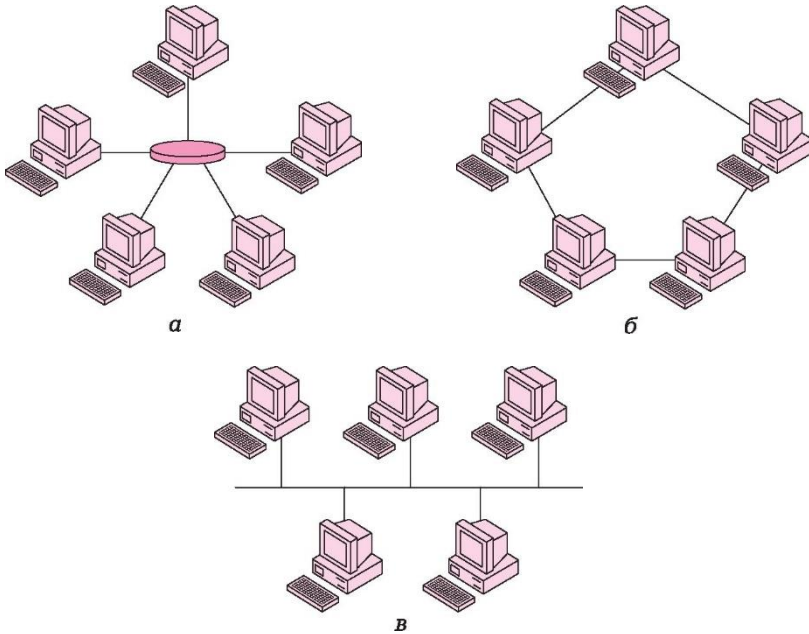
Желінің көптеген сипаттамалары байланыс топологиясын таңдауға байланысты: жеке арналардың жүктелуін теңестіруге, желіні оңай кеңейтуге, байланыс желілерінің жалпы ұзындығын қысқарту арқылы шығындарды азайтуға және т.б байланысты.

Топологияның жиынтығы шартты түрде үш негізгі топқа бөлінеді: «жұлдыз», «сакина» және «жалпы шина» (1.1 қараңыз).

«*Жұлдыз*» топологиясы бұл желі конфигурациясы, әрбір желілік компьютер концентраторға жеке байланыс желісі арқылы қосылған кезде жұмыс жасайды (1.1 а сурет). Концентратор - бір компьютерден басқа компьютерге немесе барлық компьютерлерге ақпаратты бір уақытта жіберуге қабілетті құрылғы.

Сақиналық конфигурация компьютерлерді жабық сақинаға қосады (1.1 б сурет). Бұл жағдайда, деректер бір компьютерден екіншісіне олардың тағайындалған жерін тапқанша ауыстырылады.

«*Жалпы шина*» топологиясы барлық компьютерлердің бір желіге қосылуын көрсетеді, мысалы жоғары жиілікті кабель (1.1 в суреті)



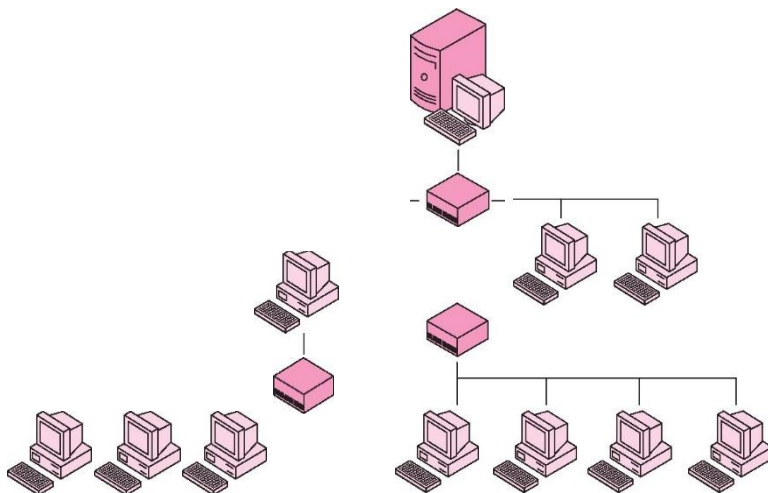
1.1. Сурет Желі типологиялары:

a — «жұлдыз»; *б* — «сақина»; *в* — «жалпы шина»

Желіні шинаға салу кезінде, әрбір компьютер терминатордың соңында орнатылған жалпы кабельге қосылады. Дабыл желі арқылы барлық компьютерлер арқылы өтіп, соңғы терминатордан көрсетіледі.

Жоғарыда біз негізгі топологияларды келтірдік. Бірақ олармен қатар, аралас (аралас) топологиялар іс жүзінде қолданылады. Олар ағаш құрылымын қамтиды (1.2 сурет). «Ағаш» типті біріктірілген топологиясы бірнеше желілерде біріктірілген, түрлі типтегі топологиясы бар ірі желілерде қолданылады.

Желілік құрылымдар «ағаштан» типті құрылымды пайдаланып, негізгі желілік құрылымдарды таза түрде қолдануға болмайды. Көптеген жұмыс станцияларын қосу үшін көпірлер, желілік күшейткіштер, қосқыштар және т.б. қолданылады. Қазіргі уақытта ағаш жергілікті және ғаламдық желілерде байланыс топологиясының ең көп тараған түрі болып табылады.



1.2 сурет. «Ағаш» типті топологиясы

1.3. КАБЕЛЬДІ БАЙЛАНЫС ЖЕЛІЛЕРІ

Кабельдік байланыс желілері бірнеше оқшаулау сызықтарымен байланысқан өткізгіштер негізінде салынған және келесі параметрлермен сипатталады: өткізу қабілеті, кедергіден кабельдік иммунитеттің дәрежесі, әлсіреу және т.б. Ақпарат беру арналарының физикалық сипаттамалары келесі тарауда толығырақ сипатталады. Сонымен қатар компьютерлік желілерде қолданылатын кабельдердің түрлерін қарастырайық (1.3-сурет):

- коаксиалды кабель;
- «бұралған жұп» типті кабель;
- Талшықты оптикалық кабель.

Коаксиалды кабель. Коаксиалды кабель - диэлектрлік және металл қасыққа салынған мыс қабырғасы (тамыры), ол сондай-ақ жоғарғы бөлігінде оқшаулағыш материалмен жабылады (1.3 а сурет).

Металл қабығы ішкі өткізгішті қорғайды, оны сыртқы электромагниттік өрістерден туындаған кедергілерден қорғайды. Коаксиалды кабель, әдетте, «жалпы шина» топологиясы бар желілерде қолданылады.

«Жұқа» және «қалың» коаксиалды кабельдер бар. «Жұқа» коаксиалды кабель 50 Ом толқындық қарсылыққа ие болады.

Ішкі мыс сымының диаметрі 0,89 мм, сыртқы диаметрі шамамен 5 мм. «Қалың» коаксиалды кабель ішкі өткізгіштің диаметрі 2,17 мм болатын толқындық қарсылыққа ие және сыртқы диаметрі шамамен 10 мм тең болады.

«Бұралған жіп» типті кабель. Бұралған жұп кабель - пластикалық қаптамамен жабылған (бірлік ұзындығына қарай аз саны бар) бір-біріне немесе бірнеше жұп оқшауланған өткізгіштеріне тиесілі (1.3 б сурет).

Бұралған жұп негізіндегі кабельдер *экрандалған* (Shielded Twisted Pair—STP) және *экрандалмаған* (Unshielded Twisted Pair—UTP) болуы мүмкін. Қорғалған «бұралған жұп» шудың иммунитетін арттыратын және «тыңдаудан» қорғауды жақсартатын металдың кілтіне орналастырылған. Қорғалмаған «бұралған жұптың» сыртқы электр магниттік әсерінен әлсіз қорғаныс бар.

Бұрылған жұп кабельдердің бірнеше санаты бар (1.1-кесте), олар жиілікте ерекшеленеді.

Коаксиалды кабельмен салыстырғанда «бұралған жұпты» пайдаланудың артықшылықтары:

- толық дуплексті режимде жұмыс істеу мүмкіндігі, яғни, ақпаратты беру және алу;
- «бұралған жұп» кабельдің төмен құны;
- Кабельде ақаулық болған жағдайда желінің жоғары сенімділігі;
- кабельдің ең аз рұқсат етілген иілу радиусы аз;
- жоғары шудың иммунитеті.

Талшықты оптикалық кабель. Талшықты оптикалық кабельдің құрылымы өте қарапайым және коаксиалды кабельдің құрылымына ұқсас болып келеді, бірақ орталық мыс сымның орнына жұқа (диаметрі шамамен 1 ... 10 мкм) шыны талшығы пайдаланылады және ішкі оқшаулау — шыны және пластикалық қаптағыштары, шынылы талшықтың сыртына шығып кетуінен сақтайды. (1.3, в сурет).



1.3 сурет. Кабель түрлері:

а — коаксиалды; б — «бұралған жұп»; в — талшықты оптикалық

1.1 кесте . «Бұралған жұп» кабелінің категориялары

Категориясы	Жиілік сызығы, МГц	Қолданылу
1	<1	Телефон және ескі модем желілері
2	1	Ескі терминалдар (IBM3270 сияқты)
3	16	10BASE-T, 100BASE-T4 Ethernet
4	20	TokenRing, қазіргі уақытта қолданылмайды
5	100	100BASE-TX Ethernet
5e	100	100BASE-TX Ethernet— Жетілдірілген 5санаты (тазартылған / жетілдірілген спецификациялар). 4 жұпты пайдаланған кезде 2 жұпты және 1000 Мбит / с дейін деректерді беру жылдамдығы 100 Мбит / с дейін. Жергілікті желілердің басым көпшілігі 5e категориясына бағытталған.
6	250	Fast Ethernet, Gigabit Ethernet (10GBASE-T Ethernet)
6a	500	Gigabit Ethernet (10GBASE-T Ethernet)
7	600	Gigabit Ethernet (10GBASE-T Ethernet)
7a	до 1200	Gigabit Ethernet (40GbE, 100GbE)

Шыны конверттің сыну индексі орталық талшықтардан әлдеқайда төмен, өйткені бұл жағдайда, сынудың түрлі көрсеткіштерінің екі материалдар шекарасынан жарық деп аталатын жалпы ішкі көрініс режимінде мүдделі болады. Сыртқы электромагниттік кедергіні қорғайды (мысалы, кабельді кейде бронды деп аталады, ол бір немесе бірнеше талшықты-оптикалық кабельдер бойынша біріктірілуі мүмкін) талап етілмейді, бірақ кейде ол ортадан механикалық қорғау үшін пайдаланылады әдетте кабелінің металл қаптауы ретінде пайдаланылмайды.

Жарықтың таралу сипатына байланысты талшықты-оптикалық кабельдің келесі түрлері қолданылады:

- бір үлгідегі талшықтары ;

- сыну көрсеткішіндегі қадам өзгерісімен көпүлгідегі талшық;
- сыну көрсеткішіне біртіндеп өзгеруімен көпүлгідегі талшық.

Сәуле үлгісі — бұл ядродағы сәулені көрсету бұрышы. Бір қалыпты талшық жарықтың толқын ұзындығымен (8 ... 9 мкм) өте төмен диаметрлі өзегін пайдалануды білдіреді. Сондықтан, мұндай кабельде тек бір ғана режим болуы мүмкін. Мысалы, 9/125 мкм бір режимдік кабель талшығы ядро диаметрі 9 мкм және шыныдан жасалған конверттің диаметрі 125 мкм болатын кабельге сәйкес келеді.

Бір мәртелік кабель өте қымбат, өйткені мұндай шағын диаметрлі шыны талшық өндірісі күрделі технологиялық процесс болып табылады. Бірақ оның сипаттамалары арзан режимі кабельдерімен салыстырғанда әлдеқайда жоғары, бұл оны ұзақ қашықтықта деректерді беру үшін пайдалануға мүмкіндік береді.

Мульти үлгідегі кабельдерде кең ядролар пайдаланылады (мысалы, 50/125 мкм, 62,5 / 125 мкм кабельдер) және олар бір режимдік кабельдерге қарағанда арзанырақ болып келеді. Үлкен диаметрі бар осындай кабельде жарық әртүрлі траектория бойынша таралуы мүмкін, сондықтан әртүрлі бұрыштарда көрініс табады, сондықтан бірден бірнеше сәулелік режимі болады. Көптеген режимдер импульстің таралуына, сәулелердің араласуына және соның салдарынан кабель сипаттамаларының нашарлауына әкеледі. Сондықтан, мультимодтық кабельдер негізінен қысқа қашықтықта (2000 м дейін) жылдамдықпен 1 Гбит / с-тан аспайтын деректерді беру үшін пайдаланылады.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Компьютерлік желі дегеніміз не?
2. Жергілікті және ғаламдық желілер арасындағы айырмашылық қандай?
3. Клиент-сервер мен файлдық сервердің сәулеті арасындағы негізгі айырмашылық қандай?
4. Компьютерлік желілерді құру үшін пайдаланылатын кабельдердің негізгі түрлері қандай?
5. Талшықты-оптикалық кабельдердің негізгі түрлерін көрсетіңіз.
6. Желі өнімділігінің негізгі сипаттамалары қандай?
7. Ауқымдылық дегеніміз не?
8. Желінің сенімділігін сипаттау.
9. Қандай құрылғы хост деп аталады?
10. Толық және толық емес желілер арасындағы айырмашылық қандай?

11. Пакеттік коммутацияланған технологияны қолданудың артықшылықтары қандай?
12. Интеграция дегеніміз не?
13. Коаксиалды кабельге қарағанда бұралған жұптың артықшылығы қандай?
14. Каналдың мүмкіндіктерін қандай факторлар анықтайды?
15. Талшықты-оптикалық сызықтардың жылдамдығына қандай шектеулер бар? Олар қалай анықталады?
16. Сымды байланыс каналдарының негізгі сипаттамаларын сипаттаңыз.
17. Кабельмен салыстырғанда оптикалық байланыс жүйелерінің артықшылықтары мен кемшіліктері қандай?

КОМПЬЮТЕРЛІК ЖЕЛІДЕГІ ДЕРЕКТЕРМЕН АЛМАСУ МЕН ТЕХНОЛОГИЯЛЫҚ АЛМАСУЛАР

2.1. ДЕРЕКТЕРДІҢ ФИЗИКАЛЫҚ АЛМАСУЫ

Деректердің алмасу ортасы — бұл компьютерлер арасында ақпарат алмасатын жолдар немесе байланыс арналары болып табылады. Ақпаратты беру каналдың физикалық сипатына сәйкес келетін дабылдармен жүзеге асырылады. Мысалы, коаксиалды кабель арқылы немесе «бұралған жұппен» деректерді беру электрлік дабылдар арқылы, талшықты-оптикалық кабель арқылы жарықты, ал сымсыз байланыстар арқылы - радиотолқындар жүзеге асырылады. Сондықтан ақпарат беру үшін оны бір немесе бірнеше басқа сигнал түріне, яғни кодтауға ауыстыру қажет.

Компьютерлік технологияда деректерді көрсету үшін екілік код пайдаланылады. Компьютердің ішіндегі мәліметтер мен нөлдер дискретті электр сигналдарына сәйкес келеді. Осы немесе кодтау жүйесін таңдау ақпаратты берудің физикалық ортасының сипаттамаларына, ақпаратты кодтау және кодтау үшін қолданылатын желілік жабдықтың құнын қамтиды. Түрлі кодтау әдістерін қолдану ақпараттың берілуінің сенімділігі мен жылдамдығына айтарлықтай әсер етуі мүмкін. Әр түрлі кодтау жүйелері бойынша бірдей физикалық тарату ортасында тасымалдау жылдамдығы айтарлықтай өзгеруі мүмкін.

Екілік сандарды кодтаудың әртүрлі жолдары бар, мысалы, сандарды көрсету үшін түрлі полярлық импульстар пайдаланылғанда, бір кернеу деңгейіне сәйкес келетін ықтимал жол, және басқа нөлге тең немесе импульстік режимдер болып табылады.

Осындай тәсілдер деректерді шифрлауға және олардың байланыс желілері бойынша екі компьютерге ауыстыру кезінде қолданылады.

Алайда, бұл сілтемелер бірінші орында әлдеқайда көп дәрежеде, сондай-ақ олар кеңістікте екенін іс жүзінде компьютер ішіндегі желілерін олардың сипаттамалары әр түрлі, көбінесе күшті электромагниттік кедергіге ұшырайды. Сондықтан, сіз әрқашан бірдей мөлшерлемесі және кодтау әдістерін пайдалану мүмкін емес компьютерге ішкі және сыртқы деректерді беру кезінде байланыс желісінің қабылдау соңында импульс сенімді анықтау үшін пайдаланады.

Модуляция - компьютерлік желілер әлеуеті және сандық деректерді, сондай-ақ компьютердің ішіндегі пайдаланылатын деректерді ұсыну үшін нақты жолын кодтау импульс ретінде пайдаланылады. Модуляцияланған кезде, дискретті ақпарат тарату үшін қолданыстағы сілтеменің жақсы болатын жиіліктің синусоидалы сигналы ретінде ұсынылады. Әлеуетті немесе импульстік кодтау арнасында берілетін сигналдарды күшті бұрмалау енгізетін жағдайда қолайлы синусоидалды сигналдар негізінде жоғары сапалы арналар мен модуляция қолданылады.

Дабылды берудің тағы бір мәселесі - бір компьютердің таратқышын екінші қабылдағышпен өзара синхрондау мәселесі. Компьютердің ішінде барлық модульдер ортақ сағаттан синхрондалады. Байланыс жүйелерінде синхрондау мәселесінің жеке желісі арқылы арнайы синхрондау сағаттарын бөлісу арқылы, сондай-ақ мерзімді синхрондау кодтары немесе алдын ала себепші бұршақ нысаны деректері арқылы ерекшеленетін өзіне тән пішіні арқылы, түрлі жолдармен шешілуі мүмкін. Блоктан кейін әр байт немесе байттар кейінгі байланыс желілері арқылы бақылау сомасын есептеу және алынған құндылықтарды беру - компьютерлер арасында деректерді беру сенімділігін жақсарту үшін, әдетте, стандартты әдіс қолданылады.

Тасымалдауды физикалық арналар арқылы берудің көптеген сипаттамалары бар. Төменде олардың кейбіреулері туралы айтылады:

Аталатын (ұсынылған) жүктеме — деректердің ағыны пайдаланушыдан желіге кіреді. Ұсынылған жүктемені желіге секундына биттерге (немесе килобитке, мегабитке және т.б.) кіретін жылдамдықпен сипаттауға болады.

Ақпарат алмасу жылдамдығы — желі арқылы берілетін деректердің нақты деректер жиілігі. Бұл жылдамдық мәлімделген жүктің жылдамдығынан аз болуы мүмкін, себебі желідегі деректер бұрмалануы немесе жоғалуы мүмкін.

Байланыс арнасының сыйымдылығы, өнімділік деп аталады, бұл арна арқылы ақпарат берудің барынша мүмкін жылдамдығы болып табылады. Бұл құрылымның ерекшелігі - ол физикалық тасымал ортасының параметрлерін ғана емес, сонымен қатар осы ортада дискретті ақпаратты берудің тандалған әдісінің ерекшеліктерін көрсетеді. Мысалы, Ethernet желісіндегі байланыс арнасының сыйымдылығы 10 Мбит / с. Бұл жылдамдық Ethernet технологиясы мен оптикалық талшықтың комбинациясы үшін ең жоғары шегінде мүмкін болып табылады. Дегенмен, бірдей оптикалық талшық үшін деректерді кодтау әдісі, сағат жылдамдығы және басқа қуатқа болатын басқа параметрлер сияқты деректерді беру технологиясын дамытуға болады. Осылайша, FastEthernet технологиясы 100 Мбит / с жылдамдығымен бірдей оптикалық талшықтан деректер беруді қамтамасыз етеді, ал Gigabit Ethernet технологиясы 1000 Мбит / с болып келеді. Байланыс құрылымының таратқышы арнаның өткізу қабілеттілігіне тең жылдамдықпен жұмыс істеуге тиіс. Бұл жылдамдық кейде таратқыштың бит жылдамдығы деп аталады.

Өткізу сызығы — бұл термин жаңылыстыруы мүмкін, өйткені ол екі түрлі мағынада қолданылады. Біріншіден, оны тасымалдау ортасын сипаттау үшін қолдануға болады. Бұл жағдайда жолдың елеулі бұрмаланусыз өткізетін жолдың өткізу қабілеттілігін білдіреді. Осы ұғымнан терминнің шығу тегі айқын болады. Екіншіден, «өткізу қабілеттілігі» термині байланыс арнасының сыйымдылығының синонимі ретінде пайдаланылады. Бірінші жағдайда, өткізу жолағының ені Герц (Гц), екіншісінде - секундына биттерде өлшенеді. Терминнің мағынасын контекстен ажыратып алу қажет, кейде бұл өте қиын болып табылады. Әрине әртүрлі сипаттамалар үшін әртүрлі терминдерді қолданған жөн болар еді, бірақ өзгертуге қиын дәстүрлер бар. «Өткізу кеңістігі» терминінің екі жақты қолданылуы көптеген стандарттардың бір бөлігіне айналды.

Байланыс арнасының сипаттамаларының тағы бір тобы бір немесе екі жақты арнаға ақпарат беру мүмкіндігімен байланысты. Екі компьютермен өзара әрекеттесу кезінде әдетте ақпаратты екі бағыт бойынша да беру керек. Пайдаланушы ақпарат алуды (мысалы, Интернеттен бейне файлды жүктеуді) ғана алатындығын немесе оны ғана (электрондық поштаны жібереді) деп санаған жағдайда, ақпарат екі бағытта ауыстырылады. Шындығында, пайдаланушыға қызығушылық тудыратын базалық деректер ағындары және кері бағыттың қосалқы ағыны - осы деректерді алу туралы хабарламалар бар. Физикалық байланыс арналары ақпараттың екі бағыты бойынша да жіберілуіне байланысты бірнеше түрге бөлінеді.

Дуплексті арна ақпараттарды екі бағыт бойынша бір мезгілде беруді қамтамасыз етеді. Ол әрқайсысы бір бағытта ақпарат беру үшін пайдаланылатын екі физикалық медиа болуы мүмкін. Немесе бір мезгілде қарсы ағындарды беру үшін бір орта болуы мүмкін;

Бұл жағдайда әрбір ағынды жалпы сигналдан оқшаулау үшін қосымша әдістер қолданылады

Жартылай дуплексті арна ақпараттың екі бағытта да берілуін қамтамасыз етеді, бірақ бір мезгілде емес, өз кезегінде белгілі бір уақыт аралығында ақпарат бір бағытта, ал келесі кезеңде - қарсы бағытта беріледі.

Симплексті арна ақпаратты бір бағытта беруге мүмкіндік береді. Дуплекстік арна жиі екі қарапайым арналардан тұрады.

2.2.

ЖЕЛІЛІК ӨЗАРА ӘРЕКЕТТЕСУДІҢ ҚИЫНДЫҚТАРЫ

Желі бір жұмыс станциясынан екіншісіне деректерді беруді қамтамасыз ететін бірқатар операцияларды орындайды. Пайдаланушы бұл қалай орын алғанына мүдделі емес, оған басқа компьютерде орналасқан бағдарламаға немесе компьютерлік ресурстарға кіру қажет. Шындығында, берілген деректер көптеген өңдеу қадамдары арқылы өтеді.

Белгілер және идентификаторлар - бақылаудың ақпараттық жабдықталған, әрқайсысы желілік пакеттер болып табылады. Біріншіден, ақпараттық блоктарға бөлінеді. Бұл деректер солардың алдында бұзылуына әкеліп соқтыруы үшін жасалған. Осы пакеттердің әрқайсысы қолданылатын байланыс ортасына сәйкес тиісті дабылдарға (кодталған) түрленеді. Таңдалған кіру әдісіне сәйкес желілерде электр немесе жарық дабылдары пайдаланылады. Хост жұмыс бекеті деректерінің кері жағында өңделеді: қабылданған, бастапқы көрінісін ескере отырып, блоктар шығу, қабылдамау жұмыстарын атқарады. Бұл, әрине, процестердің оңайлатылған сипаттамасы болып табылады.

Көрсетілген іс шаралардың бөлігі тек бағдарланады, ал қалған бөлігі— аппаратты түрде, операцияларын орындай алады

бағдарламалы да, аппаратты да жүзеге аса алады. Бірақ хабарларды жібергенде, желіні алмасудың екі жағы көптеген келісімдерді қабылдауы керек. Олар жіберілетін дабылдардың деңгейін және формасын, хабарлардың ұзақтығын анықтау әдісін, алынған ақпараттардың сенімділігін бақылау әдістерін және тағы басқаларын анықтауы керек.

Желі әртүрлі жабдықтарды қосу болып табылады, сондықтан ең өткір проблемалардың бірі үйлесімділік мәселесі болып табылады. Жабдықтарды дамытудың жалпы қабылданған ережелерінің барлық өндірушілердің талаптарын сақтамай, желілерді жобалаудағы прогреске қол жеткізу мүмкін емес еді. Сондықтан компьютерлік технологияның барлық дамуы, сайып келгенде, стандарттарда көрініс табады. Кез-келген жаңа технология тек қана тиісті стандартта бекітілген кезде ғана «занды» мәртебеге ие болады.

Ашық жүйе — бұл тұтастай қол жетімді және жалпы қабылданған халықаралық стандарттарға сәйкес әзірленген аппараттық және бағдарламалық өнімдер мен технологияларды қамтитын есептегіш орта. Ашық жүйелер - компьютер, компьютерлік желі, операциялық жүйе, бағдарламалық пакет, басқа да аппараттық және бағдарламалық өнімдер болып табылады.

Ашық сипаттамаларға сәйкес ашық жүйені кез келген жүйе деп атауға болады. Компьютерлік технологиядағы «спецификация» термині аппараттық немесе бағдарламалық компоненттердің формалды сипаттамасын, олардың жұмыс істеу әдістерін, басқа компоненттермен өзара әрекеттесуді, жұмыс жағдайларын, арнайы сипаттамаларын білдіреді.

Компьютерлік желілерде стандарттау негізі желілік құралдарды дамытуға көп деңгейлі көзқарас болып табылады. Бұл әдіс негізінде желілік мамандар үшін әмбебап тілге айналған ашық жүйелердің өзара іс-қимылының стандартты үлгісі құрылды. Желілік үлгі желіде орындалған барлық процедураларды стандарттауға мүмкіндік береді, оларды бір-бірімен өзара әрекеттесетін деңгейлерге және қосалқы деңгейлерге бөледі. Олар өзара іс-қимылдың барлық кезеңдерінде деректерді қалай беру және қабылдау туралы келісімдер жасайды. Сонымен қатар, желілік үлгілер бір желідегі абоненттер үшін де, түрлі деңгейдегі түрлі желілер үшін де ақпараттық алмасуды дұрыс ұйымдастыруға мүмкіндік береді.

Көп деңгейлі көзқараста хабарларды алмасуға екі жақ қатысады. Осылайша, әртүрлі компьютерлерде жұмыс істейтін екі «иерархияның» келісілген жұмысын ұйымдастыру қажет. Желілік алмасудың екі қатысушысы да көптеген келісімдерді қабылдауы қажет.

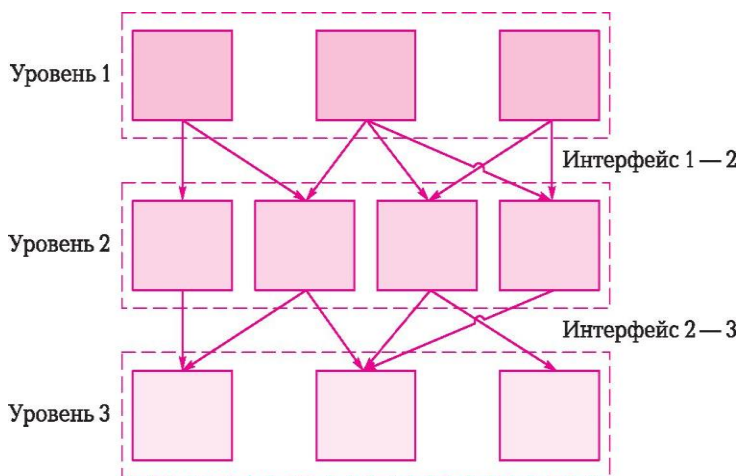
Мысалы, олар электр дабылдарының деңгейлері мен формалары туралы, хабарлардың ұзақтығын қалай анықтау керек, валидация әдістерін келісу және т.б. басқаша айтқанда, келісімдердің ең төменгі деңгейден (бит деңгейінен) ең жоғары деңгейге дейін, желі пайдаланушылары үшін қызметті жүзеге асыру болып табылады.

Хаттамалар — бұл желі деңгейінің бір деңгейін білдіретін, бірақ желідегі түрлі түйіндерде орналасқан хабар алмасу тәртібі мен пішімін анықтайтын ресімделген ережелер. Басқаша айтқанда, хаттамалар желідегі байланысты қамтамасыз ету тәртібін реттейтін рәсімдер мен ережелердің жиынтығы болып табылады.

Көршілес деңгейдегі хаттамаларды жүзеге асыратын және сол торапта орналасқан модульдер стандартталған хабар пішімдерін пайдаланып, анықталған ережелерге сәйкес бір-бірімен өзара әрекеттеседі. Бұл ережелер әдетте интерфейс деп аталады (2.1 сурет)

Интерфейс бір тораптағы іргелес деңгейде орналасқан желілік компоненттерді алмастыратын хабарлардың реттілігі мен пішімін анықтайды. Интерфейс бұл деңгеймен көршілес деңгейге ұсынылатын қызметтер жиынтығын анықтайды.

Бір қарағанда, хаттама мен интерфейс бірдей тұжырымдаманы білдіреді, бірақ шын мәнісінде оларда әртүрлі тапсырмалар бар: хаттамалар әртүрлі тораптардағы бірдей деңгейдегі модульдер өзара әрекеттесу ережелерін анықтайды және интерфейстер - бір тораптағы көрші деңгейдегі модульдер болып табылады.



2.1 сурет. Өзара әрекеттесу деңгейін ұйымдастыру

Желілердегі түйіндердің өзара іс-қимылын ұйымдастыру үшін жеткілікті иерархиялық ұйымдастырылған хаттамалар жиынтығы *байланыс хаттамаларының жинағы* деп аталады.

Байланыс хаттамалары бағдарламалық және аппараттық құралдарда да жүзеге асырылуы мүмкін. Төмен деңгейлі хаттамалар көбінесе бағдарламалық жасақтама мен аппараттық құралдардың тіркесімі арқылы жүзеге асырылады, ал жоғарғы қабат хаттамалары әдетте тек бағдарламалық жасақтама болып табылады.

Хаттамалар тек компьютерлермен ғана емес, басқа да желілік құрылғылармен - тораптармен, көпірлермен, ажыратқыштармен, маршрутизаторлармен және т.б. құрылғының түріне қарай, бір немесе бірнеше хаттамалар жиынтығын енгізуге кірістірілген құралдар болуы керек.

Стандарттар мен сипаттамалардың ашық сипаты тек байланыс хаттамалары үшін ғана емес, сондай-ақ желіні құруға арналған түрлі құрылғылар мен бағдарламалар үшін маңызды болып табылады.

2.3. OSI ЖЕЛІЛІК ҮЛГІСІ

1970-ші жылдардың соңында Халықаралық стандарттау ұйымы (ISO) стандартты желілік қарым-қатынастың барлық аспектілерін қамтиды, ол - ашық жүйе ақпаратты (OpenSystemInterconnection) осы алмасу OSI эталондық үлгісіне ұсынылды. OSI үлгісі жеті қабаттары барлық өзара және деректер желісіне бөледі: (. 2.1-кесте) өтініш, өкілі, сессия, көлік, желілік, сілтеме, физикалық. Әр деңгей үшін олардың орындалатын функциялары анықталады, негізгі ұғымдардың терминдері мен анықтамалары беріледі.

Әдебиетте, ең жиі кездесетіні 7-ші деңгейден OSI үлгісі қабаттарының сипаттамасын бастау қабылданған, пайдаланушы өтініштер желісіне кіру қолданбалы деп аталатын. OSI үлгісі 1-ші деңгеймен аяқталатын - физикалық қабат, ал үшінші тарап жеткізушілерінің деректерді беру орталарына орнатылған стандарттарын анықтайды:

- тасымалдау ортасының түрі (мыс кабельдері, оптикалық талшық, радиоэфир ж.т.б.);
- дабыл модуляциясының түрі;
- логикалық дискретті жағдайлардың дабыл деңгейлері (нөлдер мен бірліктер).

OSI үлгісінің кез келген хаттамасы өз деңгейінің хаттамалары немесе бірлікке шаққандағы хаттамаларымен өзара әрекеттесуі керек

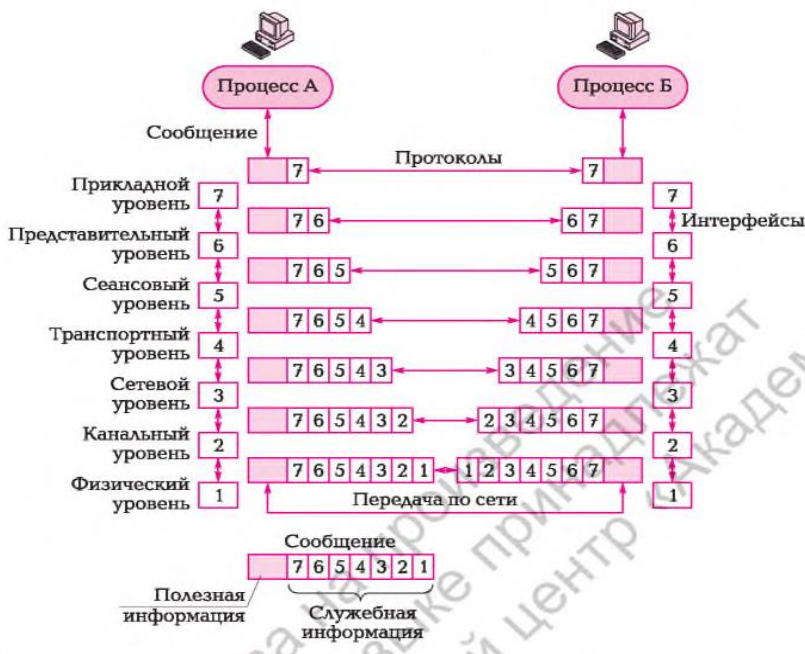
2.1 кесте. OSI үлгісі

Деректер түрі	Нөмірі	Деңгей (layer)	Функция
Деректер	7	Қолданбалы (application)	Желілік қызметтерге қол жеткізу
Ағыны	6	Көрініс деңгейі (presentation)	Деректерді ұсыну және шифрлау
Сеанстары	5	Сеанстық (session)	Сессияны басқару
Сегменттері	4	Көліктік (transport)	Соңғы нүктелер мен сенімділік арасындағы тікелей байланыс
Пакеттер / Дей-таграммалар	3	Желілік (network)	Бағытты анықтау және логикалық адресітеу
Кадрлар	2	Арналы (data link)	Физикалық адресітеу
Биттер	1	Физикалық (physical)	Дасалар ортасымен, сигналдармен және басқа жеке деректермен жұмыс істеу

және/немесе өз деңгейінен төмен (2.2 сурет). Оның деңгейіндегі хаттамалармен өзара әрекеттесу *көлденең* деп аталады, ал жоғары немесе төменгі - *тік* бірлікке шаққандағы деңгейлер болып табылады.

Әрбір деңгей шартты түрде логикалық бөлінбейтін деректер элементі арқылы сәйкес келеді, ол үлгі және қолданылған хаттамалар шеңберінде жеке деңгейде жұмыс істей алады. Сілтеме қабатының ақпараты желідегі шеңберінде біріктірілген кезде бит - физикалық деңгейде, ең кішкентай бірлігі қаптардағы, т.б. деректердің әрбір бөлігі логикалық беру (кадр, пакет, датаграмма) үшін біріктірілген, хабар болып саналады.

OSI үлгісі соңғы пайдаланушы бағдарламаларына емес, жүйенің өзара әрекеттесу құралдарын сипаттайды. Өтінімдер жүйелік құралдарға сілтеме жасайтын өздерінің өзара әрекеттесу хаттамаларын іске асырады. Өтініш OSI үлгісінің кейбір жоғарғы деңгейлерінің функцияларына қатысты болуы мүмкін, бұл жағдайда, егер желіаралық алмасу қажет болса, ол OSI үлгісінің қалған төменгі қабаттарының функцияларын орындайтын жүйелік құралдарға тікелей қатысты. Мысалы, кейбір дерек қорларын басқару жүйесі кірістірілген файлдарға қашықтан қатынасу болып келеді.



Соңғы пайдаланушы қолданбасы басқа жүйеде жұмыс істейтін басқа бағдарламамен диалогты ұйымдастыруға ғана емес, сондай-ақ белгілі бір желілік қызметтің қызметтерін алу үшін, мысалы, қашықтағы файлдарға қолжетімділікті алуға, поштаны алуға немесе ортақ принтерді қамтамасыз етеді.

Өтінім қосымшаның қабатын, мысалы, файлдық қызметке сұраса, онда осы сұраудың негізінде бағдарламалық жасақтама қызмет туралы ақпаратты (тақырып) және, мүмкін, берілетін деректерді орналастыратын стандартты форматтың байланыс форматын жасайды. Хабарламаның деректер өрісінің бос болуы немесе кез-келген деректерді қамтуы мүмкін, мысалы, жойылған файлға жазылу қажет болады. Бірақ бұл ақпаратты өз тағайындауына жеткізу үшін әлі де көптеген міндеттер шешіледі.

Өтінім деңгейінің үстіңгі деректемесінен алынған ақпаратқа негізделген өкілдік деңгейлі хаттама қажетті әрекеттерді орындайды және хабарламаға өзінің қызмет ақпаратын - тағайындалған машинаның өкілдік деңгейлі хаттамасына арналған нұсқауларды қамтитын өкілдік деңгейдің тақырыбын қосады. Алынған хабар сеанс қабатына жіберіледі, ол өз кезегінде тақырыпты қосады және т.б іс әрекеттерді жүзеге асырады. Кейбір хаттамалар қызметтің ақпаратын тақырыптың басында ғана емес, сонымен бірге соңында «соңғы нүкте» немесе кейде «терминатор» деп аталады. Ақыр соңында, хабар төменгі физикалық қабатқа жетеді, ол шын мәнінде оны алушы компьютеріне байланыс желілері бойынша жібереді. Осы уақытқа дейін барлық деңгейдегі «асқан» тақырыптар кездеседі. Хабар желіде басқа құрылғыға жіберілгенде, ол деңгейден деңгейге дейін жылжиды. Әрбір деңгей деңгейдің тақырыбын талдайды, өңдейді және жояды, берілген деңгейге сәйкес функцияларды орындайды және хабарламаны жоғарғы қабатқа жібереді (2.3-сурет).

Физикалық деңгей. Физикалық деңгей - бұл үлгінің төменгі қабаты, ол екілік пішінде ұсынылған деректерді бір құрылғыдан (компьютерден) екіншісіне ауыстыру әдісін анықтайды.

Физикалық деңгей оптикалық талшық, бұралған жұп, коаксиалды кабель, спутниктік деректер арнасы және т.б. сияқты бұқаралық ақпарат құралдарының түрлерін анықтайды. Бұл деңгейге деректерді алмасудың физикалық ортадағы сипаттамасы, сондай ақ өткізу сызығы

n - деңгей
хабарламасы

3-деңгей
хабарламасы

Тақырыбы	Деректер	Соңы 3
3		

Деректер жолы 3

2-деңгей
хабарламасы

Тақырыбы	Тақырыбы	Деректер	Соңы 3	Соңы 2
2	3			
ч				>

Деректер жол 2

1-деңгей
хабарламасы

Тақырыбы	Тақырыбы	Тақырыбы	Деректер	Соңы 3	Соңы 2	Соңы 1
1	2	3				

Деректер жолы 1

2.3. Кесте. Әртүрлі деңгейде хабарлама структурасы

шудың иммунитеті, толқындық импеданс және т.б. бірдей деңгейде дискретті ақпаратты беретін электр дабылдардың сипаттамалары - импульстік фронттардың бұрышы, берілетін дабылдардың кернеуі немесе ток деңгейі, кодтау түрі, дабыл беру жылдамдығы анықталады. Бұдан басқа, контактілердің түрлері мен әр контактінің мақсаты стандартталған болып келеді.

Қысқаша, физикалық деңгейдегі функцияларды анықтай аламыз:

- физикалық арналар арқылы биттерді беру;
- электр сигналдарын қалыптастыру;
- ақпаратты кодтау;
- үндестіру;
- модуляция.

Физикалық деңгей аппараттық құралдарда жүзеге асырылады. Физикалық деңгейдің функциялары желіге қосылған барлық құрылғыларда жүзеге асырылады. Осы деңгейде концентраторлар, дабыл таратқыштар және медиа түрлендіргіштер жұмыс істейді. Компьютер тарапынан физикалық деңгей функциялары желілік адаптер немесе сериялық порт арқылы орындалады. Физикалық деңгей екі жүйе арасындағы, физикалық электр және механикалық интерфейстер арқылы жүзеге асады.

Физикалық деңгейдің хаттамасы мысалы, кабельді 100 Ом толқындық кедергісі бар экрандалмаған бұралған жұп санаты 3 пайдаланылған, анықтайтын сипаттамасы Ethernet 1Base-технологиясындағы қызмет RJ-45 ұясы, физикалық сегментінің максималды ұзындығы 100 м, кабель туралы деректерді білдіру үшін Манчестер коды және ақ орта және электрлік дабылдардың кейбір басқа сипаттамалары беріледі.

Арналық деңгей. Арналық деңгейдің пайда болуы мүмкін қателер физикалық деңгей және бақылау өзара іс-қимылын қамтамасыз етуге арналған. Физикалық деңгейде бит жай жіберіледі. Бұл байланыс желісі кезекпен бөлінеді онда сол желілерде компьютерлердің өзара іс-қимылындағы бірнеше жұп, дене трансмиссиясының ортасы бос емес болуы мүмкін, бұл жұмыс құрамына кірмейді.

Арналық деңгей міндеттерінің бірі электр жеткізуінің орта болуын тексеру болып табылады. Байланыстыру қабатының тағы бір тапсырмасы - қателерді анықтау және түзету. Ол үшін биттер жиектелген деп аталатын топтарға топтастырылады. Арналық деңгей оны бөлу үшін әрбір жақтаудың басында және соңында арнайы бит тізбегін орналастыру арқылы әрбір кадрдың берілуінің дұрыстығын қамтамасыз етеді, сондай-ақ барлық кадр биттерін белгілі бір жолмен өңдеу арқылы бақылау сомасын есептейді және басқару элементін қосады.

Қабырға желіге келгенде, ресиверден алынған деректердің бақылау сомасын қайта есептейді және нәтижені кадрдан бақылау сомасымен салыстырады. Егер олар сәйкес келсе, кадр дұрыс деп саналады және қабылданады. Егер сомалар сәйкес келмесе, онда қате түзетіледі. Байланыстыру қабаты тек қателерді анықтап қана қоймай, сонымен бірге оларды бүлінген кадрларды қайта жіберу арқылы түзете алады.

Байланыстыру қабаты үшін қателерді түзету функциясы міндетті емес, сондықтан осы деңгейдің кейбір хаттамаларында, мысалы, Ethernet желісінде қол жетімді болмайтынын атап өткен жөн. Арналық деңгей әдетте екі деңгейге бөлінеді:

- 1) деректермен логикалық алмасу (LogicalLinkControl, LLC);
- 2) отаға қол жетімділікті басқару (MediaAccessControl, MAC). MAC қабаты ортақ ортаны дұрыс пайдалануды қамтамасыз етеді. Қоршаған ортаға қол жеткізгеннен кейін, ол интерфейс функцияларын іргелес желілік қабатпен жүзеге асыратын LLC-нің жоғары деңгейін пайдаланады. MAC және LLC қабатының хаттамалары өзара тәуелсіз. Сондықтан әрбір MAC қабатының хаттамасы кез келген LLC хаттамасымен және керісінше қолданылады.

Арналық деңгей қызметі:

- Тарату ортасына қол жеткізуді бақылау;
- Жақтау жиектерін таңдау. Бұл тапсырманың ықтимал шешімдерінің қатарында кадрдың басталуы немесе аяқталуы болып табылатын кейбір жүйелілікті сақтау табылады;
- аппараттық адресстеу (немесе байланыс қабатының мекен-жайы). Бір уақытта бірнеше алушыны қабылдай алатын жағдайда талап етіледі. Жергілікті желілерде аппараттық мекенжайлар (MAC мекенжайлары) үнемі қолданылады;
- алынған деректердің сенімділігін қамтамасыз ету. Жақтауды беру кезінде, деректер бұрмалануы мүмкін. Мұны анықтап, қате бар раманы өңдеуге әрекет жасамайды. Әдетте арна деңгейінде, қателерді анықтаудың жоғары кепілдік беретін бақылау сомасының алгоритмдері пайдаланылады;
- жоғарғы деңгей хаттамаларын адресациялау.

LLC деңгейінің функциялары әдеттегідей операциялық жүйе модулімен жүзеге асырылады және MAC қабатының функцияларын бағдарламалық жасақтама желілік адаптері және оның жүргізушісі жүзеге асырады.

Желілік деңгей. Үлгінің желілік қабаты деректерді беру жолын анықтауға арналған. Логикалық мекен жай мен физикалық атауларды көрсетуге жауап беретін, қысқа бағыттарды анықтайтын, жарамсыздықтарды қадағалау мен желідегі кейбір «кемшіліктерді» реттеу болып табылады.

Желілік деңгейдегі қателерді өңдеу, мультиплексинг (бір арнада өткізу жолағының аз ені бар бірнеше ағындарды беру) және деректер ағынын бақылауды қамтамасыз ету керек.

Жергілікті желілердің байланыс қабатының хаттамаларына сәйкес типтік топологиясы бар желіде кез келген түйін арасындағы деректерге қол жеткізуді қамтамасыз етеді. Бұл желілерді жақсы дамыған құрылыммен құруға мүмкіндік бермейтін өте қатаң шектеу, мысалы, бірнеше корпоративтік желілерді бір желіге қосатын желілер немесе түйіндер арасында артық қосылымдар бар жоғары сенімді желілер болып табылады.

Бір жағынан, типтік топологиялар үшін деректерді беру процедураларының қарапайымдылығын сақтауға, ал екінші жағынан еркін топологияларды пайдалануға мүмкіндік беру үшін қосымша желілік қабат пайдаланылады. Бұл деңгейде «желі» тұжырымдамасы енгізіледі. Бұл жағдайда желі - стандартты топологиялардың біріне сәйкес бір-бірімен байланысқан компьютерлер жиынтығы және деректерді беру үшін осы топология үшін анықталған байланыс қабатының хаттамаларының бірін пайдалану болып келеді.

Осылайша, желіде деректерді беру кабельдік қабат арқылы реттеледі, ал деректер желісі бойынша желілер арасында тасымалданады. Желілік деңгейдің басты міндеті - деректерді бағыттау, яғни желілер арасында деректерді беру. Ең жақсы жолды (маршруттауды) таңдау және оны шешу мәселесі желілік деңгейдегі басты міндеттердің бірі болып табылады. Бұл мәселе ең қысқа жол әрдайым жақсы болмағандықтан қиындайды. Жиі маршрутты таңдау критерийі деректерді беру уақыты - байланыс арналарының өткізу қабілеттілігіне және трафиктің қарқындылығына байланысты - уақыт өзгеруі мүмкін индикаторлар болып табылады. Кейбір маршруттау алгоритмдері өзгерген жүктемелерге бейімделуге тырысады, ал басқалары ұзақ уақыт бойы орташа көрсеткіштерге негізделген шешімдер қабылдайды. Бағдарды таңдау басқа сенімділікке байланысты жүзеге асырылуы мүмкін.

Жалпы алғанда, желілік қабаттың функциялары әртүрлі құрылымдармен байланыстыру арқылы хабарламаларды хабарлау функциясынан әлдеқайда кең, мысалы, бірнеше жергілікті желілердің бірлестігі болады. Желілік деңгей сонымен қатар әртүрлі технологияларды салыстыру мәселелерін шешеді, ірі желілерде адресінеуді жеңілдетеді және желілер арасындағы қалаусыз трафикке сенімді және икемді кедергілерді жасайды.

Желі деңгейіндегі хабарлар пакеттер деп аталады. Желі деңгейінде пакеттерді жеткізуді ұйымдастыру кезінде «желінің нөмірі» түсінігі пайдаланылады. Алушының мекенжайы осы желідегі желінің нөмірі мен түйін нөмірінен тұрады. Бір желідегі барлық түйіндер мекенжайдың бірдей жоғары бөлігіне ие болуы керек, сондықтан желі деңгейіндегі «желі» термині басқа, неғұрлым ресми анықтамаға ие болуы мүмкін: желі мекенжайында бірдей желі нөмірін қамтитын түйіндер жиынтығы болып табылады.

Желілік деңгейде екі түрлі хаттамалар анықталды:

- 1) желі арқылы пакеттік трафикті іске асыратын желілік хаттамалар;
- 2) бағыттық ақпараттармен алмасу хаттамалары (бағыттау хаттамасы).

Осы хаттамалардың көмегі арқылы бағыттаушылар желі аралық қосылыстардың топологиясы туралы ақпараттар жинайды. Желі деңгейінің хаттамалары операциялық жүйенің бағдарламалық модульдері арқылы жүзге асырылады, сонымен қатар бағыттаушының бағдарламалық және аппараттық құралдары арқылы жүзеге асады.

Желі деңгейінің хаттамалары мысалы IPстек TCP/Іржәне IPХстек Novell арасындағы желі аралық алмасу хаттамалары болып табылады.

Көлік деңгейі. Көлік деңгейі біршама жоғары деңгейдегі звенолар арасындағы байланысушы реттерді байланыстыру болып табылады, олар қосымшалар, төменгі деңгей мен байланыс сызықтарына біршама байланыстарын білдіреді. Көлік деңгейі сатып алушының алушыға деректерді сенімді түрде жеткізуін қамтамасыз етуге арналған болып табылады. Көліктік деңгейдің міндеттері кез келген желі өзегіндегі талап етілетін сапалы ақпараттарды жеткізуді қамтамасыз етеді, соның ішінде:

- сеанстық деңгейдегі хабарламаларды пакеттерге бөлу, оларды нөмірлеу;
- қабылданған пакеттерді буферлеу;
- келіп түсетін пакеттерді ретке келтіру;
- қолданбалы процесстерді адресациялау;
- ағынды басқару.

Ережеге сай, барлық хаттамалар, көлік деңгейінен бастап және одан да жоғары, желі өзектерінің соңғы жағдайларымен жұмыс жасайды — олардың желілік операциялық жүйесінің компоненттерімен жұмыс жасайды. Көлік хаттамалары ретінде мысалған TCP және UDPстек TCP/IP және SPXстек Novell хаттамаларын келтіруге болады.

Төрт төменгі деңгей хаттамаларын қоса алғанда желі көліктері немесе көліктік қосалқы жүйе деп атайды, себебі олар толығымен деңгеймен қосыла отырып тасымалдау міндеттерін атқарады, мысалы электронды пошта хаттамаларының көмегі арқылы жүзеге асады.

Қалған үш жоғарғы деңгей қолдағы бар көліктік қосалқы жүйе негізінде қолданбалы сервис арқылы міндеттерін шешеді.

Сеанстық деңгей. Сеанстық деңгей байланыс сеанстарын қолдауды жүзеге асырады, қосымшаларын ұзақ уақытта өзара қызмет атқаруын, басқаша айтқанда, екі жұмыс желі бекеттерінің арасындағы байланыс координациясына жауап береді.

Деңгей деректер алмасу сеанстарын ұйымдастырады, сеансты құру мен аяқталуын, тапсырмаларын синхрондау, деректермен алмасу құқықтарын анықтап, белсенді емес қосымшалар кезеңінде сеанстард қолдайды.

Алмасу синхронизациясы бақылау нүктелерінің деректер ағындарын орналастыруды қамтамасыз етеді, егер қарсылық болған жағдайда барлығын басынан бастамас үшін соңғы бақылау нүктесіне қайтуға боларлық жағдай жасалады. Тәжірибеде көптеген қосымшалар сеастық деңгейді пайдаланады, бірақ оларды жұмсай қоймайды.

Көрініс деңгейі. Көрініс деңгейі қолданбалы деңгеймен алмасатын ақпараттардың жүйенің басқа да қолданбалы дегейлерінде түсінікті болуын қамтамасыз етеді. Оның көмегімен ерекшеліктерінің алдын алады, мысалы барлық мүмкіндіктер арасындағы кодтаудың белгілері мен синтаксистері жүзеге асырылады.

Көрініс деңгейі көріністің кейбір жалпы форматтарындағы деректер форматтарын түрлендіруді жүзеге асырады, ал қабылдауда, сәйкесінше керісінше ауыстыру жұмыстарын жүзеге асырады. Бұл деңгейде деректерді шифрлау және шифрдан шешу жұмыстары жүзеге асырылады, осыған сәйкес барлық қолданбалы сервистерге арналған деректермен алмасу құпиялылығы бірден қамтамасыз етіледі. Хаттама мысалы ретінде, жұмыс жасайтын көрініс деңгейінде SecureSocketLayer (SSL) хаттамасы болып табылады, ол хаттамалар TCP/IP қолданбалы стек деңгейіндегі хаттамалар үшін құпия хабарламалармен алмасуды қамтамасыз етеді.

Қолданбалы деңгей. Қолданбалы деңгей пайдаланушы мен желілердің өзара қарым қатынасын қамтамасыз етеді. Деңгей желі қызметтеріне пайдаланушы қосымшаларының қол жетімділігіне рұқсат береді, осылайша деректер қорына сұранысты өңдейді, файлдарға қол жеткізеді, электронды мекен жайларға қайта жібереді. Сондай ақ қызметтік ақпарат алмасуларына жауап береді, қателіктер туралы ақпарат қосымшаларын ұсынады және көрініс деңгейіне сұраныстарды құрастырады. Мысалы: HTTP, POP3, SMTP.

Қолданбалы деңгей — бұл шын мәнінде, түрлі қарапайым хаттамалар жинағы, оның көмегімен пайдаланушы желілері бөлінгіш ресурстарға қол жеткізе алады, олар мынадай файлдар, принтерлер мен гипермәтіндік веб-беттер, сондай ақ

Бірыңғай жұмысты ұйымдастырады, мысалы, электронды пошта хаттамасының көмегімен жүзеге асырады. Деректер бірлігі, қолданбалы деңгейді белсенді түрде жүзеге асыратын ол — хабарлама.

OSI үлгісінің барлық деңгейінің функциялары екі топтың біріне қатысты болуы мүмкін: нақты техникалық жұмыс желісіне қатысты функция, немесе, қосымшалармен жұмысты болжау функциясына қатысты болады.

Үш төменгі деңгей (физикалық, арналық және желілік) желіге қатысты болып табылады, яғни, осы деңгей хаттамалары желінің техникалық жұмысымен тығыз байланысты және коммуникациялық құрылғыларда қолданылады. Мысалы, FDDI құрылғысына өту желінің барлық өзектеріндегі арналық және физикалық деңгей хаттамаларын толық ауыстыруды білдіреді.

Үш жоғарғы деңгей (қолданбалы, көрініс және сеанстық) қосымшаға бағытталады және желі құрылымының техникалық ерекшеліктеріне байланысы аз болады.

Осы деңгей хаттамаларына желі топологиясындағы ешқандай өзгерістер әсер етпейді. Бұл деңгейлердің хаттамалары желілік топологияның өзгеруіне, жабдықты ауыстыруына немесе басқа желілік технологияларға көшуіне әсер етпейді. Осылайша, Ethernet-тан 100 VG-АноЛАН жоғары жылдамдық технологиясына көшу қолданбалы, алдын ала тестілеу және сеанс деңгейлерінің функцияларын жүзеге асыратын бағдарламалық жасақтамадағы ешқандай өзгерістерді талап етпейді.

көлік қабаты жоғарғы төменгі деңгейдегі жұмыс істеуі барлық егжей жасырады, аралық болып табылады. Бұл тікелей хабар тасымалының техникалық құралдарына тәуелді емес қосымшаларды әзірлеуге мүмкіндік береді.

ОСИ моделі көптеген байланыс модельдерінің бірін ғана ұсынады. Бұл модельдер және олармен байланысты хаттама стеки қабаттарының, олардың функцияларын, хабар пішімдерін, жоғарғы деңгейде көрсетілетін қызметтер және басқа да параметрлер санына қарағанда әр түрлі болуы мүмкін.

2.4. ХАТТАМАЛАР МЕН СТАНДАРТТАР

2.4.1. Хаттамалардың жалпы сипаты

Желілердегі компьютерлердің өзара әрекеттесуі хабарларды алмастырудың белгілі бір ережелеріне және олардың форматтарына сәйкес, яғни белгілі бір хаттамаларға сәйкес жүзеге асырылады. Әр түрлі деңгейдегі хаттамалардың иерархиялық ұйымдастырылған жиынтығы

хаттамалық стек деп аталатын компьютерлік желідегі түйіндердің өзара әрекетін ұйымдастыру үшін жеткілікті. Жүйенің өзара әрекеттесуінің тұжырымдамалық схемасы болып табылатын желілік модельге қарағанда, хаттама стекі желілік өзара іс-қимылды жүзеге асыруға мүмкіндік беретін нақты спецификациялардың жиынтығын білдіреді.

Желілерде кеңінен қолданылатын көптеген хаттама жинақтар бар. Танымал хаттама қабаттастырып мысалдары IPX / SPS фирма Novell, Интернетте және көптеген UNIX-негізделген амалдық жүйесі, желілік стекті Стандарттау жөніндегі OSI халықаралық ұйым, стек Десна корпорациясы Digital Equipment басқаларға пайдаланылатын TCP / IP стек болуы мүмкін.

Белгілі бір жинақтағы желідегі байланыс хаттамаларын пайдалану желінің сипаттамаларын айтарлықтай анықтайды. Шағын желілерде бір стек пайдаланылуы мүмкін. Әртүрлі желілерді біріктіретін ірі корпоративтік желілерде, әдетте, бірнеше стек пайдаланылады.

Байланыс құралдары жоғарғы қабаттарының хаттамаларына қарағанда, стандартталған төменгі деңгейдегі, бір іске хаттама, және ол түрлі өндірушілердің жабдықтарын табысты бірлескен жұмыс істеуі үшін қажетті шарт болып табылады. Осы немесе осы байланыс құрылғысын қолдайтын хаттамалардың тізімі осы құрылғының ең маңызды сипаттамаларының бірі болып табылады.

Сервер мен желілік қызметтердің клиент компоненті - желілік операциялық жүйелерге тиісті бағдарламалық элементтерінің түрінде жүзеге асырылатын Компьютерлік байланыс хаттамалары, мысалы, сілтеме қабатты хаттамалары әдетте желілік адаптерлер, драйверлер, және хаттаманың жоғарғы қабаттарының түрінде жасалған болып табылады. Операциялық жүйенің ортасында жақсы жұмыс істеу мүмкіндігі байланыс жабдығының маңызды сипаттамасы болып табылады.

Жиі жарнамалық желі адаптері немесе ол арнайы NetWareіі UNIX желісінде жұмыс істеу үшін арналған хабарламаны оқи алады. Бұл осы хаттамалардың әр түрлі операциялық жүйелерінде қолданылады, егер аппараттық әзірлеушілер, желілік операциялық жүйесінде пайдаланылатын хаттама немесе оларды іске асыру осы нұсқасына қатысты, оның өнімділігін оңтайландыру дегенді білдіреді. Операциялық жүйенің ортада жұмыс істеуге мүмкіндігі оның сертификаттауды пайдаланылатын байланыс жабдықтардағы сипаттамаларының бірі ретінде түрлі операциялық жүйелерде хаттаманы іске асыру сипаты арқасында жүзеге асады.

Жергілікті желілерде ақпарат алмасуға арналған хаттамалар үш функционалдық түрге бөлінеді:

- 1) қолданбалы;
- 2) көліктік;
- 3) желілік.

Қолданбалы хаттамалар OSI-қосымшаның үш жоғарғы қабатының функциялары, көрсетілім қабаты және сеанс қабаты. Олар өзара әрекеттесуді және олардың арасында деректер алмасуды қамтамасыз етеді.

Көліктік хаттамалар OSI моделінің көліктік және сеанстық деңгейдегі функцияларын іске асырады. Олар желінің түйіндері арасындағы байланыс сеанстарын инициализациялайды және қолдайды және деректерді берудің сенімділігіне қажетті деңгейін қамтамасыз етеді.

Желілік хаттамалар OSI-желісінің, арна мен физикалық модельдің үш төменгі қабатының функциялары. Бұл хаттамалар байланыс қызметтерін ұсыну, жақтауларын шешу, бағыттау, қате тексеру және қайта таратуды басқару жекелеген орталарда, мысалы Ethernet немесе TokenRing қарым қатынас деректерінің ережелерін айқындайды. OSI хаттамалар моделінің түрлі деңгейдегі тиесілі 2.2 кестеде көрсетілген.

Әр түрлі операциялық жүйелер мен түрлі аппараттық платформаларда жұмыс істейтін және негізінен жергілікті және кеңінен екі желілер үшін ортақ мүмкіндіктерге арналған хаттамалар болып табылады.

Хаттама түрлері	OSI үлгісінің деңгейі		Хаттама
Қолданбалы хаттамалар	7	Қолданбалы	FTAM, X.400, X.500, SMPT, FPT, SNMP, Telnet, SMB, NCP, Apple Talk и Apple Share, AFP, DAP
	6	Көрініс	
	5	Сеанстық	
Көліктік хаттамалар	4	Көліктік	TCP, SPX, NetBIOS, ATP, NBP
	3	Желілік	
Желілік хаттамалар	2	Арналық	IP, IPX, NetBEUI, DDP
	1	Физикалық	

1980 жылы, Электротехника және электроника IEEE (Institute of Electrical and Electronics Engineers) институты жергілікті желілердің төменгі деңгейдегі жобалау бойынша ұсыныстары бар, стандарттарды IEEE802.x отбасында шешім қабылдайды, оның нәтижесінде, жергілікті желілердің стандарттау жөніндегі 802 комитеті ұйымдастырады. Кейінірек, Комитеттің жұмысында халықаралық стандарттар жиынтығының негізі қалыптасты ISO8802-1 ... 5. Бұл стандарттар Ethernet желілері стандарттарындағы ArcNet TokenRing өте танымал бренд негізінде құрылған болатын. Стандарттар IEEE802.x отбасы жеті деңгейлі үлгісіне OSI- физикалық және деректердің сілтемесі тек төменгі екі деңгейін қамтиды. Бұл көп дәрежеде осы деңгейлердегі жергілікті желілердің ерекшеліктерін көрсету фактісіне байланысты болып келеді.

2.4.2. OSI хаттамалар стегі

OSI протоколының стакасы мен OSI үлгісін ажыратуға болады. OSI стеки - өте нақты спецификациялардың жиынтығы - келісілген дестені құрайтын протоколдар. Бұл халықаралық, өндірушіден тәуелсіз стандарт. Корпорациялар, серіктестер мен жабдықтаушылар арасындағы өзара әрекеттесу адрестеу, аттестация және деректер қауіпсіздігі мәселелерімен қиындатады. OSI стекиндегі барлық осы проблемалар ішінара шешілді. AXI хаттамалары орталық процессордың өңдеу қуатын көп мөлшерде талап етеді, бұл оларды дербес компьютер желілеріне емес, қуатты машиналарға қолайлы етеді. Белгілі бір себептермен, OSI стеки, басқа стандартты стектерден айырмашылығы, OSI өзара әрекеттесу үлгісіне толық сәйкес келеді, ол ашық жүйелер өзара әрекеттесу моделінің барлық жеті қабатының ерекшеліктерін қамтиды.

Физикалық және байланыс қабаттарында AXI стекасы Ethernet, TokenRing, FDDI және LLC, X.25 және ISDN хаттамаларын қолдайды. Желі, тасымалдау және сеанс қызметтері OSI стекинде қол жетімді, бірақ олар кең таралған. Желілік деңгейде хаттамалар қосылуды орнатпастан және байланыс орнатпастан жүзеге асырылады. OSI үлгісіндегі OSI үлгісінде белгіленген функцияларға сәйкес тасымалдау хаттамасы байланыс желісіне қосылу және байланыс орнатпастан желілік қызметтер арасындағы айырмашылықты жасырады, сондықтан пайдаланушылар желінің негізгі қабатына қарамастан қызметтің қажетті сапасын алады. Мұны қамтамасыз ету үшін көлік қабаты пайдаланушының қызмет көрсету сапасын талап етілетінін анықтайды.

Ең төменгі жылғы 0 4-сыныбына, ең жоғары класстан кейін деректердің қате серпімділігін және қалпына келтіру талаптарының дәрежесі әр түрлі - көлік қызметінде 5 сыныптар ретінде анықталады.

Қолданбалы деңгейіндегі қызметтерге файлды тасымалдау, терминал эмуляциясы, каталог қызметтері және пошта кіреді. Олардың ең перспективалы каталог қызметі (X.500 стандарты), электрондық пошта (X.400), виртуалды терминал хаттамасы (VT), байланыс хаттамасы, қол және файл басқару (FTAM), Transfer Protocol және жұмысты басқару (JTM) болып табылады.

OSI хаттамалар стегі желілерін салуға бірліктерін қамтамасыз ету және өзара іс-қимыл жалпыға тым күрделі және аз түсінік ретінде көптеген бағытталған хаттамалар қабылданатын болды. Себебі OSI стегі көзделген қолданыстағы хаттамаларды бас тарту және дестесіндегі барлық деңгейлерінде жаңа көшу, әзірлеу және енгізу болып табылады. Бұл стеканы жүзеге асыруға кедергі келтірді және басқа желілік технологияларға елеулі инвестициялар жасаған көптеген компанияларды тастауға себеп болды.

2.4.3. TCP/IP хаттамалар стегі

TCP/IP хаттамалар стегі — соның ішінде Интернетке желілерде пайдаланылатын желілік коммуникациялық хаттамалар жиынтығы. TransmissionControlProtocol (TCP) және InternetProtocol (IP), осы Стандартта алғаш әзірленген және сипатталған болды - TCP / IP атауы екі ең маңызды хаттамалар отбасынан алынған.

Сондай-ақ, Интернет стек деп аталатын TCP / IP А стегі, ең танымал және перспективалы байланыс хаттамалар мұржаларының бірі болып табылады. Стек гетерогенді есептеу қоршаған орта үшін ортақ хаттамалар жиынтығы ретінде байланысы тәжірибелік желілік ARPAnets АҚШ Қорғаныс департаментінің (DepartmentofDefence, DoD) бастамасы бойынша басқа спутниктік желілерінде әзірленді. ARPA желісі әскери саладағы әзірлеушілер мен зерттеушілерді қолдады. ARPA желісінде екі компьютер арасындағы байланыс InternetProtocol (IP) хаттамалар арқылы орындалды, ол әлі күнге дейін TCP / IP стекіндегі негізгі болып саналады және стек атында пайда болады.

TCP / IP стек дамуына үлкен үлес оның нұсқасында хаттама дестесін іске асыру, ОС UNIX Беркли Калифорния университетінде жасады.

UNIX операциялық жүйесінің кең таралуы IP хаттамасын және басқа стек хаттамаларын кеңінен таратуды қамтамасыз етті. Сонымен қатар, жаһандық Интернет ақпараттық желісі жұмыс істейді.

TCP / IP стекинде көп деңгейлі құрылым болса да, TCP / IP интероперабельділік моделінің пайда болуына дейін TCP / IP әзірленгендіктен, OSI үлгі деңгейлеріндегі TCP / IP стек деңгейі арасындағы сәйкестік салыстырмалы түрде ерікті болып табылады.

TCP / IP хаттамаларының құрылымы 2.3. кестеде көрсетілген. TCP / IP хаттамалары 4 деңгейге бөлінеді.

Стектің ең төмен деңгейі (IV деңгейі) - шлюз интерфейстерінің деңгейі OSI моделінің физикалық және арна қабатына сәйкес келеді. TCP / IP хаттамасында бұл деңгей реттеледі, бірақ физикалық және деректер сілтеме қабатының барлық танымал стандарттарына қолдау жасамайды: жергілікті арналар үшін Ethernet, TokenRing, FDDI, жеке хаттамалардың желілері жаһандық арналар үшін / PPP SLIP аналогты-қосұлы және арнайы жұмыс істейді.

Келесі деңгей (III деңгейі) - дестесін (OSI моделінің тұрғысынан) негізгі желілік қабатының IP хаттамасы ретінде пайдаланылады және т.б. жергілікті желілермен, аймақтық желілер, арнайы байланыс желілерін, түрліше пайдаланып дейтаграмма берумен айналысатын өзара деңгейі болып табылады. Ол көптеген жергілікті желілерден тұратын композиттік желілерде пакеттерді жіберу үшін жасалған. Сондықтан, IP хаттама кешенді топологиясы бар желілерде жақсы жұмыс істейді, оларда кіші жүйелердің бар болуын тиімді пайдаланады және

2.3 кесте. TCP/IP хаттамалар стеги

OSI үлгі деңгейі	TCP/IP хаттамасы	TCP/IP деңгей
7	FTP, TFTP. Gopher, Telnet, SMTP, SNMP...	I
6		
5	TCP, UDP	II
4		
3	IP, ICMP, RIP, OSPF	III
2	Уақытпен санаспайды, бірақ барлық стандарттар қолданады.	IV
1		

жылдамдығы жоғары байланыс желілерін өткізу қабілеттілігіндегі экономикалық шығындарды реттейді. IP хаттамасы - деректер сұлбасы болып табылады.

Сонымен қатар, өзара әрекеттесу деңгейі бағыттау кестелерін жасау мен өзгертуге қатысты барлық хаттамаларды қамтиды:

- RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First) — бағыттық ақпаратты жинау хаттамасы;
- ICMP (InternetControlMessageProtocol) — бағыттау мен шлюз арасындағы қателер туралы ақпарат алмасу хаттамасы, бастапқы жүйе және қабылдағыш жүйесі, яғни кері байланыс ұйымдастыру үшін қажет.

II деңгей ол негізгі деп аталады және желі бойынша ақпаратты тасымалдау функцияларын қамтамасыз етеді. Бұл деңгейде Трансмиссияны басқару хаттамасы (TCP) (TransmissionControlProtocol) және UDP пайдаланушы дерекқоры хаттамасы (UserDatagramProtocol) жұмыс істейді. TCP қашықтағы қолданба процестері арасында логикалық байланыс жасайды және қателер бар пакеттерді автоматты түрде қайта жіберу принципін қолданады. UDP қосымшалар пакеттерін датаграм әдісімен беруді, яғни логикалық байланыс орнатпай, қателерді тексеруді және жеткізуді растауды қамтамасыз етпейді. UDP - TCP жеңілдетілген нұсқасы болып табылады.

I деңгейге (қосымша) FTP файлын көшіру хаттамасы, Telnet терминал эмуляциясы хаттамасы, Интернет электрондық поштасында пайдаланылатын SMTP пошта хаттамасы, WWW және басқа да көптеген қашықтағы ақпаратты гипермәтіндік қатынау қызметтері сияқты кеңінен қолданылатын хаттамалар болып табылады. Олардың кейбіреулері туралы мұқият болайық.

SNMP (SimpleNetworkManagementProtocol) желіні басқаруды ұйымдастыру үшін қолданылады. Басқару ақпаратын беру хаттамалары сервер мен клиент арасындағы өзара іс-қимыл тәртібін анықтайды. Олар клиенттер мен серверлер арасында алмасатын хабар пішімдерін, сондай-ақ аттар мен мекенжай пішімдерін анықтайды. Тағы бір проблема бақыланатын деректермен байланысты болып келеді. Стандарттар шлюздерде қандай деректер сақталуы керек екендігін реттейді, бұл деректердің атаулары және осы атаулардың синтаксисі болып табылады.

FTP (FileTransferProtocol) — файлға қашықтан қатынауды іске асыратын файлды тасымалдау хаттамасы. FTP сенімді тасымалдауды қамтамасыз ету үшін TCP қосылу хаттамасын көлік ретінде пайдаланады. Файлдарды беруден басқа, FTP протоколы пайдаланушыға қашықтағы компьютермен өзара әрекеттесу мүмкіндігін береді, мысалы, каталог құрамындағыны басып шығару.

FTP пайдаланушыға сақталған деректердің түрін және пішімін көрсетуге, пайдаланушының аутентификациясын жүргізуге мүмкіндік береді.

TFTP (TrivialFileTransferProtocol) — TCP қарағанда қарапайым хаттама, UDP қосылымсыз хаттаманы пайдалана отырып, қарапайым файлды тасымалдау хаттамасы деп аталады.

2.4.4. IPX/SPX стегі

Бұл стек - Novell компаниясының бастапқы хаттамалық жинағы, ол 1980-ші жылдардың басында Netvarie желілік операциялық жүйесі үшін әзірленген. Бастапқыда стек шағын көлемдегі жергілікті желілерде жұмыс істеуге және шағын өңдеу қуатына ие болды.

Novell хаттамаларының отбасы және олардың ISO / OSI моделіне сәйкестігі 2.4кестеде келтірілген.

Физикалық және арналық деңгейлерде Novell желілері осы деңгейлердің барлық танымал хаттамаларын (Ethernet, TokenRing, FDDI және т.б.) пайдаланады.

Желінің қабатында Novell IPX хаттамасында, сондай-ақ RIP және NLSP бағыттау ақпараттық хаттамаларында жұмыс істейді. IPX - бұл Novell желілерінде адрестеу және бағыттау пакеттерін қарастыратын хаттама болып табылады. IPX бағыттау шешімдері пакеттің тақырыбындағы мекенжай өрістеріне, сондай-ақ бағыттау ақпарат алмасу хаттамаларынан алынған ақпаратқа негізделген. Мысалы, IPX жеткізілген ақпаратты пайдаланады

2.4 кесте. IPX/SPX хаттамалар стегі

OSI үлгісінің деңгейі	IPX/SPX хаттама
7	NCP, SAP
6	
5	
4	SPX
3	IPX, RIP, NLSP
2	Барлық танымал стандарттар қолданылады
1	

немесе мақсатты компьютерге немесе келесі бағыттауға пакеттер жіберу үшін RIP немесе NLSP хаттамасы (NetWareLinkStateProtocol) арқылы IPX хаттаманы есептеу ресурстарын экономикалық жағынан тұтынатын хабар алмасу әдісін ғана қолдайды. IPX үш функцияны қамтамасыз етеді деп айта аламыз:

- 1) Тапсырма мекен жайы;
- 2) Бағытын орнату;
- 3) Дейтаграмма сілтемесі.

Novell стекіндегі OSI үлгісінің тасымалдау қабаты логикалық байланыстары бар хабарларды жіберетін SPX хаттамаларына сәйкес келеді.

SPX (SequencedPacketExchange) — NetWare желілерінде пайдалануға арналған байланыс хаттамасы. SPX IPX хаттамасы пайдаланылатын пакеттер ағымында хабарламалардың кепілдендірілген жеткізуді және тапсырыс беруін қамтамасыз етеді.

Жоғарғы деңгейде (қолданба, өкіл және сессия) NCP және SAP хаттамалары жұмыс істейді.

NCP (NetWareCoreProtocol) NetWare серверінің және жұмыс станциясының қабатында өзара әрекеттесу хаттамасы болып табылады. Бұл қосымшаның қабат хаттамасы OSI үлгісінің жоғарғы қабаттарында клиент-сервердің сіулетін орындайды. Осы хаттаманың функцияларын пайдалану арқылы жұмыс станциясы серверге қосылады, серверлік каталогтарды жергілікті дискілік хаттарға көрсетеді, сервердің файлдық жүйесін сканерлейді, жойылған файлдарды көшіреді, олардың атрибуттарын өзгертеді, сондай-ақ жұмыс станциялары арасында желілік принтерді бөледі.

SAP (ServiceAdvertisingProtocol) — қызмет туралы хабарландыру хаттамасы тұжырымдамалық RIP хаттамасына ұқсас. RIP хаттамасы бағыттау туралы ақпаратты алмастыруға мүмкіндік беретін сияқты, SAP хаттамасы желілік құрылғыларға қол жетімді желі қызметтері туралы ақпаратпен алмасуға мүмкіндік береді.

IPX / SPX стекінің Novell-тің оны іске асыруға арналған қасиеті оның лицензиядан өтуі қажет екендігі ұзақ уақыт бойы оны NetWare желілеріне таратуды шектеді. Дегенмен, NetWare 4.0 нұсқасы шығарылған сәтте Novell корпоративтік желілерде жұмыс істеуге бейімделуге бағытталған хаттамалардағы маңызды өзгерістер енгізді және жалғастырады. Енді IPX / SPX стәгі NetWare-ге ғана емес, сонымен қатар бірнеше танымал желілік операциялық жүйелерде - SCOUNIX, SunSolaris, MicrosoftWindowsNT жүзеге асады.

2.4.5. NetBIOS/SMB стегі

Microsoft және IBM дербес компьютерлер үшін желіде бірлесіп жұмыс істеді, сондықтан NetBIOS / SMB хаттама стекі олардың бірлескен жобасы болып табылады. NetBIOS құралдары 1984 жылы IBM PCNetwork желілік бағдарламасы үшін IBM PCPC негізгі кіріс / шығыс жүйесінің (BIOS) стандартты функцияларын кеңейту ретінде пайда болды, одан кейін NetBIOS NetBEUI хаттамасымен ауыстырылды. Бағдарлама үйлесімділігін қамтамасыз ету үшін NetBIOS сақталды (2.5-кесте).

NetBIOS ашық жүйелердің өзара әрекеттесу моделінің үш деңгейінде жұмыс істейді: желі, көлік және сессия. NetBIOS IPX және SPX хаттамаларына қарағанда жоғары деңгейде қызмет көрсете алады, бірақ бағыттау мүмкіндігі жоқ. NetBIOS кадрлар алмасу хаттамасында желі сияқты нәрсе жоқ. Бұл NetBIOS хаттамасын ішкі желімен емес, жергілікті желі арқылы пайдалануды шектейді. NetBIOS деректерді детраграмманы алмасуды және байланыс орнатуды қолдайды.

SMI, қосымшаға сай және OSI моделінің өкілдік деңгейлеріне сәйкес, жұмыс станциясының сервермен өзара әрекеттесуін реттейді. SMB функциясы төрт операцияны қамтиды.

1. *Сессияны басқару.* Жұмыс станциясы мен файлдық сервердің желілік ресурстары арасында логикалық арнаны жасау және бұзу.

2. *Файлдық қол жетімділік.* Жұмыс станциясы «файлдық серверге» каталогтарды жасау және жою, файлдарды жасау, ашу және жабу, файлдарды оқу және жазу, файлдарды қайта атау және жою, файлдарды табу, файл атрибуттарын алу және тіркеу, жазбаларды бұғаттау.

2.5 кесте. NetBIOS/SMB хаттамалар стегі

OSI үлгі деңгейі	NetBIOS/SMB хаттама
7	SMB
6	
5	
4	NetBIOS, NetBEUI
3	
2	Барлық танымал стандарттар қолданады.
1	

3. *Қолтаңба қызметі.* Жұмыс бекеті қызметте басып шығару үшін файлдарды орнатып, басып шығару кезегі туралы ақпаратты ала алады.

4. *Хабарлама қызметі.* SMB төмендегі функциялармен хабарларды қарапайым беруді қолдайды:

- Қарапайым хабарлама жіберу ;
- Кең ауқымды хабарлама жіберу;
- Хабар блогының басын жіберу;
- Хабар блогының мәтінін жіберу;
- Хабар блогының соңын жіберу;
- Пайдаланушының атын жіберу;
- Қайта жіберуді болдырмау;
- Машинаның атауын алу.

NetBIOS ұсынған API функцияларын қолданатын көптеген қосымшалардың арқасында, көптеген желілік операциялық жүйелерде бұл функциялар көлік хаттамаларына интерфейс ретінде іске асырылады.

2.5. ЖЕЛІЛІК ӨЗЕКТЕРДІҢ МЕКЕН ЖАЙЫ

2.5.1. Желілік мекен жайлар туралы жалпы түсінік

Жергілікті және ғаламдық желілерге деректер жіберу үшін жіберуші құрылғы тағайындалған құрылғының мекен-жайын білуі керек. Сондықтан әр желілік құрылғыда бірегей мекенжайы бар (идентификатор):

- физикалық немесе аппараттық (MAC-адрес);
- логикалық немесе желілік (IP-адрес);
- белгілік.

Шын мәнінде, адресаттар құрылғылардың өздері үшін жасалмайды, бірақ олардың желілік интерфейстері үшін жұмыс жасайды. Естеріңізге сала кетейік, интерфейс - ақпарат алмасуға мүмкіндік беретін құралдар мен ережелер жиынтығы. Жалғыз желі түйінінде бірнеше желі интерфейстері болуы мүмкін.

Мекенжайлар сәйкестендіру үшін пайдаланылуы мүмкін:

- жеке интерфейстерге;
- оның топтары (топтық мекен жайлар) — деректер бірнеше түйінге бір уақытта жіберілуі мүмкін;

- барлық желі интерфейстері бірден (хабар тарату мекенжайлары) - деректер желідегі барлық түйіндерге жеткізілуі керек.

Мекен жайлар іске асырады:

- сандық және белгілік;
- аппараттық және желілік;
- жазық және иерархиялық.

Жазық сандық мекенжайдың мысалы жергілікті желідегі желілік интерфейстерді бірегей түрде анықтау үшін пайдаланылатын MAC мекенжайы (MediaAccessControl).

Иерархиялық адресстеу схемасы болған жағдайда, мекенжайлар жиынтығы мекен-жайдың алаңын дәйекті түрде тарылтып, соңында бөлек желілік интерфейсін анықтайтын кірістірілген кіші топтар түрінде ұйымдастырылады. Иерархиялық ұйымдастырылған мекен-жайлардың мысалы - адресаттың мекен-жайы тұрақты көрсетілетін әдеттегі пошта мекен-жайлары: ел, қала, көше, үй, пәтер. Иерархиялық сандық мекенжайлардың типтік өкілдері желі IP- және IPX-мекен-жайлары болып табылады. Олар екі деңгейлі иерархияны қолдайды, мекенжай ең жоғары бөлігіне (желі нөмірі) және ең төменгі (торап нөмірі) болып бөлінеді. Бұл бөлім желілердің арасында тек желінің нөміріне негізделген хабарламаларды жіберуге мүмкіндік береді, содан кейін хабарды қалаған желіңізге жеткізгеннен кейін түйіннің нөмірін пайдаланыңыз. Көшенің атын почта қызметкері қалаған қалаға жеткізілгеннен кейін ғана пайдаланады.

Белгілік мекенжайлар немесе атаулар адамдар есте сақтауға арналған және сондықтан әдетте семантикалық жүктемені қолданады. Белгілік мекен-жайлар шағын және үлкен желілерде де қолданыла алады. Ірі желілерде жұмыс істеу үшін белгілік атауы иерархиялық құрылымға ие болуы мүмкін

қазіргі заманғы желілерде тораптарға бағыттау үшін, әдетте, жоғарыда аталған схемалардың үшеуі бір мезгілде қолданылады, себебі компьютердің желілік интерфейсі бір уақытта бірнеше мекенжай атауларына ие болуы мүмкін. Адресстің қандай түрін қолдану ыңғайлы екеніне қарай, тиісті мекенжайлар жиынтығы қолданылады. Пайдаланушылар желіде жіберілген хабарларда автоматты түрде сандық нөмірлерге ауыстырылатын белгілік атаулармен компьютерлерге жүгінеді. Осы нөмірлерді пайдалану арқылы хабарлар бір желіден екіншісіне ауыстырылады да, хабар мақсатты желіге жеткізілгеннен кейін, сандық нөмірдің орнына компьютердің аппараттық мекенжайы пайдаланылады. Бүгінде бұл модель кішігірім автономды желілерге тән, мұнда әбден қажетсіз болып көрінетін көрінеді – бұл

желіні үлкен желіге қосқанда, операциялық жүйенің құрамын өзгертудің қажеті жоқ.

Мекен-жайларды бір түрден екіншісіне түрлендіру үшін, арнайы адресаттың мекен-жайын шешудің арнайы хаттамалары (ARP) пайдаланылады. Адресстерді шешу хаттамалары қарастырылатын түрлі типтердегі мекенжайлар арасындағы хат-хабарларды орнату мәселесі орталықтандырылған және таратылған құралдар арқылы шешілуі мүмкін. Орталықтандырылған көзқарас болған жағдайда, желіде бір немесе бірнеше компьютерлер (атау серверлері) бөлінеді, онда атаулардың әртүрлі типтеріне, мысалы, белгінің атаулары мен сандық нөмірлерге сәйкес келетін кесте сақталады. Барлық басқа компьютерлер деректерді белгілік атымен алмасатын компьютердің сандық нөмірін табу үшін атау серверіне қол жеткізеді.

Желі арқылы жіберілген деректердің түпкі мақсаты компьютерлер немесе маршрутизаторлар емес және бұл құрылғылар бағдарламасы бойынша іске қосылған - процестер. Сондықтан, тағайындалған мекен-жайы, құрылғының портын анықтау ақпаратпен бірге, процесс деректеріне жіберіледі, мекенжайын көрсетілуі тиіс. Бұл деректер тағайындалған мекен-жайда көрсетілген желілік интерфейске жеткенде, компьютерлік бағдарламалық қамтамасыз ету оларды тиісті процедураға бағыттауы керек. Ол, мекен-жайындағы процесі бүкіл желі ішінде бірегей оны көрсетуге жоқ екенін анық компьютер ішінде оның бірегейлігін қамтамасыз ету үшін жеткілікті болып табылады. Процесс мекенжайларының мысалы TCP / IP стекінде пайдаланылатын TCP және UDP порттарының нөмірлері болуы мүмкін.

2.5.2. Физикалық мекенжай

Физикалық мекенжайы — бұл жеке торап салынған және осы түйін қосылған технологиямен анықталған түйіннің жергілікті мекенжайы.

Екі бірдей физикалық мекенжай жоқ. Бұл мекен-жайлар жабдық өндірушілермен тағайындалады және бірегей мекенжайлар болып табылады, өйткені олар орталықтан басқарылады. Әрбір желілік адаптер, маршрутизатор немесе басқа желілік құрылғы желіге қосылғанына қарамастан, MAC мекенжайы деп аталатын бірегей физикалық мекенжайға ие.

MAC-мекенжай(MediaAccessControl— қоршаған ортаға қол жеткізуді бақылау) компьютерлік желілердің белсенді құрал-жабдықтарының әрқайсысына арналған бірегей идентификатор.

Ethernet негізіндегі желілер сияқты кең ауқымды желілерде, MAC мекен-жайы әрбір торапты бірегейлендіруге және деректерді тек осы түйінге жеткізуге мүмкіндік береді. Осылайша, MAC мекенжайлары жоғары (желілік) деңгейдегі хаттамалармен пайдаланылатын OSI деректер байланысы деңгейіндегі желілердің негізін құрайды. Кері қолданбалы желі қабатының мекенжайлары, және арнайы хаттамалар (мысалы, Arp RARPv IPv4 желілері және желілерінде IPv6 негізделген NDPv) үшін MAC-мекенжайларын түрлендіру болып табылады.

MAC мекенжайы әдетте аппараттық құралдармен ғана пайдаланылады, сондықтан оны мүмкіндігінше қысқа етіп жасауға және екілік немесе он алтылық мән ретінде жазуға тырысады. MAC мекенжайлары әдетте өндіруші тарапынан жабдыққа салынады; олар сондай-ақ аппараттық мекен-жайы деп аталады. Осындай мекен-жайларды пайдалану қатаң шешім болып табылады - жабдықты ауыстырған кезде, мысалы, компьютердің желілік адаптері, компьютердің физикалық мекен-жайы өзгереді. MAC-мекен-аппараттық желілік интерфейс бекітілген, мекен-жайы, өйткені физикалық мекен-жайы бөлу әлемде еш қайталанбайтын болады. желілік жабдықтар Электр инженерлері институтының және IEEE (Institute of Electrical and Electronics Engineers) электроника мониторларына ауытқиды.

MAC мекенжайы - 48 биттің екілік саны. Қабылдауды жеңілдету және онымен жұмыс істеу үшін MAC мекенжайлары әдетте он алты санмен, мысалы, 6D-BB-AB-C7 түрінде жазылады. Бұл жағдайда бірінші алты сандар осы желілік құрылғының өндірушісін көрсетеді, ал қалған алты сан осы өндірушімен өндірілген құрылғыны анықтайды.

MAC мекен-жайына сүйене отырып, сіз желілік ресурстарға қатынауды басқара аласыз. Кәдімгі пайдаланушы, мысалы, MAC сүзгісін пайдаланатын провайдер Интернетке қол жетімділікті қамтамасыз етсе, MAC ақаулығы орын алуы мүмкін. Ең қарапайым мысал - компьютерді Интернетке қосқан кезде, содан кейін басқа компьютерді маршрутизатор арқылы жалғауға шешім қабылдады. Қосылған, бірақ Интернет жоқ. Провайдері сізге IP-мекен-жайын және сіздің желі картасының MAC-мекен-жай бойынша тіркелген сол уақытта тағайындалған және MAC-мекен-жайы ғана жабдыққа қол жеткізуді береді фактісі болып табылады. Тиісінше, желі картасының MAC мекен-жайы ерекшеленеді, және бірегей бағыттаушы MAC-мекен-жайы, ол провайдерінде тіркелген MAC-мекен-жайына сәйкес келмейді. Нәтижесінде, маршрутизатор Интернетті ала алмайды және сәйкесінше маршрутизаторға қосылған компьютерлер Интернетті іске асыра алмайды.

Арнайы резервтелген мекенжайлар бар, мысалы, жалғыз екілік бірліктерден тұратын трансляцияның мекен-жайы - FF-FF-FF-FF-FF-FF.

2.5.3. IP-мекенжай

Логикалық адресстердің мысалдары IP-стек хаттамасы TCP / IP пайдаланатын желі IP мекен-жайлары болып табылады. IP желілік деңгейге қатысты екенін ескере отырып, IP мекенжайлары жиі желі мекенжайлары деп аталады.

IP-мекенжай (InternetProtocolAddress) — IP-хаттамасына салынған компьютер желісіндегі түйіннің бірегей желілік мекенжайы. Интернет жаһандық мекенжайдың бірегейлігін талап етеді; Жергілікті желіде жұмыс жасаса, желі ішіндегі бірегей мекенжай қажет.

Жергілікті IP-мекенжай жергілікті желідегі белгілі ресурстарға қол жеткізу үшін құрылғыға тағайындалады.

Сыртқы IP-мекенжайы — бұл IP-адрес, оның бірегейлігі жергілікті желіде емес, бүкіл Интернет желісінде анықталады.

IP-мекенжайлар IP-мекенжайларының негізгі түрі болып табылады, оның негізінде желілік қабаттағы IP пакеттері деп аталатын хабарлар жіберіледі. IP мекенжайы - екілік сан. Хаттама нұсқасын IPv4 IP-мекенжайы 4 байт немесе 4 294 967 296 (2³²) ықтимал бірегей мекенжайлары шектеу мекенжай кеңістігін 32 бит ұзындығы бар. IPv4 (InternetProtocolversion4) IP хаттамасының төртінші нұсқасы, кеңінен қолданылатын бірінші нұсқасы. 6-шы нұсқада IP-адресі 128-бит болып табылады. Pv6 (InternetProtocolversion 6) - Интернет пайдаланған кезде алдыңғы нұсқасы (IPv4) алдында тұрған проблемаларды шешу үшін арналған IP хаттамасының жаңа нұсқасы болып табылады. Қазіргі уақытта орнына 32. 128-биттік мекенжай ұзындығы пайдалану арқылы, IPv6 өзінде бірнеше мың желілерінде пайдаланылады бүкіл әлемде (астам 14,000 желілер 2013 жылдың күзінде), бірақ әлі IPv4 ретінде Интернет осындай кең таралуына алған жоқ.

IP-мекен-жайын (IPv4) жазу үшін ыңғайлы нысаны кезеңдер (мысалы, 192.168.0.1) бөлінген төрт ондық сандар (0 255 дейін) түрінде енгізу болып табылады. Бұл мекенжай ұсынудың дәстүрлі ондық түрі. Бұл жағдайда төрт ондық санның әрқайсысы сегіз битпен екілік жүйеге жазылуы мүмкін. Бұл жағдайда, ең көп саны сегіз екілік бірлік тұрады болады - 11111111, ондық санау жүйесінде тең. Осылайша 255 тең, әрбір

ондық тілде берілген IP мекенжайының төрт саны 255-тен аспауы керек.

IP адресі екі бөліктен тұрады: желінің мекенжайы және түйін нөмірі.

Желі мекенжайы (желі идентификаторы) — Бұл TCP / IP протоколын пайдаланып, үлкенірек интеграцияланған желідегі (желілер желісі) бір желі сегменті. Бір желіге қосылған барлық жүйелердің IP мекенжайлары бірдей желі идентификаторына ие болады.

Өзекше нөмірі (идентификатор, өзек мекенжайы) Әр желінің ішінде TCP / IP түйінін (жұмыс бекеті, сервер, маршрутизатор немесе басқа TCP / IP құрылғысы) идентификациялайды. Түйін идентификаторы жүйені ол қосылған желі сегментінде бірегей түрде анықтайды. IP-адрес желідегі жеке түйінді емес, жеке желі интерфейсін анықтайтынын ұмытпаңыз. Торап көптеген желілерді енгізе алады, бұл жағдайда оған бірнеше IP адресі, онымен байланысты желілер саны керек. Мысалға, маршрутизатор бірден бірнеше желілерге кіреді. Сондықтан маршрутизатордағы әрбір порт өзінің жеке IP-мекенжайына ие. Соңғы түйін бірнеше IP желілерінің бір бөлігі болуы мүмкін. Бұл жағдайда, компьютерде желілік қосылыстар санына қарай бірнеше IP-адрес болуы керек.

IP-адресстердің бес саны бар, олар желі нөмірі мен торап нөмірін сәйкестендіруге тағайындалған биттер саны бойынша ерекшеленеді (2.4-сурет).

Аймақтың ірі желілерінде немесе ел ауқымында А класындағы IP мекенжайлары пайдаланылады, мұндай желілердің саны шектеулі. В сыныптарын



2.4 сурет. IP-мекенжай құрылымы

орташа мөлшері бар және әдетте ірі компаниялар пайдаланылады. С класы мекенжайлары шағын түйіндерде шағын тораптарда қолданылады. D сыныптары компьютерлердің топтарына кіру үшін қолданылады. E сыныптары болашақта пайдалану үшін сақталған. Адресінің бірінші байтының мәні бойынша, IP мекенжайының желілерінің қандай класына жататындығын анықтауға болады.

A тобында мекенжайы 0-ден бастап, желінің нөмірі бір байтты, ал қалған 3 байт желідегі торап нөмірі ретінде түсіндіріледі. A кластарында желілер 1-ден 126-ға дейінгі сандарда (сан 0 пайдаланылмайды және 127 нөмірі арнайы мақсаттар үшін сақталған). A класс желісіндегі түйіндер саны 224-ке, яғни 16 777 216 түйінге дейін жетуі мүмкін.

B кластарында желінің нөмірі мен түйін нөмірі, яғни 2 байт, 16 бит тағайындалады. Осылайша, B сынып желісіндегі түйіндердің ең көп саны - 216.

Егер мекенжай 110 ретпен басталса, онда бұл C сынып желісі. Желі нөмірі 24 бит бөлініп, түйін нөмірі - 8 бит. Бұл желілердің ең кең тарағы, олардағы түйіндер саны 28 түйінмен шектелген.

D классының мекен-жайы 1110 ретпен басталады. Бұл арнайы мультикастаулық мекенжай. Мұндай мекен-жайы бар пакет осы мекен-жай тағайындалған барлық түйіндерді алуға тиіс.

Егер адрес 11110 қатарынан басталса, онда бұл мекен жай E класында жұмыс жасайды. Бұл класс мекенжайлары IP мекенжайлық схемасына болашақ толықтырулар үшін сақталған.

2.6 кестеде желінің әрбір санатына сәйкес келетін торап сандарының ауқымын және түйіндердің максималды санын көрсетеді.

2.6 кесте. Түрлі сыныптық IP-мекенжайлардың сипаттамасы

Сыныбы	IP-мекенжайдың бірінші биттері	Желінің ең аз нөмірі	Желінің ең көп нөмірі	Әр желідегі максималды өзектер саны
A	0	1.0.0.0	126.0.0.0	2 ²⁴
B	10	128.0.0.0	191.255.0.0	2 ¹⁶
C	110	192.0.0.0	223.255.255.0	2 ⁸
D	1110	224.0.0.0	239.255.255.255	Топтық мекенжай
E	11110	240.0.0.0	255.255.255.255	Резерв

Арнайы қорғалған IP-мекенжайлар бар. Оларға мыналар жатады:

- Екі жақты нөлден тұратын мекен-жай. Мұндай IP мекенжайын қамтитын пакетті жасаған бірдей желілік түйінді білдіреді;
- Желіде барлық нөлдері бар мекен-жай. Осы желі мекен-жайы бар пакет жинақталатын компьютерге бірдей желіге, көрсетеді;
- Торап нөмірі барлық нөлдерден тұратын мекен-жайының пакетін жинақталатын түйіннің бірдей желіге жататын түйіндерін білдіреді;
- Мекен-жай екілік соңында тұрады. Бұл мекен-жайы пакеттік көзі ретінде сол желідегі барлық түйіндерде екенін көрсетеді. Бұл жағдайда, пакеттік желіден тыс баруға болады, сондықтан бұл ақпараттық бюллетень шектеулі таратылым деп аталады;
- Мекен-жайы 127.0.0.1 бар пакет желісіне берілетін және тек қабылдануы ретінде, хаттама бумасының жоғарғы қабатына қайтармайды. 127 бастап мекенжайлары, бір түйін ішінде бағдарламалық қамтамасыз ету және желілік өзара іс-қимыл процестерін сынау үшін пайдаланылады. Сондықтан, желі 127 бастап, компьютерлер үшін-мекенжайларына тыйым бөледі.

IP-адресстерді сыныптарға бөлу өте қиын.

Адресстеу жүйесі әрқашан ұтымды емес. Адресстік жүйені икемді ету үшін IP маска маскалары пайдаланылады. TCP / IP желілерін терминология ішкі желі маскасы немесе желі маска, желі мекен-жайы және желіге түйіннің мекен-жайы жатады, ол желілік түйіннің IP-мекен-жайы бөлігі анықтайды сәл маска деп аталады. Бұл әрекетті орындау үшін, бір желі бір-бірімен полигондардың бар кесіп ішіне мекен-жай кеңістігін ортақ, бірнеше қосалқы желілері бөлінеді. Маска, IP-адреснің «маска» бөлігіндегідей. Екілік жазбасында IP мекенжайындағы желі нөмірі ретінде түсіндірілуі керек биттерде бірліктер болады. Маска битінің мәні бір болғанда, түйіндерді адресстеуге тыйым салынады; мұндағы мән нөлге тең болады. «Маскирование» LSB үшін ШОБ алынған. Стандартты желілік сыныптар үшін, маска келесі мәндерге ие:

- А сынып— 11111111.00000000.00000000.00000000 (255.0.0.0);
- В сынып— 11111111. 11111111.00000000.00000000 (255. 255.0.0);
- С сынып— 11111111. 11111111. 11111111.00000000 (255. 255. 255.0).

Маски механизмі IP-маршруттауда кеңінен таратылады, ал маскалар әртүрлі мақсаттарда қолданыла алады.

Маскиді қолдану арқылы әкімші өзінің желісіндегі қызмет провайдерлерінің қосымша желілік нөмірлерін талап етпей құра алады. Қызмет жеткізушілері маска механизмін «префикс» деп аталатын бірнеше желілердің мекенжай кеңістігін біріктіру үшін пайдаланады. Осылайша, маршруттау кестелерінің көлемі азаяды, демек, маршрутизаторлардың өнімділігі артады.

Егер желі интернеттің бір бөлігі болса, желі нөмірлері орталықтандырылған түрде тағайындалады. Егер желі оқшауланғанда жұмыс істесе, желі нөмірлері еркін түрде тағайындалады.

2.5.4. DHCP өзектерін динамикалық жөндеу хаттамалары

Желі бойынша ақпарат алмасу үшін әр түйіннің желілік интерфейсіне өз IP мекенжайы қажет. IP мекенжайлары интерфейсстерге қолмен немесе автоматты түрде тағайындалуы мүмкін.

Қолмен мекенжай тағайындау әдетте желі интерфейсіннің конфигурациясы кезінде жасалады, ал жаңа тағайындалған мекенжайлар бұрын қолданылған IP адресстерін желіде қайталауға тура келмейді. Әрине, тіпті шағын желі өлшемдерімен де, бұл параметр утсыз іс рәсім болып табылады және кейбір қиындықтарды тудыруы мүмкін.

IP-адресстердің автоматты түрде тағайындалуы әрбір желі интерфейсі үшін мекен-жайларды анықтау процесін жеңілдетеді және әкімшінің желі әрекеттерін күнделікті жұмыста теңшеу үшін жауапты етеді. IP мекенжайларын автоматты түрде тағайындау үшін DHCP динамикалық хостты теңшеу протоколын қолданыңыз.

DHCP (DynamicHostConfigurationProtocol) — компьютерлері автоматты түрде TCP / IP желісінде жұмыс істеу үшін қажетті IP-мекен-жайы және басқа да параметрлерін алуға мүмкіндік беретін желілік хаттама. Бұл хаттама «клиент-сервер» үлгісінде жұмыс істейді. Ол DHCPі автоматты түрде желілік құрылғы конфигурация кезеңіне клиенттік компьютерде конфигурациялау үшін сіз оған қалаған параметрлерді алушы деп аталатын серверге сілтеме жасап, желінің әкімшісі компьютерлер арасында таратылатын мекенжайлар ауқымын көрсете алады. Бұл желілік компьютерлердің қолмен конфигурациясын болдырмайды және қателер санын азайтады. DHCP хаттамасы көптеген TCP / IP желілерінде пайдаланылады.

DNCP жұмыс станциясының TCP / IP стек параметрлерін автоматты түрде конфигурациялауға арналған. Операциялық жүйені жүктеу кезінде жұмыс станциясы DNCP сервері осы жұмыс станциясының IP мекенжайын қамтитын ақпаратты кері жіберетін конфигурация параметрлері үшін таратылатын сұрауды жібереді, сондай-ақ желінің өзара әрекеттесуін теңшеу үшін қажет басқа ақпарат болып табылады. Бұл DNCP клиенті, яғни тарату сұрауын жіберген жұмыс станциясы және DNCP сервері бірдей жергілікті желіде екенін білдіреді.

Автоматты режимде интерфейстерге мекен-жайларды бөлудің екі жолы бар - статикалық, динамикалық.

Статикалық режимде DNCP сервері жұмыс станциясының сұрауы үшін қолма-қол ақпараттар ауқымынан таңдап алынған кездейсоқ IP мекенжайын жібереді. Мекенжай ауқымы DNCP сервері конфигурацияланған кезде орнатылады. Жұмыс станциясы үшін таңдаған мекен-жайы оған шексіз уақыт беріледі, кейіннен серверге қоңырау шалып, станция сол IP-мекен-жайға ие болады.

Динамикалық бөлу кезінде, компьютерге тұрақты түрде емес, белгілі бір мерзімге беріледі. Бұл басқа компьютерге тағайындалатын осы IP мекенжайын қайта пайдалануға мүмкіндік береді. Мұндай бөлу бірдей мекен-жайы бірнеше интерфейстерді пайдаланылуы мүмкін болып, сондай-ақ байланыста, әдетте бір уақытта жұмыс істейді, бұл шын мәнінде емес, жұмыс станциясы желісінде тіркелген барлық, IP-мекенжайларды сақтау үшін мүмкін, бос тұрып қалу арқылы нақты қажеттіліктерімен емес, оларды бөлу кезіндегі түйіндер болып табылады. IP-адресін әкімші иелігінде IP-адресстердің қол жетімді санынан асатын түйіндер саны бар болып саналады.

DHTSP TCP / IPn қарамастан қолданылатын әдісті қолдану әрбір жеке желі торабы стек және икемді орталықтандырылған желісі әкімшілік саясатын жүзеге асыруға болады.

IP-адресімен жергілікті (физикалық) *мекен-жайды анықтау үшін ARP* (AddressResolutionProtocol) мекенжайын шешу пәрмені қолданылады. ARP хаттамасы байланыс қабатындағы хаттама желісінде жұмыс істейтініне байланысты түрлі жолдармен жұмыс істейді. Жергілікті желілерде қалаған мекен-жайды анықтау үшін ARP хабар тарату сұрауларын таратады. Мекен жайы ажыратымдылық хаттамасы тораптың мекен-жайын көрсететін сұрау жасайды, ол үшін тораптың тиісті физикалық мекен-жайы анықталуы керек, сілтеме деңгейінің хаттамасының,

ол осы желіде пайдаланылады және алынған кадрларды таратады.

Мұндай сұрауды алатын желілік түйін сұрауда көрсетілген желі мекенжайын желі мекенжайымен салыстырады. Мекенжайларға сәйкес болған жағдайда түйін мекен-жайын (физикалық және желілік) қамтитын жауапты жұмысты жасайды және жауапты жіберілген түпнұсқалық ARP сұрауының жіберушісіне жібереді. Егер қалаған IP-адресі жергілікті желіде болмаса және жергілікті желі шлюзмен байланыспаса, онда сұрау жіберілмейді. Мұндай пакеттер «жойылған сұраныстың мерзімі шектелген» деген хабармен жойылады. Сонымен қатар, қолданба деңгейіндегі модульдер желідегі физикалық сәтсіздіктерді және қателерді жоюды ажырата алмайды. ARP сұраулары мен ARP жауаптары бар пакеттер бірдей пішімге ие.

Кері міндеттерді шешу үшін, яғни, IP мекенжайы белгілі физикалық мекен-жайдан анықталады, *RARP* (ReverseAddressResolutionProtocol) *хаттамасы* пайдаланылады. Кері мекен-жайларды шешу хаттамасын пайдалану қажеттілігі, әдетте, дисксіз жұмыс станцияларын пайдалану арқылы, бір серверден операциялық жүйені жүктеу арқылы анықталады.

2.5.5. Домендік атаулар жүйесі

Компьютерлерді анықтау үшін, TCP / IP желілеріндегі аппараттық және бағдарламалық жасақтама IP мекенжайын пайдаланады, сонда бағдарлама қай хостқа қол жеткізу керек екенін дұрыс түсінеді. Мысалы, <http://81.19.70.3> командасы «Rambler» интернет-порталының өзіндік бетін ашады. Бірақ пайдаланушылар символдық компьютер атауларымен жұмыс істегісі келеді және, мысалы, браузердің мекенжай жолында rambler.ru сайтына кіру ыңғайлы. Сандардың жиынтығынан гөрі, адамға мағыналы есімдерді есте сақтау оңайырақ. Қазіргі уақытта көптеген W-мекенжайлары бар, сондықтан сіз олардың кем дегенде бір бөлігін еске түсіре аласыз. Сондықтан, сандық мекенжайлар схемаларына қосымша, мекенжайлардың символдық көрінісі де пайдаланылады. Сондықтан, TCP / IP желілерінде символдық хост аттары және символикалық атаулар мен IP мекенжайлары арасындағы сәйкестікті орнату механизмі болуы керек.

Бұрын символдық адресітеу желілік түйіннің физикалық мекен-жайы мен оның символдық мекен-жайы арасындағы хат-хабар кестесін сақтайтын операциялық жүйелер арқылы қамтамасыз етілді. Бұл жүйелер бастапқыда жергілікті шағын желілерде жұмыс істеуге

Өзек аттары сызықтық құрылымынан болды және бөлікке бөлінбейді. Кейбір белгілік атауы тиісті түйіннің физикалық мекенжайын анықтау мақсатында, сауалнама көп мекенжайлық сұрауының механизмі арқылы LAN бойынша барлық тораптар өткізілді. Бірақ картографиялау үшін бұл әдіс тек желі бөліктеріне бөлінген емес, шағын желісінде жақсы жұмыс істейді. Ірі желілерде, сондай-ақ бірнеше желі қамтитын желілерде, иерархиялық шешуге жүйесін неғұрлым тиімді пайдалану және, тиісінше, бірнеше жалпақ атаулары мүмкін бір алгоритімі жасауға емес, сондай-ақ, бір-біріне бөлікке «кірістірілген» тұратын мекен-жайлары, бірегей атау береді.

Атау компоненттерінің еркін санының пайдалануға мүмкіндік беретін ағаш құрылымы бар TCP / IP стек пайдаланылатын домен атауы жүйесі болып табылады.

DNS (DomainNameSystem— домен атауларының жүйесі) — бұл домен атауларын IP мекенжайларына және керісінше TCP / IP желілеріне түрлендіруге мүмкіндік беретін жүйе.

Домен(domain) — елге, ұйымға немесе басқа мақсаттарға бөлінген Интернеттің домендік атаулар (DNS) жүйесінде белгілі бір аймақ.

Домендік атаулар нүктелер арқылы бір-бірінен бөлінген бірнеше бөліктен тұрады, мысалы mail.rambler.ru. Бұл бөліктердің әрқайсысы да домен деп аталады.

Доменде сол жақта, иерархиядағы ең төмен деңгейлі доменге енетін түйін атауы жазылады және оң жақта ең жоғарғы иерархиялық деңгейге ие домен болып табылады. Сондықтан ең оңтайлы домен жоғары деңгейлі немесе бірінші деңгейлі домен деп аталады.

Нүкте бойынша бөлінген сол жақта орналасқан келесі домен - бірінші деңгейдегі доменге қатысты еншілес домен, яғни оның құрамдас бөлігі ретінде енеді. Бұл домен екінші деңгейлі домен деп аталады.

Екінші деңгейлі доменнің балалары болып табылатын домендер үшінші деңгейлі домендер деп аталады.

Мысалы, mail.rambler.com мекенжайында «ru» домені жоғарғы (бірінші) деңгейдегі домен болып табылады, екінші деңгей домені «rambler», «mail» сөзі үшінші деңгей - хост атауы (2.5-сурет).

Хост - ақпараттың белгілі бір аумағын, жеке пайдалануға арналған барлық файл түрлерін, сондай-ақ барлық Интернетті пайдаланушылар үшін ашық қатынауды орналастыруға мүмкіндік беретін сервер ретінде түсініледі.

mail.rambler.ru, к _

3денгей

2-денгей жоғарғы

денгей

2.5 сурет. Домен атауларының компоненттері

Домен атауларының иерархиясы файлдық жүйеде қабылданған файл аттарының иерархиясына ұқсас. Бірінші деңгейдегі домендердің атаулары халықаралық стандарттарға сәйкес орталықтандырылған түрде тағайындалады. Бірінші деңгейдегі домендік атаулар ұйымдардың елдерін немесе түрлерін білдіреді және әдетте екі немесе үш әріптік қысқартуларды көрсетеді (2.7-кесте)).

2.7 кесте. Бірінші деңгей домендерінің мысалы

Домен		Тағайындалуы
Жалпы (ұйым түрлері)	com	Коммерциялық
	edu	Оқу
	gov	Үкіметтік
	int	Халықаралық
	mil	Әскери
	info	Ақпараттық
	net	Желілік
	org	Коммерциялық емес
Аймақтық (елдер мен аймақтар)	ru	Ресей Федерациясы
	ua	Украина
	us	АҚШ
	jp	Жапония
	de	Германия
	gb	Ұлыбритания
	au	Австралия
	za	Оңтүстік Африка

Екінші деңгейлі домен әдетте осы доменді қолданатын корпоративтік желі немесе хост компьютері болып табылатын ұйымның бүркеншік аты болып табылады. Үшінші және кейінгі деңгейдегі домендер екінші деңгейлі домендердің бөлігі болып табылады және практикада әдетте басқа ұйымдарға немесе жеке тұлғаларға сатылатын немесе басқа ақысыз берілген кейбір субнокттерді немесе субхосттарды көрсетеді. Мұндай хосттар жиі Интернетті пайдаланушылардың үй беттерін қабылдайды. Егер бір домен басқа доменнің құрамдас бөлігі болса, онда мұндай поддомен деп аталуы мүмкін немесе субдомен(subdomain).

Атаудың бөлімдерге бөлінуі оның иерархиясының деңгейінде әртүрлі адамдар немесе ұйымдар арасында бірегей аттарды беру үшін әкімшілік жауапкершілікті бөлісуге мүмкіндік береді. Бұл ұйымдардың өзара келісімсіз бірегей атауды қалыптастыру мәселесін шешуге мүмкіндік береді. Әлбетте, иерархияға жоғарғы деңгейдегі аттарды беру үшін жауапты бір ұйым болуы керек.

DNS - таратылған дерекқор. CSN дерекқорында *домендік атаулар кеңістігі* деп аталатын ағаш құрылымы бар (2.6-сурет). Домендік атау осы дерекқорда домен түйініне сәйкес бөліктерді бөліп тұратын атаудағы нүктелермен бірге ата-ана доменіне қатысты өз орнын анықтайды.

Түбірлік атаудың (root'sname) нөлдік белгісі бар және бір нүкте («.») Арқылы белгіленеді. Ағаштың әрбір түйін ортақ дерекқордың немесе доменнің бөлігін білдіреді. Әрбір домен одан әрі бөлімшелерге бөлінуі мүмкін - субдомены - ата-аналық тораптарының ұрпақтары (parentnodes). Әр домен

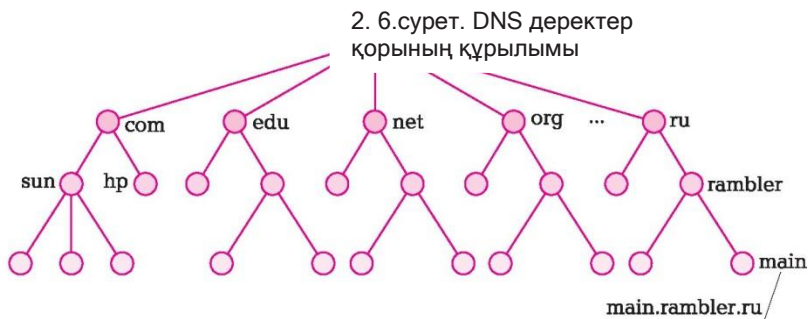
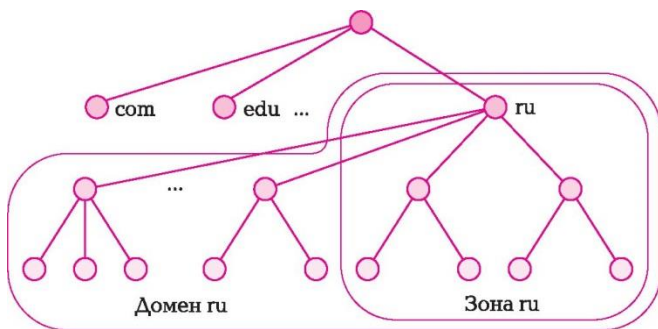


Рис. 2.6. Структура базы данных DNS

оның негізгі доменіне қатысты орналасуын анықтайтын белгісі бар. FQDN символы бөлінген түбірлік доменнің, бастап тізбегі тегтер болып табылады «.», Мысалы main.rambler.ru. DNS қызметінің көмегімен домен атауларының желі мекенжайларына сәйкестігін белгілеу орталықтандырылған. DNS серверімен басқарылатын домен атауларының кеңістігі *аймақ* деп аталады (2.7-сурет).

Арт-сәулет пайдалану туралы операция DNS қызметіне негізделі көрсету принципі DNS-серверлерді және DNS- клиенттерді анықтайтын «клиент-сервер», сондай-ақ қызмет болып табылады. DNS-сервер (атау сервері) - домендік аттар кеңістігі туралы ақпаратты сақтайтын бағдарламалық жасақтама екені анық. DNS клиенттері IP мекенжайындағы домен атауын көрсету үшін сұраулары бар серверлерге қол жеткізеді. Кез келген түйін кез келген желі түйінінің қажетті IP мекенжайы туралы ақпаратты ала алады. Ол іздеу орындау түйінінің және қалаған түйінге ортақ доменде орналасқан серверге жеткенше осы торап әріптері үшін, жоғарыда жұмыстары бойынша орналасқан барлық DNS-серверлерді қарастырады. Қалаған түйіні бар домен табылған, ал одан әрі дәйекті, доменнің иерархиясының төмен орналасқан серверлер атына. іздеу кәшінде олар туралы ақпарат, егер, DNS-серверлерінің жұмыстарына жақын басталады және бүгінгі күнге дейін, сервер DNS-серверлерді сұрау мүмкін емес болып табылады.

Жоғарыда айтылғандай, DNS-ді дамытудың негізгі мақсаттарының бірі орталықтандырылмаған басқарумен бөлінген деректер базасын құру болды. Бұл өкілеттіктері арқылы қол жеткізіледі. Доменді басқаратын ұйым оны қосалқы домендерге бөліп, оларды басқару құқығын береді.



2.7 сурет. «Ш» доменіндегі «ш» аймағының мысалы

Басқа ұйымдарға қосалқы домендер беріледі. Бұл басқару құқығын беретін өкілетті органның осы субдоменнің барлық деректерін сақтауға және басқаруға жауапты екенін білдіреді. Ол деректерді еркін өзгерте алады, субдоменді төменгі деңгейдегі субдомендерге бөліп, оларды басқа ұйымдарға жібере алады. Ата-доменде осы ішкі домендердің ресурстарына сұрауларды қайта бағыттау үшін, оның қосалқы домендерінің деректер көздеріне ғана (әдетте атын серверлерге нұсқайды) қатысты көрсеткіштер бар.

Бір IP-адресіте бір компьютерде (виртуалды хостинг) көптеген веб-сайттарды қолдауға мүмкіндік беретін көптеген атаулар болуы мүмкін. Кері дұрыс - жалғыз атауды жүктемені теңестіруді жасауға мүмкіндік беретін көптеген IP мекенжайларымен салыстыруға болады.

Жүйенің тұрақтылығын арттыру үшін бірдей ақпаратты қамтитын көптеген серверлер пайдаланады және хаттамаларда әртүрлі серверлерде орналасқан ақпаратты синхрондауды қолдайтын құралдар бар.

DNS сипаттамаларын төмендегідей анықтай аласыз.

1. *Ақпаратты сақтауды тарату.* Әрбір торап торабы тек жауапкершілік аймағына кіретін деректерді және (мүмкін) түбірлік DNS серверлерінің мекен-жайларын сақтауы керек.

2. *Иерархиялық құрылым,* онда барлық түйіндер ағашқа біріктіріледі және әрбір түйін бағынышты түйіндердің жұмысын дербес анықтай алады немесе оларды басқа түйіндерге жібере алады.

3. *Қауіпсіздік және артықшылық.* Тораптарының (аймақтардың) сақталуы мен қызмет көрсетуі үшін әдетте (және әдетте) бірнеше физикалық және логикалық түрде бөлінген серверлер бар, бұл деректердің сақталуын қамтамасыз етеді және тіпті түйіндердің біреуі істен шыққан жағдайда жұмыс жалғасады.

Барлық DNS шешімдері қорғауды талап етеді, себебі хакерлік DNS серверіне шабуыл жасалса, пайдаланушылар тіпті оны іске асырмай-ақ, тұзаққа түседі. Шабуылдардың осы кіші түрлерімен келесі қауіп-қатерлер байланысты:

- DNS шабуылдары нәтижесінде пайдаланушы дұрыс бетке түсіп кету қаупін тудырады. Веб-сайт адресін енгізген кезде шабуылданған DNS сұрауды алдыңғы беттерге бағыттайды;
- Пайдаланушының жалған IP мекенжайына ауысуының нәтижесінде, хакер өзінің жеке ақпаратына қол жеткізе алады. Бұл жағдайда, пайдаланушы ақпараттың құпиялығын қамтамасыз етпейтініне күмәнданбайды;

■ DDoS-шабуылдар, DNS серверінің жұмыс істемеуіне алып келеді. пайдаланушы DNS-серверге алу мүмкін емес болады, оның браузер сайттың мекен-жайына сәйкес келетін IP-мекен-жайын таба алмай-ақ, оған қажетті бетінде қол жетімді болмайды. DNS серверлеріне DDoS шабуылдары DNS серверінің жұмысының төмен болуына байланысты және байланыс арнасының ені жеткіліксіз болғандықтан жүзеге асырылуы мүмкін.

Домендік атауды алу тәртібі доменді тіркеу деп аталады. Ол DNS дерекқорында домен әкімшісіне нұсқайтын жазбаларды жасаудан тұрады. Тіркеу тәртібі мен талаптары таңдалған домен аймағына байланысты. Доменді тіркеуді таңдалған домен аймағының ережелері рұқсат етсе, тіркеуші ұйым және жеке тұлға жүзеге асыра алады.

2.5.6. URL — әмбебап ресурстар идентификаторы

URL (UniformResourceLocator) — Бұл Интернетте ресурстың мекен-жайын жазудың стандартталған тәсілі. URL-ті 1990 жылы Тим Бернерс-Ли Женевадағы Ядролық зерттеулер жөніндегі Еуропалық Кеңесте ойлап тапты. URL Интернеттегі іргелі инновацияға айналды. Бастапқыда URL мекенжайы World Wide Web-те ресурстарды (көбінесе файлдар) орналасқан жерін көрсетуге арналған.

Енді URL барлық дерлік интернет-ресурстардың мекен-жайларын көрсету үшін пайдаланылады және жалпы URI ресурсының сәйкестендіру жүйесінің бөлігі ретінде орналастырылады (UniformResourceIdentifier - дерексіз немесе физикалық ресурсты анықтайтын таңбалар тізбегі). *URL* стандарттары IETF және оның аффилиирленген тұлғалары арқылы реттеледі.

URL бұл бастапқыда желі ресурстарының жері табиғи нұсқауларды арттыру жүйесі ретінде жобаланған. URL мекенжайының келесі дәстүрлі формасы бар:

<кесте>://<логин>:<құпия сөз>@<хост>:<порт>/<URL-жол>?<параметрлері>#<якорь>

Бұл жазбадағы:

кесте — ресурсқа қолжетімділік схемасы; көбінесе желі хаттамаларын білдіреді;

логин — ресурсқа кіру үшін қолданылатын пайдаланушы аты;

Құпия сөз - көрсетілген пайдаланушының құпия сөзі; Хост - DNS жүйесінде хосттың толық тіркелген домендік атауы немесе хосттың IP мекенжайы кезеңдермен бөлінген ондық сандардың төрт тобы түрінде; сандар 0-ден 255 аралығындағы бүтін сандар; қосылу үшін хосттың порты;

URL-жол - ресурстың орны туралы ақпаратты көрсету, хаттамаға байланысты;

Параметрлер - серверге жіберілген параметрлермен сұрау жолы. Параметр сепараторы & белгісі;

Якорь - ашылған құжаттың кейбір бөлігіне (бөлігіне) сілтеме жасайтын «якорь» идентификаторы. Бет браузерде көрсетілген якорьге байланысты әр түрлі болуы мүмкін. Жалпы хаттамалар (протоколдар) URL қамтиды: ftp-FTP файлды тасымалдау хаттамасы; http - гипермәтінді тасымалдау хаттамасы HTTP; rtmp-RealTimeMessagingProtocol - деректерді ағынға арналған хаттама, ол көбінесе Интернеттегі веб-камералардан бейне және аудио ағынды ағынмен тасымалдау үшін пайдаланылады;

rtsp - нақты уақыттағы ағымдық хаттама; https - шифрлауды (әдетте SSL немесе TLS) пайдалану арқылы HTTP хаттамасының арнайы іске асуы; gopher-хаттама Gopher; mailto - электрондық пошта мекенжайы; news-news Usenet;

nntp - NNTP хаттамасы арқылы Usenet жаңалықтары; Telnet сеансына интерактивті телнет-байланыс; файл - жергілікті файлдың атауы; деректердің тікелей деректері (Деректер: URL); tel - көрсетілген телефонға қоңырау шалу.

Мұнда қарапайым URL-нің мысалы келтірілген:

http://academia-moscow.ru/about/elektronnyy_kontent/index.php#sel

Бұл жағдайда, жол HTTP сервері орнатылған машина доменінің мекен-жайын және серверлік ағаштың түбірінен индекс файлында жолды қамтиды. php HTTP кестесі, www- үшін бастысы идентификатор, машина мекенжайы, TCP порты, сервер каталогына жол, іздеу критерийі және жапсырма. Бұл құжаттарда пайдаланылатын URL мекенжайының ең кең таралған түрі, IP мекенжайларын машинаның мекенжайы ретінде пайдалануға рұқсат етіледі.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Деректер беру ортасы дегеніміз не?
2. Байланыс арнасының мүмкіндіктері қандай?
3. Өткізу қабілетін не анықтайды?
4. Қандай жүйе ашық деп аталады?
5. Желілік байланысқа көп деңгейлі көзқарас дегеніміз не?
6. Хаттама дегеніміз не?
7. OSI желісі моделінің мақсаты қандай?
8. 7-деңгейдегі OSI үлгісінде 1-3-ші деңгейдегі хаттамалар қандай?
9. 7 деңгейлі OSI үлгісіндегі 5-7 деңгейдегі хаттамалар қандай?
10. Трансмиссиялық ортамен байланыс қандай деңгейде болады?
11. Қандай деңгей желі арқылы өтеді?
12. Қателерді анықтау және түзету қандай деңгейде болады?
13. Жіберілген ақпаратты қажетті желілік пішінге беру тәртібі қандай деңгейде болады?
14. Физикалық деңгейдегі хаттамалардың мысалдарын келтіріңіз.
15. Сіз қандай Интернет-хаттамаларды білесіз?
16. Домендік атау жүйесі қандай қолданылады?
17. Неліктен Интернетте байланыс хаттамаларының ең перспективті стекі түр?
18. IPX / SPX стекінің ерекшеліктері қандай?
19. Желілік құрылғының физикалық мекенжайы қандай?
20. IP-адресі дегеніміз не?
21. Желілік құрылғының символикалық мекенжайы қандай? Бұл қалай калыптасады?
22. TCP / IP хаттама архитектурасындағы деңгейлерді тәртіпке келтіріңіз.
23. TCP хаттамасының қызметі қандай?
24. IP хаттамасының қызметі қандай?
25. Интернеттегі мекен-жай жүйесіне не кіреді?
26. Ethernet мекенжайының қандай құрылымы бар?
27. IP-адресінің құрылымы қандай?
28. IP мекенжайлары қандай?
29. Домен атауы сервері дегеніміз не?

ЖЕРГІЛІКТІ ЖЕЛІНІҢ БАЗАЛЫҚ ТЕХНОЛОГИЯСЫ

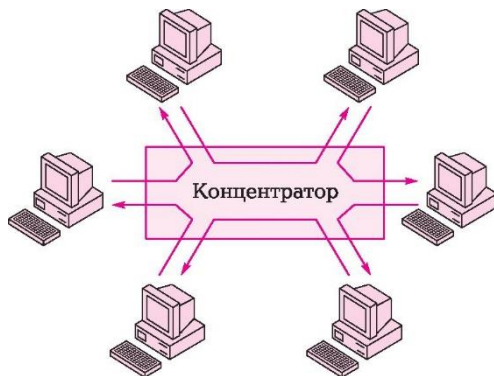
Бірінші жергілікті желілер пайда болған сәттен бастап, бірнеше жүздеген желілік технологиялар әзірленді, бірақ бірнеше желілер өте танымал болды, бұл ең алдымен танымал фирмалармен осы желілерді қолдауға және олардың ұйымдық қағидаларын стандарттаудың жоғары деңгейіне байланысты болды.

3.1. TOKENRING желілері

Token Ring желісін IBM компаниясы 1985 жылы ұсынды (бірінші нұсқасы 1980 жылы пайда болды). TokenRing-тің мақсаты IBM шығарған компьютерлердің барлық түрлерін желіге біріктіру болды. Token-Ring қазіргі уақытта IEEE802.5 халықаралық стандарты болып табылады. Бұл желі Ethernet мәртебесі бойынша бір деңгейге қояды.

Ethernet жабдығына қарағанда, TokenRing жабдығы әлдеқайда қымбат, өйткені ол айырбастаудың аса күрделі әдістерін пайдаланады, сондықтан Token-Ring желісі әлдеқайда таралған кезде (үлкен компьютерлерге мысалы, салдарынан) ірі айырбас бағамы мен шектеулі қолдың уақыты ішінде, оны қолдану ақталған болып табылады.

Трансмиссиялық ортаға кіруді бақылау станцияларының әдісі ретінде сақина топологиясындағы желіге арналған Token Ring әдісі (TokenRing), пайдаланылған. Ол сақина айналасында қозғалатын маркер деп аталатын арнайы үш байттық кадрды пайдаланады.



3.1 сурет. Token Ring негізіндегі желі: логикалық топология «сақина», физикалық - «жұлдыз»

Маркердің иесі меншік иесіне ортаға ақпарат беру құқығын береді. Егер токенді алатын түймеде ешқандай ақпарат жоқ болса, онда ол тек келесі соңғы станциясына өтеді. Әрбір станция маркерді белгілі бір уақытқа дейін ұстай алады, түрлі жұмыс станцияларына әртүрлі басымдықтарды тағайындауға болады.

Жалпы алғанда, TokenRing желілік топологиясы сақина топологиясына қарағанда жұлдыздың топологиясына ұқсас (3.1-сурет). Бір-бірімен байланыстырудың орнына сақинаны қалыптастырудың орнына, токенді сақиналық жұмыс станциялары концентраторға радиалды қосылды. Бірақ бірнеше хабарламалар болуы мүмкін, және бұл жағдайда хабарламалар арнайы қосқыштар арқылы сақинаға біріктіріледі. Егер сіз бір хабарламаны қолдансаңыз, онда біріктіруші коннекторлар тоқтауға болмайды.

Ауыстырушы орта ретінде талшықты немесе талшықты-оптикалық кабельдер қолданылады. Деректерді беру жылдамдығы 4 немесе 16 Мбит / с.

Token Ring - сенімді желі және кешігу болжанатын және желінің тұрақтылығы маңызды қосымшалар үшін өте қолайлы. Технология кез келген жерде жылдамдықтың маңызы бар, ақпараттың шынайы жеткізілуі сияқты жауапты өтініштер бар жерде кеңінен таралған. Бірақ қазір бұл желі Ethernet технологиясымен алмастырылды, оның сенімділігі Token Ring-тен кем емес және өнімділігінен әлдеқайда жоғары.

3.2. FDDI ТАЛШЫҚТЫ ОПТИКАЛЫҚ ИНТЕРФЕЙС

FiberDistributedDataInterface технологиясы (FDDI) ANSI комитеті 1980 жылы әзірледі. Бұл тасымалдау ортасы ретінде талшықты-оптикалық кабельді қолдануға арналған бірінші LAN технологиясы болды. Оның дамуына әкелген себептер желілердің өткізу қабілеттілігі мен сенімділігіне қойылатын талаптарды күшейтеді. Хаттаманың Token Ring негізделген ұзақ қашықтыққа (200 км-ге дейін), үстінен созылып жергілікті желіге жұмсалады. Бұл деректерді беру стандарты. Сонымен қатар, FDDI желісі бірнеше мың пайдаланушыларды қолдай алады.

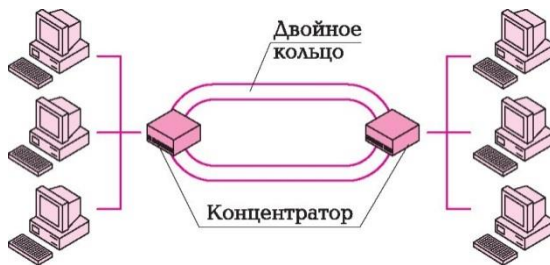
Деректер беру құралы ретінде FDDI талшықты-оптикалық кабелі пайдаланылатын, бірақ пайдаланылуы болады.

FDDI стандарты IEEE802.5 халықаралық стандарты Token Ring ұсынатын таңбалауышқа қол жеткізу әдісіне негізделген. Осы стандарттан шағын айырмашылықтар ұзақ қашықтықта ақпарат берудің жоғары жылдамдығын қамтамасыз ету қажеттілігімен анықталады.

FDDI- желі топологиясы сақина, (бұл жағдайда 100 Мбит / с кезінде жұмыс екі арналар әрбір) 200 Мбит / с екі рет тиімді ставкасы бойынша ақпарат толық дуплексті беруді пайдалануға, негізінен, мүмкіндік береді, онда екі түрлі қолданбалы талшықты кабель болып табылады.

Сондай-ақ, сақинадағы концентраторлармен бірге жұлдызды-топологиялы топология қолданылады. Сақиналардың деректері түрлі бағытта айналып отырады. Бір сақина негізгі болып саналады, ақпарат әдеттегі күйде беріледі; екінші - қосалқы, оған бірінші сақинадағы үзіліс кезінде деректер беріледі (3.2-сурет). Бұл FDDI-ге толығымен мәлімделген өткізу қабілеттілігіне қол жеткізуге мүмкіндік береді. Қалыпты жұмыс кезінде негізгі сақина деректерді беру үшін пайдаланылады, екінші сақина бос емес. Осылайша, сақина желінің түйіндері арасында деректерді берудің негізгі және резервтік жолын құрайды. Екі сақина болуы - осы бекемдік FDDI желісі жақсарту басты жолы болып табылады, және әлеуетін осы артты сенімділігі артықшылығына келетін түйіндер екі сақина қосылғанда болуы керек.

FDDI желісі өзінің элементтерінің біреуі істен шыққан жағдайда оның жұмысын толық қалпына келтіре алады. Егер көптеген ақаулар болса, онда желі бірнеше қосылмаған желілерге бөлінеді. FDDI технологиясы ақаулықтарды анықтау механизмдерінің технологияларын толықтырады



3.2 сурет. FDDI желісі

сақтық көшірме байланыстары болуына негізделген желісінде деректерді беру траекториясын Token Ring реконфигурация тетіктері, екінші сақинасын қамтамасыз етеді.

FDDI желілеріндегі қоңыраулар жалпы ортақ деректер ортасы ретінде қарастырылады, сондықтан оған арнайы кіру әдісі анықталған. Бұл әдіс желілік қатынау әдісі TokenRing өте жақын, сондай-ақ Token Ring әдісі деп аталады. Айырмашылықтар әдісі қол уақыт маркерін сақтау FDDI желілік TokenRing желісінде ретінде тұрақты табылатындығында болып келеді. Шағын жүктеме артты кезде, сондай-ақ жоғары жеделдету кезінде нөлге дейін азайтуға болады - Бұл жолы ринг жүктемеге байланысты болып келеді.

Түрлі өндірушілердің жабдықтар үйлесімділігін жақсы дәрежесін көрсетеді, сондықтан қазіргі уақытта, дене қабаты нұсқалаының бірі, және FDDI стандарттар ретінде желілік технологияны қолдау талшықты-оптикалық кабельдерінің көпшілігі, уақыт сынынан тұратын болып табылады.

3.3. ETHERNET ТЕХНОЛОГИЯСЫ

Ethernet— Негізінен жергілікті желілерде пайдаланылатын пакеттік деректерді беру технологиясы. Ағылшын тілінен «ethernet» термині «жердегі желі», яғни басқа компьютерге (немесе серверге) қосылған желі «тікелей арнада» қолданылады.

Ethernet— бұл жергілікті желілердің ең кең тараған стандарты. Қазіргі уақытта миллиондаған жергілікті желілер Ethernet хаттамаларын қолданады. LAN портына қосылу үшін компьютерде Ethernet желілік картасы орнатылған болуы керек.

және байланыс өзі сым арқылы өтеді, ол бұралған жұп болып табылады, коаксиалды немесе оптикалық кабель деп аталады.

Бұл деректерді беру технологиясын 1973 жылы Xerox компаниясының қызметкері Роберт Меткалфен ойлап тапты. Бұл ең танымал стандартты EthernetDIX (DEC, Intel, Xerox) құру болып табылады. Шын мәнінде бұл жергілікті желілердің көпшілігі үшін қажетті хаттама болып саналады.

Ethernet технологиясы құрылғаннан бері көптеген жылдар бойы өзгерді, атап айтқанда, жылдамдықты арттырды. Бастапқы нұсқаларда тек 3 Мб / с жылдамдығын беру немесе қабылдау жылдамдығы болған және жұмыс істеудің жарты дуплексті режимі (яғни, беру немесе қабылдау) қолданылады. Содан кейін толық дуплексті қолданатын неғұрлым озық технологиялар келді:

- 10-мегабитті Ethernet — 10BASE5 (қалың Ethernet), 10BASE2 (жұқа Ethernet), StarLAN 10 ж.б.;
- жылдам Ethernet (100 Мбит/с) — 100BASE-T, 100BASE-S ж.б.;
- гигабитті Ethernet — 1000BASE-X, 1000BASE-T ж.б.;
- 10-гигабитті Ethernet — 10GBASE-CX4, 10GBASE-LR ж.б.,
- нәтижесінде 40-гигабитті және 100-гигабитті Ethernet— 40GBASE-SR4, 100GBASE-SR10 ж.б.

3.3.1. Ethernet алғашқы технологиясы

Егер Ethernet желісі алғаш рет 1973 жылы пайда болса, оны әзірлеуші компания Xerox сол жылдары белгілі болды. Желілік жұмысы өте табысты болды, сондықтан оны 1980 жылы, DEC (DigitalEquipmentCorporation) және Intel сияқты ірі компаниялары қолдады. Комитет 802 IEEE (InstituteofElectricalandElectronicEngineers), және ESMA (EuropeanComputerManufacturersAssociation): 1985 жылы компаниялар арасында ынтымақтастық нәтижесінде, Ethernet бір желілік халықаралық стандартына айналды, ол ірі халықаралық стандарттар ұйымдарына қатысты.

Ethernet Arcnet және Tokenring ескірген технологияларын жойды. Ол қол әдісімен, яғни, соқтығысуды анықтау және Transmission Control бар «шина» атты моно арналарында бірнеше қолды анықтайды. CSMA / CD (CarrierSenseMultipleAccesswithCollision) Стандартты IEEE802.3 деп аталады. Сондай-ақ, CSMA / CD кестесін пайдаланатын желілік хаттама деп аталады. CSMA / CD хаттамасы OSI үлгісіндегі деректер байланысының деңгейінде жұмыс істейді. Бұл кіру әдісі келесідей сипатталуы мүмкін:

- желіге қосылған барлық құрылғылар бірдей, яғни кез келген бекет кез келген уақытта беріле бастайды (егер жіберуші ортасы бос болса);
- Бір станция арқылы тасымалданатын деректер желідегі барлық станцияларға қол жетімді болып келеді. Ортақ тасымалдаушы қосылған барлық дерек басқа бір уақытта қабылданған бір жұмыс станциясы, жіберілген хабарламаға сай жұмыс атқарады. Тек бір станцияға арналған Message (ол тағайындау станциясы мекен-жайы және жіберуші станцияның мекенжайын қамтиды) станциясы алған жұмысқа еленбейді және басқалар болып табылады. Жабдық қақтығыстарды (қақтығыстарды) автоматты түрде таниды. Жанжал анықталғаннан кейін станциялар беруді біраз уақытқа созады. Бұл уақыт қысқа және әрбір станция үшін жұмыс жасайды. Кідірістен кейін қайта беру жалғасады. Іс жүзінде, егер шамамен 80-100 станция жұмыс істесе, қақтығыстар желі жұмысының төмендеуіне әкеледі;
- Пайдаланушылар саны артып, желінің тиімділігі төмендейді. Бұл жағдайда желідегі сегменттердің санын аз қолданушылар саны барынша арттыру оңтайлы болып саналады. Бірақ желідегі сегменттердің барынша көп саны, сондай-ақ желіге қосылған жұмыс станцияларының ең көп санында шектеулер бар.

Стандартты Ethernet жобалау ол әрбір желілік карта, сондай-ақ орнатылған желілік интерфейс зауытта оған тігілген бірегей алты-байттық нөмірі (MAC мекенжайы), болуы тиіс, бұл көзделген болатын. Бұл сан жақтаудың жіберуші мен алушысын анықтау үшін пайдаланылады.

Үлкен дәрежеде әлемде Ethernet танымалдығы барлық ерекшеліктері, опциондардан бастап, желілік хаттамалар бүкіл әлемде өндірушілердің үлкен саны, нәтижесінде барлық ашық хат болады, бұл шын мәнінде үлкен үлес қосты бірімен толық үйлесімді Ethernet жабдықты шығара бастады.

Ең Ethernet-карта және басқа да құрылғылар автоматты анықтау екі құрылғы арасында үздік ықтимал қосылу үшін жылдамдығы және дуплекстеуін пайдаланып, бірнеше деректер ставкаларын қолдау болады. Авто-қиыстыру сәтсіз болса, жылдамдық серіктестігін реттейді және жартылай дуплексті беру режимін қосады. Мысалы, құрылғы болуы Ethernet 10 порты / 100 ол арқылы технологиясы 10Base-T және 100BASE-TX, және портына Ethernet10 / 100/1000 бойынша жұмыс істей алады, - деп жазған 100BASE-TXi 1000BASE-T, стандарттар 10Base-T қолдайды. Бұл Ethernet енгізу стандарттары, бұралған жұпты қолданады.

1. 10BASE-T - компьютерлердің «бұралған жұп» кабелімен байланысуға мүмкіндік беретін физикалық Ethernet интерфейсі. 10BASE-T атауы кабельдің физикалық негіздерінің кейбір қасиеттерінен келеді. 10 нөмір 10 Мбит / с деректерді беру жылдамдығына сілтеме жасайды. BASE сөзі базалық байланысы үшін қысқа (модуляциясыз деректерді беру әдісі). Бұл белгілі бір уақытта желіде тек бір Ethernet сигналы болуы мүмкін дегенді білдіреді. Басқаша айтқанда, мультиплексинг кең жолақты каналдарда қолданылмайды. «Т» әрпі пайдаланылған кабельдің түрін белгілейтін twistedpair (бұралған жұп) сөз тіркесінен келеді.

2. 100BASE-TX - «жұлдыз» топологиясы желілерінде пайдалану үшін 10BASE-T стандартын әзірлеу. 5-санаттағы жұптық жұп пайдаланылады, тек екі қорғалмаған жұп жұптар пайдаланылады, дуплекстік деректерді беруді қолдайды, қашықтық 100 м дейін.

3. 5е санаттағы бұралған жұпты пайдаланатын 1000BASE-T-стандарты. Деректерді беру кезінде 4 жұп тартылады. Деректерді беру жылдамдығы бір бу үшін 500 Мбит / с. Қашықтығы - 100 м дейін.

Осылайша, деректерді беру жылдамдығына және тасымалдау ортасына байланысты, Ethernet технологиясы үшін бірнеше опциялар бар.

3.3.2. Fast Ethernet

Мультимедиялық сияқты қазіргі заманғы қосымшалар, желілік өнімділігі талаптарына қосымша жаңа талаптар, таратылған есептеу, топтық операцияларды онлайн өңдеу жүйесі, тиісті стандарттар кеңейту қажеттілігін шақырды.

90-жылдардың басынан бастап көп таралған Ethernet нұсқасы кейінірек талшықты-оптикалық кабель желісін пайдалану үшін стандарт ретінде анықталды, берудің медиа бұрмалау жұбы ретінде пайдаланылады. Стандарттарға тиісті толықтырулар енгізілді. 1995 жылы, жылдам Ethernet нұсқасы үшін стандартты пайда, (FastEthernet деп аталатын, IEEE 802.3u стандарт) 100 Мбит / с жылдамдықпен жұмыс істейтін электр беру ортасы, бұралған жұп немесе талшықты-оптикалық кабель ретінде пайдаланылады. Кейінірек төменде жылдамдығы 1000 Мбит / с (GigabitEthernet, IEEE 802.3z стандартты), арнайы нұсқа талқыланатын болады.

802.3u стандарты тәуелсіз стандарт емес, бірақ қолданыстағы стандартқа қосымша болып табылады.

802.3. Бұл мыс пен талшықты-оптикалық кабельдерді 100 Мб / с жылдамдықта жұмыс істейтін желілер үшін байланыс деңгейінің хаттамасының стандарты. Мүдделі өзгерістер бірнеше элементтері конфигурациясы пайдаланылуы кабелі түріне, сегменттерінің ұзындығы мен фабрикаларына саны, оның ішінде өткізу мүмкіндігін өсіретін физикалық қабатын білдіреді.

1. Шын мәнінде FastEthernet ерекшелігі әр түрлі физикалық медиа үшін трансмиссиялық схемаларының үш түрін қамтиды (3.1 кесте).

2. 100Base-TX - ең танымал және 10BASE-T сияқты өте ұқсас сымды байланыспен байланысты. 5-санатындағы мыс бұрылыс жұбы жалпы 10Base-T RJ45 коннекторы арқылы торларды, қосқыштарды және соңғы түйіндерді қосу үшін қолданылады.

3. Оптикалық талшықты кабель пайдаланатын 100Base-FX-FastEthernet іске асуы мүмкін. Бұл стандартты екі жіпшелер арқылы берілетін спектрін ұзақ толқын бөлігін пайдаланады: бір - қабылдау, және басқа үшін - беру үшін. Көп үлгілі талшықты пайдаланған кезде желілік сегмент ұзындығы жоғары соқтығысты анықтау және толық дуплекс 2 шақырым жартылай дуплексті режимде 400-м қол жеткізуге болады. Қашықтан жұмыс режимі бір реттік талшықпен жүзеге асуы мүмкін.

3.1 кесте. FastEthernet технологиясының спецификалық сипаттамасы

Сипаты	Спецификация		
	100Base-TX	100Base-T4	100Base-FX
Кабель түрі	UTP5, STP1	UTP3	Көп үлгілі талшықты оптикалық
Қол жетімділік әдісі	CSMA/CD	CSMA/CD	CSMA/CD
Желідегі өзектердің максималды саны	1 024	1 024	1 024
Сегменттің максималды ұзындығы, м	100	100	Толық дуплекстік беру үшін 2 000, жартылай дуплекс беру үшін 412
Топология	Жұлдыз	Жұлдыз	Жұлдыз
Желі диаметрі, м	205	205	—

4. 100Base-T4 - FastEthernet жүйесінің ең соңғы нұсқасы, кейіннен 100Base-TX және 100Base-FX техникалық сипаттамалары пайда болды. FastEthernet сипаттамасының қалған бөлігі сияқты IEEE 802.3u стандартында сипатталған. Бұл технология үшінші санаттағы төрт бұралған жұптан тұратын кабельді пайдаланады. Хабарламадан бастап, басқа да екі деректерді беру ағымдағы бағытына байланысты қосылған. Сонымен қатар төрт жұп әрқашан хабарламада бағытталған. Осылайша, төрт жұп әрбір уақытта беру үшін пайдаланылады және бір коллизияларды анықтау, тыңдау тасымалдаушы жиілігі үшін пайдаланылады. Тек екі бұралған кабель жұп беру үшін пайдаланылады. 100Base-TX стандартына, айырмашылығы, бұл 100Base-T4 стандарты барлық төрт жұпта пайдаланады. Сонымен қатар, ретранслятор жұмыс бекетінің тікелей кабель деректері арқылы жұмыс бекетін және қайталағышын қосқан кезде 1, 3 және 4 немесе қарама-қарсы бағытта бұрмалап жұп арқылы өту - жұп 2, 3 және 4-жұп 1 және 2 Ethernet стандарты сияқты коллизияларды анықтау үшін пайдаланылады. Басқа екі жұп 3 және 4 пәрмендерге байланысты кезек-кезек сигналды бір немесе басқа бағытта бере алады.

Физикалық орта коаксиалды кабель өзінде деректерді беру арқылы ғана бұралған жұп мен талшықты оптикалық түрін пайдаланылатындықтан, қосылыстардың қажетті жылдамдығын сақтауға мүмкіндік береді және шешім осылайша неғұрлым ыңғайлы және үнемді болып табылады.

3.3.3. Gigabit Ethernet жоғары жылдамдықты технологиясы

GigabitEthernet желісі— стандартты Ethernet желісін дамыту ұғымдардың табиғи, эволюциялық жолы. сабақтастықты сақтай отырып, ең бастысы, бірте-бірте тек желі жүретін бөліктерінде Gigabit сегменттерін енгізудің жаңа ставкаларына қадам жасап, Ethernet желі сегменттерінің FastEthernetі GigabitEthernetv қосылуы жүзеге асады және ол өте оңай жүзеге асады. Бұған қоса, бұл жерде жоғары жылдамдықты қажет етпейді. Бәсекелес Gigabit желілер туралы айтар болсақ, олардың пайдалануын дереу қаражаттың үлкен құнының желілік жабдықты толық ауыстыруды талап етуі мүмкін.

Онлайн GigabitEthernet барлығы бірдей Access әдісінің CSMA / CD алдыңғы нұсқаларында, сондай-ақ дәлелденген, сол пакеттік пішімдерін (кадрлар) және сол пайдаланатындай сақталған өлшемдері

яғни, Ethernet және FastEthernet сегменттері бар байланыс нүктелерінде хаттамаларды қайта өңдеу қажет емес. Жалғыз нәрсе айырбас бағамдарының келіссөзі. Сондықтан Gigabit Ethernet желісін қолданудың негізгі саласы ең алдымен Ethernet және FastEthernet хабарларын бір-бірімен байланыстырады.

Серверлер мен ең озық дербес компьютерлерді тарату неғұрлым айқын жұмысқа айналды GigabitEthernet SUPERFAST. Нақты стандартқа айналған 64 биттік PCI негізгі желісі мұндай желі үшін қажетті деректер жиілігіне жетеді.

Және бұл Development Company GigabitEthernetAllyanke Ethernet сияқты мүмкіндігінше, бірақ 1000 Мбит / с бит жылдамдығы отырып, хаттама 802.3z дамыту топты құру туралы 1996 жылдың жазында жариялай басталды. GigabitEthernet стандарттарының бірқатарын бекіту 1999 жылдың жазында аяқталды.

Неғұрлым жоғары жылдамдықпен жұмыс істейтін желілер оңай көшуді беретін, жоғарыда жұмыс деректеріне (10 және 100 Мбит / с) қатысты GigabitEthernet сабақтастығында аталған. Барлық үш Ethernet жылдамдығы бірдей IEEE802.3 деректер кадрын пішімін, толық дуплексті операциялар мен ағындарды бақылау әдістерін қолданады. Жартылай дуплекстік режимде Gigabit Ethernet операторы дауысты сұраумен және шиеленістерді жоюмен бірдей әдісті пайдаланады. Ол Ethernet және FastEthernet пайдаланылатын кабельдерде барлық ірі түрлерін қолдайды: (3.2 кесте) талшықты-оптикалық, бұралған жұп 5 санаттағы кабельді, коаксиалды кабельді іске қосады. Дегенмен, GigabitEthernet технологиясының әзірлеушілері жоғарыда аталған қасиеттерді сақтауға тек FastEthernet-ғы жағдайдағыдай ғана емес, сонымен қатар MAC деңгейіне де физикалық деңгейде өзгерістер енгізуге мәжбүр болды.

Стандартты GigabitEthernet әзірлеушілері алдында бірнеше қиын мәселелер болды. Олардың біреуі жартылай дуплекстік жұмыс үшін қолайлы желі диаметрін қамтамасыз ету мәселесі болды. Байланысты жақтау өлшемін және барлық параметрлері CSMA / CD өзгермейтін әдісін сақтай отырып, тек қана 25 м рұқсат етілген сегмент ұзындығы ортақ БАҚ үшін CSMA / ҚДНК кабель ұзындығының нұсқасы GigabitEthernet шектеулерге тән болып келеді. Желілік диаметрі кемінде 200 м болатын көптеген қосымшалар бар болғандықтан, FastEthernet технологиясындағы минималды өзгерістердің салдарынан осы мәселені шешуге тура келді.

3.2 кесте. GigabitEthernet технологиясының спецификалық сипаттамасы

Сипаты	Спецификация		
	1000Base-LX	1000Base-SX	1000Base-T
Кабель түрі	Ұзын толқынды лазерлі мультимодальды және бір режимді кабель (1 300 нм)	Талшықты оптикалық	4-жұптық экрандалмаған UTP5
Сегменттің максималды ұзындығы, м	Көп үлгілі: 550 м (62,5 мкм); 550 м (50 мкм) бір үлгілі: 5 км (9 мкм)	220 м (62,5 мкм); 500 м (50 мкм)	100

Тағы бір күрделі міндет - кабельдердің негізгі түрлеріне 1000 Мбит / с жылдамдықпен шығу. Тіпті талшықтар үшін ол FibreChannel технологиясы, оптикалық-талшықты нұсқасы GigabitEthernet үшін негіз құрал ретінде алынды, оның физикалық қабаты ретінде кейбір проблемаларды ұсынады осы жылдамдығына жету үшін, тек 800 Мбит / с максималды деректерді беру жылдамдығын ұсынады. Бірақ бұл мәселелер қазіргі уақытта шешілді.

GigabitEthernet үшін алғашқы өнімдердің ішінде серверлерге арналған қосқыштар мен желілік карталарға арналған гигабит модульдері болды. Құрастырмалы FastEthernetро GigabitEthernet күрт жүйе құраушы LAN өткізу, көтеру және қосулы FastEthernet сегменттерді бөлісетін екі санын көбейту нәтижесінде сақтауға болады, сол жұмыстардан ауысады. NIC GigabitEthernetна серверін орнату серверге арнаны кеңейту және осылайша, пайдаланушылар қуатты жұмыс станциялары өнімділігін арттыруға мүмкіндік береді.

Деректерді беру технологиясы бойынша физикалық орта ретінде GigabitEthernet 802.3z стандарт және көпүлгілі талшықты-оптикалық кабель анықтайды және 75 Омға тән кедергісі бұралған жұпта экрандалған. Біраз уақыттан кейін, 5-ші санаттағы Twisted жұп кабеліне негізделген кабель үшін Gigabit Ethernet іске қосылды, сондай-ақ физикалық орталар үшін спецификациялар белгіленеді.

1000-BASE-LX стандарты, толқын ұзындығы 1 310 нм болатын екінші мөлдір терезеде бір режимді немесе мультиүлгілі оптикалық талшықты пайдаланады. Сигнал қозғалысының диапазоны тек қана байланыста болады.

әдетте өткізу қабылдау түрі мен 550 м көп үлгілі оптикалық талшық үшін 5 км бір-режим оптикалық талшыққа дейін жұмыс жасайтын болып табылады.

1. 850 нм толқын ұзындығы бар мөлдірліктің бірінші терезесінде мультимодтық талшықты пайдаланатын 1000Base-SX-стандарт. Сигналды жіберу қашықтығы - 550 м дейін.

2. 5 санаттағы кабельді беру барлық төрт бұралған жұп жіберуге 2. 1000BASE-tuse бір әр жұптың параллель 250 Мбит / с жылдамдықпен жүреді. Қосылған түйіндерінің ең көп саны, сегмент ұзындығының көрсеткіштері, бұралған жұптың бесінші категориялы кабельі үшін стандартты желілік диаметрінде қалады.

3. Жоғары жылдамдықты деректерді беру технологиясы қайта стандартты болады IEEE 802.3 10 GigabitEthernet (10 Gb, Ethernet 10G, 10 Гбит / с), оның ауданы-Niemi пайда болатын қалалық шкала Ethernetrasshirilas пайдалануға (маусым 2002 жылы мақұлданды) және жаһандық (WAN) желілер стандартты Толқындардың ұзындығы және пайдаланылатын әдістерді кодтау, біртекті және көп үлгілі талшықты-оптикалық беріліс БАҚ кабельдер ретінде пайдалануды анықтаудың бірнеше техникалық сипаттамаларын сипаттайды.

4. 10-Gigabit Ethernet жаңа стандарты LAN, MAN және WAN үшін физикалық орта үшін жеті стандартты қамтиды.

5. 1. 10GBase-CX4 - (15 м-ге дейін) технологиясы 10 Gigabit Ethernetdlya үшін қысқа арақашықтық, мыс кабель жалғағыштар InfiniBand және CX4 пайдаланып жұмыс жасайды.

6. 2. 10GBase-SR- технологиясы 10-Gigabit-Ethernetdlya қысқа қашықтықта (26 немесе 82 м дейін, кабель түріне байланысты), көп үлгілі талшықтарда пайдаланылады. Сондай-ақ, ол жаңа мульти үлгілі талшықты (2 000 МГц / км) пайдалану арқылы 300 м дейінгі қашықтықты қолдайды.

7. 3. 10GBase-LX4 – көп үлгілі талшықтар үшін 240 және 300 м арасында қашықтыққа Wavelength Division Multiplexing пайдаланады. Сондай-ақ, бір реттік талшықты қолданғанда 10 км дейінгі қашықтықты қолдайды.

8. 4. 10GBASE-LR және 10GBASE-ER - бұл стандарттар сәйкесінше 10 және 40 км дейінгі қашықтықты қолдайды.

9. 5. 10GBase-SW, 10GBase-LWi 10GBase-EW- осы стандарттар физикалық интерфейстің үйлесімді форматы және жылдамдықты деректер интерфейсі OC-192 / STM-64 SONET / SDH пайдаланылады. Бұл стандарттар 10GBase-SR ұқсас, 10GBase-HAA 10GBASE- ER сәйкесінше кабельдер мен электр беру қашықтығында бірдей түрлері қолданылады.

10. 10GBASE-T, IEEE 802.3ap-2006 - 2006 жылдың маусым айында қабылданғаннан кейін 4 жыл жұмыс жасайды. 6-санатындағы бұрылыс жұпты (максималды қашықтық - 55 м) және 6а (максималды қашықтық - 100 м) пайдаланады.

11. Ethernet (baskplane / midplane) модульдік ажыратқыштар / маршрутизаторлар және серверлер (модульдік / Blade) 10GBase-KR-технологиясы 10 Gbps объединительная.

HARTING RJ Industrial 10G - Harting компания орнату үшін құралдар қажет етпейді, әлемдегі алғашқы 10-Gigabit RJ-45 коннекторына құру туралы жариялады.

2010 жылы IEEE халықаралық қауымдастық IEEE 802.3ba 40 Gbps және 100 Гбит / с Ethernet стандартының ресми мақұлдауын жариялады. Бұл стандарттар 2010 жылға дейін 10 Гбит / с жоғары жылдамдыққа ие болатын Ethernet стандарттарының топтарын дамытудың келесі кезеңі болып табылады. Жаңа стандарт IEEE Std 802.3ba-2010 жылы бірнеше 10 Гбит / с немесе 25 Гбит / с сілтемелер алмасу 40 және 100 Гбит / деректерді беру жылдамдығын қамтамасыз етеді.

Бір қызығы, бұл стандарт бірінші болып екі түрлі жылдамдықты (40 және 100 Гб / с) анықтауға арналған. IEEE айтуынша, жаңа стандартты басы-бойынша кең жолақты желілердің әлемде өсуде сұранысты қанағаттандыру үшін көмектеседі және технологиялық инновацияларды келесі толқынына негізі болып табылады.

Шын мәнінде, жаңа стандартты IEEE 802.3ba Ethernet IEEE 802.3-стандартына түзету болып табылады, бірақ мұндай стандарттау артықшылықтарының бірі, бәлкім, олардың өнімділігін арттыру және, ол деректер орталықтарында 10-Gigabit-Ethernet байланыстарды іске асыру мүмкіндігі болар еді.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Token Ring технологиясындағы концентраторлар принципін түсіндіріңіз.
2. Arcnet технологиясының жетіспеушілігі қандай?
3. FDDI желілерінде жоғары қол жетімділік қалай қамтамасыз етіледі?
4. Ethernet технологиясы неге кеңінен таралған?
5. IEEE 802.3 стандартын нені айқындайды?
6. CSMA / CD үшін қол жеткізу әдісін сипаттаңыз.
7. 10BASE-T, 100BASE-TX, 1000BASE-T стандарттары арасындағы айырмашылық қандай?

8. IEEE 802.3ba стандартының ерекшеліктерін сипаттаңыз?
9. FastEthernet спецификациясында қандай тасымалдау схемалары сипатталады?
10. CSMA / CD қол жеткізу әдісін Gigabit Ethernet желісінде пайдалану мүмкін бе?
11. 1000Base-LX, 1000Base-SX, 1000Base-T техникалық сипаттамалары қандай?

КОМПЬЮТЕРЛІК ЖЕЛІНІҢ ТЕХНИКАЛЫҚ ҚҰРАЛДАРЫ

Компьютерлік желі - компьютерлер мен басқа құрылғыларды бір-бірімен байланыстыратын бағдарламалық және аппараттық құралдардың күрделі жиынтығы. Желілердің техникалық құралдары желілік адаптерлермен, байланыс арналарымен (кабельдік, спутниктік, телефондық, сандық, талшықты-оптикалық, радиоарналар және т.б.), түрлі сигналдық түрлендіргіштермен және басқа желілік жабдықпен жабдықталған компьютерлерден тұрады.

4.1. ЖЕЛІЛІК АДАПТЕРЛЕР

Желілік адаптер (желі картасы, желі платасы, Ethernet- адаптер, NIC— NetworkInterfaceController) — бұл компьютердің басқа желілік құрылғылармен өзара әрекеттесуіне мүмкіндік беретін перифериялық құрылғы.

Желілік адаптер деректерді беру ортасына қолжетімділікті және желілік құрылғылар арасында деректер алмасуды басқарады. Желілік адаптер сыртқы байланыс желілері бойынша тиісті электромагниттік сигналдармен ұсынылған екілік деректерді алмастыру мәселесін шешеді. Кез келген компьютердің контроллері сияқты, желілік адаптер амалдық жүйе драйверін де іске қосады және желілік адаптер мен драйвер арасында функцияларды бөлу іске асыруға дейін өзгеруі мүмкін. Адаптер орындаған функциялар жиынтығы нақты желі хаттамасына байланысты. Желі адаптері құрылғы мен желі ортасы арасындағы физикалық және логикалық орналасуы фактісін ескере отырып, оның функцияларын желі құрылғысымен интерфейс функцияларына бөлуге болады және

айырбастау функциясы желімен жұмыс жасайды. Желілік құрылғы компьютер ретінде әрекет етсе, желілік ортаға қосылу екі жолмен жүзеге асырылуы мүмкін: желі жүйесі (шина) арқылы немесе сыртқы интерфейстер арқылы. Ең таралған әдіс шина арқылы интерфейс. Бұл жағдайда адаптер жүйелік арқаулық жүйеден келетін деректерді буферге шығарады және ішкі бақылау сигналдарын шығарады.

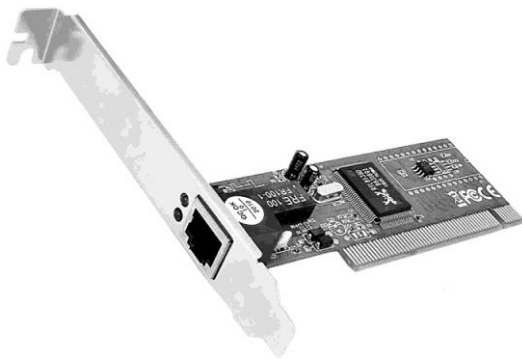
Әдетте, желілік адаптерлер клиенттік компьютерлер мен серверлерге арналған адаптерлерге бөлінеді. Клиенттік компьютерлер үшін адаптерлер, жұмыс көп жүргізушісіне беріледі, сондықтан адаптер оңай және арзан болып табылады. Серверлерге арналған адаптерлер әдетте дербес желісінде еске және керісінше кадрларды аудару жұмысының ең жұмысы көп өз процессорларында қамтамасыз етілген.

Желілік адаптерлер желіні құрастыру технологиясы бойынша, сондай-ақ жобаны іске асыруда бір-бірінен ерекшеленеді:

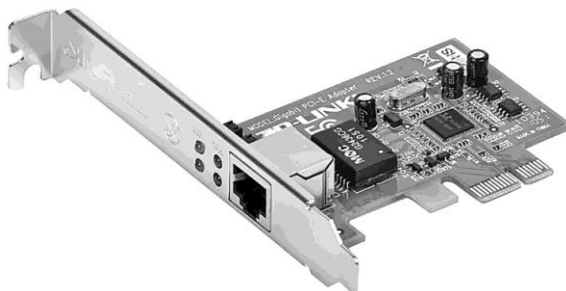
- Сыртқы желілік карта (орнатылатын ISA, PCI, PCI-E-слот немесе USB-бөлу);
- Енгізілген немесе аналық платаға біріктірілген.

Желілік адаптер картасының бір немесе бірнеше сыртқы түрі бар: желі кабелін оған қосуға арналған қосқыштары бар.

Сыртқы желілік карталар компьютердің жүйелік бөлігіне қосымша (жеке кеңейту картасы) немесе басқа сыртқы құрылғы ретінде орнатылады. PCI карталары PCI интерфейсi бар желілік карталармен жабдықталған (4.1-суретті қараңыз). Өз кезегінде, PCI интерфейсi жиілікте жұмыс жасайтын 32 биттік нұсқасы үшін ең жоғарғы өткізу қабілеттілігіне ие болады.



4.1 сурет. PCI желілік платасы



4.2. сурет PCIExpress-1x желілік платасы

33.33 МГц 133 МБ / с, ағытпаның кернеуі 3.3 немесе 5 В құрайды.

Ауыстырғыштар және басқа да сорттарда - сымсыз желілерді ұйымдастыру үшін Wi-Fi құрылғысы қосылады. Қосылудың интерфейсі, PCI және жұмыс принципі мүлде басқаша. Олар келесі тарауда егжей-тегжейлі талқыланады.

Енді PCI интерфейсінің біртіндеп «жоғалып кетуіне» байланысты PCI Express-1c желілік карталары шығарылады (4.2-сурет).

PCMCIA стандарт (Personal Computer Memory Card International Association - компьютерлік жад карталарын халықаралық қауымдастығы) бастапқыда жадты кеңейту үшін әзірленген болатын, бүгінгі сипаттамасы кеңейтілген және түрлі сыртқы құрылғыларды қосу үшін пайдаланылады. Енді, осы интерфейс арқылы желілік карталарды қосу мүмкін (4.3-сурет). Бұл құрылғылар ноутбуктер немесе желілік карта жоқ басқа құрылғылар үшін жарамды, бірақ PCMCIA қосқышы бар (кейбір PDA үлгілері).



4.3 сурет. PCMCIA интерфейсі бар желілік карта



4.4 сурет. USB-интерфейсі бар желілік адаптер

USB-желілерінің карталары PCI аналогтарымен ұқсас функцияларға ие, бірақ олар әдетте қымбатқа түседі. Олардың басты артықшылығы әмбебаптығы: бұл адаптерді USB порты бар кез-келген жүйеге қосуға болады (4.4-сурет). Дегенмен, біздің заманымызда шеткі құрылғылардың саны көп емес - бұл қарапайым емес, PCI-қосқышы жоқ кезде өте маңызды болып табылады.

Көптеген желілік аналық платаларды қазірдің өзінде кірістірілген желілік карталары бар. Желілік карта жүйелік тақтаға біріктірілген болса, сіз PCI слотын сақтайсыз. Кейбір аналық платалард кіріктірілген гигабит адаптерлері бар. Гигабит жабдықтарының бағасын ескере отырып, бұл керемет қосымша екендігін мойындауымыз керек. Кейбір үлгілерде тіпті екі желілік карта бар.

4.2. КОНЦЕНТРАТОРЛАР

Желілік концентратор, немесе *хаб* (hub— орталық) — мұндай «бұралған жұбы» (4.5 сурет) ретінде кабельдік инфрақұрылымы бар желіге Ethernet компьютерлерін үйлестіретін аппарат. Қазіргі уақытта хабтар желі ауыстырғыштарымен ауыстырылады.

Желілік хабтардың сондай-ақ «қалың» немесе «жіңішке» коаксиалды кабель негізінде қолданыстағы желілерге қосылу үшін коннекторларда болуы мүмкін.

Хабы барлық құрылғылар арасында желісінің өткізу жолағын бөле отырып, осылайша, өзіне тән Ethernet топологиясының «ортақ шинасын» түсіне отырып, барлық басқа (қосылған) порттарында сигнал порттарының біреуіне кіріс сигналын ретрансляциялау, бірінші (физикалық) деңгейі OSI желілік моделінің жұмысын істейді және жартылай дуплексті режимде жұмыс жасай алады. Бір мезгілде екі немесе одан да көп құрылғыны жіберуді (соқтығысуды) өңдеу



4.5 сурет. Концентратор

Ethernet басқа бұқаралық ақпарат құралдарына ұқсас желілер - жалғыз құрылғыны тоқтату, беруші және кездейсоқ үзілістен кейін әрекет ұзартуға, қазіргі заман тілінде, концентратор құрылғысы бір соқтығысу доменінде біріктіреді.

Құрылғының, мысалы, айырмашылығы, порттардың немесе кабель кінәсінің бірінен бұл жағдайда толығымен тоқтатады, коаксиалды кабель желісі, ажыратылған кезде желілік хаб, сондай-ақ үздіксіз желілік жұмысын қамтамасыз етеді.

Хабтың жалғыз артықшылығы - төмен құндылығы - Ethernet желілерін дамытудың алғашқы жылдары ғана орынды болды. Электрондық микропроцессорлық компоненттер жақсарып, арзан болғандықтан, концентратордың бұл артықшылығы толығымен жойылды.

Хаб кемшіліктер «ортақ шина» топологиялық кемшіліктерді, тораптар санын өткізу қабілеті, атап айтқанда қысқартудың қисынды жалғасы болып табылады. Сілтеме қабатының түйіндері бір-бірінен оқшауланған емес, өйткені, олардың барлық нашар түйіннің деректер қарқынымен жұмыс істейтін болады. Егер желі 100 Мбит / с бағамы және 10 Мбит / с жылдамдықпен ғана бір торабында түйіндердің құрамында мысалы, содан кейін барлық бөлімшелер торабы 10 Мбит / с кез келген ақпарат қызметін көрсетсе, онда 10 Мбит / с жылдамдықпен жұмыс істейтін болады. Тағы бір кемшілігі - желі деңгейін төмендететін барлық порттарға желілік трафикті беруі.

Кейінірек OSI үлгісін қолданып, екінші (арна) деңгейінде жұмыс істейтін зияткерлік құрылғылар пайда болды. Хабтар айырмашылығы, олар кадр тақырыбындағы ашылуына байланысты порттар арасындағы шеңберінен Ethernet тәуелсіз және селективті беруді қамтамасыз етуге қабілетті болып табылады, және дұрыс порттарына оларды экспедиторлық MAC мекенжайларына сәйкес алушыламен жұмыс жасайды.

Коммутаторлар алдымен ірі Ethernet желілерін босату және оңтайландыру үшін пайдаланылды, содан кейін хабты ауыстырды.

4.3. РЕПИТЕРЛЕР

Деректерді тасымалдаудың кез-келген технологиясына тап болатын алғашқы міндеттердің бірі - оларды мүмкіндігінше алыс қашықтыққа жіберу мүмкіндігі. Физикалық орта осы процесте өзін шектейді - алыстатылған қашықтықта сигнал күші азайып, қабылдау мүмкін болмайды.

Репитерлар немесе қайталағыштар әлсіреген сигналдарды (олардың амплитудасы мен пішінін) қалпына келтіреді, оларды бастапқы формасына дейін жеткізеді. Мұндай сигналдық реле мақсаты желінің ұзындығын ұлғайту болып табылады. Егер желінің ұзындығы желі сегментінің максималды ұзындығынан асып кетсе, желіні бірнеше сегменттерге бөліп, оларды қайталағыш арқылы жалғау керек.

Репитер желінің сенімділігін жақсартады, өйткені бір сегментінің сәтсіздігі (мысалы, кабельдік үзіліс) басқа сегменттердің жұмысына әсер етпейді. Алайда, әрине, деректер бүлінген сегменттен өтпейді. Үлгі бойынша AXI физикалық деңгейде жұмыс істейді.

Репитер сигнал деңгейін немесе олардың физикалық сипатын өзгертпейді. Аналогтық жүйелерге тән пайда жоғары жиілікті сандық сигналдарға жарамсыз. Әрине, оны қолдану арқылы кейбір шағын әсерге қол жеткізуге болады, бірақ бұрмалаудың қашықтығы артуымен деректердің тұтастығы тез бұзылады. Мұндай жағдайларда күшейтпейді, бірақ сигналды қайталайды. Бұл жағдайда кірістегі құрылғыдан сигнал алады, содан кейін оның бастапқы формасын танып, оның нақты көшірмесін шығарғанда шығарады.

Мұндай схема теориялық деректерді еркін қашықтықтарға жібере алады (Ethernet-дың физикалық ортасының ерекшеліктерін ескермеген жағдайда).

Бастапқыда Ethernet топологиясы бар «жалпы шинамен» коаксиалды кабель пайдаланылды, және бірнеше ұзын сегменттердің өзара байланысы болуы керек еді. Осы мақсатта әдетте екі порты бар қайталанатын (қайталағыштар) пайдаланылды. Кейінірек концентраторлар деп аталатын көп портты құрылғылар пайда болды. Олардың физикалық орналасуы бірдей болды, бірақ қалпына келтірілген сигнал, сонымен қатар, сигнал пайда болған барлық белсенді порттарға таратылды.

4.4. 10 базалық хаттаманың (бұралған жұп) пайда болуымен, терминологиялық шатасуларды болдырмау үшін, мульти-порты бұралған жұптың ретрансляторлары әдетте желілік хабтар және коаксиалды қайталама немесе қайталама құрылғылар деп аталады.

4.5.

4.6.

4.7. ЖЕЛІЛІК КӨПІРЛЕР

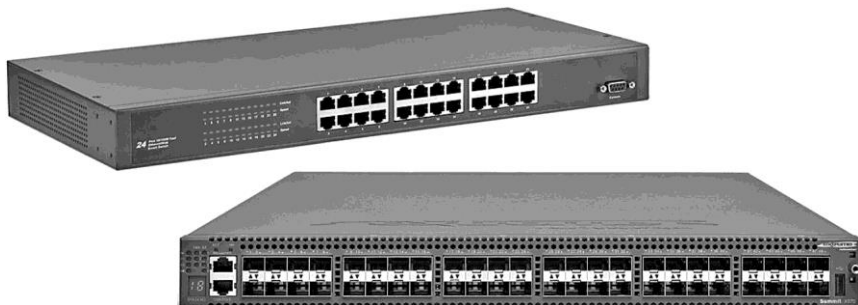
Түйіндердің жүздеген және мыңдаған үлкен желілерін, тіпті осындай GigabitEthernet сияқты жоғары жылдамдықты технологияларды мүмкіндіктерін ескере отырып, ортақ қоршаған ортаның негізінде салынған мүмкін емес жүйенің іс жүзінде барлық технологияларының ортақ ортада түйіндердің саны шектеулі екендігі айтылады. Өйткені негізгі кемшіліктер тораптардың белгілі бір шектік санынан асып кеткен кезде пайда болады. Арттыру - осы шектің желі қуатына шыққанда бөлісті ортаға кірмес бұрын тез азаяды, және кешігу ұзақтығын бастайды. Бұл мәселені шешу үшін, желі көпірді біріктіреді бірнеше трансмиссиялық ақпарат құралдарында, бөлуді бастайды. Көпір жергілікті желілер арасындағы байланыс үшін қызмет етеді, кадрларды бір желіден екіншісіне аударады.

Желі көпір, (көпір) - желі компьютерлік желі сегменттерін үйлестіруге арналған екінші деңгейдегі OSI моделінің, желілік құрылғысы.

Функционалдық көпірлер - концентраторларға қарағанда, ең озық құрылғылар. Көпірлер желі шуылдарын, қателерді немесе бұзылған кадрларды қайталамайды. Әрбір қосылған желі үшін көпір желі торабының абоненті болып табылады. Бұл жағдайда көпір жақтауды алады, оны буфер жадында есіне түсіреді, кадрдың тағайындалған мекен-жайын талдайды. Егер рамка алынған желіге тиесілі болса, көпір осы кадрға жауап бермеуі керек. Егер жақтауды басқа желіге жіберу қажет болса, онда ол барады. Қоршаған ортаға қол жеткізу қалыпты торап ережелеріне сәйкес жүзеге асырылады.

4.5. ЖЕЛІ КОММУТАТОРЫ

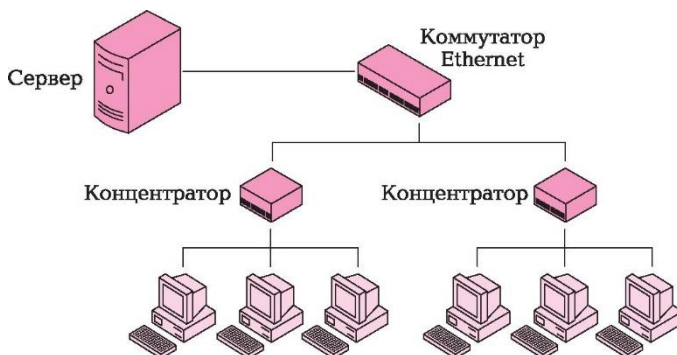
Желілік коммутатор(switch— қайта қосқыш) — бір немесе бірнеше желі сегменттері ішінде компьютерлік желінің бірнеше түйінін жалғауға арналған құрылғы.



4.6 сурет. Желі коммутаторы

Қосқыш OSI моделінің арна (екінші) деңгейінде жұмыс істейді. Ауыстырғыштар көпір технологиясы бойынша жасалған және жиі көбелек көпірлері ретінде қарастырылады (4.6-сурет).

Бір қосылған құрылғыдан трафикті басқаларға тарататын тордан айырмашылығы коммутатор тек деректерді алушыға ғана жібереді. Ерекшелікте коммутатордың шығыс портын білмейтін құрылғылар үшін желінің барлық түйіндеріне және трафикке трафик таратылады. Бұл желінің өнімділігі мен қауіпсіздігін арттырады, желінің қалған сегменттері деректерді өңдеуден бас тартады. Ethernet жергілікті желі стандарттары арасында ең танымал болғандықтан, осы технологияға арналған қосқыштар төменде қарастырылады.



4.7 сурет. Коммутаторға қосу кестесі

Ethernet қосқышына екі түрлі жергілікті желілер 4.7 суретте көрсетілген.

Коммутатор түйіннің MAC-мекен-жайының коммутатор портына сәйкестігін білдіреді, ол қауымдастырылған жадта сақталатын коммутация кестесін жадында сақтайды. Коммутатор қосылғанда, бұл кесте бос, ол жаттығу режимінде жұмыс істейді. Бұл режимде портқа келетін деректер коммутатордағы барлық басқа порттарға тасымалданады. Бұл жағдайда коммутатор фреймдерді (кадрларды) талдайды және жіберуші хосттың MAC-мекен-жайын анықтайды, оны кестеге біраз уақыт қояды. Кейін коммутатор порттарының біреуі MAC мекенжайы кестеде тұрған хостқа арналған жақтауды алса, онда бұл кадр тек кестеде көрсетілген порт арқылы тасымалданады. Егер тағайындалған тораптың MAC мекенжайы кез келген қосқыш портымен байланыспаса, онда ол қабылданған порттан басқа барлық порттарға жіберіледі. Уақыт өте келе коммутатор барлық белсенді MAC мекенжайлары үшін кестені құрастырады, нәтижесінде трафик локализацияланған болып табылады.

Қосқыштар басқарылатын және басқарылмаған (қарапайым) бөлімдерге бөлінеді. Көптеген басқарылатын қосқыштар сізге қосымша функцияларды теңшеуге мүмкіндік береді.

4.6.

ЖЕЛІЛІК ШЛЮЗДЕР БАҒЫТТАУШЫЛАР

МЕН

Барлық Ethernet желісінде жұмыс істеу үшін арналған құрылғылар немесе IEEE802.3 - жұмыс тобының екі желі сегменттері қосылу және олардың әрқайсысы шеңберінде трафикті локализациялау көпірлері, сондай-ақ бірнеше LAN сегменттерді қосылуға мүмкіндік беретін қосқыштарды ұйымдастыру концентраторлары жұмыс жасайды (хабтар).

Алайда, жаһандық (WAN) желілерге қол жеткізуді қоса алғанда, әр түрлі желілік хаттамалар, оның байланыс сайттар үшін, сондай-ақ күрделі конфигурациясы бар желілерде қолданылады. *Маршрутизаторлар* немесе *роутерлер* деп аталатын жабдықтардың арнайы түрі, (^ го е ^), бар неғұрлым тиімді трафик бөлу және желі тораптары арасындағы балама бағыттарды пайдалану. Бағыттаушылардың негізгі мақсаты - әр текті желілер одағы және қызмет көрсету балама жолдары.

Бағыттаушылардың түрлі түрлері іс жүзінде пайдалану нүктесін анықтайды. Порттарының түрлері және саны, әр түрлі болып келеді. Бағыттаушы, мысалы, тиімді қозғалысын басқару үшін LAN Ethernet пайдаланылуы мүмкін.

Интернет пайдаланудың басқа түріне желінің Etnernet желілерінде, мысалы Токер Кольцо, FDDI байланысты желілік сегменттерінің үлкен санының қатысуымен және жергілікті желілермен WAN шығу қамтамасыз ету.

Бағыттаушылар ғана желілерін түрлі қатынас және жаһандық желіге қол жеткізуді қамтамасыз етеді, сонымен қатар, (үшінші OSI моделінің) желілік қабатының хаттамаға Т трафик басқаруға болады. Жоғары деңгейде ажыратқыштармен салыстырғанда мұндай бақылау қажеттілігі туындайтын кезде желілік топологиясы күрделене түседі және оның тораптар саны артады,

Бағыттаушы хаттамаға негізделген болғандықтан, ол осындай құнын, жеткізу жылдамдығы және т.б. осы сияқты факторларды басшылыққа алып үздіксіз бағыт деректер жеткізу туралы шешім қабылдай алады, бағыттаушылар ғана қажетті порттардың деректерді беруді қамтамасыз ететін, тиімді трафик таратылымды басқаруға мүмкіндік береді.

Кесте бағыттаушыдан одан әрі пакеттерін экспедиторлық туралы шешім қабылдайды, ол осыған негізделген ақпаратты қамтиды. Кесте жазбалардағы бірқатар мақсаттағы желісін мекенжайын қамтитын әрқайсысының бағдарлары, пакеттеріне тиісті келесі түйіннің, мекенжайы, және т.б. тұрады. Жиі графтар теориясын пайдалана отырып, кестелер маршруттау салу жұмысы жүргізіледі.

Жаңа заманауи бағыттаушылар сонымен қатар қорғаныс функциясын орындайды - жергілікті желіні сыртқы қауіптен қорғайды. Олар желідегі пайдаланушылардың Интернет ресурстарына кіруіне, IP-адресстерді таратуға, трафикті шифрлаумен айналысатын және т.б. көптеген реттеушілер болып табылады.

Кейде «маршрутизатордың» орнына «желілік шлюз» түсінігі пайдаланылады. Шын мәнінде бағыттаушылардың аппараттық желілік шлюздің бір ғана мысалы болып табылады.

Желі шлюзі (шлюз) - осындай LAN және WAN сияқты әртүрлі хаттамаларды қолдана отырып, компьютерлік желінің бағдарламалық қамтамасыз ету өзара үшін аппараттық бағыттаушы немесе бағдарламалық қамтамасыз ету болып табылады.

Бейтарап бағыттаушы өзі қабылдайды, өткізеді және тек сол хаттамаларын пайдаланатын желілер арасында пакеттерді жібереді. Хаттамалар қоршаған ортаның бір түрі басқа да физикалық орта (желі) хаттамалық айырбастауды қамтамасыз етеді. Бірақ, содан кейін біз желілік шлюз туралы айтып отырмыз. Мысалы, Интернетке жергілікті компьютерге жалғаған кезде әдетте желілік шлюз пайдаланылады.

Желілік шлюздер барлық белгілі операциялық жүйелерде жұмыс істейді. Желілік шлюздер аппараттық шешімдер, бағдарламалық жасақтама немесе екеуі де болуы мүмкін. Бірақ әдетте бағыттаушыда немесе компьютерде орнатылған бағдарламалық құрал болуы мүмкін. Желілік шлюз роутер пайдаланатын барлық хаттамаларды түсінуі керек. Әдетте желілік шлюздер желілік көпірлерге, қосқыштарға және дәстүрлі маршрутизаторларға қарағанда баяуырақ. Желілік шлюз - басқа желіге шығыс ретінде қызмет ететін желі нүктесі болып табылады. Интернетте түйін немесе соңғы нүкте желілік шлюз немесе хост болады.

Интернет-пайдаланушылар мен веб-беттерді пайдаланушыларға жеткізетін компьютерлер хосттар болып табылады және әр түрлі желілер арасындағы түйіндер желілік шлюздер болып табылады. Мысалы, компанияның жергілікті желісі мен Интернет арасындағы трафикті бақылайтын сервер желілік шлюз болып табылады.

Ірі желілерде желілік шлюз ретінде әрекет ететін сервер әдетте прокси-серверлермен және брандмауэрмен біріктірілген. Бұл ұғымдар кейінгі тарауларда талқыланады.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Серверлік желілік адаптер дегеніміз не?
2. Желілік адаптерлерде қандай интерфейстер қолданылады?
3. Желілік хаб дегеніміз не?
4. Трансиверлер мен қайталағыштар қалай қолданылады?
5. Желілік порттар қандай болып табылады?
6. Желілік коммутатор дегеніміз не?
7. Бағыттаушылар дегеніміз не?
8. Желілік шлюздерде бағыттаушылардан қандай айырмашылық бар?

СЫМСЫЗ ТЕХНОЛОГИЯЛАР

Сымсыз жергілікті желі дәстүрлі сымды желіге қарағанда аппараттық компоненттердің сәл өзгеше жиынтығын қажет етеді. Әлбетте, ең үлкен айырмашылық - желілік сервер, сымсыз клиенттік компьютерлер және желіні құрайтын басқа құрылғылар арасындағы сымдардың болмауы.

Бұл тарау сымсыз желіні құрайтын компоненттерді анықтайды, кіру нүктелерінің түрлі сипаттары мен функцияларын, желілік адаптерлерді, антенналарды сипаттайды.

5.1. СЫМСЫЗ БАЙЛАНЫС ЖЕЛІСІ

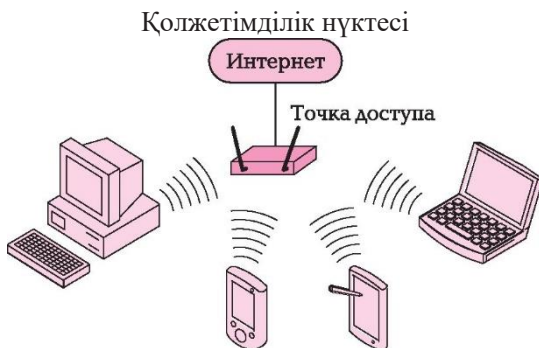
Компьютерлік желілерде кабельдік арналардан басқа сымсыз байланыстар жиі қолданылады. Кейде кабельді төсеу кезінде қиын немесе экономикалық жағынан пайда болмайтын жағдайлар туындайды. Мұндай жағдайларда *сымсыз жергілікті желілер* (WirelessLAN-WLAN) желіні барлық пайдаланушылар үшін барлық ақпараттық ресурстарға қол жетімділікті қамтамасыз ету үшін бірыңғай ақпараттық жүйеге біркелкі жергілікті желілер мен компьютерлерді біріктіруге мүмкіндік беретін қосымша байланыс желілерін орнатпай желіні ұйымдастыру үшін пайдаланылады.

Радио толқындар - осындай желілердегі ақпарат тасымалдаушы. Байланыс арналарында тиісті радиотолқындар таратқыштары мен қабылдағыштары қолданылады. Технология радио жиілігінен 2,4 ГГц жиілігінде деректерді беру әдістемесіне негізделеді, сигналды жұмыс жиіліктері мен арнайы қосымшалар кодтау арқылы жүзеге асады.

Сымсыз желі құрылымы әдеттегі Ethernet желісіне қосылған кіру нүктесі (немесе кейде шлюз) деп аталатын негізгі WLAN бекеті арқылы анықталады. Кіру нүктесі WLAN клиенті ретінде жұмыс істейтіндіктен, бірнеше қатынасу нүктелерінің желісі сымсыз байланыс ретінде жұмыс істей алады (5.1-сурет).

Сымсыз байланыс байланыстары қарапайым сымды желілер үшін стандарттарға толығымен сай келетін, кабельді қолданбастан, компьютерлік желілерді жасауға мүмкіндік береді деп айтуға болады. Бұл технология пайдаланушыға таныс Ethernet желісінде сезіну мүмкіндігін береді.

Сымсыз байланыс желілері жиілік диапазонында және тарату ауқымында ерекшеленеді. Қатынасу нүктесінен клиентке максималды мүмкін қашықтық жүйенің пайдаланылатын WLAN ішіндегі радиотолқын тарату ортасына байланысты. Белгілі болғандай, темірбетонды едендер, сондай-ақ әртүрлі жабдықтардың шығаратын шу деңгейі көптеген байланыс ауқымын айтарлықтай төмендетеді. Сонымен қатар, радиотолқындардың таралуына кедергі құбылысы бар, бұл WLAN-жүйелерінің ауқымын азайтады. Қабылдағыштың сәулелену қуаты мен сезімталдыққа байланысты ашық алаңға қатынау нүктесі мен клиент арасындағы ең үлкен қашықтық 300 метрге дейін, кеңсе ғимараттарында - 50 метрге дейін болуы мүмкін. Зауыт ғимараттарында әдетте 50 метрден аз болады. WLAN станциясында да, клиенттік жабдықта да арнайы антенналар максималды ауқымды айтарлықтай арттыра алады. Тар кішкентай антеннаны жоғары қуаттылықпен пайдалану кезінде 20 км қашықтықта жұмыс істеуге болады. Сымсыз желіні ұйымдастыру



5.1 сурет. Сымсыз технологиямен компьютерлерді біріктіру

көптеген антенналары бар таратқыштар жабық кеңістікте қолданылады.

Сымсыз желілер әртүрлі деп аталады. Мысалы, WirelessLocalAreaNetwork, яғни WLAN, немесе орыс тілінде - сымсыз жергілікті желі. Ең танымал сымдарсыз желілердің атауы Wi-Fi болды. Бір қызығы, термин «Wi-Fi» бастапқыда тартуға болатын сөздер ойынына айналды. Wi-Fi (HighFidelity-high- дәлдігі) бастапқыда WECA пресс-релиздеріне қарамастан және WirelessFidelity сөзі пайда болды. Мұндай тұжырымнан бас тартқан, Wi-Fi қысқартулары жоқ шифрланған, бірақ қысқартылған атау болып табылады.

Wi-Fi Alliance сауда белгісі. Сымсыз деректерді беру технологиясы артықшылықтары:

- кабельді пайдаланбай желіні қолдану мүмкіндігі, бұл ұйымның және / немесе одан әрі желінің құнын төмендетеді. Бұл әсіресе маңызды емес жерлерде маңызды кабель төсеу мүмкіндігі болып саналады;

- мобильді құрылғыларға арналған желіге кіруді қамтамасыз ету;

- Wi-Fi құрылғыларын кеңінен қолдану міндетті түрде кепілдік берілген үйлесімдігі Wi-Fi Alliance жабдығын сертификаттау;

- Клиенттердің мобильділігі және Интернетті пайдалану мүмкіндігі кез келген жағдайда жұмыс жасайды;

- Wi-Fi желісінің қамту аймағында желіге қосылу мүмкіндігі әртүрлі құрылғылардың қанша пайдаланушысы - телефондар, компьютерлер, ноутбуктар және т.б білдіреді;

Wi-Fi құрылғыларынан радиацияның төмен деңгейі, деректер (ұялы телефоннан 10 есе аз).

Технологияның кемшіліктерінің арасында Wi-Fi желісіндегі нақты беру жылдамдығы құрылғылар арасындағы физикалық кедергілерге, басқа электронды құрылғылардан кедергі болуына, құрылғылардың салыстырмалы орналасуына және бір мезгілде пайдаланушылар санына байланысты екенін атап өткен жөн.

5.2.

СТАНДАРТТЫ СЫМСЫЗ БАЙЛАНЫС

Сымсыз желілер стандартта сипатталған радиоарнамен байланысқан жергілікті компьютерлік желілердің технологиясына негізделген.

IEEE 802.11. Стандарты байланыс пен жиілік диапазонын анықтайды.

Қолданыстағы стандарттар тізімі 5.1 кестеде келтірілген.

802.11 стандарты құрылғылар үшін жиіліктер мен жиілігі әрбір елде уәкілетті мемлекеттік органмен бөлінеді және реттеледі. Сондай-ақ, жергілікті нормативтік актілер өздерінің құрылғыларының жұмыс ережелерін, олардың қуатын, жиіліктердің диапазонын бөлуін, таратқыштың қуатын және басқа да сипаттамаларын реттейді.

Біздің елімізде осындай орган - Ресей Федерациясының Байланыс және бұқаралық коммуникациялар министрлігі болып табылады. Министрлік барлық базалық жиілік үшін барлық нұсқаларын жүзеге асыру стандарттары 802.11 (А, В, г, н) РФ пайдалануға рұқсат деп белгіленген соңғы нормативтік құжатта көрсетіледі. Осы отбасының барлық стандарттарына ұқсас Wi-Fi802.11 ISO / OSI үлгісінің физикалық және арнадағы төменгі екі деңгейінде жұмыс істейді. Сондықтан, мұндай TCP / IP сияқты желі Ethernet (802.3) жұмыс желілік қосымшалар мен желілік хаттамалар, сондай-ақ Wi-Fi-802.11 пайдаланылуы мүмкін. Басқаша айтқанда, бірнеше кірістері бар Ethernet-бағыттаушылар бар болса, желі, ол сымды құрылғының 802.3 немесе сымсыз Wi-Fi-құрылғыға 802.11 қосылғаны маңызды болмайды: барлық перифериялық құрылғылар бір-бірін көріп және дұрыс қарым-қатынас жасайтын болады.

Wi-Fi құрылғыларын рұқсатсыз кіруден қорғау үшін, WiredEquivalentPrivacy (WEP) шифрлау тетіктері пайдаланылады. WEP шифрлау статикалық немесе динамикалық болуы мүмкін. Статикалық WEP шифрлауы арқылы кілті өзгермейді.

5.1 кесте. Сымсыз байланыс стандарттары

Стандарт атауы	Өткізу жылдамдығы, Мбит/с	Жиілігі, ГГц
802.11	1, 2	2,4
802.11a	54	5
802.11b	5,5 и 11,0	2,4
802.11g	54	2,4
802.11n	600	2,4 — 2,5, 5

динамикалық шифрлау әдісі шифрлау кілтін мезгіл-мезгіл өзгертеді. 2004 жылы жаңа WPA және WPA2 қауіпсіздік алгоритмдері 802.11 стандартына өзгерістер енгізілді. WEP технологиясы ескірген деп саналды. Қауіпсіздіктің жаңа әдістері WPA және WPA2 (Wi-Fi Protected Access) бағдарламалық деңгейлердегі сияқты аппараттық деңгейлерде де көптеген сымсыз байланыстармен өзара байланыса алады.

5.3. СЫМСЫЗ ЖЕЛІНІҢ ҰЙЫМДАСТЫРУҒА АРНАЛҒАН ҚҰРЫЛҒЫЛАР

5.3.1. Wi-Fi-адаптерлер

Wi-Fi адаптері сымды желідегі желілік карта сияқты бірдей функцияны орындайды. Ол пайдаланушы компьютерін сымсыз желіге қосу үшін қызмет етеді, яғни ол компьютер мен желі арасындағы интерфейс. Сымсыз желіде адаптерде деректерді компьютерден желіге жіберетін радио таратқыш бар, ал кіріс сигналдарын желінің деректерімен анықтайтын және оларды компьютерге жіберетін ресивер бар.

Wi-Fi PCI-адаптерлері PCI-интерфейстері тіркелген компьютерлер желісіне сымсыз қосылуды қамтамасыз етеді. Wi-Fi PCI-адаптері ең кең таралғандардың бірі болып саналады.



5.2 сурет. PCI интерфейсі бар Wi-Fi желілік адаптер



5.3 сурет. Wi-Fi Card-Bus адаптері, 5.4 суреттің PCMCIA адаптерлері бар сыртқы Wi-Fi USB интерфейсі

Дербес компьютерлерге арналған желілік контроллер түрлері (5.2 суретте көрсетілген).

PCI-адаптері кеңейтілген-диапазонда және күрделі конфигурациясы салаларында, әсіресе маңызды болып табылатын жоғары пайда, бір, екі немесе үш сыртқы барлық бағытқа арналған антенналармен жабдықталған. Қолдау Wi-FiPCI- адаптерлер стандарттарында 802.11 ^, б, г, н) 300 Мбит / с дейін тұрақты сигнал қабылдау және деректерді беру жылдамдығы қамтамасыз етеді. Желілік деректерді қорғау WEP, WPA және WPA2 шифрлауын қамтамасыз етеді.

Wi-FiCard-Bus-интерфейс адаптерлер PSMSYA кеңейту слоттары РККard (5.3 сурет) бар желілік ноутбук немесе портативті компьютер үшін сымсыз жоғары жылдамдықты жалғануды қамтамасыз ететін кіріктірілген антеннасы бар, жинақы және орнатуы оңай, Wi -Fi қатынау желілеріне икемді және ыңғайлы қосылуын қамтамасыз етеді.

Сыртқы адаптерлер - Wi-Fi USB-адаптері сымсыз жергілікті желі үшін тез сымсыз желіге USB- интерфейсін бар кез келген жұмыс үстелін немесе ұялы компьютерді қосу үшін және сыртқы электрмен жабдықтау үшін талап етпейтін мүмкіндік туады (5.4 сурет).

5.3.2. Сымсыз қолжетімділік нүктесі

Сымсыз қолжетімділік нүктесі (WirelessAccessPoint, WAP) — бұл барлық желіге (сымсыз немесе сымды) сымсыз кіруді қамтамасыз ететін немесе жаңа сымсыз желі жасайтын сымсыз базалық бекет (трансивтік жабдықтардың жиынтығы). Көбінесе олар Wi-Fi кіру нүктелері деп аталады.

Кіру нүктесі арқылы сымсыз адаптерлер арасында байланыс пен ақпарат алмасу, сондай-ақ желінің сымды сегменті арқылы байланыс жүзеге асырылады. Осылайша, кіру нүктесі қосқыш рөлін атқарады. Wi-Fi кіру нүктесінде бір немесе бірнеше желі интерфейстері бар, бұл арқылы осы нүктені әдеттегі сымды желіге қосуға болады. Сол интерфейссті нүктені теңшеу үшін пайдалануға болады.

Кіру нүктесі компьютерлер топтарын (олардың әрқайсысы сымсыз желі адаптері бар) жеке желіге (Ad-hoc режимі) қосуға және желінің сымсыз және кабельдік бөліктері (Infrastructure mode) арасындағы көпір функциясын орындау үшін пайдаланылуы мүмкін.

Кіру нүктесі SSID (ServiceSetIdentifier) арнайы сигнал пакеттерін пайдаланып жібереді. SSID желісін білу үшін, клиент кіру нүктесіне қосылу мүмкіндігі мүмкін екенін анықтай алады.

Егер бірдей SSID-лері бар екі қол жеткізу нүктесі қамту аймағына кірсе, ресивер сигнал деңгейіне қарай олардың арасында таңдай алады. Стандартты Wi-Fi қосылуға арналған өлшемдерді таңдау кезінде клиентке толық еркіндік береді. Дегенмен, стандартта сымсыз Wi-Fi желілерін құрудың барлық аспектілері сипатталмаған, сондықтан әрбір жабдық өндірушісі бұл мәселені бір немесе басқа тұрғыдан жақсы деп санайтын тәсілдерді қолдана отырып шешеді.

Енгізу нүктелері ашық ауада пайдалану үшін де, қоршаған ортаға әсер етуден қорғалған, сондай-ақ ғимарат ішінде қолданылуы мүмкін.

5.3.3. Антенналары

Антенналар сымсыз желінің ауқымын кеңейту үшін және сигналдарды қабылдау мен қабылдауды бір уақытта жақсарту үшін қолданылады. Wi-Fi антенналары бағдарламалық жасақтаманы қажет етпейді.



5.5 сурет. Бағытталған (а) және көп бағытты(б) антенналар

конфигурация немесе орнату үшін түрлі факторлар пайда болуы мүмкін.

Wi-Fi байланысының радиусы антенналарға және қоршаған ортаға байланысты. Капиталдың қабырғалары сигналдың күшін айтарлықтай азайтады. Күрделі кедергі металл торы болуы мүмкін. Іс жүзінде, Wi-Fi байланысының қашықтығы бірнеше ондағандан бірнеше жүз метрге дейін, ал арнайы антенналарды, тіпті бірнеше километрді құрайды. Ғимарат ішінде кішігірім желіні құру үшін Wi-Fi адаптерлерімен және кіру нүктелерімен жабдықталған стандартты антенналар жеткілікті, бірақ кейде сатып алынған сыртқы антенналарды пайдалану қажеттілігі бар. Антенналардың негізгі сипаттамасы - антеннаның пайда болуы. Әлбетте, пайда неғұрлым жоғары болса, соғұрлым жақсы.

Антенналардың екі түрі бар - *бағытты және көп бағытты* (сурет 5.5, а, б). Көп бағытты антенналар кіріктірілген Wi-Fi-желілік адаптерлермен, кіру нүктелерімен, маршрутизаторлармен жабдықталған.

5.3.4. Wi-Fi-бағыттаушылары

Wi-Fi бағыттаушылары сымды немесе сымсыз желі үшін жалпыға қолжетімді Интернет желісіне жылдам және оңай орнатуға мүмкіндік береді.



5.6 сурет. Wi-Fi-роутер

Wi-Fi-роутері (сымсыз бағыттаушы), сондай-ақ сымды бағыттаушылар компьютер желісінде трафикті бағыттау үшін арналған. Wi-Fi бағыттаушысында желі ішіндегі пайдаланушыларды және сыртқы байланыс арнасын жалғайтын VAN желісін қосу үшін Wi-Fi желісі мен жергілікті желі кабелімен жұмыс істеу үшін бір немесе екі антенна бар (5.6-сурет).

Осылайша, Wi-Fi бағыттаушысы сымды желілер мен Wi-Fi желілерін біріктіреді және олардың арасында бағытты жүргізеді. Барлық Wi-Fi бағыттаушылары веб-интерфейс арқылы басқарылады.

Wi-Fi арқылы берілетін деректердің қорғау, хаттамалар WPA / WPA2-RADIUS, WPA / WPA2-PSK, WEP қамтамасыз етеді. MAC мекенжай сүзгісі желіге рұқсатсыз кіруді болдырмайды.

5.3.5. Wi-Fi ADSL-модемдері

xDSL (DigitalSubscriberLine, сандық абоненттік байланыс) — заманауи микроэлектроника жетістіктері мен цифрлық сигналдарды өңдеу әдістері негізінде сызықтық кодтарды және желілік бұрмалануларды түзетудің бейімделген әдістерін қолдана отырып, жалпыға қолжетімді телефон желісінің абоненттік желісіне өткізу қабілетін айтарлықтай арттыратын технологиялардың отбасы.

XDSL технологиялары 90-шы жылдардың ортасында пайда болды. Сандық абонентті ISDN тоқтататын балама ретінде жұмыс жасады.

XDSL акронимінде «x» таңбасы белгілі бір технологияның атауында бірінші таңбаны белгілеу үшін пайдаланылады.

XDSL технологиялары жылдамдығымен мәліметтерді жоғары сапалы аналогтық және цифрлық модемдерге қарағанда әлдеқайда жоғары деңгейге жеткізуге мүмкіндік береді.

Бұл технологиялар абоненттерге және провайдерлерге айтарлықтай пайда әкеле отырып, дауысты, жоғары жылдамдықтағы деректерді және бейнені беруді қолдайды. Көптеген xDSL технологиялары деректерді жоғары жылдамдықпен жіберуді және дауысты беруді сол мыс жұбынан біріктіруге мүмкіндік береді. XDSL технологиясының қолданыстағы түрлері негізінен пайдаланылатын модуляция түрінде және деректерді беру жылдамдығымен ерекшеленеді.

ADSL (AsymmetricDigitalSubscriberLine, асимметриялық сандық абоненттік желі) - бұл қол жетімді арналарды өткізу қабілетін шығыс және кіріс трафигі арасында асимметриялық түрде таратылатын модем технологиясы. Көптеген пайдаланушылардың кіріс трафигі шығыс трафигінің көлемінен айтарлықтай асып түскендіктен, шығыс трафигінің жылдамдығы әлдеқайда төмен. Бұл шектеу тең-теңімен байланыс желілері мен бейне байланыстардың таралуына қолайсыз болады.

Wi-FiADSL модемдері телефон желісі арқылы интернетке шығу үшін жасалған және Интернет желісіне қатынау мүмкіндігі бар жергілікті желі ішінде сымсыз қатынауды ұйымдастыратын көп функциялы құрылғылар болып табылады (5.7-сурет).

ADSL модемдері төрт топқа бөлінеді:

1) ішкі PCI модемдері;



5.7 сурет. Wi-FiADSL-модем

- 2) USB интерфейсі бар сыртқы модемдер;
- 3) Ethernet интерфейсі бар сыртқы модемдер;
- 4) Ethernet интерфейсімен сыртқы бағыттаушылар (бағыттаушылар).

ADSL бағыттаушыларының көпшілігі RFP хаттамасын қолдайды.

PPPoE (нүкте-`pointprotocoloverEthernet`) - желілік хаттама арна МЖӘ жақтау беру деңгейі Ethernet арқылы (нүкте-`PointProtocol`, хаттама сілтеме қабатын OSI пиринг) жүзеге асады. PPPoE аутентификация, деректерді қысу, шифрлау сияқты қосымша мүмкіндіктерді береді. ADSL-маршрутизаторлар дербес қажет болған жағдайда, пайдаланушының компьютерінде PPPoE-клиентті орнату қажетінсіз провайдеріне қабілетті болып келеді. Сондықтан, ADSL-бағыттаушысы жеке жергілікті серверді оңай ауыстыра алады, ол жергілікті желідегі Интернетке қол жеткізуді қамтамасыз етеді. Бағыттаушылар Ethernet интерфейсі арқылы қосылады және осындай ADSL модемдерінің конфигурациясы веб-интерфейс арқылы кез-келген шолғышты қолданады.

Сондай-ақ кіріктірілген қосқыштары бар ADSL-бағыттаушылар, Wi-Fi кіру нүктелері, басып шығару серверлері және басқа да мүмкіндіктері болады. Мүмкін, олар ADSL модемінің ең дамыған түрі болар. Осындай жабдықтың көмегімен шағын үй желісін ұйымдастыруға, кез-келген қосымша құрылғыларды пайдаланбастан байланыс қызметтерін қосуға болады, ол тек қана ыңғайлы емес, сонымен бірге бірден бірнеше жабдықты сатып алуға арзанырақ болып келеді.

Wi-Fi-қайталағыштар - сымсыз желінің қамту аймағын кеңейтуге арналған құрылғылар және кіру нүктелері мен сымсыз желілік адаптерлер арасындағы қашықтықты ұлғайтуға арналып орнатылған жабдықты пайдалануға мүмкіндік береді.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Сымсыз желі қандай компоненттерді құрайды?
2. Сымсыз деректерді беру технологиясының артықшылықтары қандай?
3. 802.11, 802.11a, 802.11b, 802.11g, 802.11n беру жылдамдығы қандай?
4. 802.11 және 802.3 стандарттары арасындағы негізгі айырмашылық қандай?

5. Wi-Fi адаптерлері дегеніміз не?
6. Wi-Fi PCI адаптерлері мен Wi-Fi сыртқы адаптерлер арасындағы айырмашылық қандай?
 7. Сымсыз кіру нүктесі дегеніміз не? Оның функциялары қандай?
 8. Wi-Fi маршрутизаторының қандай мүмкіндіктері бар?
 9. xDSL технологиясы қандай мүмкіндіктер береді?

ЖЕЛІЛІК БАҒДАРЛАМАЛЫҚ ҚАМТАМАСЫЗ ЕТУ. ҚЫЗМЕТ ЖӘНЕ ҚЫЗМЕТ КӨРСЕТУ

6.1. «ТҰТЫНУШЫ СЕРВЕР» КӨПДЕНГЕЙЛІ СӘУЛЕТІ

Ақпараттық жүйенің бөлігі болып табылатын компьютерлер мен бағдарламалар, әдетте, тең емес. Олардың кейбіреулері өздерінің ресурстары (файлдық жүйе, процессор, принтер, дерекқор және т.б.), басқалары осы ресурстарға қол жеткізе алады. Кез келген ресурсты басқаратын компьютерге немесе бағдарлама, ресурс сервер (және т.б. «файл сервер», дерекқор сервері, компьютер сервер.) деп аталады. Ресурстың клиенті мен сервері сол компьютер жүйесінде және желі арқылы қосылған түрлі компьютерлерде орналаса алады.

«Тұтынушы-сервер» (тұтынушы-сервер) - компьютерді немесе желіні сәулет тапсырмаларын немесе желілік жүктемені клиенттер деп аталады, қызмет көрсетушілер арасында бөлінетін серверлер деп аталатын және клиенттерге қызмет көрсету болып табылады.

Кез келген соңғы құрылғы тұтынушы немесе сервер ретінде әрекет ете алады. Сервер тұтынушыға кейбір қызметтер немесе ресурстар ұсынады. Тұтынушы желі туралы ақпаратты сұрайтын нысан болып табылады. Әдетте, бұл серверден ақпаратты сұрайтын дербес компьютер немесе жұмыс бекеті болып табылады.

Тұтынушы пен сервер арасындағы айырмашылықты және олардың қандай бағдарламалық жасақтамаға орнатылғанын және оны қалай теңшегенін түсіну маңызды. Егер сіз үй компьютерін алып, оған Apache веб-сервер қолданбасын салсаңыз, мысалы, ол компьютер сервер ретінде әрекет етеді деп айтуға болады. Үлкен IBM серверін алсаңыз, оған MSWindows қойыңыз және веб-браузерді қолдансаңыз, онда бұл жабдық клиенттік құрылғы ретінде пайдаланылған деп айтуға болады. Осылайша, клиент арасындағы айырмашылық

және сервер OSI моделінің қолданбалы деңгейінде; осы модельдің төменгі деңгейлері тұрғысынан олардың арасындағы айырмашылық жоқ.

Көптеген серверлер сол хостта жұмыс істей алады. Кішігірім ұйымдарда жиі бір мезгілде жұмыс істейтін бірыңғай физикалық сервер бар, мысалы, веб-сервер, пошта сервері, файлдық сервер және т.б. Бұл серверлердің әрқайсысы өз клиенттеріне қосыла алады.

Тұтынушы-сервердің сәулетін қолданудың артықшылықтары анық:

- тұтынушы бағдарламалары бойынша сервер кодын қайталамау;
- барлық есептеулер серверде орындалғандықтан, тұтынушы орнатылатын компьютерге қойылатын талаптар азаяды;
- барлық деректер әдетте көптеген тұтынушылардан әлдеқайда жақсы қорғалған серверде сақталады. Серверде тек қана қол жетімділік қатынасы бар тұтынушыларға деректерге рұқсат беру үшін рұқсатты басқаруды қамтамасыз ету оңайырақ болып келеді.

Кемшіліктердің ішінде жұмыс істемейтіндігіне назар аударуға болады.

Сервер бүкіл компьютер желісінде жұмыс істемейді. Сервер тиімді емес сервері барлық тұтынушыларға қызмет көрсету үшін жеткіліксіз болатын сервер, сондай-ақ жөндеу, алдын-алу және т.б. болатын сервер болып саналады. Сонымен қатар, осы жүйенің жұмысын қамтамасыз ету үшін бөлек маман - жүйелік әкімші қажет.

Көптеген деңгейдегі сәулет «тұтынушы-сервер» деп аталатын - деректерді өңдеу функциясы бір немесе бірнеше жеке серверлерде көрсетілетін «тұтынушы-сервер» сәулетінің бір түрі. Бұл серверлер мен тұтынушылардың мүмкіндіктерін тиімді пайдалану үшін деректерді сақтау, өңдеу және ұсыну функцияларын бөлуге мүмкіндік береді. Атап айтқанда, іс клиент сервер қосымшалары арқылы дереккөз серверіне сұрау етеді, онда үш-деңгейлі сәулет болып табылады.

Үшдеңгейлі, немесе үшбөлімді, сәулет — тұтынушының қосымшасы (әдетте жұқа тұтынушы деп аталады), қосылған қосымшалар сервері және бағдарлама сервері жұмыс істейтін дереккөз сервері: үш компонентті қабылдайтын бағдарламалық жасақтама пакетінің сәулеттік үлгісі.

Үш деңгейлі тұтынушы-сервер сәулетінің механизмдері осы оқулықтың екінші бөлімінде толығырақ сипатталады.

6.2. ЖЕЛІЛІК ҚЫЗМЕТ ПЕН ЖЕЛІЛІК ҚЫЗМЕТ КӨРСЕТУ

Функционалдық тұрғыдан алғанда, компьютерлік желі пайдаланушыға белгілі бір қызметтер жиынтығын ұсынатын желілік қызметтер жиынтығы ретінде қарастырылады, мысалы, аппараттық немесе бағдарламалық жасақтамаға қол жеткізу немесе басқа желі түйінінің ақпараттық ресурстары болып табылады.

Қызмет — бұл белгілі бір қызметтер жиынтығын іске асыратын желі компоненті.

Желілік қызмет көрсету — Бұл қызмет тұтынушысы (мысалы, пайдаланушы) және қызмет провайдері (қызмет көрсету) арасындағы интерфейс).

Желілік қызметке қатысты:

- деректерді сақтау;
- ақпарат іздеу;
- пошталық қызметтер (электронды пошта);
- желідегі өзектер арасындағы деректерді беру;
- қолданбалы процестер арасындағы өзара іс-қимыл сессияларын ұйымдастыру.

Желілік қызметтерді тұтынушылар - пайдаланушылар, бағдарламалар, операциялық жүйелер, функционалдық блоктары, есептеу үдерістері және т.б. Желілік қызметтерді жеткізуші - бұл қызметтерді немесе қызметтер жиынтығын сататын желі қызметі. Мысалы, қызмет жеткізушілері жергілікті және қашықтағы ресурстар мен қызметтерді жалпы қолжетімділік пен пайдалануды қамтамасыз ету құралы болып табылады.

Жабдықтарға, бағдарламалық жасақтамаларға және деректерге қол жеткізуден басқа, желі қызметтері басқа да нақты міндеттерді шешеді, мысалы, таратылған деректерді өңдеуге қатысты тапсырмалар. Бұл тапсырмалар деректердің бірнеше көшірмелерін синхрондауды, түрлі түйіндерде (репликалау қызметі) немесе бір тапсырманы орындауды ұйымның желісінің бірнеше машиналарында (қашықтағы процедураларға қоңырау шалу қызметі) қатарлас қамтамасыз етуді қамтиды.

Желілік қызметтің қызмет көрсету бөлімі осы желі қызметі қосылған желі ресурстарына қол жеткізу немесе оған қол жеткізу туралы сервистің клиенттік бөлігінен алынған сұраныстарды өңдеуді және орындауды жүзеге асырады. Әдетте, желілік операциялық жүйе компьютерлік желінің стандартты функцияларын қамтамасыз ететін базалық желілік қызметтердің қызметтерін ұсынады. Қосымша қызметтерді жүйелік желілік бағдарламалар немесе амалдық жүйе арқылы іске қосылатын утилиталар арқылы жүзеге асырылатын желілік қызметтер қамтамасыз ете алады.

Файл қызметі тұтынушыға қашықтағы файлдық жүйелермен өзара әрекеттесуге мүмкіндік береді. Желілік басып шығару қызметі принтерді пайдалану арқылы басып шығаруға мүмкіндік береді. Сервер бөлігі осы тұтынушының бастамасы бойынша клиенттік бөлімге желілік қызметтерді ұсына алады.

Желілік қызметтің сапасының негізгі көрсеткіштерінің бірі оның ыңғайлылығы, ең алдымен пайдаланушымен жұмыс істеу ыңғайлылығы. Сондықтан желілік сервистер арнайы интерфейспен, көп жағдайда пайдаланушыға интуитивті түрде беріледі.

Сервис арқылы көрсетілетін қызметтерді пайдалану үшін, тұтынушының сервис бөлігі пайдаланушы компьютерінде орнатылуы керек. Бір қызмет үшін көптеген функциялары бар және олардың функциялары мен интерфейс ыңғайлы болуы мүмкін. Ең бастысы, тұтынушы серверге (сервистің серверлер жағына) сұраулармен сервер түсінетін ережелерге сәйкес қол жеткізеді. Осыған байланысты, желілік қызметтерді дамыту кезінде тұтынушы пен серверлік бөліктер арасындағы желілік өзара әрекеттесудің хаттамасын (ережелерін), олардың арасындағы функцияларды бөлуді және әртүрлі компьютерлік желілердің хаттамаларын үйлестіруді ерекше маңызды міндет болып табылады. Хаттамаларды келісу мәселесі компьютерлік желінің тораптары көбінесе әртүрлі аппараттық платформалармен және сәйкесінше желілік өзара әрекеттесудің көптеген хаттамаларын талап ететін әртүрлі желілік бағдарламалық жасақтаманың болуымен байланысты.

Желілік қызметтің операциялық жүйеге интеграциялану дәрежесі желілік қызметтің бағдарламалық жасақтамасын енгізудің келесі түрлерін ажыратады:

- желілік қызмет операциялық жүйенің бөлігі болған кезде интеграцияның жоғары деңгейі;
- желілік қызмет амалдық жүйеге қосымша болғанда интеграцияның орташа деңгейі;
- желілік қызмет дербес бағдарламалық өнім болғанда интеграцияның төмен деңгейі.

Пайдаланушы ұсынатын желі қызметтерінің жиынтығы желінің қол жетімді аппараттық және бағдарламалық құралдарына байланысты. Жақында, әсіресе Интернетті дамыту кезінде, желілік қызметтер саны айтарлықтай өсті. Пайдаланушылар желілік қызметтермен нақты айналысады, ал пайдаланушының негізгі желілік бағдарламалық жасақтамасы мен аппараттық құралдарының жұмысынан байқалады.

Серверлік бағдарламалық қамтамасыз ету — сервистік функцияларды орындайтын компьютерлік жүйенің бағдарламалық компоненті

тұтынушының сұранысы бойынша белгілі бір ресурстарға немесе қызметтерге қол жеткізуге мүмкіндік береді.

Тұтынушы сұраулары мен сервердің жауаптарының пішімі хаттама арқылы анықталады. Ашық хаттама сипаттамалары ашық стандарттармен сипатталады, мысалы, Интернет хаттамалары RFC құжаттарында анықталады.

Орындалған тапсырмаларға байланысты, кейбір серверлер, қызмет көрсету туралы өтініш болмаған жағдайда, жұмыс істемейді. Басқалары, мысалы, ақпарат жинау бойынша жұмыс жасай алады. Мұндай серверлерде тұтынушылармен жұмыс істеу қосымша тапсырма болуы мүмкін.

Прокси сервер сияқты қызмет көрсететін сервер әрдайым жабдықтың қандай да бір түрімен жұмыс істейтін бағдарлама немесе бағдарлама модулі екенін түсінуі маңызды. Бұл бағдарлама болмаса, аппараттық құрал ештеңе бере алмайды.

6.3. ЖЕЛІЛІК ОПЕРАЦИЯЛЫҚ ЖҮЙЕ

Компьютердің операциялық жүйесі (ОС) компьютерлік ресурстардың (жад, процессор, сыртқы құрылғылар, файлдар және т.б.) тиімді басқарылуын қамтамасыз ететін басқару және өңдеу бағдарламаларының жиынтығы ретінде ұсынылуы мүмкін, сондай-ақ пайдаланушыға компьютерлік аппаратурамен жұмыс істеу және қосымшаларды әзірлеу үшін ыңғайлы интерфейсті ұсынады. Бұл анықтама қазіргі заманғы жалпы мақсаттағы операциялық жүйелерге қолданылады.

Желілік операциялық жүйе компьютерлік желілерде жұмыс істеуге арналған кірістірілген мүмкіндіктерге ие операциялық жүйе болып табылады.

Желілік операциялық жүйе қашықтағы компьютерлер мен желілік құрылғылардың ресурстарына қол жетімділікті қамтамасыз етеді және қашықтағы компьютерлерге ресурстар ұсынады. Компьютерлік желінің операциялық жүйесі көп жағдайда жергілікті компьютердің операциялық жүйесімен бірдей.

Желіні ұйымдастыру кезінде операциялық жүйе пайдаланушыдан төмен деңгейлі желілік бағдарламалық жасақтама мен аппараттық құралдардың жұмысының барлық мәліметтерін жасыратын интерфейсін рөлін атқарады. Мысалы, MAC мекенжайы және IP мекенжайы сияқты желілік компьютерлердің сандық мекенжайларының орнына, компьютер желісінің амалдық жүйесі есте сақтауға ыңғайлы символикалық атаулармен жұмыс жасауға мүмкіндік береді. Нәтижесінде, пайдаланушының көзқарасында, оның желісі бар.

күрделі нақты мәліметтер көп жалпыға ортақ ресурстардың жиынтығына айналады.

Желідегі әр түрлі компьютерлерде бірдей немесе түрлі операциялық жүйелер жұмыс істей алады. Мысалы, сол UNIX OS желідегі барлық компьютерлерде жұмыс істей алады. Неғұрлым шынайы опция - бұл әртүрлі операциялық жүйелер жұмыс істейтін желі, мысалы, кейбір компьютерлер UNIX OS басқаруында жұмыс істейді, екіншісі - NetWare жұмыс істейді, қалғандары Windows операциялық жүйесінде жұмыс істейді. Барлық осы операциялық жүйелер бір-бірінен тәуелсіз жұмыс істейді, олардың әрқайсысы өздерінің процестерін құру және тоқтату және жергілікті ресурстарды басқару туралы тәуелсіз шешімдер қабылдайды. Бірақ кез-келген жағдайда желіде жұмыс істейтін компьютерлердің операциялық жүйелері желідегі әртүрлі компьютерлерде жұмыс істейтін процестердің өзара әрекеттесуін ұйымдастыруға және осы компьютерлердің ресурстарын желілік пайдаланушылар арасында бөлісу үшін өзара байланысқан хаттамалар жиынтығын қамтуы тиіс.

Жергілікті компьютердің операциялық жүйесі желіде жұмыс істеуге мүмкіндік береді, яғни оның ресурстарын жалпы пайдалану үшін қамтамасыз ету және / немесе желідегі басқа компьютерлердің ресурстарын пайдалану, осы компьютердің операциялық жүйесі-ақ желілік OS жұмыс жасайды.

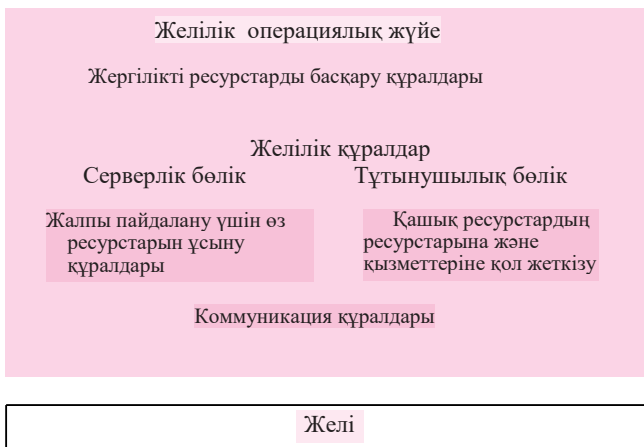
6.1 суретте желілік операциялық жүйенің негізгі функционалдык компоненттерін көрсетеді.

Жергілікті ресурстарды басқару құралдары ЖЖҚ процестер, жоспарлау және диспетчерлік процестер, көп процессорлық бақылау және жергілікті операциялық жүйелердің ресурстарын басқарудың басқа функциялары арасында бөлу функцияларын жүзеге асырады.

Желіні өз кезегінде үш компонентке бөлуге болады:

- 1) жалпы пайдаланылатын жергілікті ресурстар мен қызметтерді ұсыну амалдары - операциялық жүйенің серверлік бөлімі;
- 2) қашықтағы ресурстарға және қызметтерге - операциялық жүйенің клиенттік бөлігіне кіруге өтініш жасау құралдары;
- 3) коммуникациялық жүйемен бірге желілік компьютерлер арасында хабарларды беруді қамтамасыз ететін операциялық жүйе құралдары.

Операциялық жүйенің клиенттік бөлігі басқа компьютердің ресурстарына тікелей қол жеткізе алмайды. Ол тек «сұрайды», ол осы көлемде жұмыс істейтін операциялық жүйенің серверлік бөлігі болуы мүмкін.



6.1 сурет. Желілік операциялық жүйенің негізгі функционалдык компоненттері

Осы ресурстарға тиесілі компьютер. Бұл «сұраулар» желі арқылы жіберілген хабарлар түрінде көрсетіледі. Хабарлар белгілі бір әрекеттерді орындауға пәрмен, сонымен қатар осындай файл мазмұнын ретінде нақты деректерді ғана, болуы мүмкін.

Клиент пен серверлік бөліктер арасындағы хабар алмасуды желілік коммуникациялық жүйе ОС көлік құралдары арқылы басқарады. Бұл құралдар және т.б. хабарлар, бөлікке бөлшектемей хабарлар (бумалары, кадрлар), сандық мекенжайлары компьютерлік атауларының конверсиялау, сенімді хабар жеткізуді ұйымдастыру, күрделі желісінде бағытын анықтау, қалыптастыру сияқты функцияларды орындауға желі арқылы берілетін хабарлар компьютерлік өзара іс-қимыл ережелері және т.б. IPXі Ethernet, TokenRing, IP, сондай-ақ байланыс хаттамаларына жазылады. Екі компьютерлер желісі арқылы байланысу үшін, олардың операциялық жүйелерді көлік құралдары коммуникациялық хаттамаларды жалпы жиынтығын қолдау керек. Осы байланыс хаттамаларын іске асыру операциялық жүйе бағдарламалық модульдер, олардың мазмұнына бармайтын желілік операциялық жүйесі серверлер және клиенттер арасында хабар беру болып табылады.

Сонымен қатар, желілік операциялық жүйелердің клиенттік бөліктері сұрау форматтарын ресурстарға қайта өңдеуді жүзеге асырады. Жергілікті желілік ресурстарға қол жеткізу үшін қолданбалы бағдарламалардан сұраулар қабылданады.

Нысаны, яғни операциялық жүйенің жергілікті бөлігінде қабылданған пішінде жұмыс жасайды. Сұрау клиенттік бөлік бойынша сұралған ресурс орналасқан компьютерде жұмыс істейтін серверлік ОЖ бөлігінің талаптарына сәйкес келетін басқа пішінде желіге жіберіледі. Клиент бөлігі сонымен бірге серверлік бөліктен жауап алады және оларды жергілікті форматқа түрлендіреді, сондықтан бағдарлама үшін жергілікті және қашықтағы сұраулардың орындалуы ажыратылмайтын болып табылады.

Желілік операциялық жүйе пайдаланушыға желілік қызметтерді пайдаланатын желі қызметтерінің жиынтығын ұсынады. Осылайша, желілік қызмет желі пайдаланушыларын белгілі бір қызметтер жиынтығымен қамтамасыз ететін желілік компонент болып табылады. Бұл қызметтер кейде желілік қызметтер деп аталады. Айта кету керек, бұл термин техникалық әдебиетте «сервис», «қызмет көрсету», «қызмет» ретінде аударылады. Бұл терминдер кейде синонимдер ретінде пайдаланылуына қарамастан, кейбір жағдайларда осы терминдердің мағыналарының айырмашылығы іргелі сипатта екенін есте ұстау керек. Нақтырақ айтқанда, «сервис» шеңберінде біз белгілі бір қызметтер жиынтығын іске асыратын желілік компонент, ал «қызмет көрсету» дегеніміз - бұл қызмет ұсынатын қызметтер жиынтығының сипаттамасы. Осылайша, қызмет - бұл тұтынушы мен қызмет көрсетуші арасындағы (қызмет) интерфейс.

Әрбір қызмет желі ресурстарының нақты түрімен және / немесе осы ресурстарға қол жеткізудің белгілі бір әдісімен байланысты. Мысалы, басып шығару қызметі желінің пайдаланушыларына желінің ортақ принтерлеріне қатынауды қамтамасыз етеді және басып шығару қызметін ұсынады, ал пошта қызметі желі ақпараттық ресурсына - электрондық поштаға қол жетімділікті қамтамасыз етеді. Желілік қызметтердің арасында қарапайым пайдаланушыға бағытталмағандарды таңдауға болады. Мұндай қызметтер желіні ұйымдастыру үшін қолданылады.

Пайдаланушыларға, қосымшаларға және желілік әкімшілерге операциялық жүйе арқылы қызметтердің бай жиынтығы дәрежесі желі операциялық жүйелерінің жалпы жиынынан оның орнына байланысты.

Бірінші желілік операциялық жүйелер қазірдің өзінде бар жергілікті операциялық жүйенің және ондағы желілік хабтың жинағы болды. Сонымен қатар, жергілікті ОЖ негізгі желілік функцияларды орындайтын желілік қаптаманы басқару үшін қажетті ең аз желілік функциялармен жабдықталған. Дегенмен, болашақта желілік операциялық жүйелерді әзірлеушілер жұмыс істей бастағаннан бастап желілік операциялық жүйе барынша тиімді тәсіл деп санайды.

оның үстіне желіні құрастыру үшін арнайы ойластырылған және жобаланған. Осы операциялық жүйелерде желі функцияларын терең оның логикалық келісімді, пайдалану және модификациялау, ыңғайлығын, сондай-ақ жоғары өнімділікті қамтамасыз ететін жүйесі, негізгі модульдер кірістіріледі. Бұл тәсіл артықтығы жоқтығы маңызды. Барлық желі қызметтері сондай-ақ біріктірілген болса, яғни операциялық жүйенің құрамдас бөлігі ретінде қарастырылады, операциялық жүйенің барлық ішкі жұмыс желілік функцияларды орындауға оңтайландыруға болады.

Арнайы серверлері бар желілерде серверлер рөлінде жұмыс істеу үшін оңтайландырылған және серверлік операциялық жүйелер деп аталатын желілік операциялық жүйелердің арнайы нұсқалары қолданылады. Осы желілердегі қолданушы компьютерлер клиенттің операциялық жүйелерімен жұмыс істейді.

Сервер ретінде жұмыс істеу үшін операциялық жүйенің мамандануы - серверлік операциялардың тиімділігін арттырудың табиғи жолы. Мұндай өсімге деген қажеттілік жиі өте қатты сезіледі, әсіресе үлкен желіде. Егер желіде жүздеген немесе тіпті мыңдаған пайдаланушылар бар болса, ортақ ресурстарға сұраныстың қарқындылығы өте жоғары болуы мүмкін және сервер бұл ағындардың үлкен кешігуіне жол бермеуі керек.

Бұл мәселенің айқын шешімі - компьютерді қуатты аппараттық платформасы бар сервер ретінде пайдалану және сервер функцияларына оңтайландырылған операциялық жүйе. Аз функциялары, операциялық жүйесі жүзеге асырылуы тиімдірек. Олар ОС әзірлеушілер кейде олардан толық бас тартуға дейін, оның басқа да ерекшеліктерін кейбір ұстап тұруға мәжбүр сервер операцияларын оңтайландыру, жүзеге асырылуы мүмкін.

Дегенмен, кейбір серверлік ОС-дың тым тар мамандандырылуы олардың әлсіз жағы болып табылады. Мысалы, әмбебап бағдарламалау интерфейсінің болмауы және кейбір ОС-дағы қосымшаларды қорғау оны қолданбалы бағдарламаларды орындау ортасы ретінде пайдалануға мүмкіндік бермейді, бұл желідегі басқа серверлік операциялық жүйелерді қосу қажеттілігіне әкеледі. Сондықтан, көптеген серверлік операциялық жүйелерді әзірлеушілер функционалды шектеулерден бас тартады және серверлік ОС-ға оларды әмбебап сервер ретінде пайдалануға мүмкіндік беретін және тіпті клиент ОС ретінде пайдалануға мүмкіндік беретін барлық компоненттерді қамтиды. Мұндай серверлік операциялық жүйелерге озық әмбебап графикалық пайдаланушысына интерфейс арқылы беріледі.

Келесі түрде серверлік ОС класының айырықша ерекшеліктерін қалыптастыруы мүмкін:

- Мульти-процессорлық платформаларды қоса алғанда қуатты аппараттық платформаларды қолдау;
- бір мезгілде орындалған процестер мен желілік қосылыстардың көп санын қолдау;
- желілік жабдықты қолдау;
- желілік протоколдарға, маршруттау хаттамаларына қолдау көрсету;
- желілік трафикті сүзуді қолдау;
- орталықтандырылған желіні басқару үшін компоненттердің операциялық жүйесіне қосу (мысалы, анықтама қызметі немесе желіні пайдаланушыларға аутентификация және авторизациялау қызметтері);
- Желілік қызметтердің кең ауқымы.

Арнайы серверлері бар желілердегі клиенттік операциялық жүйелер, әдетте, олардың ұйымдастырылуын айтарлықтай жеңілдететін сервер функцияларынан босатылады. Клиент ОС әзірлеушілері пайдаланушы интерфейсіне және желілік қызметтердің клиенттік бөліктеріне назар аударады. Ең қарапайым клиенттік қосымшалар негізгі желілік қызметтерді ғана қолдайды - әдетте файлдық қызмет және баспа қызметі болып бөлінеді. Сонымен қатар, әмбебап деп аталатын клиенттер бар, олар желідегі барлық серверлермен жұмыс істеуге мүмкіндік беретін клиенттік бөліктердің кең спектрін қолдайды.

Көптеген компаниялар желі операциялық жүйелерін дамытады, сол операциялық жүйенің екі нұсқасын жасайды. Олардың біреуі серверлік операциялық жүйе ретінде жұмыс істеуге, ал екіншісі клиент ретінде жұмыс істеуге арналған. Бұл нұсқалар көбінесе бірдей негізгі кодқа негізделеді, бірақ қызметтер мен утилиталар жиынтығынан, сондай-ақ конфигурациялық параметрлермен ерекшеленеді, олардың кейбіреуі әдепкі бойынша орнатылады және өзгертілмейді.

6.4. БҰЛТТЫ СЕРВИСТЕР ЖӘНЕ ОНЫҢ МҮМКІНДІКТЕРІ

Бұлтты есептеулер (cloudcomputing) — жалпы есептегіш ресурстарға (деректерді беру желілері, серверлер, сақтау құрылғылары, қосымшалар және қызметтер) сұраныс бойынша кез келген жерде кең таралған және ыңғайлы желіге қолжетімділікті қамтамасыз ету моделі, ол дереу жедел қамтамасыз етілуі және ең аз операциялық шығыстар болып табылады.

Бастапқыда байланысты деректерді ішіндегі суреттер бұлтта сол желісінің бір бөлігі болып табылады компьютерлер жеңілдетілген белгілерді түскен атауы «бұлттық» компьютерлік жүйелер деп аталады. Бұл жағдайда, әрбір компьютер тәуелсіз бірлік ретінде қарастырылады және тұтастай бөлігі болып табылады. Тұтастай алғанда, пайдаланушымен жұмыс істеу кезінде тікелей қатысатын жүйенің аумағы. Сонымен, серверлер арқылы қол жетімді барлық желілік ресурстарды, шын мәнінде, «бұлтты» деп атауға болады.

Ең танымал файлдарды сақтау компьютердің жадында, және арнайы онлайн сақтау емес, мүмкіндік қызметтерді дамыту арқылы алынған. Мұндай функционалдық барлық бұлт иесі файлдар сақталатын сияқты, таңдау бұрын-соңды болмаған еркіндік пайдаланушыларға қамтамасыз ететін кез келген уақытта және Интернетке қосылған кез келген құрылғыдан қол жетімді болып табылады. Сондай-ақ бұлтты қызметтер жеке деректердің резервтік сақтауы ретінде қолданыла алады. Қатты диск жетіспеушілігінен туындаған жағдайда сіз бұрын бұлтқа сақталған барлық қажетті ақпаратты қайтара аласыз.

Сонымен қатар, бұлтты сақтау орны әртүрлі компьютерлік құрылғылар арасында деректерді синхрондау үшін пайдаланылуы мүмкін. Мысалы, сіз компьютердегі файлға өзгертулер жасауға және бұлттық қызмет синхрондалатын басқа құрылғыларда сақтай алуға, бұл түзету барлық данада енгізілетін болады сенімді бола аламыз.

Ең танымал бұлтты сақтауға жататындар: тіркелгеннен кейін GoogleDisk, Yandex.Disk, MicrosoftSkyDrive, Dropbox, iCloud т.б. Олардың барлығы дерлік файлдар (әдетте 5-тен 10 Гб) сақтау үшін бос белгілі бір сомаға пайдаланушыларға береді. Ол сәл көрінетін болса, онда қосымша ақыға, терабайт бірнеше ондаған дейін 20 Гб дейін сақтау сыйымдылығы ұйымдастыра алады.

Бұлтты бағдарламалар өкілдерінің бірі MicrosoftOffice365 кеңсе жиынтығы болып табылады. Office орнатылған жоқ екенін, тіпті, мұндай кеңселік пакеттің Word, Excel, PowerPoint ретінде ең танымал кеңсе бағдарламалардың барлығын қамтитын. Бұл кез келген жерден құжаттарға қол жеткізуді қамтамасыз етеді, онда Интернетке қол жеткізу болып табылады, және құрылғылардың түрлі оларға жұмыс істеуіне мүмкіндік береді. Сонымен қатар, пайдаланушылар, лездік хабар алмасу құжаттарға ынтымақтасуға қабілетті, және олардың деректер қауіпсіздік туралы алаңдамайды.

Заманауи пайдаланушысы бар барлық «бұлт» жүйелерін төрт санатқа бөлуге болады:

- 1) SaaS үлгісі (Software-as-a-Service) — бағдарламалық жасақтама немесе бағдарламалық қамтамасыз ету ретінде сұраныс бойынша;
- 2) PaaS үлгісі (Platform-as-a-Service) — платформа қызмет көрсету ретінде;
- 3) DaaS үлгісі (Desktop-as-a-Service) — жұмыс үстелі қызмет көрсету ретінде;
- 4) IaaS үлгісі (IaaS or Infrastructure-as-a-Service) — инфрақұрылым қызмет көрсету ретінде.

SaaS — бағдарламалық жасақтаманы сатуға және пайдалануға арналған бизнес үлгісі, онда сатушы веб-қосымшаны әзірлеп, басқарады, бұл Интернетте бағдарламалық қамтамасыз етуге тапсырыс берушінің кіруіне мүмкіндік береді. Қызметтің тұтынушысы үшін SaaS үлгісінің басты артықшылығы - онымен жұмыс істейтін аппараттық және бағдарламалық қамтамасыз етуді орнату, жаңарту және қызмет көрсету шығындарының болмауы.

SaaS үлгісінде қолданба қашықтан пайдалануға бейімделген және бір бағдарламаны бірнеше клиенттерді қолданады.

PaaS— пайдаланушы бұлттық провайдерінде орналастырылған ақпараттық технологиялар, платформалар, операциялық жүйелер, осы жүйелердің, аралық, дамыту құралдары мен сынауды басқару деректер базасын, пайдалануға қол жеткізе алады. Онда бұлтты есептеулердің моделі болады. Бұл модельде, компьютерлік желілер, соның ішінде ақпараттық технологиялардың инфрақұрылымын, барлық, серверлер, толығымен провайдері арқылы басқарылатын сақтау жүйесі, жеткізуші тұтынушылар мен платформа жиынтығы бақыланатын параметрлері платформалар үшін қол жетімді түрлерінің жиынтығымен анықталады, және тұтынушылық олардың виртуалды көшірмелерін жасауға алаңын пайдалануына мүмкіндік беріледі, тұтынылатын випи санын динамикалық түрде өзгерткен кезде, қолданбалы бағдарламалық жасақтаманы орнату, әзірлеу, тестілеу, пайдалану Ресурсын қабылдайды.

DaaS моделі SaaS-тың логикалық жалғасы болып табылады. DaaS қызметі клиенттерге толық дайын жұмыс орнында стандартталған виртуалды жұмыс орнын ұсынады, ол әрбір пайдаланушы өзінің тапсырмаларын реттеуге мүмкіндік алады. Осылайша, пайдаланушы жеке бағдарламаға емес, толыққанды жұмыс істеу үшін қажетті бағдарламалық жасақтамаға қол жеткізеді. Жұмыс орнындағы пайдаланушы физикалық тұрғыдан қол жеткізу үшін

жергілікті желі немесе Интернетті пайдалана алады. Қол жететін құрылғы жұқа клиент ретінде пайдаланылады және оған қойылатын талаптар аз болып келеді.

Бүгінгі таңда DaaS технологиясы перспективалы бұлт бағыты болып саналады және қажетті бағдарламалық жасақтамамен тәулік бойы жұмыс істеуге дайын толыққанды жұмыс станциясын орналастыру мүмкіндігін береді. Сонымен қатар, барлық қосымшалар арнайы деректер орталығында сақталады, бұл тіпті төмен қуатты компьютерлерде, ноутбуктерде, планшеттерде және тіпті смартфондарда бұл құралды тиімді пайдалануға мүмкіндік береді.

IaaS қызмет моделі пайдаланушыларға жабдықты (серверлерді, клиенттік жүйелерді, желілік жабдықтарды және т.б.), сондай-ақ операциялық жүйелерді және қажетті бағдарламалық қамтамасыз етуді лизинг арқылы бұлт қорларын дербес басқаруға мүмкіндік береді. Көбінесе, бұл технология корпоративтік клиенттерге өздерінің бұлтты есептеулер қызметін жасау үшін қолданылады. Тұтынушы операциялық жүйелерді, виртуалды сақтау жүйелерін және орнатылған қолданбаларды, сондай-ақ қол жетімді қызметтер жиынтығын шектеуді бақылай алады. Бұлттың негізгі физикалық және виртуалды инфрақұрылымын, соның ішінде желіні, серверлерді, қолданылатын операциялық жүйелердің түрлерін бақылау және басқару, бұлтты провайдер қамтамасыз етеді.

Бұлтты қызметтерді пайдалану оң және теріс жағына ие. Артықшылықтары шақырылуы мүмкін:

- интернет желісіне қосылған кез-келген құрылғыдан жеке деректерге кіру және кіру мүмкіндігі;
- резервтік деректерді сақтауды ұйымдастыру мүмкіндігі;
- барлық пайдаланушы құрылғыларындағы деректерді синхрондау;
- бұлттық қызметтерде жұмыс істейтін құрылғылардың бастапқы бағдарламалық жасақтамасына және аппараттық құралдарына қойылатын төмен талаптар;
- төмен қуатты жабдықтарға кешенді бағдарламалық қамтамасыз етуді қолдану мүмкіндігі.

Қиындықтарға байланысты болуы мүмкін:

- сақталған деректерге қол жеткізуді бақылау мүмкін болмайды;
- файлдарға немесе қосымшаларға кіру қажет болған сайын қызметке Интернет-байланыс орнату қажеттілігі туындайды;
- қызметті иеленетін нақты әзірлеушінің қызметтері мен ұсыныстарын пайдалану қажет.

6.5. ЖЕЛІЛІК ҚАУІПСІЗДІК ҚЫЗМЕТІ

Интернет-технологиялар желілік қауіпсіздікті қамтамасыз етуге деген көзқарасты өзгертеді. Ақпараттық қауіпсіздіктің кең ауқымы бар, Интернет арқылы ақпараттық ресурстарға қолжетімділікті басқарады, сондай-ақ ұйымның ішкі желілері арқылы және оның серіктестері арқылы ақпараттар мен желілік байланыстардың тұтастығы мен түпнұсқалығын басқарады. Артып келе жатқан сұранысты қанағаттандыратын ең жиі келтірілген сыртқы ақпаратты қорғау құралдарының ішінде келесі екі жүйені атап өту керек: брандмауэрлер және прокси-серверлер.

6.5.1. Прокси-сервер

Прокси-сервер (проху — өкіл, уәкілетті) — сервис немесе компьютерлік желілердегі бағдарламалар жиынтығы, бұл тұтынушыларға басқа желілік қызметтерге жанама сұрауды орындау мүмкіндігін береді.

Клиент прокси серверге қосылып, басқа серверде орналасқан ресурсты (мысалы, электрондық пошта) сұрайды, одан кейін прокси сервері көрсетілген серверге қосылады және одан ресурсты алады немесе ресурсты өзінің жеке кәшінен қайтарады. Прокси-сервері клиенттің компьютерін желілік шабуылдардан қорғауға мүмкіндік береді және клиенттің анонимділігін сақтауға көмектеседі.

Прокси серверлерді пайдаланудың мақсаттары келесі жолдармен тұжырымдалуы мүмкін.

1. *Деректерді кәштеу* — егер бірдей сыртқы ресурстарға жиі кіре алатын болсаңыз, олардың көшірмесін прокси-серверлерге сақтауға және сұраныс бойынша беруге болады, осылайша арнадағы жүктемені сыртқы желіге азайтады және клиент ақпарат алатын жылдамдығын арттырады.

2. *Деректерді сығу* — прокси сервер интернеттен ақпаратты жүктеп, оны сығылған нысанда соңғы пайдаланушыға жібереді. Мұндай прокси серверлер сыртқы желілік трафикті сақтауға мүмкіндік береді.

3. *Жергілікті желіні сыртқы қол жеткізуден қорғау*: мысалы, прокси-серверлерді жергілікті компьютерлерге сыртқы ресурстарға ғана қол жетімді етіп конфигурациялауға болады және сыртқы компьютерлер мүлдем жергілікті желілерге кіре алмайды (олар тек прокси серверді көреді).

4. *Жергілікті желіден сыртқыға қатынауды шектеу* (кейбір веб-сайттарға кіруге тыйым салуға, Интернетті кейбір жергілікті пайдаланушыларға шектеуге, трафикке шектеулер қоюға, қауіпті мазмұнды сүзуге болады).

5. *Әр түрлі ресурстарға қол жеткізудің құпиялылығы*. Прокси сервері сұраудың немесе пайдаланушының көзі туралы ақпаратты жасыра алады. Сондай-ақ, шынайы пайдаланушы туралы мақсатты серверге жалған ақпарат жіберетін прокси-серверлерді бұрмалайды.

6. *Шектеулі қолжетімділікті айналып өту*. Прокси серверлер белгілі бір ресурстарға қол жетімділік заңмен шектелген елдердің пайдаланушылары арасында танымал.

Кез-келген интернет-қолданушы қол жеткізе алатын прокси-сервер ашық деп аталады.

Прокси сервердің жұмысы төмендегідей қысқаша сипатталуы мүмкін. Белгілі бір хаттамаға арналған нақты бағдарлама немесе операциялық жүйенің барлық желілік қосылымдары сервердің IP-мекен-жайына емес, прокси-сервердің IP-мекен-жайына жіберіледі. Клиент компьютері прокси сервермен (қажет портта) желі байланысын ашады және тікелей ресурсқа сілтеме жасағаны секілді қалыпты сұрау жасайды. Сұрау деректерін танып, клиент компьютерінің дұрыстығын және рұқсаттарын тексеріп, прокси-сервері қосылымдарды бұзбастан, ресурспен тікелей жаңа желі қосылымын ашады және сол сұрауды жасайды. Деректерді (немесе қате туралы хабарды) алған прокси-сервер оны клиенттік компьютерге жібереді. Прокси сервері әр қолдау көрсетілетін хаттамаға арналған толық мүмкіндігі бар сервер және клиент болып табылады және хаттаманың әр сатысында әкімші анықтайтын кіру саясаттарын қолдануға мүмкіндігі бар.

Қазіргі уақытта, басқа желілік хаттамалардың ролінің өсуіне қарамастан, Интернетке қатынау жылдамдығына тарифтеуге көшу, сондай-ақ NAT функциясымен арзан аппараттық бағыттаушылардың пайда болуына қарамастан, прокси серверлер кәсіпорындарда кеңінен қолданылады.

6.5.2. Брандмауэр

Брандмауэр интернетте немесе желімен келген деректерді тексеретін бағдарламалық жасақтама немесе аппараттық кешен болып табылады және брандмауэр параметріне байланысты блоктар немесе оларды компьютерге көшіруге мүмкіндік береді.

Брандмауэр хакерлердің немесе зиянды бағдарламалық құралдардың компьютерге желі немесе интернеттен кіруіне жол бермейді және зиянды бағдарламалардың басқа компьютерлерге жіберілуіне жол бермейді.

Брандмауэр желіні екі немесе одан көп бөліктерге бөлуге және пакеттердің бір бөлігінен екіншісіне өту шарттарын анықтайтын ережелер жиынтығын іске асыруға мүмкіндік береді.

Әдетте, бұл шекара кәсіпорынның жергілікті желісі мен Интернеттің арасында болса да, ол кәсіпорынның жергілікті желісінде жүзеге асырылуы мүмкін. Осылайша, брандмауэр барлық трафиктің өзінен өтеді.

Әрбір бума үшін брандмауэр оны өткізіп жіберуге немесе оны тастауға шешім қабылдайды. Брандмауэр осы шешімдерді қабылдау үшін, бірқатар ережелерді анықтауы керек.

Барлық брандмауэрлер үш түрге бөлінеді:

- 1) Пакеттік сүзгілер;
- 2) Қолданбалы деңгей сервері;
- 3) Қосылу деңгейінің серверлері.

Барлық түрлері бір мезгілде бір брандмауэрде кездеседі.

Пакеттерді сүзу брандмауэрлері пакетті өткізіп жіберуге немесе осы пакеттің тақырыбындағы IP мекенжайларына, жалаушаларына немесе TCP порт нөмірлеріне қарап тастау туралы шешім қабылдайды. IP адресі және порт нөмірі - желілік және тасымалдау қабаты туралы ақпарат, тиісінше, бірақ пакеттік сүзгілер ақпараттар қабатының ақпаратын пайдаланады, себебі TCP / IP-дегі барлық стандартты қызметтер белгілі бір порт нөмірімен жұмыс жасайды.

Қолданбалы деңгейдегі серверлері бар брандмауэрдер брандмауэрде іске қосылған және осы қызметке қатысты барлық трафиктен өтетін серверге тән қызметтерді (FTP, электрондық пошта, www және т.б.) пайдаланады.

Осылайша, екі қосылыс клиент және брандмауэр үшін, және брандмауэрден бастап баратын клиент сервер арасындағы қарым қатынас қалыптасады.

Қолдау көрсетілетін серверлердің толық жиынтығы әр нақты брандмауэр үшін ерекшеленеді. Қолданбалы деңгейдегі серверлерді пайдалану сыртқы пайдаланушылардан жергілікті желі құрылымын жасыруға мүмкіндік береді, сондай-ақ пайдаланушы деңгейінде түпнұсқалықты растауға (сәйкестендіруді растауға) мүмкіндік береді. сыртқы әлеммен өзара іс-қимыл - қолданбалы деңгейдегі хаттама серверлер ең жоғары қорғау деңгейін қамтамасыз етуге барлық кіріс және шығыс трафикті толығымен бақылайтын бірнеше қосымшалар арқылы мүмкіндік береді.

Қосылыс деңгейінің сервері TCP қосылым аудармашы болып табылады. Пайдаланушы брандмауэрдегі белгілі бір портқа қосылым жасайды, содан соң брандмауэрдің басқа жағында тағайындалған жерге қосылуды жүзеге асырады. Сеанс барысында бұл аудармашы байтты екі бағытта сымға айналдырады. Әдетте, тағайындалған орын алдын-ала орнатылған, көптеген көздер болуы мүмкін. Өртүрлі порттарды пайдалану арқылы түрлі конфигурацияларды жасауға болады. Сервердің бұл түрі TCP негізіндегі кез-келген пайдаланушы анықтайтын қызмет үшін аудармашы жасауға, осы қызметке қатынасуды басқаруға, оны пайдалану статистикасын жинауға мүмкіндік береді.

Бірқатар брандмауэрлер виртуалды корпоративтік желілерді ұйымдастыруға, яғни интернеттегі бірнеше жергілікті желілерді бір виртуалды желіге біріктіруге мүмкіндік береді. Виртуалды желілер пайдаланушыларға жергілікті желілердің ашық байланысын ұйымдастыруға, шифрлау арқылы берілетін ақпаратты құпиялылықты және тұтастығын сақтауға мүмкіндік береді. Бұл жағдайда интернетті таратқанда, тек пайдаланушы деректері шифрланбайды, сонымен қатар желі мекенжайлары, порт нөмірлері және т.б. сияқты желілік ақпарат болып табылады.

Брандмауэр сыртқы желіге қосылу үшін құрылғылардың қолдау көрсетілетін түрлерін қолдана отырып, сыртқы бағыттаушы ретінде пайдалануға болады.

БАҚЫЛАУ СҰРАҚТАРЫ

1. «Клиент» және «Сервер» ұғымдарын анықтаңыз.
2. Клиент-сервердің сәулетін пайдаланудың артықшылықтары қандай?
3. Үш деңгейлі «клиент-сервер-өлеңдер» сәулетінің ерекшелігі қандай?
4. Желілік қызмет және желі қызметі дегеніміз не?
5. Желілік қызметтерге не жатады?
6. Пайдаланушы ұсынатын желі қызметтерінің жиынтығы неге байланысты?
7. Серверлік бағдарламалық қамтамасыз ету дегеніміз не?
8. Желілік операциялық жүйе дегеніміз не?
9. Желілік операциялық жүйенің негізгі функционалдық компоненттерін көрсетіңіз.

10. Операциялық жүйе мен сервер операциялық жүйесі арасындағы айырмашылық қандай?
11. ОС сервердің ерекшелігі қандай?
12. Желілік қосымшалар қалай жіктеледі?
13. Бұлтты қызметтердің анықтамасын қалыптастырыңыз.
14. Бұлтты жүйелердің негізгі санаттарын көрсетіңіз.
15. SaaS, PaaS, DaaS, IaaS үлгілерін анықтаңыз.
16. Бұлтты сервистерді пайдалану артықшылықтары мен кемшіліктері қандай? Қандай кемшіліктер бар?
17. Прокси-сервер не үшін пайдаланылады?
18. Брандмауэр дегеніміз не?



БӨЛІМ

ДЕРЕКТЕР ҚОРЫН ҚОРҒАУ МЕН ӨНДЕУ ТЕХНОЛОГИЯСЫ

- 7 Тарау. Деректер қоймасы, білім қоры, деректер қоры теориясының негізгі ережелері.
- 8 Тарау. Деректер үлгісі. 9 Тарау. Деректердің реляциялық үлгісі.
- 10 Тарау. Тұжырымдамалық, логикалық және физикалық деректер үлгісін құру принциптері.
- 11 Тарау. Деректер сәулеті.
- 12 Тарау. SQL негіздері
- 13 Тарау. Деректердің бүтіндігін қамтамасыз ету.
- 14 Тарау. Деректер қорын басқарудың

ДЕРЕКТЕР ҚОЙМАСЫ, БІЛІМ ҚОРЫ, ДЕРЕКТЕР ҚОРЫ ТЕОРИЯСЫНЫҢ НЕГІЗГІ ЕРЕЖЕЛЕРІ

7.1. ДЕРЕКТЕР ҚОРЫ ЖӘНЕ АҚПАРАТТЫҚ ЖҮЙЕ. НЕГІЗГІ АНЫҚТАМАЛАРЫ

Қазіргі әлемде ақпарат кез-келген қызмет саласының тиімділігін анықтайтын факторға айналады. Ақпараттық ағындар көбейіп келеді, деректерді өңдеу жылдамдығына қойылатын талаптар өсуде, енді операциялардың басым бөлігі қолмен орындалмайды, олар ең озық және перспективалық компьютерлік технологияларды қолдануды талап етеді. Қазіргі уақытта ақпараттық сақтау және өңдеу жүйесі үлкен рөл атқарады.

Ақпарат — нысандар, құбылыстар, процестер, айналадағы әлемдегі оқиғалар туралы ақпарат, олар туралы білімнің белгісіздігін азайтады.

Ақпарат толық, сенімді, уақтылы, дәйекті, барабар болуы керек. Ақпараттық жүйелердің негізгі мақсаты шешімдерді қабылдау үшін сенімді ақпаратты уақытында шығару болып табылады.

Ақпараттық жүйе — бұл кез келген өрістің міндеттерінде ақпаратты жинау, сақтау, өңдеу, іздеу және беруді қамтамасыз ететін техникалық және бағдарламалық құралдар жиынтығы.

Ақпараттық жүйелер ұзақ мерзімді сақтауды және ақпаратты сапалы талдауды қамтамасыз етеді, проблемаларды шешуге және жаңа ақпараттық өнімдерді жасауға көмектеседі.

Пәндік аумақ — ақпараттық жүйеде сақталатын және пайдаланылатын нақты әлемнің бөлігі.

Тақырыпты басқаруды ұйымдастыру және сайып келгенде автоматтандыру мақсатында зерттеу жұмыстары жүргізіледі. Пәндік аймақ нысандарды, осы нысандарды пайдаланатын процестерді, сондай-ақ көптеген пайдаланушыларды сипаттайды, олар

тақырып аймағының бірыңғай көрінісі болады. Тақырыптық аумақты талдау кез-келген ақпараттық жүйені құру алдында жұмыс жасайды. Белгілі бір ақпараттық жүйенің тақырыптық алаңы пайдаланушыларға қызығушылық тудыратын нақты нысандар жиынтығы ретінде қарастырылады. Бұл нысандардың әрқайсысы белгілі бір қасиеттер мен атрибуттар жиынына ие.

Ақпараттық нысан — тақырыптық аймақтың кейбір мәнін сипаттау - нақты әлемде бар немесе орын алған нысан, үрдіс, құбылыс немесе оқиға. Ақпараттық нысан - бұл логикалық байланысты ақпараттың жинағы, яғни ақпараттық нысандар арасындағы байланыстардың әр түрлі түрлері болып табылады.

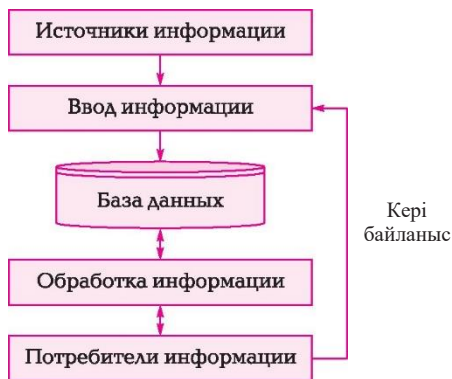
Ақпараттық жүйелерде өте күрделі құрылымы бар ақпараттық айналымның үлкен көлемі бар. Ақпараттық жүйе ең жалпы нысанда мынадай блоктан тұратын кесте түрінде шартты түрде ұсынылуы мүмкін (7.1-сурет):

- ақпаратты енгізу бірлігі (сыртқы ортадағы нысандардың жағдайы туралы ақпарат жинағы);
- деректер қоры (деректер қоймасы);
- ақпарат өңдеу бөлімі (іздеу, сүзу, сұрыптау, біріктіру, талдау, ақпарат шығару);
- кері байланыс бөлімі (енгізу туралы ақпаратты түзету үшін тұтынушы өндеген ақпараттың берілуі).

Дерекқор ақпараттық жүйенің негізі болып табылады, оның орталық байланысы болады.

Деректер — бұл ақпарат кейінгі өңдеу, беру және сақтау үшін жарамды кейбір формада бекітілген,

Источники информации- Ақпарат көздері, Ввод информации -Ақпаратты енгізу
База данных -Деректер қоры, Обработка информации - Деректерді өңдеу,



7.1 кесте. Ақпараттық жүйенің жалпы көрінісі

мысалы, компьютерде сақталған немесе компьютерге енгізу үшін дайындалған.

Деректер қоры (ДҚ) — бір-бірімен байланысты деректер жиынтығы, бірнеше пайдаланушы пайдаланатын нысандардың жай-күйін және олардың кейбір субъектілердің қарым-қатынастарын бейнелейді.

Деректер құрылымы деректердің жекелеген бөліктері (элементтері) арасындағы қатынастарды көрсететін ережелер мен шектеулер жиынтығы.

Деректерді өңдеу — бұл деректер жиындарын түрлендіруді орындайтын міндеттер жиынтығы. Деректерді өңдеу деректерді компьютерге енгізуді, деректерді кез-келген критерий бойынша іріктеуді, деректер құрылымын түрлендіруді, компьютердің сыртқы жадында деректерді беруді, есептеу немесе басқа өзгерістердің нәтижесі болып табылатын деректерді шығаруды (кестедегі немесе кейбір басқа пайдаланушы үшін ыңғайлы) білдіреді.

Деректерді өңдеу жүйесі (ДӨЖ) — бұл деректерді басқару тапсырмаларын орындайтын аппараттық және бағдарламалық жасақтама жиынтығы.

Деректерді басқару — деректерді өңдеу жүйесінің табысты жұмыс істеуі үшін қажетті деректердің бүкіл ауқымын қамтиды.

Метадеректер — бұл дерекқордың өз құрылымының сипаттамасы. Олар сондай-ақ «деректер туралы деректер» деп аталады. Бұл барлық дерекқор нысандары туралы ақпаратты қамтитын жүйелік кестелер. Метадеректер өңдеу бағдарламаларының деректерінің тәуелсіздігін қамтамасыз етеді. Деректер сипаттамасы деректермен бірге сақталған болса, қосымша деректер құрылымын өңдеу бағдарламаларын жазбастан деректерді сұрауға және өзгертуге болады.

Әрине, сақталатын және өңделген деректер құрылымдарының сапасы мен сенімділігіне қойылатын талаптар өте жоғары. Дерекқордың құрылымы деректердегі толық, дәйекті, толық болуы керек. Бұдан басқа, деректер базасын дәл деректерді дұрыс енгізуге және сенімді ақпаратты уақытында алу мүмкіндігін қамтамасыз етуге мүмкіндік беретін етіп жасау керек. Сондықтан дерекқор құрылымын жобалау, сондай-ақ оған қол жеткізу әдістерін таңдау өте маңызды міндет. Ақпараттық жүйеде дұрыс емес деректерді ұйымдастыру осы шарттарға сай келмейді және жүйенің өзін пайдалану үшін жарамсыз.

Дерекқорды басқару жүйесі туралы түсінік дерекқор тұжырымдамасымен тығыз байланысты.

Деректер қорын басқару жүйесі (ДҚБЖ) — деректер базасын құру және пайдалану бақылау үшін тілі мен бағдарламалық қамтамасыз ету жиынтығы.

Деректер базасын басқару жүйесін деректер алу, өңдеу, деректер құрылымын өзгерту және оларды талдау құралы болып табылады. ДҚБЖ деректерге қол бақыланатын, ыңғайлы, жылдам және өте маңыздысын қамтамасыз етеді, деректер қорын ақпарат жазбаны және өзгертуге мүмкіндік береді, сондай-ақ пайдаланушыға мөлшерлеу, деректер тұтастығын қамтамасыз ету үшін және пайдалануға базаны қамтамасыз ету құралдары бар. ДҚБЖ әдетте құрылымдау автоматтандырылған құралдарын біріктіретін және құпиялылығын қамтамасыз ету құралдарымен деректерді манипуляциялау, бірнеше ортада ақпаратты қалпына келтіру және сақтау.

Дерекқорды басқару жүйесі ең алдымен кәсіби әзірлеушілерге арналған. Орташа пайдаланушы дерекқорды басқарады және қосымшалар деп аталатын арнайы қосымшаларды пайдаланып деректермен жұмыс істейді. Қолданбалар деректер базасына кіру құралдарын пайдаланатын бағдарламалау жүйесімен қоршаған ортада немесе ДҚБЖ ортасынан тыс жасалуы мүмкін.

Жалпы жағдайда көптеген әртүрлі бағдарламалар сол дерекқормен жұмыс істей алады. Мысалы, егер дерекқор белгілі бір кәсіпорынды модельдесе, онда онымен жұмыс істеу үшін бірнеше бағдарламалар жасалуы мүмкін: қызметкерлерді есепке алудың кіші жүйесі, қызметкерлердің жалақысын есептеудің шағын жүйесі немесе қоймалық есепке алудың кіші жүйесі және т.б. Бір дерекқормен жұмыс істейтін қосымшаларды қарастыру кезінде деректердің бір-біріне параллель және бір-біріне тәуелсіз жұмыс жасайтындығы болжанады. Бұл әртүрлі өтімдердің бір дерекқормен жұмыс істеуін қамтамасыз етуі тиіс, олардың әрқайсысы дұрыс орындалады, бірақ басқа қосымшалармен жасалған дерекқордағы барлық өзгерістерді ескереді (7.2 сурет)).

Бір пайдаланушылық жүйесі — бұл бір уақытта бір пайдаланушыдан дерекқорға қол жеткізе алмайтын жүйе.

Көп пайдаланушылық жүйесі — бұл бірнеше пайдаланушылар дерекқорға бір уақытта кіруге болатын жүйе.

Көптеген пайдаланушылардың жүйелерінің негізгі міндеті әрбір жеке пайдаланушының бір жүйедегі жүйе ретінде жұмыс істеуіне мүмкіндік беру болып табылады. Бір пайдаланушылық жүйесі мен

көп пайдаланушылық жүйелер олардың ішкі құрылымында; олар іс жүзінде көрінбейді. Ұйымның қазіргі заманғы формасы сақтау және ақпаратқа қол жеткізу болып табылады. Деректер банкі орналасқан. Әртүрлі деректер банкі анықтау. Олардың ең толықтығы құрастырылған.

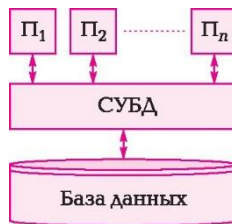
Деректер банкі (ДБ) — бұл жүйе арнайы ұйымдастырылған деректер (дерекқор), бағдарламалық қамтамасыз ету, тіл ұйымдастырушылық, әдістемелік орталықтандырылған қамтамасыз ету үшін арналған деректер жинақтау және ұжымдық көп мақсатты пайдалану болып табылады.

Деректер банкі автоматтандырылған деректерді өңдеу үшін қажетті барлық қолдау көрсету үшін кіші жүйелерін қамтитын күрделі жүйе болып табылады. Бұл анықтамада деректер банктерінің негізгі ерекшеліктері көрсетілген. Ең алдымен, дерекқорлар әдетте бір пайдаланушы үшін бір тапсырманы шешу үшін емес, көп мақсатты пайдалану үшін жасалады. Деректер банктерінің тағы бір ерекшелігі - бұл пайдаланушыларға деректерді сақтау, реттеу және кіруге байланысты барлық операцияларды орындауды жеңілдететін арнайы тіл мен бағдарламалық құралдардың болуы. Осылайша, деректер банкі деректер базасын жинау тиісті деректер базасын басқару жүйесі болып табылады.

Деректер базасы - белгілі бір тәртіпте ұйымдастырылған ақпаратпен компьютердегі орталықтандырылған сақтау орны. ДҚБЖ - деректер қорын құру функцияларын орындайтын, оны жұмыс тәртібінде ұстайтын, өңдеу бағдарламаларына қажетті ақпаратты беретін бағдарламалар жиынтығы.

Деректер сөздігі (ДС) деректер құрылымы туралы деректерді орталықтандырылған сақтауға, дерекқор файлдарының бір-бірімен өзара байланысы, оларды ұсыну түрлері мен форматтарының түрлері, деректерге қол жеткізуді шектеу және т.б. деректер банкінің кіші жүйесі болып табылады. Деректер сөздігі - деректер туралы ақпаратты сақтау және бақылау үшін деректер базасын пайдалану және дамыту.

Кейбір дерекқорларда бұл деректерді орталықтандырылған сақтамайды, компьютерлік желіде таратылады.



7.2 сурет. Қолданбалы бағдарламалар (бағдарламалар) мен

База данных Деректер қоры

Бөлінген деректер қоры (БДҚ) — компьютерлік желінің әртүрлі түйіндерінде орналасқан бірнеше дерекқордан үзінділерді қамтитын дерекқор және әртүрлі ДҚБЖ басқарады.

Бөлінген дерекқорлар қарапайым жергілікті дерекқор ретінде пайдаланушылардың және бағдарлама бағдарламаларының көзқарасынан көрінеді. Бөлінген дерекқор қашықтағы деректерді, сенімділікті, қол жетімділікті және жүйенің өнімділігін ортақ пайдалану мүмкіндігін жақсартады, бұл шығындарды үнемдеуге және масштабтауға әкеледі.

Заманауи ақпараттық жүйелер тек дерекқорларды ғана емес, білім базаларында қолданады.

Білім — бұл осы саладағы мәселелерді шешуге мүмкіндік беретін тақырыптық саланың анықталған заңдылықтары (принциптер, сілтемелер, заңдар).

Білім қоры — пәндік аумақтағы білім деңгейі.

Білімді пайдаланушы жүйені *зияткерлік* деп атайды.

Деректер базасының теориясының негізгі ұғымдарының мағынасын ашқаннан кейін оларды дамыту және енгізу сұрақтарына жүгіне алады. Бірақ жоғарыдағы тұжырымдар мен анықтамаларға мерзімді түрде (курстың келесі бөлімдерін оқып болғаннан кейін) дәлірек ассимиляциялау үшін қайтару ұсынылады.

Қорытындылай келе пайдаланушылардың негізгі санаттарын және дерекқор әзірлеушілерін және олардың деректер банкінің жұмысындағы рөлін анықтайды.

Соңғы пайдаланушылар — бұл мүдделер базасында жасалатын пайдаланушылардың негізгі санаты. Кейбір ақпаратты алу үшін (мысалы, өнімдер немесе қызметтер каталогын қарайтын фирмалардың тұтынушылары) деректерді алу үшін кездейсоқ пайдаланушылары болуы мүмкін, тұрақты пайдаланушылары болуы мүмкін (мысалы, өздері үшін арнайы әзірленген бағдарламалармен жұмыс істейтін қызметкерлер (қызметтік міндеттер)). Соңғы пайдаланушылардан компьютерлік технологиялар мен бағдарламалау саласында арнайы білім болмауы керек.

Қосымшалардың әзірлеушілері мен әкімшілері — бұл дерекқорды жасау және қайта құру кезінде жұмыс істейтін пайдаланушылар тобы. Қолданба әкімшілері функционалды шағын жүйеге біріктірілген нақты қосымшаны немесе қосымшаларды өңдеу кезінде әзірлеушілердің жұмысын үйлестіреді. Арнайы қосымшаларда әзірлеушілер жұмыс істейді

деректер базасынан белгілі бір ақпарат алу үшін қажет.

Пайдаланушылардың барлық түрлері таңдалуы мүмкін емес. Жеке ДҚБЖ көмегімен ақпараттық жүйелерді әзірлеу кезінде DBA әкімшісі, бағдарлама әкімшісі және әзірлеуші жиі бір адамда бар. Дегенмен ірі фирмаларда немесе корпорацияларда бизнес-процестердің барлығын немесе көпшілігін автоматтандыру үшін пайдаланылатын күрделі корпоративтік дерекқорларды құру кезінде қосымшалар әкімшілері мен даму бөлімдері топтары болуы мүмкін.

Ең күрделі міндеттер дерекқордың әкімшілік тобына тағайындалады.

Дерек қорларының әкімшілері — дерекқорды әзірлеудің бастапқы кезеңінде оны оңтайлы ұйымдастыру және соңғы пайдаланушыларды бір уақытта пайдалану үшін жауапты пайдаланушылар тобы. Даму және қайта құру кезеңінде бұл пайдаланушылар тобы осы операцияны өзгертпей немесе тоқтатпай дерекқорды дұрыс қайта құру мүмкіндігіне жауапты болып табылады.

Әкімшілік топтың құрамында ДҚ болуы керек:

- жүйелік талдаушылар;
- деректер құрылымдарының дизайнерлері және сыртқы ақпаратқа қатысты деректер банкі;
- деректерді өңдеу процесін жобалаушылар;
- жүйелік және қолданбалы бағдарламашылар;
- операторлар және техникалық қызмет көрсету бойынша мамандар.

7.2. ДЕРЕКТЕР ҚОРЫН БАСҚАРУ ЖҮЙЕСІ. СУБД НЕГІЗГІ ФУНКЦИЯСЫ

Қолданбалы бағдарлама тұрғысынан файл - бұл деректерді оқи алатын және жаза алатын сыртқы жады деп аталатын аумақ.

Файлдарды атау ережелері, файлда сақталған деректерге қол жеткізу жолы және осы деректер құрылымы нақты файлды басқару жүйесі мен файл түріне байланысты болады. Файлды басқару жүйесі сыртқы жадты бөлу функцияларын қабылдайды, файл есімдерін сыртқы жадыдағы сәйкес мекенжайларға салыстыру, деректерге қол жетімділікті қамтамасыз етеді. Алайда бұл қарапайым ақпараттық жүйелерді құру үшін жеткіліксіз. Файлды басқару жүйелері мүмкіндік бермейді.

логикалық түрде үйлестірілген файлдар жиынтығын қолдау, деректерді басқаруға арналған тіл қамтамасыз ету, әр түрлі ақаулардан кейін ақпаратты қалпына келтіру және бірнеше пайдаланушылардың бір уақытта жұмыс істеуі. Бағдарлама бағдарламасы осы қасиеттері бар кейбір деректерді басқару жүйесіне сүйенуі керек. Мұндай жүйе дерекқорды басқару жүйесі болып табылады.

Деректерді өңдеу жүйелері нақты деректерді, дерекқорды басқару жүйесін және дерекқорды басқару жүйесі арқылы деректерге қол жеткізетін қолданбалы бағдарламаны қамтиды.

Деректер қорын басқару жүйелері мынадай функциялары бөлінуі мүмкін:

- сыртқы жадыдағы деректерді басқару;
- белсенді жадыдағы деректерді басқару;
- транзакцияларды басқару;
- тіркеу, сақтық көшірме жасау және қалпына келтіру;
- ДҚ тілдерін қолдау.

Сыртқы жадыдағы деректерді басқару (дискідегі). ДББЖ пайдаланушыларға мүмкіндікті қамтамасыз етуі тиіс:

- дерекқорда деректерді сақтау, алу және жаңарту;;
- деректерге қол жеткізуді бақылау;
- бірнеше қолданушылардың қатарлас жұмысын қамтамасыз ету;;
- деректер тұтастығын сақтау.

Деректерді алу, сақтау және жаңалау деректер қорында — ДҚБЖ-ның ең негізгі функциясы. Және бұл функцияны орындау жолы соңғы пайдаланушыдан (ДББЖ файлдық жүйені пайдаланады ма, файлдар қалай ұйымдастырылғандығын және т.б.) жасырын болуы керек.

Деректерге қол жетімділікті бақылау — бұл құпия сөзді қорғауды, дерекқорға қол жеткізу деңгейін және оның жекелеген элементтерін және т.б. қолдана отырып, дерекқорға рұқсаты бар тек рұқсат беру мүмкіндігі. Әрбір пайдаланушы өзінің дерекқорынан оның пайдаланушының құқықтарына сәйкес қол жетімді деректермен жұмыс істей алады. Осылайша, деректерді басқару жүйелерінде қауіпсіздік қамтамасыз етіледі.

Параллелизмді басқару ДҚБЖ-нің бір мезгілде қол жетімділігімен көптеген пайдаланушылардың деректерді дұрыс жаңартуға кепілдік беретін механизмі бар. Бірлескен операциядағы қақтығыстар мәліметтердің логикалық тұтастығын бұзуға әкелуі мүмкін, сондықтан жүйе деректерді басқа біреу пайдаланған кезде пайдаланушыны деректерді жаңартуға кедергі келтіретін шараларды қамтамасыз етуі керек. Сипатталған жағдайда «бұғаттау» сияқты атаулар пайдаланылады.

Бұғатталған жазбалардың саны бойынша бір-бірінен ерекшеленетін құлыптардың әртүрлі түрі - кесте, бет, жол және басқалары бар.

Деректер бүтіндігін қолдау ол деректер мен олардың өзгерістері белгіленген ережелерге сәйкес келетін бақылау құралдары арқылы жүзеге асырылады. Дерекқордың тұтастығы дерекқордың сипаты болып табылады, яғни ол толық, дәйекті және тиісті түрде көрсетілетін тақырыптық аймақ туралы ақпаратты қамтиды. Дерекқордың тұтастығын сақтау дерекқордағы сәйкессіздік болған жағдайда тұтастықты тексеруді және оны қалпына келтіруді қамтиды.

Дерекқордың тұтастығы дерекқорда сақталған деректер қанағаттандыруы қажет шарттар түрінде тұтастығы шектеулерімен сипатталады. Осындай жағдайлардың мысалы дерекқорда сақталатын нысандар атрибуттарының ықтимал мәндерінің ауқымын шектеу немесе салыстырмалы дерекқорлар кестелеріндегі қайталанатын жазбалардың болмауы болып табылады.

Белсенді жадыда деректерді басқару. ДҚБЖ жұмыс істейтін деректер базасының көлемі өте үлкен - әдетте, ол ЖЖҚ қол жетімді мөлшерінен әлдеқайда көп. Әрбір деректер элементіне қол жеткізу кезінде сыртқы жадымен алмасу орындалса, бүкіл жүйе сыртқы жады құрылғысының жылдамдығымен жұмыс істейтін болады. Іс жүзінде бұл жылдамдықты көбейтудің жалғыз жолы - ЖЖҚ деректерін буферлеу.

Буферлер - сыртқы және оперативті жады арасында алмасуды жылдамдатуға арналған ЖЖҚ аумақтары. Аралық деректер базасына уақытша деректер қорын сақтайды, деректерді деректер базасына кіргенде пайдалануға арналған немесе өңдеуден кейін дерекқорға жазуға жоспарланған.

Дамып келе жатқан ДҚБЖ аралық буферлерді өздерінің жиынтығын қолданады, олар аралық буферлерді өзгертуге арналған ережелерімен ерекшеленеді. Болашақта компьютердің саны соншалықты үлкен болады, бұл буферлинг туралы алаңдамауға мүмкіндік береді.

Транзакцияны басқару. *Транзакция*— бұл деректер базасы бойынша ДҚБЖ-нің бір бүтін деп есептейтін әрекеттер жиынтығы, яғни барлық операциялар орындалатын деректермен операциялардың кезектілігі немесе олардың ешқайсысы («барлығы немесе ештеңе» принципі) болып табылады.

Егер транзакция сәтті болса, ДҚБЖ осы транзакциямен жасалатын дерекқордың өзгерістерін сыртқы жадыда анықтайды,

немесе операциядағы барлық өзгерістер жойылып, олардың ешқайсысы дерекқор күйіне әсер етпейді.

Транзакцияның тұжырымдамасы дерекқордың логикалық тұтастығын сақтау үшін қажет. Транзакцияның мысалы банк жүйесінің бір шотыдан екіншісіне ақша аудару операциясы болып табылады. Сіз барлық әрекеттерді орындауыңыз керек (бір клиенттің есебін көбейтіңіз және басқа есептік жазбаны азайтыңыз) немесе осы әрекеттердің біреуін орындамайды. Бір шот бойынша ақша сомасын азайта алмайды, бірақ екінші жағынан ақша көлемін арттырмайды.

Алғашқы іс-әрекеттерден кейін (шотқа ақша сомасын азайту) сәтсіздікке ұшырады. Мысалы, клиенттік компьютерді дерекқордан ажыратуға болады немесе клиенттік компьютерде жүйелік қате орын алуы мүмкін, нәтижесінде операциялық жүйенің қайта іске қосылуына әкелді. Бұл жағдайда дерекқормен не болды? Алғашқы клиенттің есебінен ақшаны азайтуға арналған пәрмен жасалды, ал екінші пәрмен - басқа шотқа ақшаны көбейту үшін - жоқ, бұл қайшы, сәйкессіз дерекқор күйіне әкеледі. Алдымен олар бір шоттан ақша алып, содан кейін оларды басқа есептік жазбаға қосады. Егер кем дегенде бір әрекеттің сәтсіздігі орын алса, операция нәтижесі дұрыс болмайды және шоттар арасындағы теңгерім бұзылады. Демек, бұл жағдайда жүйе операциялар тізбегі басталғанға дейін бұрынғы күйге оралуы керек.

Транзакцияны басқару бір операциялық және көп мәртелік деректер базасында маңызды, мұнда транзакциялар параллельде орындалуы мүмкін.

Егер параллельді бірнеше мәміле жасалса, онда қарама-қайшылықтар да ДББЖ функциясы болып табылады. Осындай жағдайларды анықтаған кезде транзакция әдетте кері қайтарылады, яғни бір немесе бірнеше мәмілелер жасаған өзгерістерді жою. Мәліметтер тұтастығы мәселелерін зерттегенде мәміле механизмі мен мәмілені басқару туралы толығырақ қарастырылады.

Тіркеу, сақтық көшірме жасау және қалпына келтіру. Компьютер жұмыс істеп тұрған кезде ақаулық орын алуы мүмкін (мысалы, электр қуаты өшіп қалуына байланысты) және компьютерді сақтау құрылғысына зиян тигізуі мүмкін. Деректер бұзылуы мүмкін, бұл одан әрі жұмыс істеу мүмкін емес. ДҚБЖ-ға қойылатын негізгі талаптардың бірі деректерді физикалық және логикалық тұтастығын қорғауды қоса алғанда сыртқы жадыда сақтаудың сенімділігі болып табылады. Сақтау сенімділігі ДББЖ деректер базасының соңғы келісілген жағдайын қалпына келтіруге қабілетті болуы тиіс

кез-келген аппараттық құрал немесе бағдарламалық құрал ақаулығынан кейін (логикалық немесе физикалық ақаулық) жүзеге асады.

Физикалық тұтастықтың сақталуы кіргеннен кейін дерекқорды резервтеу және қалпына келтіруді қамтиды.

Өзгерістерді тіркеу ең қарапайым жағдайда дерекқорда жасалған барлық өзгерістердің сыртқы жадына дәйекті түрде жазу кіреді. Төмендегі ақпарат жазылады: өзгерудің реттік нөмірі, түрі және уақыты; мәміле идентификаторы; өзгертілген нысан (сақталатын файлдың нөмірі және оған деректер блогының нөмірі, блоктың ішіндегі жолдың нөмірі); алдыңғы және жаңа нысанның жағдайы.

Дерекқорды өзгерту журналы - осылайша жасалған ақпарат. Журнал мәміленің бастапқы және соңғы белгілерін және бақылау нүктесінің қабылдауын қамтиды. Бұл ДҚБЖ пайдаланушылары үшін қол жетімсіз дерекқордың ерекше бөлігі. Тіркеу деректер базасының негізгі бөлігіндегі барлық өзгерістерді жазатын үлкен қамқорлықпен қамтамасыз етіледі. Кейде әртүрлі физикалық дискілерде орналасқан журналдардың екі данасы жасалады және сақталады.

Деректер қорына сақтақы көшірме жасау — бүлінген немесе бұзылған жағдайда түпнұсқа немесе жаңа орынға деректерді қалпына келтіруге арналған ортаға арналған деректердің көшірмесін жасау процесі.

Деректер қорын қалпына келтіру — бұл логикалық және физикалық сәтсіздіктер жағдайында дерекқорды ағымдағы күйге келтіретін ДҚБЖ функциясы. Жоғарыда айтылғандай, жүйенің немесе сақтаудың сәтсіздіктен, аппараттық және бағдарламалық жасақтама қателіктерінен, ДҚБЖ жұмысын тоқтатуына әкелуі мүмкін қателер орын алуы мүмкін. Барлық осындай жағдайларда ДҚБЖ дерекқорды қалпына келтіру және оның дәйекті күйіне оралу механизмін қамтамасыз етуі керек.

Физикалық сәтсіздік жағдайында, егер өзгерістер журналы мен дерекқордың өзі зақымдалған болса, қалпына келтіру тек сақтық көшірме жасау кезінде ғана мүмкін болады.

ДҚ тілдерін қолдау. Дерекқорлармен жұмыс істеу үшін арнайы тілдер пайдаланылады, әдетте дерекқор тілдері деп аталады. Заманауи ДҚБЖ әдетте деректер базасымен жұмыс жасаудың барлық қажетті құралдарын қамтитын біртұтас интеграцияланған тілін қолдайды, оны құрудан бастап, деректер базасымен негізгі пайдаланушылық интерфейсті қамтамасыз етеді.

Қазіргі уақытта ең кең таралған реляциялық ДҚБЖ стандартты тілі - SQL тілі (StructuredQueryLanguage).

Негізгі функциялардан басқа, ДҚБЖ түрлі қосалқы қызметтерді ұсынады. Көмекші утилиталар, әдетте, деректер базасын тиімді басқаруға арналған:

- деректер экспорты/импорты;
- дерекқордың мониторингі - дерекқор жұмысының сипаттамаларын бақылау және пайдалану;
- деректер базасын пайдалану дәрежесін немесе дәрежесін бағалау үшін статистикалық талдау;
- индекстерді қайта құру;
- «қоқыс» жинау (пайдаланылмаған жазбалар) және сақтау құрылғысынан жойылған жазбаларды физикалық тұрғыдан жою, бос орынды біріктіру және қажет болса, жадты қайта бөлу үшін жадты қайта бөлу. Деректер базасын басқару жүйесін ұйымдастыру және оның компоненттерінің құрамы біз қарастыратын функциялар жиынына сәйкес келеді.

Деректер базасын басқару жүйесінде дерекқордың қозғалтқышын, деректер базасының тілдік процессорын, уақытты қолдаудың ішкі жүйесін және қосымша дерекқорға техникалық қызмет көрсету мүмкіндігін беретін қызметтік бағдарламаларды (сыртқы утилиталар) таңдай алады. Кейбір жүйелерде, бұл бірлік анық ерекшеленеді, бірақ қисынды бұл бөлу барлық деректер базасын басқару жүйелерінде жүзеге асырылуы мүмкін.

Сыртқы жадта деректерді басқару үшін жауапты, жад басқару, операциялық басқару және журналға тіркеу болып табылады. Басқаша айтқанда, ДҚБЖ-ның негізі - дерекқорды жасау және қолдау үшін қажетті және жеткілікті бағдарламалық модульдер жиынтығы - пайдаланушылардың ақпараттық қамтамасыз етудің стандартты міндеттерін шешетін әмбебап бөлік.

Сервистік бағдарламалық сипатталған пән аймағында және нақты пайдаланушы қажеттіліктеріне байланысты қосымша мүмкіндіктер мен қызметтерге бірқатар пайдаланушыларға береді.

ДҚ тілі процессор машинасының кодында кейбір орындалатын бағдарламаларына деректер базасының тілі есептілігін құрайды.

Соңында жеке дерекқор коммуналдық статистикасын жинау, жаһандық деректер базасының тұтастығын тексеру және т.б. тиеу және түсіру сияқты рәсімдерді, оқшауланған жүйеде қарастырады.

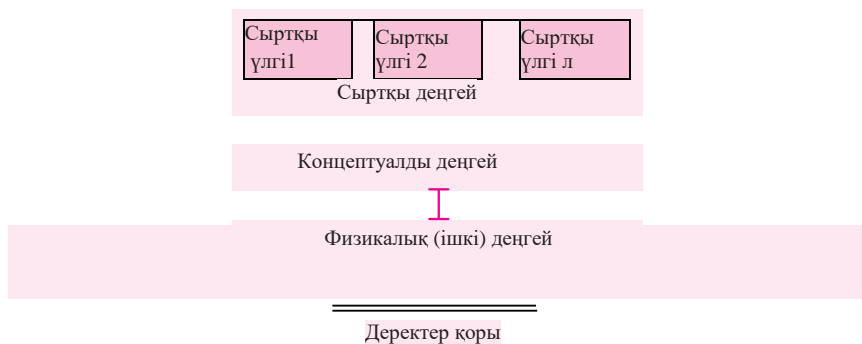
Утилиттер ДҚБЖ ядросы интерфейсін, ал кейде ядроға ену арқылы бағдарламаланған.

7.3. ДЕРЕКТЕР ҚОРЫНЫҢ СӘУЛЕТІ. ФИЗИКАЛЫҚ ЖӘНЕ ЛОГИКАЛЫҚ ТӘУЕЛСІЗДІК

Көп деңгейлі дерекқордың сәулет идеясы бірінші 1975 жылы жарияланған, Американдық ұлттық стандарттар институты (ANSI) стандарттау жөніндегі комитетінің баяндамасында тұжырымдалған болатын. Ол, сыртқы тұжырымдамалық және физикалық (ішкі) деңгейдегі (7.3 сурет), оның ішінде жалпылама үш деңгейлі сәулет үлгісі деректер базасына ұсынған болатын. Осы сәулет енгізудің басты мақсаты оның физикалық өкілдігінің жылдық деректер базасын пайдаланушы көрінісін бөлу болып табылады. Дерекқор сәулеті түрлі деңгейлерде түрлі деңгейдегі абстракцияның деректерін қолдайды.

Дерекқор сәулетін пайдаланушы және жүйенің деңгейдегі бірінші аражігін қамтамасыз етуі тиіс.

Сыртқы үлгі деңгейі — әр үлгіде өздерінің деректерді көрсетуі бар ең жоғары деңгей. Пайдаланушылардың жеке топтары тек осы бағдарлама ішінде қол жетімді деректермен жұмыс істейді, яғни әр қолданба тек осы бағдарламада талап етілетін деректерді көреді және өңдейді.



7.3 сурет. Деректер базасын басқарудың үш деңгейлі үлгісі

Дерекқор компаниясына ұқсайтын мысалы, егер, содан кейін онымен жұмыс істеуге бірнеше бағдарламалары болуы мүмкін. Ішкі жүйесі бухгалтерлік кадрлардың қызметін қолдану, қызметкерлердің жеке деректерін пайдаланады - және т.б. оның туған жері, төлқұжат нөмірі, мекен-жайы, күні және қоймасындағы тауарларды қадағалап, қызметкерлер ретінде қарастырылады және тауарлар мен жеке ақпарат сатпайды, бірақ тауарлар мен жеткізушілер туралы ақпаратты қажет етеді. Сонымен қатар, бұл ақпарат бір дерекқорда сақталады.

Түрлі бағдарламаларды әртүрлі бағдарламалау тілдері арқылы жасауға болады. Қолдану бағдарламашылар жиі жоғары деңгейдегі тілдерді немесе арнайы тілдер ДҚБЖ пайдаланады.

Бір дерекқормен жұмыс істейтін қосымшаларды қараған кезде, олар бір-бірінен параллель және бір-біріне қарамастан жұмыс істей алатынын болжайды. Олардың әрқайсысы дұрыс орындалады, бірақ назарға басқа деректер бағдарламалар арқылы жасалған базасында барлық өзгерістерді қабылдайды, сондықтан, бұл дерекқор бір деректер базасымен бірнеше қосымшаларды қамтамасыз етуі тиіс.

Пайдаланушы тобына арналған деректерді ұсыну сипаттамасы *сыртқы кесте* деп аталады. Дерекқор жүйесінде бірнеше сыртқы схемалар әр түрлі пайдаланушылар немесе тапсырмалар топтары үшін бір уақытта қолдана алады.

Сәулеттің концептуалды деңгейі негізгі болып табылады және оның барлық қосымшалары үшін және олардан тәуелсіз дерекқорды жалпы түрде көрсету үшін қызмет етеді. Тұжырымдамалық деңгей - бұл үш деңгейлі сәулеттегі аралық деңгей және дерекқордың барлық мәліметтерін дерексіз түрде ұсынуды қамтамасыз етеді. Бұл доменнің формалды ақпараттық логикалық моделі, яғни ол ұсынылған және сақталатынына қарамастан, домендегі деректерге қойылатын талаптардың толық көрінісі. Бұл деңгейдегі дерекқордың сипаттамасы *концептуалдық кесте* деп аталады. Тұжырымдамалық құрылымға нысандар мен олардың атрибуттары, объектілер арасындағы қатынастар, деректерге қойылған шектеулер, деректер туралы семантикалық ақпарат, қауіпсіздікті қамтамасыз ету және деректер тұтастығын сақтау кіреді.

Концептуалды кесте— бұл барлық деректер элементтерінің бірыңғай логикалық сипаттамасы және олардың арасындағы қатынастар, бүкіл дерекқордың логикалық құрылымы. Кез-келген үлгі сияқты, концептуалды кесте өңдеудің тұрғысынан нақты әлемдік нысандардың маңызды ерекшеліктерін көрсетеді.

Сәулеттің физикалық (сыртқы) деңгейі сақтау ортасындағы деректер қорын ұсынуды қолдайды. Дерекқордың физикалық деңгейдегі сипаттамасы *ішкі кестесі* немесе *сақтау кестесі* деп аталады.

Физикалық деңгей файлдарда немесе сыртқы сақтау құралында орналасқан бет құрылымдарында орналасқан деректер болып табылады. Бұл деректер базасын жеке іске асыру. Бұл деңгейде оңтайлы өнімділікке қол жеткізіп, дискілік кеңістікті үнемді пайдалануды қамтамасыз етуі керек.

ДҚБЖ физикалық деңгейде сақтау құрылғыларындағы деректерді сақтау, индекстерді жасау, деректерді шығару және т.б. үшін операциялық жүйеге қол жеткізу әдістерімен өзара іс-қимыл жасайды.

Деректер қысу туралы ақпарат және шифрлау әдістері, оларды таңдалған, дене деңгейінде ақпарат, деректер және индекстерді сақтау үшін дискілік кеңістік бөлу сақталады (сақталған деректер элементтерінің нақты мөлшерін көрсете отырып) жазбаларды сақтау деректемелері сипаттамасы, жазбаларды жері туралы ақпарат.

Физикалық деңгей ДҚБЖ бақылауында операциялық жүйемен басқарылады. Қазіргі уақытта физикалық деңгей ДҚБЖ-нің толықтай қамтамасыз етілген. Дерекқорды жобалауда басты назар тұжырымдамалық деңгейлік үлгіні құруға ауысады.

ANSI архитектуралық үлгісінде деректерді салыстырудың деңгейін қамтамасыз ететін ДҚБЖ механизмдері бар деп болжануда. Бұл тетіктердің функционалдық мүмкіндіктері деректерді абстракциялауды қамтамасыз етеді және барлық деңгейлердегі деректердің тәуелсіздік дәрежесін анықтайды.

Күрделі деңгейдегі салыстыруларға сәйкес, деректердің логикалық және физикалық тәуелсіздігін бөлу қажет.

Логикалық тәуелсіздік концептуалдық кестеге енгізілген өзгерістерден сыртқы тізбектерді қорғауды білдіреді. Сонымен қатар, логикалық тәуелсіздік бір деректер базасымен жұмыс істейтін басқа да қосымшаларды түзбей, бір деректерді өзгерту мүмкіндігін және физикалық деректерге қол жеткізу механизмін қайта құруды білдіреді.

Физикалық тәуелсіздік концептуалдық кесте сақтау кестесіне өзгерістерден қорғалған дегенді білдіреді. Физикалық тәуелсіздік сондай-ақ, деректер базасымен жұмыс істейтін барлық қосымшалардың денсаулығын сақтай отырып, сақталған ақпаратты бір медиадан екіншісіне тасымалдауға мүмкіндік береді.

БАҚЫЛАУ СҰРАҚТАРЫ

1. «Ақпарат», «ақпараттық жүйе», «ақпараттық нысан» және «ақпараттық технологиялар» сияқты ұғымдарды анықтаңыз.
2. Дерекқор дегеніміз не және ақпараттық жүйеде оның орны қандай?
3. Мәліметтер мен метадеректер арасындағы айырмашылық неде?
4. Дерекқорды басқару жүйелерінің мақсаты қандай?
5. Қолданбалы бағдарламалар дерекқормен өзара қалай әрекеттеседі?
6. Деректер базасы дерекқордан қалай ерекшеленеді?
7. Дерекқорда қандай компоненттер бар?
8. Мәліметтер сөздігі дегеніміз не?
9. Пайдаланушылардың негізгі санаттарын және деректер базасын әзірлеушілерді көрсетіңіз. Деректер банкінің жұмысында олардың рөлі қандай?
10. Дерекқорды басқару жүйелерінің функциялары қандай?
11. Дерекқорға не үшін кіруім керек?
12. Мәміле дегеніміз не?
13. ДҚБЖ қандай қолдау қызметтері ұсынады?
14. Дерекқордың тұтастығы нені білдіреді?
15. ДҚБЖ-ның үш деңгейлі сәулеті қандай?
16. Сыртқы үлгілердің ерекшелігі қандай?
17. Концептуалды деңгейдің ерекшелігі қандай?
18. Дене деңгейінің ерекшелігі қандай?
19. Дерекқор кестесі дегеніміз не?
20. Деректердің физикалық және логикалық тәуелсіздігі дегеніміз не?

ДЕРЕКТЕР ҮЛГІСІ

8.1. ДЕРЕКТЕР ҮЛГІСІНІҢ ТҮСІНІГІ

Деректер қорының классикалық теориясында деректер үлгісі дерекқорды басқару жүйесіндегі деректерді ұсыну және өңдеудің ресми теориясы болып табылады.

Ұзақ уақыт бойы «деректер үлгісі» термині ресми анықтамаларсыз қолданылған. Бұл тұжырымдаманы дәл анықтай алатын алғашқы мамандардың бірі Э.Кодд болды. Деректер үлгісінің тұжырымдамасын енгізгенімен, деректер үлгісінің ең көп тараған түсіндірмесі Кристофер Деректеріне жатады. Оның анықтамасына сәйкес, *деректер үлгісі* үш аспектіні қамтиды:

- 1) Құрылым аспектісі — деректер базасында типтерді және логикалық деректер құрылымдарын сипаттау әдістері;
- 2) Манипуляция аспектісі — деректерді манипуляциялау әдістері;
- 3) Бүтіндік аспектісі — дерекқордың тұтастығын сипаттау және сақтау әдістері.

Құрылым аспектісі логикалық дерекқордың не екенін анықтайды.

Манипуляция аспектісі дерекқорға сауал қоюға арналған бір немесе бірнеше тілдердің сипаттамаларын қамтиды. Бұл тілдер дәл синтаксисті жоқ, дерексіз болуы мүмкін. Деректер үлгісінің манипуляциялық бөлігінің басты мақсаты - бұл үлгіге сәйкес келетін ДҚБЖ енгізуде жеделдік дәрежесі қолдау көрсетілетін анықтамалық деректер базасының тілін қамтамасыз ету. Манипуляция аспектісі дерекқор күйлері арасында ауысу жолдарын анықтайды - деректерді модификациялау әдістері және дерекқордан деректерді шығару әдісі.

Бүтіндік аспектісі дерекқордың дұрыс күйін сипаттау құралдарын анықтайды. Деректер үлгісінің тұтас бөлігі сипатталған

бұл үлгіге сәйкес ДҚБЖ-нің барлық іске асырылуында қолдау көрсетілетін тұтастықты шектеу тетіктері болып табылады.

Дерекқор теориясында деректер үлгісі кез-келген дерекқордың негізі болып табылады. Деректер үлгілері доменнің ақпараттық нысандарын, олардың өзара байланысын сипаттайды және рұқсат береді:

- дерекқорды басқарудың логикалық және физикалық аспектілері арасындағы шекараны анықтау (деректер тәуелсіздігі);
- соңғы пайдаланушылар мен бағдарламашыларға деректердің мәнін жалпы түсіну қабілеті мен құралдары (коммуникативтік дағдылар);
- бірыңғай операция ретінде үлкен жиынтық жазбалар жиынтығында ұқсас операцияларды орындау мүмкіндігін беретін жоғары деңгейлі тілдік тұжырымдамаларды анықтау (өңдеу жиынтығы).

Деректер үлгісі олар басқаратын барлық арнайы ДҚБЖ мен дерекқорларға осы үлгіге негізделгенін сипаттайтын түсініктер мен сипаттамалар жиынтығын сипаттайды. Деректер үлгісінің қолжетімділігі бір жалпы тіл арқылы арнайы бағдарламаларды салыстыруға мүмкіндік береді.

Келесі деректер үлгілері классикалық болып табылады:

- иерархиялық;
- желілік;
- реляционды.

Бұдан басқа, соңғы жылдары деректер үлгілерін іс жүзінде белсенді енгізіле бастады:

- көп өлшемді;
- нысанды-бағытталған.

Сондай-ақ белгілі үлгілердегі кеңейтетін басқа да деректер үлгілері негізінде түрлі жүйелер жасалуда. Олардың арасында нысанды-реляциялық, ауытқушылықпен нысанды, семантикалық, тұжырымдамалық және бағдарланған үлгілер деп атауға болады. Осы үлгілердің кейбірі дерекқорларды, білім базаларын және бағдарламалау тілдерін біріктіруге қызмет етеді. Кейбір ДҚБЖ бірнеше деректер үлгісін бір уақытта қолдайды.

8.2. ТЕОРЕТИКАЛЫҚ-КЕСТЕЛІК ДЕРЕКТЕР ҮЛГІСІ

Иерархиялық деректер үлгісі дерекқор доменінің өкілін иерархиялық ағаш түрінде пайдаланады, түйіндері тектік байланысты «ата-баба ұрпаққа» тізбегімен қарастырады.

Ағаш — бұл циклдарды қамтымайтын қосылмаған бағытталмаған диаграмма. Ағашпен жұмыс жасағанда, белгілі шыңдарды таңдап, оны ағаштың түбірі ретінде анықтаңыз және оны нақты қарастырыңыз - бұл шыңға ешқандай жиектер кірмейді. Бұл жағдайда ағаш бағдарланады, бағдар түбірден анықталады.

Терминал шыңдары, яғни доғалы жапырақтары жоқ шыңдар ағаштың *жапырағы* деп аталады. Ағаштың әр түрлі тармақтарындағы түбірден жапыраққа дейінгі жолдағы шыңдар саны басқаша болуы мүмкін. Иерархиялық деректер үлгілерінде ағаш құрылымын тамырдан жапырақтарға бағыттайды. Дерекқор схемасының графикалық диаграммасы *анықтама ағашы* деп аталады.

Иерархиялық құрылым құрылымның деректеріне сәйкессіздікті болжайды - кейбіріне қатаң бағынышты. Иерархиялық үлгіні жасау шынайы әлемде бір нысан ата-ана ретінде әрекет ететін кезде көптеген байланыстар иерархияға сәйкес келетіндігімен негізделген және көптеген бағынатын нысандар онымен байланысты болуы мүмкін.

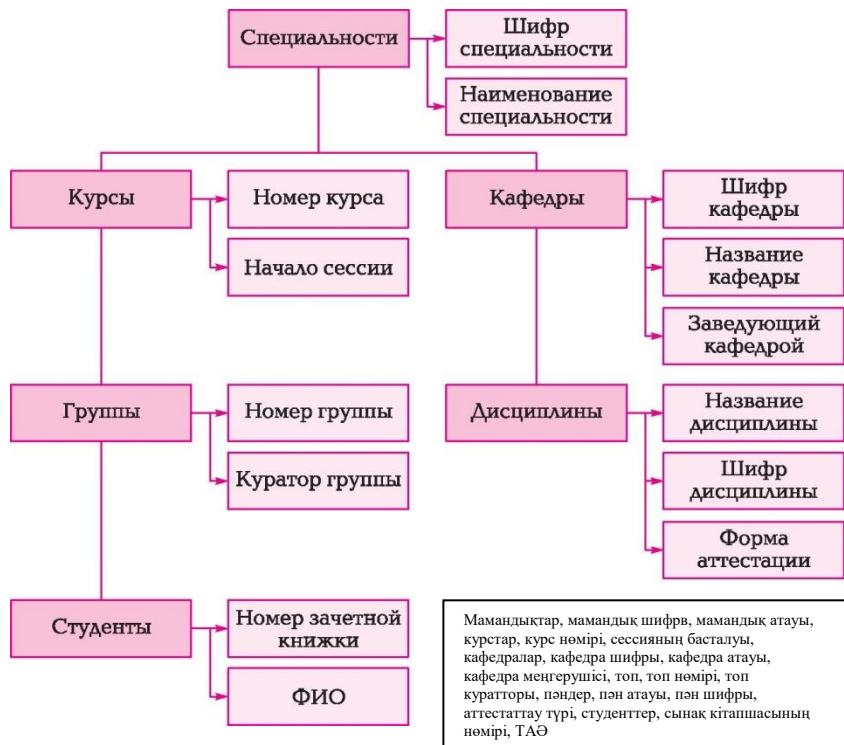
Ағаштың әр шыңы доменнің мәніне сәйкес келеді. Бұл субъект атрибуттардың ерікті санымен сипатталады. Иерархиялық дерекқордың мысалы 8.1 суретте көрсетілген.

Ағаш бойымен жылжу әрдайым түбір шыңынан басталады, одан кейін келесі деңгейдегі кез-келген шыңның жазбасының белгілі бір данасына өтуге болады. Бұл шыңы ағымдағы шыңдарға айналады және данасы ағымдағы данаға (жазба) айналады. Осы жазбадан осы шыңның басқа жазбасына, бас шыңында жазылған данаға немесе шыңның жазба данасына өтуге болады.

«Ағаш» түрінің анықтамасына сәйкес ата-бабалар мен ұрпақтар арасындағы сілтемелердің тұтастығын бақылау автоматты түрде сақталған деген қорытынды жасауға болады.

Адалдықты бақылаудың негізгі ережесі келесідей тұжырымдалған: ұрпақ ата-анасыз өмір сүре алмайды, ал кейбір ата-аналар ұрпақтар болмауы мүмкін. Түрлі ағаштардың жазбалары арасындағы байланыстың тұтастығын сақтау механизмдері жоқ.

Иерархиялық деректер үлгісінің еңбегі компьютерлік жадты тиімді пайдалануды және деректер бойынша негізгі операцияларды орындау уақытының жақсы индикаторларын қамтиды. Иерархиялық деректер үлгісі иерархиялық тәртіпті реттелген ақпаратпен жұмыс істеу үшін пайдалы.



8.1 сурет. Иерархиялық үлгідегі байланыс көріністері

Иерархиялық деректер үлгісінің негізгі кемшілігі ақпарат жеткілікті күрделі логикалық байланыстары бар қайта өңдеу және орта пайдаланушы үшін түсіністік күрделі, ауқымдылығы, артық деректер болып табылады.

Деректерді ұйымдастырудағы иерархиялық көзқарасты кеңейту желілік тәсіл болып табылады. Желілік деректер үлгісінде кез-келген объект басты және бағынышты болуы мүмкін және басқа нысандармен қарым-қатынастардың кез-келген санының қалыптасуына қатыса алады. Желілік деректер жазбалар жиынтығы және жазбалар арасындағы қарым-қатынастардың жиынтығы тұрады, және одан дәл болуы - деректер кестесінің рекордтық түрлері жиынтығымен анықталады және байланыс түрлерінің берілген жиынтығы жағдайларда әрбір түрін орнату әрбір түрі көшірмелерін жиынтығын көрсетеді.

Желілік деректер үлгісі ерікті график түріндегі деректер элементтерінің әртүрлі өзара байланысын көрсетуге мүмкіндік береді, осылайша иерархиялық деректер үлгісін жинақтайды.

Желілік деректер бірнеше артықшылықтарға ие.

- *Икемділік.* Бірнеше ата-ана / бала қарым-қатынастары желілік дерекқорға қарапайым иерархиядан гөрі құрылымы күрделі болатын деректерді сақтауға мүмкіндік береді.
- *Стандарттау.*
- *Жылдамдық.* Үлкен күрделілігіне қарамастан, желілік дерекқорлар иерархиялық деректер базасының жылдамдығына салыстырмалы өнімділікке қол жеткізді.

Желілік деректер моделінің жетіспеушілігі оның негізінде салынған дерекқор схемасының күрделілігі мен қатаңдығы, сондай-ақ қарапайым пайдаланушы дерекқорда ақпаратты өңдеуді түсіну мен орындаудың күрделілігі болып табылады. Бұдан басқа, желілік деректер үлгісінде сілтемелердің тұтастығы жазбалар арасындағы еркін қарым-қатынасты орнату рұқсат етілгендігінен әлсіреді.

8.3. РЕЛЯЦИОНДЫҚ ҮЛГІ

Реляциялық үлгі деректердің жиынтық-теориялық үлгілеріне жатады және қолданушылар арасында және кәсіби әзірлеушілер арасында ең танымал. Реляциялық үлгі деректер базасын мүмкіндігінше икемді етуге ұмтылудың арқасында пайда болды. Бұл үлгі деректер байланыстарын сақтаудың қарапайым және тиімді механизмін қамтамасыз етті. Байланысты дерекқорды басқару жүйесі деректерді өңдеудің ең сәтті технологиялары болып табылады. Бизнес әлеміндегі деректердің көпшілігі реляциялық түрде сақталады.

Реляциялық үлгі деректердің бірыңғай ұсынуын қамтамасыз ететін жалғыз ғана үлгі. Реляциялық деректер үлгісінде ақпараттық нысандар кестелер түрінде ұсынылған. Әрбір дерекқор кестесі жолдар мен бағандар жиынтығы ретінде ұсынылады. Жолдар (жазбалар) нысанның данасына, нақты оқиға немесе құбылысқа сәйкес келеді. Бағандар (өрістер) атрибуттарға, яғни сипаттамаға, сипаттамаға немесе объектінің, оқиғалардың немесе кейбір құбылыстардың параметрлеріне сәйкес келеді (8.2-сурет).

Кестеде жалғыз жол болмауы керек, яғни ол бос болуы мүмкін. Сол кестенің барлық жолдары бірдей құрылымға ие. Реляциялық дерекқордағы адам кем дегенде бір бағанды қамтуы мүмкін. Әрбір бағанда нақты деректер түрі бар. Деректер базасында әдеттегі бағдарламалау тілдерінде қолданылатын деректерге ұқсас деректер типтері пайдаланылады. Кесте өрістері бөлінбейтін ақпаратты қамтуы керек.

Жолы (баған)

П			
1			
1			
1			
1			

Жазба (жол)

8.2 сурет. Деректер қорының кестесі

Әрбір дерекқордың кестесінде кемінде бір бірегей кілт қажет.

Бірегей кілт — бұл өрісті немесе нысанның әрбір данасын (жазба) бірегей түрде анықтайтын өрістер жиынтығы. Кестедегі бірегей кілттер бірнеше болуы мүмкін. Бірегей кілттердің бірі *бастапқы кілт* ретінде анықталады. Бірегей кілттермен қатар, кесте бастапқы кілттің бірдей мәндері бар екі немесе одан көп жазбалардың болуына мүмкіндік бермейді. Ол минималды болуы керек, сондықтан оны жою бірегейлігіне әсер етпейтін өрістерді қамтымайды.

Бір кесте жазбаға немесе кесте жолына сәйкес келетін, олардың әрқайсысы деректер, атап айтқанда ақпараттық кестеде сақталады бір нысан (және т.б. құбылыс мәні, жүйесі,) бөлу, көптеген деректер арасындағы қарапайым нысан сілтемелерін сипаттауға ыңғайлы. Бұл жағдайда нысандар өкілдерінің әрқайсысы жазбалар өрістерінің тиісті мәндерімен сипатталған бірдей құрылымға немесе қасиетке ие. Мысалы кестеде мынадай сипаттамалары белгілі, оның әр колледж студенттер тобының туралы ақпаратты қамтуы мүмкін тегі, аты және әкесінің аты, жынысы, жасы және білім. Себебі бір кестедегі доменнен неғұрлым күрделі логикалық деректер құрылымдарын сипаттау мүмкін болмаса, кесте байланыстары қолданылады.

Сыртқы бұқаралық ақпарат құралдарында деректерді физикалық орналастыру деректерді қарапайым файлдармен оңай жүзеге асырылады.

Реляциялық деректер моделінің артықшылығы компьютерде физикалық іске асырудың қарапайымдылығы, түсінікті және қолайлылығы болып табылады. Пайдаланушылардың кеңінен қолданылуының негізгі себептеріне айналған қарапайымдылығы мен айқындығы. Деректердің бұл түрін өңдеудің тиімділігі проблемалары техникалық тұрғыдан толықтай шешілді. Реляциялық үлгінің басты кемшілігі - иерархиялық және желілік қатынастарды сипаттаудың күрделілігі.

Реляциялық ДҚБЖ-ның ең соңғы нұсқалары нысанды-бағытталған жүйелердің кейбір қасиеттеріне ие. Мұндай ДҚБЖ жиі нысан-реляциялық деп аталады. Мұндай жүйенің мысалы - Oracle8.x өнімдері болып табылады.

8.4. ҚАРЫМ ҚАТЫНАСТЫҚ ДЕРЕКТЕР ҮЛГІСІ

Классикалық қарым-қатынас үлгісі кестелердегі жазбаларда сақталатын деректердің бөлінбеуін қабылдайды және деректердің артық болуына жол бермейді. Сайланғаннан кейінгі үлгі деректердің бөлінбейтіндігіне шектеу жоқ кеңейтілген реляциялық үлгі болып табылады. Бұл үлгі құрамды (көп мәнді) өрістерді қабылдайды, яғни өрістің мәні ішкі сөздерден тұруы мүмкін. Көп мәнді өрістерге арналған мәндердің жиынтығы негізгі кестеге ендірілген жеке кесте болып саналады.

Қарым қатынастық үлгілердің артықшылығы реляциялық салыстырмалы кестелер жиынтығын біріктірілген реляциялық кестемен бірге ұсыну мүмкіндігін береді. Бұл ақпараттың жоғары көрсетілуін қамтамасыз етеді және оны өңдеудің тиімділігін арттырады.

Сайланғаннан кейінгі модельдің кемшілігі сақталатын деректердің тұтастығы мен дәйектілігін қамтамасыз ету проблемасын шешудің күрделілігі болып табылады. Бұл проблема ДҚБЖ-да тиісті механизмдерді қосу арқылы шешіледі.

8.5. КӨП ӨЛШЕМДІ ДЕРЕКТЕР ҮЛГІСІ

Ақпараттық жүйелердің тұжырымдамасын әзірлеу кезінде келесі екі саланы бөлуге болады: операциялық өңдеу және аналитикалық өңдеу жүйелері (шешімдерді қолдау жүйесі). Байланысты ДҚБЖ деректерді өңдеудің ақпараттық жүйелеріне арналған және бұл салада ең тиімді болып табылады. Алайда, аналитикалық өңдеу жүйелерінде олар жеткілікті икемді болмады. Мұнда көп өлшемді деректерді басқару жүйесі тиімдірек. Көп өлшемді жүйелер талдау және шешімдер қабылдау үшін ақпаратты жылдам өңдеуге мүмкіндік береді. Реляциялық модельмен салыстырғанда, көп өлшемді деректерді ұйымдастыру аса жоғары көрінетін және ақпараттылығы бар.

Көп өлшемді ДҚБЖ-да деректер реттелген көп өлшемді массивтер түрінде ұйымдастырылады - гиперкуба. Олар деректердегі сұрауларға жылдам жауап береді, себебі қоңыраулар белгілі бір пайдаланушылар тобы үшін қажетті шағын блоктарға қабылданады.

Көп өлшемді дерекқорлардың технологиясы жаңа технология болып табылады. Деректер базасында деректерді ұсыну үшін көп өлшемді көзқарас реляциялық, сонымен бірге шын мәнінде жұмыс істейтін көп өлшемді ДҚБЖ-мен бір мезгілде пайда болғанына қарамастан, бастапқыда өте аз болды. 90-жылдардың ортасынан бастап ғана, оларға қызығушылық массивтік сипатқа ие болды.

Деректер үлгісінің көп өлшемділігі визуальды ұсынуды білдірмейді, бірақ деректерді сипаттау және өңдеу кезінде ақпарат құрылымының қисынды ұйымы болып табылады. Жоғарыда айтылғандарды суреттеу үшін, 8.3 сурет осы деректердің реляциялық (а) және көп өлшемді (b) ұсыныстарды ұсынады (кестелерде жиһазды сату көлемі туралы ақпарат бар).

Көп өлшемді деректер үлгілерінің негізгі ұғымдарын – өлшеу және қысқарту туралы қарастырайық.

Өлшеу — бұл гиперкубаның бетінің бірін құрайтын ұқсас деректер жиынтығы. Басқаша айтқанда, өлшемдер көп өлшемді координат жүйесінің осьтері болып табылады. Ең жиі қолданылатын уақыт өлшемдеріне мысалдар «Күндер», «Айлар», «Кварталдар» және «Жылдар». Географиялық өлшеулер ретінде «Қалалар», «Аймақтар», «Аймақтар» және «Елдер» кеңінен қолданылады. Өлшем деректерінің көп өлшемді үлгісінде индекстің рөлі гиперкубадағы жасушалардың нақты мәндерін анықтауға қызмет етеді.

Өлшеуіш (ячейка, көрсеткіш) — бұл өріс мәні белгіленген өлшемдер жиынтығымен анықталған.

Үлгі	Ай	көлемі
Шкаф	маусым	12
Шкаф	шілде	24
Шкаф	тамыз	5
Диван	маусым	2
Диван	шілде	18
Үстел	шілде	19

a

Үлгі	Маусы	Шілде	Тамыз
Шкаф	12	24	5
Диван	2	18	
үстел		19	

b

8.3 сурет. Деректердің реляционды және көп өлшемді көріністер

Февраль				
Январь				
	Склад 1	Склад 2	Склад 3	Склад 4
Мебель	1 000	200	100	200
Аксессуары	500	50	25	50
Ткань	500	50	25	50

8.4 сурет. Гиперкуб мысалы

Қаңтар, ақпан.
Жиһаз, аксессуарлар, мата. Қойма

Өріс түрі көбіне сандық деп анықталады. Бұл сатылым көлемін сандық немесе ақшалай түрде, қоймадағы тауар қалдықтары, шығындар және т.б. болуы мүмкін.

3-суретте көрсетілген үш өлшемді текшедегі шаралар ретінде. 8.4, сатылымдардың сомалары пайдаланылады, ал өлшеу - уақыт, тауар және қойма болып табылады. Өлшемдер топтаудың белгілі бір деңгейлерінде ұсынылған: тауарлар түріне қарай топтастырылған, ал мәміленің жасалған уақыты туралы мәліметтер ай сайын жүргізіледі.

Текше мүлдем үш өлшемді болуы керек емес. Ол екі және көп өлшемді болуы мүмкін - мәселенің шешілуіне байланысты.

Дегенмен, текше өзі талдау үшін қолайлы емес. Тіпті үш өлшемді текше компьютерлік экранда көрсетілуі қиын, өйткені қызығушылық шараларының мәндері көрінеді. Текшеде сақталған деректерді көрсету шін, күрделі иерархиялы жол және баған тақырыптары бар екі өлшемді кесте көрсеткіштері пайдаланылады. Бұл операция текшені «кесу» деп атайды. Термин қайтадан бейнелі деп аталады.

Сарапшы, қалай болғанда, текше сызығының өлшемдерін қызықтырады және «қысқартады». Пайдаланушы табиғи, интуитивті деректер үлгісін алады. Бұл бизнес процесі туралы қысқаша немесе егжей-тегжейлі ақпарат алуға және талдау процесінде пайдаланушымен болатын басқа да манипуляцияларды жасауға мүмкіндік береді. Осылайша, талдаушы текшенің екі өлшемді кесегін алады және онымен жұмыс істейді. Сонымен, логерлер журналда жыл сайынғы сақиналарды қарастырады. Тиісінше, кестенің өлшемдеріне сәйкес әдетте, тек екі өлшеулер «жойылмай» қалады.

Бөлім бір немесе бірнеше өлшемдерді бекіту нәтижесінде алынған гиперкубаның ішкі жиыны болып табылады. «Бөлімді» қалыптастыру пайдаланушының мәндерін шектеу үшін орындалады, өйткені гиперкубаның барлық мәндері іс жүзінде

	Қойма 1	Қойма 2	Қойма 3	Қойма 4
Қаңтар	2000	300	150	300
Ақпан	3000	1000	200	1000

8.5 сурет. Бір шараның екі өлшемді бөлімі

ан бір уақытта пайдаланылмайды. 8.5 суретте бір өлшем үшін текше өлшемді бөлімі - «Саны» және екі «кесілмеген» өлшем - «Қойма» және «Уақыт» көрінеді. 8.6 суретте тек қана «біркелкі емес» өлшемі бар - «қойма», бірақ мұнда екі шараның мәндері көрсетіледі - «Саны», «Сумма». Текшедегі екі өлшемді көрсетілім тіпті екі өлшемнен артық «жабық» болып қалса да мүмкін болады. Бұл жағдайда «кесілген» текшедегі екі немесе одан да көп өлшемдер бөлік осіне (жолдар мен бағандар) орналастырылады (8.7-сурет).

Өлшемді деректер үлгісінің жетіспеушілігі - ақпаратты әдеттегі жедел өңдеудің ең қарапайым міндеттері үшін оның қиындықтары болып табылады.

Көп өлшемді ДҚБЖ негізгі артықшылықтары:

- көпөлшемді деректер базасының технологияларын қолданбаларға тез интеграциялауға мүмкіндік беретін жүйенің жалпы қарапайымдылығы. Көп өлшемді дерекқорларға негізделген жүйелерді әзірлеу және басқару үшін арнайы дағдылар аз;
- иеленудің жиынтық құны, сондай-ақ инвестициялардың тез қайтарымы;
- көп өлшемді ДҚБЖ болған жағдайда, деректерді іздеу дерекқордың көп өлшемді тұжырымдамалық көрінісіне қарағанда әлдеқайда тезірек болады, өйткені көп өлшемді дерекқор сұралған жасушаларға оңтайландырылған қолжетімділікті қамтамасыз етеді;
- көп өлшемді ДҚБЖ ақпараттық үлгіге түрлі кіріктірілген функцияларды енгізу міндеттерін оңай жеңе алады, ал объективті SQL тілінің шектеулері осы міндеттерді реляциялық ДҚБЖ негізінде жүзеге асыруға мүмкіндік береді, бұл өте қиын және кейде мүмкін емес.

	Қойма 1	Қойма 2	Қойма 3	Қойма 4
Саны	200	100	50	300
Сомасы	3000	1000	200	1000

8.6 сурет. Бірнеше өлшемге арналған екіөлшемді шаршы бөлімі

	Қаңтар				Ақпан			
	Қойма 1	Қойма 2	Қойма 3	Қойма 4	Қойма 1	Қойма 2	Қойма 3	Қойма 4
Саны	200	100	50	300	200	100	50	300
Сомасы	3000	1000	200	1000	3000	1000	200	1000

8.7 сурет. Бір осьте бірнеше өлшемдері бар текшедің екі өлшемді бөлімі

Деректерді көп өлшемді құрылымдарда сақтау кезінде «ісіну» ықтимал мәселесі бос мәндерді сақтауға байланысты туындайды. Көп өлшемді массив өлшеу белгілерінің барлық ықтимал құрамаларынан кеңістік қорғалған, бірақ шын мәнінде өте аз бөлігін ғана қорғайды (мысалы, тек бірнеше аймақтарда сатылатын өнімдердің саны) Егер орын бос болса да, кейін барлық, текшенің көпшілігі, бос болады. Қазіргі заманғы жүйелер осы проблеманы шеше алады.

Матрица (Speedware), OracleExpressServer (Oracle), Cache (InterSystems) - көп өлшемді деректер үлгісін қолдауға жүйелердің мысалдары Essbase (ArborSoftware), MediaMulti болып табылады. Мұндай Media / MR (Speedware) сияқты кейбір бағдарламалық қамтамасыз ету, бір мезгілде көп өлшемді және реляциялық деректер базасымен жұмыс істей алады. Ішкі деректер үлгісі көпөлшемді үлгі, (түйіндердің көп өлшемді массивтерді деңгейінде) деректерге кіру желісінің үш әдістері және нысаны реляциялық болып табылады.

8.6. НЫСАНДЫ БАҒДАРЛАУ ҮЛГІСІ

Нысанды-бағытталған деректер базасын басқару жүйелері дерекқордың мүмкіндіктері мен мүмкіндіктерін және объектілі-бағытталған бағдарламалау тілдерінің мүмкіндігін үйлестірудің нәтижесі болып табылады. Нысанды-бағытталған ДҚБЖ деректер базасының нысандармен нысанды-бағдарланған бағдарламалау тілдеріндегі нысандармен жұмыс істеуге мүмкіндік береді. Олар бағдарламалау тілдерін кеңейтеді, ұзақ мерзімді деректерді ашық түрде енгізеді, параллель деректер жұмысын басқарады, деректерді қалпына келтіруге, байланысты мәселелерге және басқа мүмкіндіктерге ие болады.

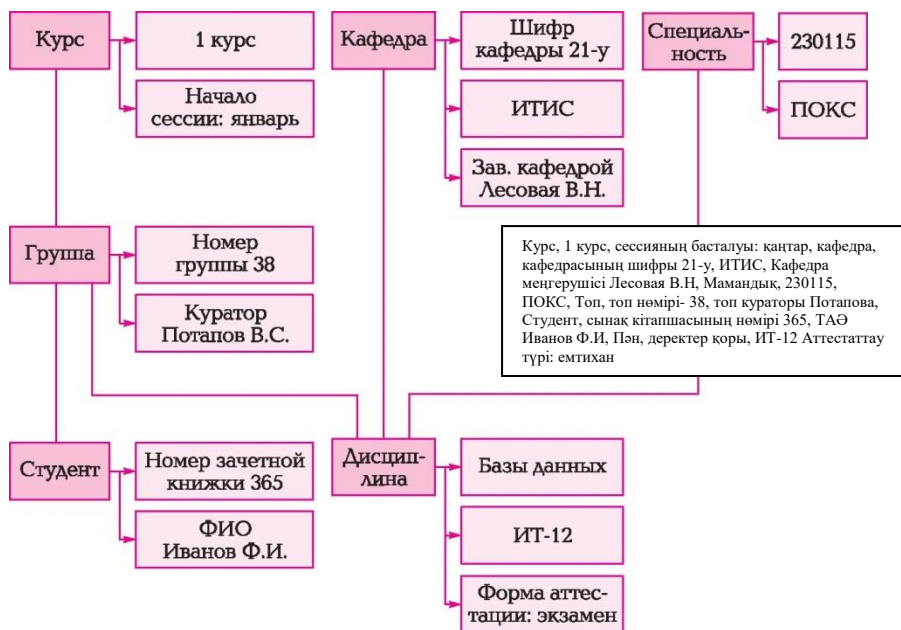
Нысанды бағдарланған ДҚБЖ пайда болуы компьютерлік ЖЖҚ-на сәйкес келмейтін нысандарды сақтау құралдарында нысанды-бағытталған тілдердегі программистердің қажеттіліктеріне байланысты болды. Сондай-ақ, сақтау қажет болды.

Қолданбалы бағдарламаның қайта басталуы арасындағы нысандардың жай-күйі. Сондықтан нысанды бағдарланған дерекқорды басқару жүйелерінің көпшілігі деректерді басқару процедуралары бағдарламасына қосылған кітапхана болып табылады.

Нысанаға бағдарланған үлгіде, деректерді ұсынғанда, дерекқордың жеке жазбаларын анықтауға болады. Деректер базасының жазбалары мен оларды өңдеу функциялары арасындағы қарым-қатынас нысанды-бағытталған бағдарламалау тілдерінде тиісті құралдарға ұқсас механизмдер арқылы анықталады.

Нысанаға бағдарланған дерекқордың құрылымы графикалық түрде ағаш түрінде бейнеленеді, оның түйіні нысандар болып табылады. Нысандардың сипаттары кейбір стандартты типтерде (мысалы, string - string) немесе пайдаланушы жасаған (сынып ретінде анықталған) түрімен сипатталады.

«Колледж» нысанды-бағдарланған дерекқорының логикалық құрылымының мысалы 8.8 суретте келтірілген.



8.8 сурет. «Колледж» нысанды -бағдарланған дерекқорының логикалық құрылымы

Нысанға бағдарланған дерекқордың логикалық құрылымы көрінісінде иерархиялық дерекқордың құрылымына ұқсас. Олардың арасындағы айырмашылық деректерді өңдеу әдісі болып табылады. Дерекқор моделіндегі деректер бойынша әрекеттерді орындау үшін логикалық операциялар нысанды-бағытталған механизмдермен күшейтіледі.

Нысанды бағдарланған деректер үлгісінің басты артықшылығы реляциялық бірмен салыстырғанда нысандардың күрделі қарым-қатынасы туралы ақпаратты көрсету мүмкіндігі болып табылады. Нысанды -бағытталған деректер үлгісі дерекқордың жеке жазбаларын анықтауға және оларды өңдеу функцияларын анықтауға мүмкіндік береді.

Нысанға бағдарланған үлгінің кемшіліктері өңдеудің жоғары тұжырымдамалық күрделілігі мен қолайсыздығы болып табылады.

Реляциялық ДББЖ өндірушілерін нысанды технологиялардың танымалдығы реакциясы - әмбебап серверлер деп аталатын нысанды-реляциялық деректер қорының пайда болуы болды. Нысанды-бағытталған DBMS - нысанға бағытталған әдісті іске асыратын кейбір технологияларды қолдайтын реляциялық ДҚБЖ: нысандар, сыныптар және мұра дерекқор құрылымы мен сұрау тілінде жүзеге асырылады. Нысанды-реляциялық DBMSs, мысалы, кеңінен танымал OracleDatabase, Informix, DB2.

Нысанды-реляциялық үлгінің негізгі артықшылықтары - компоненттерді қайта пайдалану және бөлісу мүмкіндігі.

Нысанды-реляциялық DBMS әдісінің айқын кемшілігі оның күрделілігі мен байланысты шығындар болып табылады. Реляциялық модельге тән қарапайым және айқын, кеңейтімнің осы түрлерін қолданғанда жоғалады.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Деректер үлгісі дегеніміз не?
2. Деректер үлгісі не үшін қажет?
3. Деректер үлгісі қандай аспектілерді қамтиды?
4. Мәліметтерді ұсынудың классикалық және заманауи үлгілерін көрсетіңіз.
5. Деректердің иерархиялық үлгісінің еңбегін және құндылығын көрсетіңіз.
6. Иерархиялық деректер базасында деректерді физикалық орналастыру қалай ұйымдастырылған?
7. Желі деректерінің үлгісін сипаттаңыз.

8. Реляциялық деректер үлгісін сипаттаңыз.
9. Реляциялық деректер моделінің ерекшелігі қандай?
10. Көп өлшемді деректер үлгілері қайда жатады?
11. Көп өлшемді үлгілердің құндылықтарын көрсетіңіз.
12. Көпөлшемді деректер үлгісін сипаттаңыз.
13. Көп өлшемді үлгіде деректер бойынша орындалатын операциялардың мағынасын атаңыз және түсіндіріңіз.
14. Көп өлшемді кестелердің мысалдарын келтіріңіз.
15. Деректерді ұсынудың нысанды-бағытталған моделінің артықшылықтары мен кемшіліктерін көрсетіңіз.
16. Көп өлшемді ДҚБЖ пайдаланылатын ұғымдардың мағынасын ашыңыз: «өлшеу», «ұяшық», «тілім».

РЕЛЯЦИОНДЫ ДЕРЕКТЕР ҮЛГІСІ

9.1. РЕЛЯЦИОНДЫ ДЕРЕКТЕР ҮЛГІСІНІҢ ЕРЕКШЕЛІКТЕРІ

9.1.1. Негізгі компоненттері мен түсініктері

Реляциялық деректер үлгісінің теориялық негізі қарым-қатынас теориясы болды. Қатынастар жиынтығы кейбір арнайы операцияларға байланысты жабылады, яғни бұл операциялармен бірге дерексіз алгебра құрайды. Байланыстың бұл маңызды қасиеті бастапқы алгебра байланысты деректерді өңдеу тілін дамыту үшін реляциялық үлгіде пайдаланылды. 1970 жылы американдық математик Е.Ф. Кодд бірінші реляциялық үлгінің негізгі түсініктері мен шектеулерін тұжырымдады. Кодд ұсыныстары дерекқор жүйелеріне өте тиімді болды, бұл үлгі үшін ол компьютерлік техниканың теориялық негіздері саласындағы беделді Тьюринга сыйлығына ие болды.

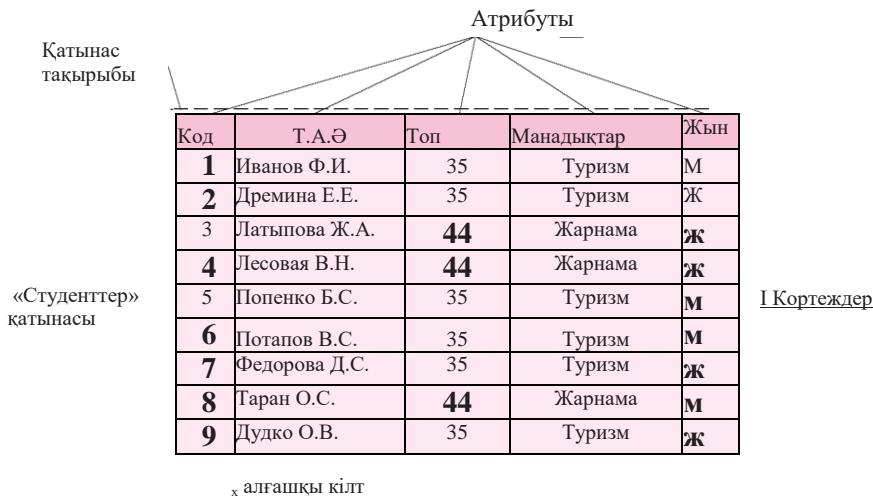
Реляциялық дерекқорлардың негізгі түсініктері бар:

- қатынасы;
- деректер түрі;
- домен;
- атрибут;
- кортеж;
- алғашқы кілт.

Алдымен біз осы түсініктердің мәнін колледж оқушылары туралы ақпаратты қамтитын «Студенттер» үлгісінде көрсетеміз (9.1-сурет).

Байланыстыру реляциялық деректер үлгісінің негізгі тұжырымдамасы. Осы себепті модель реляциялық деп аталады (лат. Relatio - «қатынас», «байланыс»).

Қатынас физикалық деңгейде деректердің сақталатын ұяшықтарында жолдар мен бағандар түрінде екі өлшемді кестелер бар. Әр кестеде біреудің нысандары туралы ақпарат бар



9.1 сурет. Реляциялық тәсілдің негізгі ұғымдарының ара қатынасы

түрі мен барлық кестелердің жиынтығы бірыңғай дерекқорды құрайды. Кестенің жолы *жазба* деп аталады, кестенің бағаны - *өріс*. Әр өріс кестеде бірегей атау болуы керек. Кестелерді ұйымдастырудың ерекшеліктері дерекқорды құру және қолдау үшін қолданылатын арнайы ДҚБЖ-ға тәуелді болады.

Деректер — бұл дерекқорда бірнеше түрлер түрінде сақталатын жиынтық ақпарат. Деректер түрлерін қолдана отырып, кестенің белгілі бір бағанындағы деректерге, оның ішінде бөлінген жады көлеміне қатысты негізгі ережелер белгіленеді.

Реляциялық үлгідегі «деректер типінің» түсінігі бағдарламалаудағы «деректер түрі» ұғымына толығымен сәйкес келеді. Реляциялық дерекқорда сақталған деректер мәндері теріледі, яғни әр сақталған мәнің түрі белгілі. Әрбір бағана өзінің жеке деректер түрін жазады. Әдетте, заманауи реляциялық дерекқорлар кейіпкерлерді, сандық деректерді, арнайы сандық деректерді (мысалы, «ақша»), сондай-ақ арнайы деректерді (күн, уақыт, уақыт аралығы) сақтауға мүмкіндік береді. Сонымен қатар, реляциялық жүйелер пайдаланушыларға өздерінің деректер түрлерін анықтау мүмкіндігін қолдайды.

«Домен» термині дерекқорлар үшін нақты болып келеді, бірақ кейбір бағдарламалау тілдерінде түрлерімен ұқсастығы бар.

Кейбір авторлар «домен» және «деректер түрі» ұғымдарын анықтайды. Сонымен қатар, кейбір реляциялық ДҚБЖ-да «домен» түсінігі тіпті пайдаланылмайды.

Жалпы алғанда, домен домен элементтері тиесілі кейбір негізгі деректер түрін көрсетумен анықталады және олардың шектеулерінің сипаттамалары. Әрбір домен сәйкес дерекқордың барлық домендерінің атаулары арасында бірегей болып табылатын атпен байланысады. Домен белгілі бір типтегі мәндердің рұқсат етілген шектеулі жиыны ретінде анықталуы мүмкін. Мысалы, біздің мысалдағы ТАӘ домені таңбалар жолдарының негізгі түрінде анықталады, бірақ оның мәндерінің саны тек орыс алфавитінің таңбаларын ғана қамтуы мүмкін және сандарды және т.б. қамтуы мүмкін. ЖЫНЫСЫ домені да символдық, бірақ ұзындығы шектеулі (тек 1 таңба) және рұқсат етілген мәндердің жиынын (тек «е» немесе «ж» таңбалары бар).

Сондай-ақ, «домен» тұжырымдамасының семантикалық құрамдас бөлігін атап өткен жөн: деректер бірдей доменге жатқанда ғана салыстырылады деп саналады. Мысалы, ТАӘ мен МАМАНДЫҚТАР домендерін салыстыруға ешқандай мән бермейді. Олар таңба түріне жататынына қарамастан, олардың мәндері салыстырмалы емес.

Атрибут — деректер базасында сақталатын доменнің нысанының осы сипаты (сипаттамасы). Атрибут кейбір доменге тиесілі атаумен және мәнмен сипатталады. Әрбір сәтте нысанның әр данасы ерекше сипаттамалар жиынтығымен ерекше сипатталады (9.1-суретті қараңыз)).

Қатынас кестесі (қатынас тақырыбы) — бұл {атрибут атауы, домен атауы} жұп аталатын жиынтығы. Байланыстыру схемасының дәрежесі немесе «арифметика» бұл жиынның күші, яғни атрибуттардың саны. Байланыстыру схемасы кесте бағанының үстіңгі деректемелері қатарынан ұсынылуы мүмкін (9.1 суретті қараңыз).

Кортеж, осы қатынас схемасына сәйкес келетін, қатынас схемасына тиесілі әр атрибут атының бір пайда болуын қамтитын жұптардың {attribute name, value} жиынтығы. «Мән» бұл атрибуты үшін жарамды домен мәні болып табылады. Осылайша, кестелеу кесте жолы ретінде ұсынылуы мүмкін (9.1-суретті қараңыз). Сонда бұл қатынас өзара қарым-қатынас схемасына сәйкес келетін кесінділер жиынтығы ретінде ұсынылуы мүмкін. Реляциялық орган - бұл теңгерімсіз жиынтығы.

Алғашқы кілт (қатынас кілті) — бұл құндылықтар байланыстың қосындысын бірден анықтайтын атрибуттардың ең аз жиынтығы. Әрбір қатынас үшін, оның атрибуттарының толық жиынтығы толыққанды болып табылады.

Айқын анықтау қолайлы - Алайда, бастапқы кілт ресми анықтамасының негізгі кілттер атрибуттар жиынтығының негізгі қасиеттері қозғамай алып тастауға болады, мысалы, атрибуттары қамтуы тиіс емес, яғни, «ең аз» талап етеді. «Кодекс», бірегей идентификациялық әрбір студент (әрбір кортеж немесе жазба) - Бұл әрбір кортеж бірегей болып табылатындығын білдіреді, бірақ ол бір атрибуттары жинағының барлығы мүмкін болған жағдайда, (9.1 сурет. қараңыз) «Студенттер» қарым-қатынасын көруге болады .

Кесте үшін анықтау негізгі автоматтық жазбаларды сұрыптау мүмкіндігі, сондай-ақ жазбалардың негізгі салаларында рений мәндерін қайталау арқылы бақылаудың жоқтығы болып табылады және үстелге іздеу операцияларын орындау жылдамдығын арттыру. Негізгі атрибут белгілі бір мағынада болуы мүмкін. Дегенмен, негізгі өріс өте семантикалық емес жүктемені орындамайды және жай кестеде нысан идентификаторы болып табылады. Жиі кестелердегі бастапқы кілттерді анықтау үшін автоматты мән жасайтын өрістер («counters», autoincrementfield) қолданылады. Бұл жағдайда, кілттік өрістің бірегейлігін сақтау үшін барлық жауапкершілік пайдаланушыдан жойылып, дерекқор процессорына жіберіледі. Есептегіш өріс төрт байт бүгін сан болып табылады және пайдаланушы кестеге жаңа жазбаны қосқанда, автоматты түрде көбейтіледі.

Осылайша, деректердің реляциялық үлгісі оның элементтерінің тізімімен және олардың мәндерінің тізбесі арқылы берілетін қатынас түсінігіне негізделеді. Бұл коэффициент кестемен ұсынылуы мүмкін. Кестенің бағандарының тақырыптары атрибуттар деп аталады. Олардың атауларының тізімі қатынас схемасы деп аталады. Әрбір төлсипат оның мәндерінің доменімен бірге домен деп аталатын деректердің түрін анықтайды. Толық кесте қатынасы деп аталады, ал кестенің әр жолы өзара қатынастар деп аталады. Әрине, реляциялық деректер моделін терминология сәл ерекше болып табылады, және жиі тұрақты үстелдер (нысанын «қарым-қатынас атрибуттары» білдіретін «, кестедегі бағандар» туралы әңгіме) шарттарын пайдаланады. біз Реляциялық деректер базасының және басқарушы қаражат-нұсқаманың ұйымдастыру практикалық мәселелерді шешу үшін және қашан келіп, бізге таныс осы терминологияны пайдаланады. Осындай терминология көптеген коммерциялық реляциялық деректер базасында пайдаланылады.

Біз өзара түсінікке қатысты бірқатар математикалық анықтамаларды енгіземіз.

N -трансшекаралық қатынасы R , немесе ара R деңгейі n , жинақтарын $D_1, D_2, \dots, D_n (n > 1)$,

әр түрлі түрлері міндетті емес. Бастапқы көпшілігі D_1, D_2, \dots, D_n домен үлгісінде аталады. *Декартовті (тура) өндірісі*— бұл жиынтығы, оның элементтері түпнұсқалық екі жиынтығының элементтерінің реттелген жұптары болып табылады.

$A_1(D_1), A_2(D_2), \dots, A_n(D_n)$ — атрибут атаулары. Ондай болса, кесте \mathcal{Z} қатынасы R атрибут атауларындағы соңғы көпшілік еп аталады $\mathcal{Z} = \{L(A_1, A_2, \dots, A_n)\}$.

Кестедегі қатынасы \mathcal{Z} соңғы көпшілікте D_1, D_2, \dots, D_n деп аталады көптеген R декартивті өндіріс $D_1 \times D_2 \times \dots \times D_n$ деп белгіленеді.

Қатынастар элементі (d_1, d_2, \dots, d_n) , жоғарыда аталып өткендей, кортеж деп аталады. Кортеж (d_1, d_2, \dots, d_n) n компоненттерін иеленеді. Кортежді белгілеу үшін қысқартулар мен белгілеулер ұсынылады d_1, d_2, \dots, d_n . Реляциялық деректер үлгісіндегі қатынастарды анықтау үшін декарттық өнім тұжырымдамасын қолдану модельді сындарлы етеді. Математикалық тұрғыдан алғанда, бұл модельдің барлық басқа тұжырымдамалары картециялық өнімге негізделген қатаң математикалық құрылыстың шеңберінде анықталады.

9.1.2. Қатынас сипаттары

Енді бұрын берілген анықтамалардан кейінгі қатынастардың кейбір маңызды қасиеттерін қарастырайық.

Дубликаның болмауы жиым жиынтығы ретінде қатынастың анықтамасынан туындайды. Бұл сипат бастапқы кілттің әр қатынастарының болуын қадағалайды.

Математикалық түрде оның анықтамасы бойынша жиынтығы сәйкес элементтерге ие болмайды, және, тиісінше, олардағы элементтерді олардың компоненттерінің мәнімен ғана бөлуге болады. Бұл модель үшін өте маңызды жағдай - реляциялық модель домендегі нысандар туралы деректердің қайталануын толығымен жояды.

Атрибуттарды ресімдеудің болмауы. Байланыс қатарларындағы төлсипат мәніне сілтеме жасау үшін атрибут атауы әрқашан пайдаланылады. Байланыстың атрибуттары тапсырыс бермейді, себебі анықтама бойынша қатынас тақырыбы - жұптар жиынтығы {атрибут атауы, домен атауы}. ДҚБЖ-нің өзі физикалық тапсырмада күпірлік атрибуттардың мәндерін сақтауы туралы шешім қабылдайды (әдетте, бірдей физикалық тәртіп әр қатынастың барлық буындарына қолдау көрсетіледі). Бұған қоса, бұл сипат қолданыстағы түрлендіру жұмысын жеңілдетеді.

атрибуторды қосу арқылы ғана емес, сонымен бірге барлығын жою арқылы жүзеге асады.

Атрибутор мәндерінің атомы атрибуттың жоқтығын білдіреді мысалы, «Иванов И.И., ул, Ленина, 8» атрибутының мәні дұрыс емес, себебі ол көп сипаттамалары (аты, көше, үй). Бұл сипат келесі деректерен тұрады, домен анықтамасынан әлеуетті мәндердің жиынтығы ретінде қарапайым деректер түрі.

Кортеж тәртібінің болмауы — олар физикалық енгізілген тәртіпте сақталады. Бұл қасиет ара қатынасының корпусының жиынтығы ретінде анықтау салдары болып табылады. Жоқ топтамалар жиынтығында тәртіпті сақтауға қойылатын талаптар; ДҚБЖ қосымша сақтау икемділігін береді. Сыртқы жадтағы дерекқорды және дерекқорды сұрау кезінде деректер жұмыс жасайды.

Қатынастардың қасиеттерін қолданудың нақты мысалдары болатын төмендегі бөлімдерде, реляциялық деректер базасын дамыту және қалыпқа келтіру болып табылады.

9.2. НЕГІЗГІ РЕЛЯЦИОНДЫ АЛГЕБРА

Реляционды алгебра (Коддтың реляционды алгебрасы) — Бұл реляциялық деректер үлгісіндегі қатынастар бойынша жабық операциялық жүйе. Реляциялық алгебра операциялары *реляциялық операциялар* деп аталады.

Реляциялық алгебраның негізгі идеясы, егер қарым-қатынастар жиынтығы болса, қарым-қатынастарды манипуляциялау құралдары дерекқорға тән кейбір арнайы операциялармен толықтырылған дәстүрлі теориялық операцияларға негізделуі мүмкін. Реляциялық алгебра қатынастарда осындай операциялардың жиынтығы болып табылады, олардың әрқайсысының нәтижесі де өзара қатынас болып табылады. Алгебраның бұл қасиеті *жабық* деп аталады.

Кодд реляциялық алгебра операцияларын екі топқа бөлуге болады: *негізгі теориялық және арнайы реляциялық*. Операциялардың бірінші тобы көптеген теориялық жұмыстарды қамтиды. Екінші топ нақты деректерді манипуляциялау тапсырмалары бойынша дәстүрлі жиынтық-теориялық операцияларды жасау болып табылады.

Теориялық операциялардың жиынтығы операцияларды қамтиды:

- қатынастарды біріктіру;
- қарым-қатынастан өту;;
- қатынастарды алып тастау;
- қатынастардың тікелей өнімі.

Арнайы реляциондық процестер дегеніміз:

- қатынастарды шектеу;
- қарым-қатынас жасау;;
- қатынастарды бөлу;
- қатынастарды қосу.

Сондай-ақ, дұрыс қорытқы қатынасының үстіңгі (диаграмма) генерациялау мүмкіндік, операцияны қайта атау алгебралық өрнектерді мен атрибуттарды есептеу деректер базасын сақтауға мүмкіндік береді алгебраны тағайындау пайдалануға, енгізілген. Реляциялық алгебра операциялар бірдей көзқарасы (мысалы, проекциялық) бойынша немесе екі қарым-қатынас (мысалы, кәсіподақ) бойынша жүзеге асырылуы мүмкін. Бірінші жағдайда операция біртұтас, екіншісі - екілік деп аталады. Бинарлық операцияны орындау кезінде операцияларға қатысты қатынастар құрылымға сәйкес болуы керек.

Кейбір қарым-қатынастар арқасында атрибуттарының атаулары айырмашылықтарға ресми үйлесімді емес, бірақ атрибуттары операцияны қайта атау қолданғаннан кейін, сондықтан айналды.

N -ary реляциялық ϕ функциясы қатынасты қайтаратын және қатынастары бар дәлел ретінде берілген функциямен ұсынылуы мүмкін:

$$R = f(R_1, R_2, \dots, R_n).$$

Реляциялық алгебра жабылғандықтан, реляциялық алгебраның басқа өрнектерін рациональдық операциялардағы:

$$R = f[f_i(R_{i1}, R_{i2}, \dots), f_2(R_{21}, R_{22}, \dots), \dots].$$

Қатынастық құрылымдардың үйлесімділігі атрибуттардың атауын және сәйкес домендердің түрлерінің үйлесімділігін білдіреді. Үйлесімділіктің ерекше жағдайы - сәйкестік (сәйкестік).

Бірдей өлшемдегі R_1 және R_2 ($R_1 \cup R_2$) екі үйлесімді қатынастардың тіркесімі - бұл қайталануды жоюмен бастапқы қатынастардың барлық элементтерін қамтитын R қатынасы, яғни, алынған қатынастар кем дегенде бір операнд қатынастарының біріне енетін барлық түйіндерді қамтиды.

Мысалы, R_1 - 35 топтағы оқушылар жиынтығы және R_2 қатынасы - студенттер жиынтығы. Содан кейін

Л қатынасы

ТАӘ	Топ	Жын
Иванов Ф.И.	35	Е
Кириллова Е.Е.	35	Ә
Потапов В.С.	35	Е
Дудко О.В.	35	Ә

 R_2 қатынасы

ТАӘ	Топ	Жын
Иванов Ф.И.	35	Е
Потапов В.С.	35	Е
Таран О.С.	44	Е
Ильин Г.С.	44	Е

 $R (R_1 \cup R_2)$ қатынасы

ТАӘ	Топ	Жын
Иванов Ф.И.	35	Е
Кириллова Е.Е.	35	Ә
Потапов В.С.	35	Е
Дудко О.В.	35	Ә
Таран О.С.	44	Е
Ильин Г.С.	44	Е

9.2 сурет. Қатынастарды біріктіру

R коэффициенті 35 топтағы студенттерді қоса алғанда, жиынтықты білдіреді, немесе ерлер немесе екеуі де жұмыс жасайды (9.2 сурет).

Бірдей өлшемдегі R_1 және R_2 ($R_1 \cap R_2$) екі үйлесімді қарым-қатынас қиылысы бастапқы қатынастарды енгізетін қосындыларды қамтитын R қатынасын тудырады. Алдыңғы мысалдан R_1 және R_2 қатынастары үшін (9.2-суретті қараңыз), нәтижесінде алынған R коэффициенті ер жыныстағы 35 топтағы барлық оқушыларды білдіреді (9.3-сурет).

Сол өлшемнің R_1 және R_2 ($R_1 - R_2$) үйлесімді коэффициенттерін алып тастау - бұл R_1 -ге тиесілі, бірақ R_2 -ге жатпайтын көптеген топтардан тұратын байланыс. Алдыңғы мысалдан R_1 және R_2 -нің ұқсас көрсеткіштері үшін (9.2-суретті қараңыз) R -ге қатынасы әйел жынысындағы 35 топтың оқушысы болады (9.4-сурет). A_1 дәрежесіндегі R_1 қатынасы және A_2 ($R_1 \cap R_2$) R_2 қатынасы өнімі бірдей атрибут аттары болмайды —

 R қатынасы ($R_1 \cap R_2$)

ТАӘ	Топ	Жын
Иванов Ф.И.	35	Е
Потапов В.С.	35	Е

 R қатынасы ($R_1 - R_2$)

ТАӘ	Топ	Жын
Кириллова Е.Е.	35	Ә
Дудко О.В.	35	Ә

9.3 сурет. Қатынастардың қиылысуы

9.4 сурет. Қатынастарды есептеу

R₁ Қатынасы

ТАӘ	Топ	ЖЫН
Иванов Ф.И.	35	Е
Таран О.С.	44	Е

R₂ қатынасы

Топ	Куратор	Курс
35	Кирсанова Л.Н.	3
44	Никитина Л.П.	4

Қатынас R (R, TIMESR₂)

ТАӘ	Топ	жын	Топ 1	Куратор	Курс
Иванов Ф.И.	35	Е	44	Никитина Л.П.	4
Таран О.С.	44	Е	35	Кирсанова Л.Н.	3
Иванов Ф.И.	35	Е	35	Кирсанова Л.Н.	3
Таран О.С.	44	Е	44	Никитина Л.П.	4

9.5 сурет. Қатынасты орнату мысалы

ол кімнің атауы ілінісу тақырыптар болып табылады R_x және R₂ қарым-қатынастар дәрежесін қатынасы R (A₁ + a₂), болып табылады, және дене түрлі қосылыстар луын P_x және бірінші және W элементтері R_j жиынтығы, және соңғы a₂ элементтеріне тиесілі сияқты, бұл R₂ қарым-қатынас бар - R₂ көптеген элементтерге ие болады. Егер сіз бір немесе бірнеше атрибуттардың атауына ие болатын екі қатынастағы өнімді алғыңыз келсе, атын өзгерту операциясы қолданылады. Байланыстың өнімі 9.5 мысалында көрсетілген.

Кодтың реляциялық алгебра жұмыстарының жабылуын пайдаланудың негізгі мағынасы оның негізі операциясымен шынымен пайдалы қосылыстарда анықталатын болып табылады.

Реляциялық алгебраның алғашқы арнайы операциясы - көлденең іріктеу немесе сүзгілеу операциясы немесе шектеу қатынастарының жұмысы болып келеді.

Шектеу әрекеті (алып келу) WHERE үшін екі операцияны қажет етеді: шектелген қатынас және қарапайым шектеу шарты. F (RWHEREf) формуласы бойынша R қатынасын шектеу операциясы (таңдау) - формулада келтірілген логикалық өрнектің шынайылығын қанағаттандыратын R қатынасы сияқты бірдей тақырып пен денемен жаңа қатынасты білдіреді. Атрибут атаулары (немесе бағандардың санын) жазу үшін пайдаланылатын Формула, тұрақты, логикалық операциялар (дербес шешімін таңдау Ал, әдісті НЕМЕСЕ, Not- ЕМЕС), реляциялық операторлары («=», «F», «>», «^», « <«,» ^ «) Және жақшалар көрсетіледі. 9.6 суретте R коэффициентін шектеу операциясының мысалы келтірілген - 44 топтағы студенттер тобының үлгісі (UNDER group = 44) берілген.

R қатынасы

ТАӘ	Топ	Жын
Иванов Ф.И.	35	Е
Кириллова Е.Е.	35	Ә
Потапов В.С.	35	Е
Дудко О.В.	35	Ә
Таран О.С.	44	Е
Ильин Г.С.	44	Е
Федорова Д.С.	35	Ә
Медведева Ж.А.	44	Ә
Пушкина А. А.	44	Ә

R WHERE қатынасы ТОП = 44

ТАӘ	Топ	Жын
Таран О.С.	44	Е
Ильин Г.С.	44	Е
Медведева Ж.А.	44	Ә
Пушкина А. А.	44	Ә

9.6 сурет. Қатынастарды шектеу

Атрибуттарының $\{a_1, a_2, \dots, \text{бір}\}$ (PROJECTA $\{A_1, A_2, \dots, \text{бір}\}$) жиынтығына проекция қарауы салдарынан атрибуттары $\{a_2, a_1, \text{тақырыбында анықталған жиынтығының қатынасы ... Егер}\}$, және нысаны жинақтарын тұратын органмен $\langle AJ: VT, A2: v2, \dots, \text{бір}: \rangle$, бір қатынасы мәндері бар кортеж атрибуты A_1 бар, a_2 атрибуттары мән $v2$ бар екенін осындай көлемдегі \dots , төлсипат vn мәніне ие. Осылайша, проекцияны орындау кезінде қайталанатын кортежи алып тастау арқылы үлгідегі «тік» қарым-қатынас болады. Басқаша айтқанда, өзара қарым-қатынас туралы проекциясы атрибуттары $\{A_1; a_2, \dots, \text{бір}\}$, онда жиынтығы бар $\{A_1; a_2, \dots, \text{бір}\}$ атрибут тізімнің толық үстіңгі бөлігі болып табылатын қарым-қатынастың атағы $\{AJ$ бар қарым-қатынас болып табылады; $a_2, \dots, \text{бір}\}$ және қайталанатын жинақтарын қоспағанда, қарым-қатынас a кортежи бар орган болып табылады.

Жобалау операциясы мынадай қосымшаны жазуға мүмкіндік береді:

- атрибуттар тізімінің жоқтығы барлық атрибуттардың көрсетілуін білдіреді (бірдей проекция жұмысы);
- бос проекцияның нәтижесі бос жиын болып табылады;
- Проекциялау операциясы үлгінің нәтижесін қоса алғанда еркін коэффициентке қолданылуы мүмкін.

Мысалы, алдыңғы мысалдан R қатынасы үшін (9.6 суретті қараңыз), проекцияның нәтижесі **ЖОБАНЫҢ СТУДЕНТІ** {Group} екі кесіндіден тұратын қатынас болады (9.7-сурет).

Проекциялау жұмысы көбінесе көлденең іріктеу немесе сүзгілеу операцияларында аралық қадам ретінде пайдаланылады. Сонымен қатар, сұранысқа жауап алудың соңғы кезеңінде дербес қолданылады.

Бөлу қатынастарына дейін

Бұл статистикалық тұрғыдан қиын, қарау. Мысалы, екі қатынастар: A және B тақырыптары бар $\{a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_m\}$ $\{b_1, b_2, \dots, b_m\}$ сәйкесінше. Атрибуттары $bi (i = 1, 2, \dots,$

$m)$ бір доменде анықталады және бірдей атауға ие болады, яғни, екі қатынас үшін ортақ болып табылады. Көптеген атрибуттарды $\{aj\}$ құрамды атрибуттар деп атаймыз, ал көп атрибуттарды $\{bj\}$ — b құрамды атрибуттар дейміз. Ол кездегі қатынастар бөлінісі A дан B (ADIVIDEBYB) тақырып қатынастары a және a ғзасы, барлық кортеждер құрамындағы $\{a\}$, ондай кортеждер $\{a, b\}$, қатынастары A барлық кортеждерде $\{b\}$, B қатынасына тең болады. Басқа сөзбен айтқанда, A дан басталатын мәндердің нәтижелері, B дан басталатын нәтижелер қатынастары үшін сәйкес келетін мән болып табылады.

Қатынасы
RPROJECTCCTУДЕНТТЕР {Топ}
Топ
44
35

9.7 сурет. Жоба қатынасының нәтижесі

Д қатынасы

ТАӘ	Пәні	Бағасы
Иванов Ф.И.	Ақпараттық жүйе	5
Кириллова Е.Е.	Ақпараттық жүйе	4
Дудко О.В.	Ақпараттық жүйе	5
Федорова Д.С.	Ақпараттық жүйе	5
Ильин Г.С.	Ақпараттық жүйе	4
Федорова Д.С.	Математика	5
Иванов Ф.И.	Математика	3
Кириллова Е.Е.	Математика	5
Федорова Д.С.	Экология	5
Иванов Ф.И.	Экология	4
Дудко О.В.	Экология	4
Федорова Д.С.	Деректер қоры	5
Медведева Ж.А.	Деректер қоры	4
Иванов Ф.И.	Деректер қоры	3

Қатынасы

Пән
Ақпараттық жүйе
Экология
Деректер қоры

R қатынасы
{Ri DIVIDE BYR2}

ФИО
Иванов Ф.И.
Федорова Д.С.

9.8. Бөлу операциясы

($R_1 \div R_2$) R_2 қатынасы бойынша барлық пәндер бойынша балл алған барлық студенттер туралы деректер береді (операция нәтижесі 9.8-суретте көрсетілген).

Қосылыстар $C_f(R_1; R_2)$ формуласы F берген қатынастар R_1 және R_2 жағдайы, формула F сәйкес сынамаларды іріктеу операция нәтижесінде кейіннен қолдана отырып қарым-қатынастардың R_1 және R_2 декарттық өнім арқылы алынуы мүмкін қатынасы R болып табылады.

Бұл операция белгілі бір жағдайларға немесе формулаларға негізделген екі қатынастарды біріктіру қажет болған жағдайларда қолданылады. ережелерін жазу F формуласы дискретизация жұмыс істеуі үшін бірдей болып табылады.

Бір (немесе бірнеше анықталған атрибуттарға логикалық өрнек - - қосылыстар үшін басқаша айтқанда, төлсипаты арқылы құрама R_1 қатынасы және (ешқандай қарым-қатынас ортақ атрибуты аттары бар) төлсипат қатынасы R_2 операция түрінде $Q(R_1 \text{ TIMES } R_2)$ мұндағы Q , атрибут нәтижесі болып табылады). Құрастырмалы $C_f(R_1; R_2)$, Формула F , сондай-ақ *Q-қосылыстар* деп аталатын еркін қарау (ерекше жағдайларда қарағанда), болып табылады. Байланыс операцияларды көрсету үшін, біз сәл бұрын мысалдарда пайдаланылған тақырыптардың және дене қарым-қатынастарды өзгерту. Дерекқор қатынасы $R \wedge \{M.O., \text{ тобы, Пол}\}$, және $R_2 \{группа, куратор, курстык\}$ арақатынасы (9.9 сурет) бар деп есептейік. Бұл құрама R_1 және R_2 қарым-қатынас Топ (R_x , оларды таңбалау арқылы тобын және R_2 тобын, тиісінше, әрбір қатысты үшін) атрибуттар табу қажет.

Нәтижесі бірінші және екінші коэффициенттерінің болуы қортеждерді үйлестіре жасаған және шартты ($R_x \text{ Group} = R_2 \text{ Group}$) қанағаттандыратын қатынасы R , m жасалған болатын. Оның студенттері мен жетекшілері топтардың тізімдерін ($R(R_1 \text{ TIMES } R_2)$ ($R_1 = R_2$ тобы топ) және (курс = 3) ..): 3 курс студенттер ғана таңдауға - басқа шартты қолдануға болады.

Практикалық тұрғыдан маңызды, тікелей байланыстар жағдайлары тепе-тең қосылатын табиғи қосылыс болып табылады.

Бірлесе жұмыс істеу формулада операнды теңдестіруді сипаттайтын фактісі сипатталады. Кейде екі қатынастардың *эквивалентті орналасуы* сол бағандарда орындалады, олардың атрибуттары сәйкесінше аттар мен домендерге ие. Бұл жағдайда біз ортақ атрибутқа қатысты біріктіру туралы айтады. Жоғарыда келтірілген мысалда бір бағанмен біріктірілген әрекеттің ерекше жағдайлары көрсетілген.

Қатынасы R_1

ТАӘ	Топ	Жын
Иванов Ф.И.	35	Е
Кириллова Е.Е.	35	Ә
Потапов В.С.	35	Е
Дудко О.В.	38	Ә
Таран О.С.	44	Е
Ильин Г.С.	44	Е
Федорова Д.С.	35	Ә
Медведева Ж.А.	44	Ә
Пушкина А. А.	44	Ә

Қатынасы R_2

Топ	Куратор	Курс
35	Кирсанова Л.Н.	3
44	Никитина Л.П.	4
38	Петрова Е.М.	3
24	Долинская Н.А.	2

Қатынасы $R (R_1 \text{ TIMES } R_2)$ WHERE $R_1 \text{ Top} = R_2 \text{ Top}$

ТАӘ	Топ	Жын	Куратор	Курс
Иванов Ф.И.	35	Е	Кирсанова Л.Н.	3
Потапов В.С.	35	Е	Кирсанова Л.Н.	3
Дудко О.В.	38	Ә	Петрова Е.М.	3
Таран О.С.	44	Е	Никитина Л.П.	4
Ильин Г.С.	44	Е	Никитина Л.П.	4

9.9 сурет. Қосылу операциясы

Табиғи қосылыс операциясы (JOIN операциясы) жалпы атрибутқа ие (қарапайым немесе композициялық) екі қатынастарға қолданылады. Бұл қатынаста атрибут ұқсас атауға ие (атаулар жинағы) және сол доменде (домендерде) анықталады.

Табиғи біріктіру операциясының нәтижесі - R қатынасы, ол R_1 және R_2 қатынастарының эквиваленттік орналасуын екі қарым-қатынастың біріктірілген жиынтығына жалпы атрибуциялау болып табылады.

Енді Родуляциялық Кодд алгебраның жұмысының анықтамаларын қысқаша тұжырымдауға болады.

1. Бірлескен іс-шараларды орындау кезінде бірдей тақырыптармен екі қарым-қатынас операциясы кезінде, кем дегенде бір операндалық қарым-қатынасқа енетін барлық топтармен байланыс жасалады.
2. Бірдей тақырыптармен екі қатынастардың қиылысу (INTERSECT) операциясы өзара байланыс жасайды, соның ішінде операндалық қатынастарға кіретін барлық түймелер жүзеге асырылады.
3. Бірдей тақырыптармен екі қатынастардың айырмашылығы (MINUS) қатынасы қамтылған барлық топтамаларды қамтитын

қатынас - бұл бірінші операнд, сондықтан олардың ешқайсысы екінші операнд болып табылмайды.

1. Екі қатынастардың өнімі (TIMES) орындалғанда, тақырыптардың қиылысы бос, байланыс жасалады, олардың қосындылары бірінші және екінші операндтердің кесінділерін біріктіру арқылы жасалады.

2. Белгілі бір шарт бойынша шектеудің (WHERE) нәтижесі - бұл шартты қанағаттандыратын ара-операнды кесінділерін қамтитын қатынас болып табылады.

3. Оның қатынасы атрибуттардың жиынтығын алдын ала белгіленген ішкі жиыны арқылы проекциясы (жоба) қатынасуына тиісті көптеген операнд қарым-қатынастарды кортеж болып табылады.

4. Күрделі (JOIN) белгілі бір жай-күйі туралы екі қарым-қатынас, оның луын бірінші және екінші коэффициенттерді үйлестіре жасаған және осы шартты қанағаттандыратын қатынасы, кортежи нәтижесінде қалыптасады.

5. Реляциялық бөлу қадамы (DIVIDEBY) коэффициентінің нәтижесінде (алғашқы төлсипатының тіркелген құны бойынша), екінші төлсипатының орнатылған мәндер екінші операнд құндылықтарының көптеген қамтиды, мұндай бірінші операнд жинақтарын бірінші Атрибут мәндерінде, соның ішінде жинақтарын тұрады.

Кодд алгебрасы артық емес; Сонымен қатар, алгебраның негізгі операциясы ретінде Картесиялық өнімді пайдалану тәжірибесіз оқырмандарды адастыруы мүмкін. Алайда, бұл сәл ескірген және жетілмеген теориялары Реляциялық деректер бойынша барлық дерлік оқулықтарда реляциялық деректер моделін айла-шарғы жасау негізгі тетіктерін талқылау басталады. Себебі SKL (құрылымдық сұрау тілі) тілінің семантикасы көбінесе осы алгебраға негізделеді және біз алдымен Кодд алгебрасымен танысқаннан кейін SQL-ні зерделеу оңай болады.

9.3.

Деректер кестеде сақталады, олар пайдаланушы енгізеді. Бұл жазбалардың физикалық тәртібі деп аталады. Дегенмен, жиі кез-келген өрістер бойынша деректерді сұрыптап, физикалық тәртібіне қарағанда деректерді басқа тәртіпте көрсету қажет. Сұрыптау тәртібін белгілейді

белгілі бір өріске енгізілген құндылықтарды көбейту немесе азайту тәртібінде жүзеге асады. Мысалы, студенттік деректерді топтық нөмірмен және / немесе алфавит бойынша реттеуге болады (9.10 сурет).

Сонымен қатар, белгілі бір критерийге сай келетін ақпараттың үлкен көлемінде, мысалы, оның туған күніне қарай оқушыны табу үшін жиі қажет. Үлкен кестеде іздеу кезінде жазбаларды қарапайым іздеу көп уақытты қажет етпейді, демек, тиімсіз болады. Осы проблемаларды шешудің тиімді құралы индекстерді пайдалану болып табылады.

Индекстер кестедегі нақты өрістер немесе өрістер жиынтығы бойынша іздеуді және сұрыптауды жылдамдатуға мүмкіндік беретін деректер қорларындағы арнайы құрылымдар ретінде ұсынылуы мүмкін. Индекстер деректердің бірегейлігін, яғни бастапқы немесе бірегей кілттерді жасау үшін қолданылады.

Физикалық түрде индекс - жазба мекен-жайын анықтау үшін пайдаланылатын кесте. Индекстер болған жағдайда, көптеген жағдайларда деректерді индекстеудің болмауына қарағанда тезірек жасауға болады, себебі индекстегі мәндер реттеледі және индекстің өзі аз болады. Индекс пайдаланушы немесе жүйе үшін арнайы кесте үшін жасалады.

Индекстелген кестелерде әртүрлі операцияларды орындау жылдамдығын көбейту көбінесе кестелердің өзімен емес, шағын индекс файлдарымен орындалуымен байланысты. Ең үлкен әсер жоғары-

Топ	ТАӨ
35	Иванов Ф.И.
35	Кириллова Е.Е.
35	Потапов В.С.
35	Дудко О.В.
48	Таран О.С.
44	Ильин Г.С.
35	Федорова Д.С.
44	Медведева Ж.А.
44	Пушкина АА.

a

Топ	ТАӨ
35	Дудко О.В.
35	Иванов Ф.И.
35	Кириллова Е.Е.
35	Потапов В.С.
35	Федорова Д.С.
44	Ильин Г.С.
44	Медведева Ж.А.
44	Пушкина АА.
48	Таран О.С.

б

9.10 сурет. Жазбалардың физикалық тәртібі (а); кестедегі жазбалар «Топ» өрісі бойынша сұрыпталады (б)

Индекстелген кестелермен жұмыс істеу үлкен кестелер үшін қол жеткізіледі. Индекс деректер кестесіндегі әрбір жазба үшін негізгі мәндерді қамтиды. Негізгі мәндер бір немесе бірнеше кесте өрістері негізінде анықталады. Сонымен қатар, индекс кестеде тиісті жазбаларға бірегей сілтемелерді қамтиды және осылайша іздеу шартына сәйкес келетін жолдарды іздеуге мүмкіндік береді. Индекстерді пайдалана отырып, жұмыстарды жеделдету индексі іздестіруге оңтайландырылған құрылымға (мысалы, теңгерілген ағаш) байланысты болады.

Индекстерді кітаптағы мазмұн кестесімен салыстыруға болады. Мәтіннің оң орына іздегенде біз ол орналасқан тарауды немесе бөлімінде іздейді, содан кейін мәтінді іздеу үшін қазірдің өзінде мазмұнында дұрыс бетке кітабын ашу және жатырмыз. Индекс кесте көрсеткіші ретінде әрекет етеді, оның алдын ала қарауы кесте жазбаларына сілтеме жасайды. айтуынша индекс мәні, жүйесі пайдаланушыны шығарады деректер алапта, қажетті деректер блогын табады. Осылайша, индекс кестеде деректерді іздеу үдерісін жеделдетуге мүмкіндік береді, ал кейде пайдаланушының сұранысы бойынша алынған деректердің реттелуін жылдамдатуға мүмкіндік береді. Енді біз бастапқы кілтін кестедегі әрбір жазбаны анықтауға мүмкіндік беретін индекс ретінде анықтай аламыз.

Кестелердегі ақпаратқа жеке қол жеткізуді ұйымдастыру әдістері негізінен келесі факторларға байланысты:

- Индекс файлы жазбаларының кілттік өрісіндегі мазмұн түрі;
- негізгі кестені жазу үшін пайдаланылатын анықтамалардың түрі (индекстер);
- қажетті жазбаларды іздеу әдісі.

Дерекқорлармен жұмыс істейтін қосымшаларды әзірлеу кезінде ең қарапайым индекстер болып табылады. Қарапайым индекстер бір кесте өрісінің мәндерін пайдаланады. «Оқушылар» кестесіндегі қарапайым индекстің мысалы «Идентификатор» өрісі (жеке нөмір) болуы мүмкін (9.11-сурет).

Көп жағдайда деректерді белгілі бір тәртіпте көрсету үшін, бір өрістегі қарапайым индексті пайдалану жеткілікті, кейде күрделі индекстерден аулақ болу мүмкін емес жағдайлар. Композиттік индекс екі немесе одан да көп кесте өрістерінің мәндеріне негізделген. Композиттік индекстерді пайдалану туралы жақсы мысал «Қызметкерлер» кестесі болуы мүмкін (9.12-сурет). Бұл адамның атымен қарапайым индекс ретінде пайдалану түсінікті болып табылады.

Идентификатор	ТАӘ	Топ	Жын
101	Иванов Ф.И.	35	Е
102	Кириллова Е.Е.	35	Ә
103	Потапов В.С.	35	Е
104	Дудко О.В.	35	Ә
105	Таран О.С.	44	Е
106	Ильин Г.С.	44	Е
107	Федорова Д.С.	35	Ә
108	Медведева Ж.А.	44	Ә
109	Пушкина А. А	44	Ә

9.11 сурет. «Идентификатор» бағаны бойынша қарапайым индекс мысалы

Бұл жағдайда кестенің келесі өрістеріне негізделген композиттік индексті пайдалануға болады: «Бөлімше», «Табельді нөмір».

Дерекқордағы кестелерде индекстер болмауы мүмкін. Бұл жағдайда үлкен кесте үшін белгілі бір жазбаның іздеу уақыты өте маңызды болуы мүмкін және индекстің қолданылуы қажет болады. Екінші жағынан, сіз тым көп индекстерді құруға қатыспауыңыз керек. Индекстердің санын арттыру кесте жолдарын қосу, жаңарту және жою операцияларын баяулатады, себебі индекстің өзі жаңартылуы қажет. Оңтайлы сұрау өнімділігі үшін әдетте сұрауларда жиі пайдаланылатын кесте бағандарында индекстер жасалады. Сонымен қатар, индекстер қосымша көлемде жұмыс атқарады.

Бөлімше нөмірі	Табельді нөмір	ТАӘ	Қызметі	Жалақысы
001	1	Иванов Федор Иванович	Директор	50000
001	2	Потапов Валерий Самуилович	Директордың	40000
001	3	Таран Олег Сергеевич	Бас инженер	40000
002	1	Ильин Иван Васильевич	Бас есепші	45000
002	2	Берковская Ольга Николаевна	Есепші	20000
002	3	Дремина Елена Евгеньевна	Есепші	20000
003	1	Попенко Борис Сергеевич	Бас механик	35000
003	2	Федорова Дарья Сергеевна	Инженер	30000

9.12 сурет. «Бөлімше» және «Қызметкерлер саны» бағандарындағы композиттік индекстің мысалы

Сондықтан, индексті жасамас бұрын, сұрау жұмысының жоспарланған пайдасы индексті сақтау үшін компьютер ресурстарының қосымша құнынан асатынына көз жеткізу керек.

9.4.

КЕСТЕЛЕРДІ БАЙЛАНЫСТЫРУ. СІЛТЕМЕЛІК БҮТІНДІК ТҮСІНІГІ

Нақты әлемдік нысандар арасындағы байланыстар дерекқор деректерінің құрылымында көрініс табуы мүмкін. Әдетте, өзара байланысты дерекқор өзара байланысты кестелер жиынтығынан тұрады. Екі немесе одан көп кестелер арасындағы байланыс ұйымдастыру *кестелерді байланыстыру* деп аталады.

Кестелер арасындағы байланыстарды СУБД арқылы ұсынылған құралдарды пайдаланып, кестені жасағанда және бағдарламаны іске қосқанда орнатуға болады. Екі немесе одан көп кестені байланыстыруға болады. Бұл басты кестеде (ата-аналық кесте деп те аталады) әрбір жазба үшін, бағынатын кестеде (балалық кесте деп аталатын) бір немесе одан да көп жазбалар болуы мүмкін сияқты теңдей қатынастар немесе бағыныстылық қатынастары болуы мүмкін. Реляциялық дерекқорда байланыстырылған кестелерге қоса, кез-келген басқа кестеге қосылмаған бөлек кестелер болуы мүмкін.

Кестелерді байланыстыру үшін қосылымның өрістерін - «сәйкестік» өрістерін пайдаланыңыз. Байланыс өрістері индекстелуі және бірдей түрі болуы керек. Үстелдік кестеде индексі сыртқы кілт деп аталатын негізгі кестеге қосылу үшін шақырылады. Бұл көрсеткіштің құрамы негізгі кестенің индекстік өрістерінің құрамымен толығымен немесе ішінара сәйкес келуі керек.

Дерекқор кестелері арасында байланыстың үш түрі бар:

- 1) «көпшілігіне біреу»;
- 2) «біреуге біреу»;
- 3) «көпшілікке көбі».

«Көпшілігіне біреу» қатынасы ата-аналық кестенің жалғыз жазбасы бірнеше баланың жазбаларына сәйкес келуі мүмкін болған кезде көп қатынастар пайда болады. Көптеген қарым-қатынастар кейде көптеген қатынастар деп аталады. Кез келген жағдайда кестелер арасындағы өзара қарым-қатынас мәні өзгеріссіз қалады. Бір-көпшілік қарым-қатынас реляциялық дерекқорлар үшін кең таралған болып табылады. Ол деректер құрылымын моделдеуге және иерархияға мүмкіндік береді.

ID	NAME_GROUP	K_RUK
1	38-пкс	Федоров С.В.
2	18-пкс	Иванов Ф.И.
3	44-э	Попенко Б.С.
4	36-тг	Филин О.К.

9.13 сурет. «Топтар» кестесі

«Біреуден көпшілікке» қатынастарын көрсету үшін екі кестені қолданамыз. Біреуі оқу орындарының топтарының тізімін (9,13 сурет), ал екінші жағынан - студенттердің тізімін көрсетеді (9,14 сурет).

«Топтар» кестесінде келесі құрылым бар:

- ID— жазба идентификаторы, бүтін түрдің негізгі өрісі;
- NAME_GROUP— топ атауы, таңба түрі өрісі;
- K_RUK— сынып мұғалімі, таңба түрі өрісі. «Студенттер» кестесі келесі өрістерді қамтиды:
- ID— жазба идентификаторы, бүтін түрдің негізгі өрісі;
- NAME_STUD— Студенттің аты, таңба түрінің өрісі;
- ID_GROUP— топ идентификаторы, бүтін түрдегі индекс өрісі.

Бұл жағдайда «Топтар» кестесі негізгі болып табылады, ал бағынатын кесте - «Студенттер» кестесі. Бұл жағдайда, кестелер тиісінше «Топтар» және «Оқушылар» кестелерінен, яғни, топтық идентификатор бойынша, «ID» және «ID_GROUP» өрістерімен байланысты, студенттің белгілі бір топқа тиесілі екенін анықтай алады.

ID	NAME\STUD	ID_GROUP
1	Иванов Петр Иванович	1
2	Петров Иван Сергеевич	1
3	Таран Ольга Сергеевна	2
4	Дудко Олеся Владимировна	3
4	Дремина Елена Евгеньевна	1
5	Федорова Дарья Сергеевна	2
6	Лесовая Вера Николаевна	2
7	Семенова Елена Геннадьевна	3
8	Никитина Любовь Петровна	4
9	Курганский Владимир Иванович	4

9.14 сурет. «Студенттер» кестесі

«Топтар» кестесінен топпен жазбаны таңдаған кезде екі кестені байланыстырғаннан кейін, осы топтағы студенттермен бірге «Студенттер» кестесінде ғана қол жетімді болады. Мысалы, егер бірінші кестеде көрсеткіш «18-дана» (ID = 2) тобының жазбаларына орнатылған болса, онда кестеде ID_GROUP = 2 пайда болады (9.15 сурет).

ID тобы яғни топтың осы студенттердің басын анықтау мүмкін емес болады, бізге тиісті жазбалар үшін пән үстелінің «Студенттер» байланысын жасайды, ал жазба (кездейсоқ немесе әдейі) жойылатын үстел «Топ» деп аталады. Яғни, ақпараттық қолайсыз болып табылатын деректер, жоғалады. Демек, ол пән кестеге байланысты жазбада бар болса, мастер (ата-ана) кестеде жазбаны тыйым салу немесе жою үшін қажетті болып табылады, немесе шебер кестенің жазбасымен бірге осы кестедегі тиісті жазбаны жояды. Сол сияқты, бос өрісті қалдыру немесе негізгі кестеде болмайтын мәндерін енгізу үшін, негізгі кестеде кез келген жазбаға оларды байланыстырмайды. Яғни соның салдарынан осы кестедегі деректерді енгізу мүмкін емес. Егер негізгі кестенің негізгі өрісіндегі немесе бағынатын кестенің сілтеме өрісінде мәнді өзгертсеңіз, байланыс ақаудың жоғалуына немесе бүлінуіне әкеледі.

Ата-ана кестесіндегі бір жазба бала кестесіндегі «*бірден бір*» қатынасының жазбасына сәйкес келгенде, бір-бірімен қатынас болады. Бұл қатынас бірден көп қатынастарға қарағанда әлдеқайда аз. Дерекқор кестесінің қосымша ақпаратынан өсуін қаламаса, ол қолданылады. Мысалы, кез-келген кестеде қызметкерлер туралы ақпараттың мұрағаты бар,

ГО	NAME_GROUP	K_R.UK	ГО	NAME_STUD	ID_GROUP
1	38-пкс	Федоров С.В.	1	Иванов Петр Иванович	1
2	18-пкс	Иванов Ф.И.	2	Петров Иван Сергеевич	1
3	44-э	Попенко Б.С.	3	Таран Ольга Сергеевна	2
4	36-тг	Филин О.К.	4	Дудко Олеся Владимировна	3
			4	Дремина Елена Евгеньевна	1
			5	Федорова Дарья Сергеевна	2
			6	Лесовая Вера Николаевна	2
			7	Семенова Елена Геннадьевна	3
			8	Никитина Любовь Петровна	4
			9	Курганский Владимир Иванович	4

9.15 сурет. «Топтар» және «Студенттер» кестелерінің жазбалары арасындағы сәйкестік үлгісі

онда жүзден астам өріс (сипаттамалар) және жүздеген мың жазбалар бар. Бұл мұрағатта үнемі «ТАӘ», «туған күні» және «мекен жайы» пайдаланылады. Деректерді пайдалану тұрғысынан бұл деректерді оқып-үйреніп, барлық кестені ашып, жадқа көптеген деректерді жадыға жүктеуге болады. Сондықтан оны екіге бөлуге болады - бір дүкенде жиі сұралатын ақпарат, ал екінші жағынан - сирек пайдаланылатын ақпарат. Содан кейін сіз осы кестелерді бір-бірімен байланыстыра аласыз. Кейде кейбір ақпарат «құпиялылық» болуы керек, содан кейін барлық пайдаланушыларға қол жетімді болмауы тиіс деректер жеке кестеде орналастырылуы және құпия сөзді енгізу арқылы «қауіпсіз» болуы мүмкін.

«Көпшіліктен көпшілікке» қатынасы келесі жағдайларда қолданылады:

- ата-аналар кестесіндегі бір жазба баладан бірнеше жазбаға сәйкес келеді;
- бала кестесіндегі бір жазба ата-аналық кестеде бірден артық жазбаларға сәйкес келеді.

Іс жүзінде «көпшілікке көп» қатынасы өте сирек қолданылады. Себептер кестелер мен олардың жазбаларының өзара әрекеттесуін ұйымдастырудың күрделілігі болып табылады. Сонымен қатар, «көптеген адамдарға» қатынасы үшін негізгі және бағынышты кестелердің ұғымдары мағынасы жоқ. Реляциялық дерекқордағы кез-келген «көпке-көпше» қарым-қатынасы қосымша кестелерді енгізу арқылы бір-көп қарым-қатынас (бір немесе бірнеше) ауыстырылуы керек.

Сілтемелік бүтіндік деректердің реляциялық қорында — бұл байланыстырылған кестелер арасындағы сәйкестік. Анықтамалық тұтастық әдетте негізгі кілт пен шетелдік кілтін біріктіру арқылы орындалады. Сілтеме тұтастығын қолдау үшін, кестедегі кез келген өріс шетелдік кілтпен бірге жариялануы тек ата-аналық кестенің бастапқы кілт өрісінен мәндерді ғана қамтуы керек. Сілтемелік тұтастық пайдаланушылар немесе қолданбалар бойынша теңдесі жоқ деректердің енгізілуіне жол бермейді. Көптеген реляциялық СУБД-да, екі кесте арасында байланыс жасау кезінде қолдануға болатын әртүрлі сілтеме тұтастығы ережелері бар.

Байланыстырылған кестелермен жұмыс істеу келесі мүмкіндіктерге ие:

- Байланыс өрісін өзгерткенде, екі кестенің жазбалары арасындағы байланыс бұзылуы мүмкін. Сондықтан, байланыс өрісін редакциялағанда, негізгі кестенің жазбалары тиісті түрде болуы керек

байланыстыру өрісін және бағынатын барлық кестелердің байланыс өрісінің мәндерін өзгерту (каскадты өзгерту);

- Негізгі кесте жазбасын жойсаңыз, сіз оны жоюыңыз және сәйкес келуіңіз керек. Бағынысты кестедегі жазбалар (каскадты жою);
- Бағыныңқы кестеге жазбаны қосқанда, оның мәні байланыс алаңындағы негізгі кестенің байланыс өрісінің мәніне тең болуы керек.

Каскадты жоюға, каскадты өзгертуге, жаңа мән орнату мүмкін оларды құру немесе қайта құрылымдау болып табылады. Бұл шектеулер, сонымен бірге өрістер, индекстер және басқа элементтер құрылымға енгізілген кестелер мен осы дерекқормен жұмыс істейтін барлық қосымшаларға қолданылады деректер. Осындай шектеулерді жүзеге асыру, сондай-ақ грамматикалық: үстелдер арасындағы байланыстарды орнату және жою, байланыс алаңдарын редакциялауға тыйым салу немесе тыйым салу және т.б..

9.5. ДЕРЕКТЕРДІҢ РЕЛЯЦИЯЛЫҚ ҚОРЛАРЫНДАҒЫ БҮТІНДІК ПРИНЦИПТЕРІН ҚОЛДАУ

Деректердің тұтастығы деректер базасының технологиясындағы негізгі ұғымдардың бірі болып табылады. Бұл сипаттамалар дерекқордағы ақпарат әрдайым дұрыс және толық болуын қамтамасыз етуге мүмкіндік беретін құралдардың қолжетімділігін білдіреді. Адалдық ережелері орнатылуы керек және олар деректер базасымен бірге сақталып, жаһандық деңгейде құрметтелуі керек. Деректер сақталғанына қарамастан (интерактивті режимде, импорт арқылы немесе арнайы бағдарлама арқылы) деректердің тұтастығы қамтамасыз етілуі керек.

Реляциялық үлгіде нақты әлемдік нысандар өзара байланысты қатынастар жиынтығы ретінде ұсынылады. Адалдық доменнің ақпараттық моделінің шынайы әлемдік нысандарға сәйкес келуі және олардың әрбір сәтте өзара байланысы ретінде анықталуы мүмкін. Доменде салынған модель үшін маңызы бар кез-келген өзгеріс дерекқорда көрініс табуы тиіс және домен тұрғысынан ақпараттық моделді нақты түсіндіру керек.

Міндеттілік шектеулері деректер базасының жүйесінен басқа мемлекетке аудару деректерінің дәйектілігін қамтамасыз етеді және деректер доменіне дерекқорда сақталған деректермен тиісті түрде көрсетілуге мүмкіндік береді. Адалдықты шектеу анық және айқын емес болып бөлінеді.

Жасырын шектеулер деректер құрылымының өзін анықтайды. Мысалы, Студенттік типтік жазбалар «Топ» деректер жиынтығының міндетті түрде мүшелері болып табылатыны тұтастай тұтастық шектеу ретінде қызмет етеді, бұл әрбір оқушының міндетті түрде топтың мүшесі болуы керек.

Ашық шектеулер дерекқор кестесіндегі деректер сипаттамасының тіл құралдары (DDL, DataDefinitionLanguage) арқылы көрсетіледі. Ашық шектеулер ретінде, деректер мәндеріне қатысты шарттар жиі пайдаланылады.

Мысалы, қызметкердің жалақысы теріс болуы мүмкін емес, ал қызметкерді жұмысқа қабылдау күні міндетті түрде басқа жұмысқа ауысу күнінен аз болады. Осы шектеулерді жүзеге асыру СУБД-мен оның жұмыс істеуі барысында бақыланады.

Статикалық және динамикалық тұтастығы шектеулері бар. Статикалық шектеулер доменнің барлық жайларына тән және динамикалық шектеулер доменді бір мемлекеттен екіншісіне көшіру мүмкіндігін анықтайды. Статикалық тұтастығы шектеулерінің мысалдары төлқұжаттың бірегей санының немесе туған күніндегі шектеулердің талаптары болуы мүмкін, бұл ағымдағы күннен көп болуы мүмкін емес. Динамикалық тұтастығын шектеудің мысалы банк жүйесінің шектеулілігі болып табылады, оған сәйкес клиенттің ақпаратын жоймайынша, оны жоюға болмайды.

СУБД деңгейіндегі деректердің тұтастығын басқару құралдары кіреді:

- бастапқы кілтті тағайындау үшін енгізілген құралдар, соның ішінде автоматтандырылған кадаммен өрістердің түрімен жұмыс жасау құралдары, СУБД өзі жаңа бірегей мәнді болып келеді;
- кесте қатынастары туралы ақпаратты беретін және сілтеме тұтастығын бұзатын кез-келген операцияны автоматты түрде басатын деректер тұтастығын сақтау құралдары.

Реляциялық деректер моделіндегі адалдықты қолдау келесі аспектілерге ие.

Біріншіден, бұл реляциялық дерекқордың «реляциялық қатынас» түрінің біртекті деректер құрылымымен жұмыс істеуіне мүмкіндік беруі керек деген тұжырымдаманың құрылымдық тұтастығын қолдау болып табылады. Бұл жағдайда «реляциялық қатынас» тұжырымдамасы классикалық реляциялық дерекқорда оған қойылған барлық шектеулерді қанағаттандыруы керек.

Реляциялық СУБД тек типтің деректер құрылымымен қатынасы арқылы жұмыс істейді. Реляциялық кестеге сәйкес келетін ережелерді сақтау қажет:

- кестеде біркелкі кортеждер жоқ;
- бағандар қатынастар атрибуттарына сәйкес келеді;
- әрқашан алғашқы кілттер бар болып табылады;
- кез келген атрибут бірегей атауға ие болады;
- өндірістік кестедегі жолдар тәртібі;
- бағандар тәртібімен ғана ерекшеленетін екі қатынас бірдей болып саналады.

Екіншіден, бұл тілдік тұтастығын қолдау болып табылады, ол реляциялық дерекқор SQL стандартынан төмен емес деректерді сипаттау және өңдеу үшін тілдерді қамтамасыз етуі керек. Стандарттарға сәйкес келмейтін төмен деңгейлі басқа деректерді өңдеу құралдары қол жетімді болмауы керек.

Үшіншіден, бұл сілтеме тұтастығын қолдау болып табылады. Өзара байланыстырылған кестелердің атрибуттарының арасындағы қатынастардың белгілі бір принциптерін қамтамасыз етуді білдіреді:

- балалар кестесінің жолдары олармен байланыстырылған негізгі кестенің жолы жойылған кезде жойылады;
- негізгі кестенің жолдары олармен байланыстырылған негізгі кестенің жолы жойылғанда, ал ата-аналық кесте кілті анықталмаған нөлдік мәнмен ауыстырылады.

Реляциялық тұтастығы кезінде қосымша немесе жою операциялар деректер модификациялау кезінде дәйекті деректер базасын мемлекеттік қолдау көрсетуді көздейді. Құрылымы, тілдік және деректер базасын тұтастығы реляциялық деректер құрылымдармен жұмыс ережелерін анықтайды. Тұтастығының осы үш түрін қолдауға қойылатын талаптар әр СУБД осы ережелерді керек етеді және реляциялық моделі және реляциялық деректер базасын қосымшалар әзірлеу деректер базасын пайдалана отырып құру кезінде әзірлеушілер осы ескермеуге тиіс деп болжайды.

9.6.

ДЕРЕКТЕРДІҢ РЕЛЯЦИОНДЫ ҮЛГІСІНІҢ КЕМШІЛІКТЕРІ МЕН ЕРЕКШЕЛІКТЕРІ

Деректер базасын ұйымдастырудың салыстырмалы көзқарастары 1960 жылдың соңында жасалды. Эдгар Кодд зерттеулері бірден дереу алыс және кең таралған болды. Ал негізгі теориялық

осы саладағы нәтижелер 1970-ші жылдардың басында алынды және сол уақытта Реляциялық СУБД-нің алғашқы прототипі пайда болады, ұзақ уақыт бойы мұндай жүйелердің тиімді іске асырылуына қол жеткізу мүмкін емес деп саналады. Дегенмен, реляциялық тәсілдің артықшылықтары мен реляциялық дерекқорларды ұйымдастыру және басқару әдістері мен алгоритмдерін жасауы реляциялық жүйелер жаһандық дерекқор нарығындағы үстем жағдайды иеленді. 1980-жылдардың ортасында. реляциялық жүйе әлемдік нарықтағы ерте деректер базасын дерлік тоқтатты. Ал енді реляциялық СУБД ең тиімді болып қала береді.

Бұған реляциялық деректер моделінің маңызды жетістіктері ықпал етті.

1. Ең алдымен бұл қарапайым кесте ұсыну, оның негізінде ең көп тараған тақырыптық салаларды модельдеуге болады. Реляциялық модель тек табиғи құрылымына негізделген деректерді сипаттау құралын ұсынады. Машина деректерін көрсету үшін қосымша құрылымдарды енгізудің қажеті жоқ. Тиісінше, бұл модель бір жағынан бағдарламалардың максималды тәуелсіздігін қолдайтын жоғары деңгейлі деректер тілін, сондай-ақ машина ұсыну және деректерді ұйымдастыруды негіздейді. Бұл абстракцияның нақты және ресми түрде анықталуы мүмкін.

2. Реляциялық деректер үлгісінің тағы бір артықшылығы - көптеген теориялар мен математикалық логикаға негізделген қарапайым және бір мезгілде күшті математикалық аппараттың болуы және деректер базасын ұйымдастырудағы реляциялық тәсілдің теориялық негізін қамтамасыз ету. Сонымен қатар, реляциялық көзқарас деректерді сыртқы жадыда белгілі бір физикалық деректер базасын ұйымдастыру қажеттілігінсіз манипуляциялау мүмкіндігін береді.

Жалпы танылған еңбектермен қатар, реляциялық модельде бірқатар кемшіліктер бар

Реляциялық жүйелердің тән шектеулері, олардың қарапайымдылығының тікелей салдары болып табылады. Бұл әсіресе өте күрделі деректер құрылымдарын талап ететін дәстүрлі емес қолданбаларда (мысалы, автоматтандырылған жобалау жүйелері) жүзеге асады.

1. Доменнің семантикасын дұрыс көрсете алмау. Бұл сондай-ақ олардың құрылымының қарапайымдылығы үшін төлем болып табылады.

Қарудан кейінгі жүйелер саласындағы қазіргі заманғы зерттеулер негізінен бұл кемшіліктерді жоюға арналған.

БАҚЫЛАУ СҰРАҚТАРЫ

1. «Байланыстыру», «деректер типі», «домен», «атрибут», «өтпелі», «қатынас схемасы», «бастапқы кілт» негізгі түсініктерін анықтаңыз.
2. «Қатынас», «деректер типі», «домен», «атрибут», «өтпелі», «байланыс сұлбасы», «бастапқы кілт» түсініктерінің математикалық анықтамасын беріңіз.
3. Өзара қарым-қатынастың негізгі қасиеттерін сипаттаңыз.
4. Реляциялық алгебра деп аталады.
5. Реляциялық алгебраның негізгі теориялық операцияларын сипаттаңыз.
6. Арнайы реляциялық операцияларды сипаттаңыз.
7. Біртұтас деп аталатын және екілік операция дегеніміз не?
8. Бірлестік, қиылысу, қарым-қатынастарды алып тастау мысалдарын келтіріңіз.
9. Неліктен қарым-қатынастың нәтижесі сирек пайдаланылады?
10. Жұмыстың және байланыстың жұмысы жалпыға ортақ болып табылады?
11. Тікелей байланыс пен табиғи байланыстың жұмысы туралы не ерекше?
12. Дерекқордағы индекс дегеніміз не?
13. Индекстелген кестедегі деректермен жұмыс жылдамдығының өсуі неге байланысты?
14. Кестелерді дерекқорларға байланыстыру механизмін түсіндіріңіз.
15. «Деректердің тұтастығы» дегеніміз не?
16. Байланыстырылған кестелермен жұмыс істеу ерекшеліктерін сипаттаңыз.
17. «Бір-біріне», «біреуіне көпшілігіне», «көпшілігіне көп» деген байланыс түрлерін сипаттаңыз.
18. Қандай жағдайларда жеке қарым-қатынас қолданылады?
19. Неліктен «көптеген адамдарға» байланысты болдырмау керек?
20. Реляциялық деректер моделінің артықшылықтары мен кемшіліктерін келтіріңіз

ТҰЖЫРЫМДАМАЛЫҚ, ЛОГИКАЛЫҚ ЖӘНЕ ФИЗИКАЛЫҚ ДЕРЕКТЕР ҮЛГІСІ ҚҰРЫЛЫСЫНЫҢ ҚАҒИДАТТАРЫ

10.1. ДЕРЕКТЕР ҚОРЫН ЖОБАЛАУ МІНДЕТТЕРІ

Деректер базасын құру процесі ұзақ, тапсырыс берушімен, білім саласындағы мамандармен талқылауды қажет етеді. Корпоративтік маңызды ақпараттық жүйелерді әзірлеу кезінде деректер жобасы - бүкіл жүйенің негізі болып табылады.

Деректер базасын жобалау өзара байланысты деректер үлгілерінің жиынтығын құрудан тұрады. Деректер моделін құру процесі деректерді өңдеуден және манипуляциялаудан ажырамайды.

Деректерді жобалау екі негізгі кезеңнен тұрады: логикалық және физикалық модельдеу. Логикалық модельдеу кезеңінде әзірлеуші дамыған деректерге қойылатын талаптарды жинайды, домен сипаттамасын жасайды және нақты ДҚБЖ-ға тәуелді емес модель жасайды. Физикалық модельдеу кезеңінде әзірлеуші ДББЖ мен нақты қолданушы қосымшалары үшін оңтайландырылған үлгі жасайды.

Ақпараттық жүйе үшін деректерді құру кезінде ең маңызды міндеттер ақпараттық жүйенің барлық қажетті функцияларының жиынтығын қамтамасыз ететін дұрыс логикалық деректер құрылымын жасауға байланысты. Нашар әзірленген деректер, әдетте, тиімсіз және тіпті пайдасыз.

Деректерді әзірлеу өте қиын. Көбінесе бұл көптеген қайшы талаптармен берілген. Дұрыс логикалық құрылымды құру деректерді қалыптастыруға және өңдеуге әсер ететін барлық факторларды кешенді талдауды қарастырады.

Жақсы жобаланған деректер қоры:

- дерекқордың мазмұнына арналған барлық пайдаланушы талаптарын қанағаттандырады. Дерекқорды жасамас бұрын, дерекқордың жұмыс істеуі үшін пайдаланушы талаптарын зерттеуді жүргізу қажет;
- деректердің дәйектілігі мен дәйектілігін қамтамасыз етеді;
- ақпараттың табиғи, түсінуге оңай құрылымын қамтамасыз етеді. Деректер қорының сапалы құрастырылуы дерекқорға көбірек «мөлдір» болып келеді және оңай түсінуге мүмкіндік береді; Сондықтан қате деректерді енгізу ықтималдығы төмендейді және деректер базасына қызмет көрсету сапасы жақсарады;
- Дерекқордың өнімділігі үшін пайдаланушы талаптарын қанағаттандырады. Көптеген ақпарат көлемінде өнімділікті сақтау мәселесі барлық негізгі кемшіліктерді дереу «жыпылықтайды». Дизайнердің міндеті ең оңтайлы дерекқорды әзірлеу үшін осы факторлардың барлығын ескеру болып табылады. Деректер қорын жобалаудың негізгі міндеттері төменде көрсетілгендей болуы мүмкін:
- барлық қажетті ақпаратты дерекқорда сақтауды қамтамасыз ету;
- сұралған барлық деректердің қол жетімді болуын қамтамасыз ету;
- деректердің қайталануын және қайталануын азайту;
- дерекқордың тұтастығын қамтамасыз ету.

Бұл мәселелерді шешу жолында біз түсінеміз. Бірақ алдымен кейбір терминдердің анықтамасын есте сақтау қажет.

Пәндік аумақ — нақты әлемнің бөлігі, деректер қорын көрсету. Мысалы, пән аймағы ретінде, біз студент кітапхананы, кәсіпорын бухгалтерлік есеп, адами ресурстарды, банк, дүкен және т.б таңдай аламыз.

Мамандану маңызсыз немесе тіпті мағынасыз, сондай-ақ, ақ маңызды ұғымдар мен деректерді қамтиды. Сіз қоймада тауарларды есепке таңдай Сондықтан, егер, содан кейін «термин партиясы» тақырыбы аймағы ретінде маңызды болып табылады. Сонымен қатар тауарларды есепке алу үшін шот алған қызметкерлерінің балалары саны маңызды емес ақпарат болып табылады. Бірақ жалақыны есептеу үшін балалардың қол жетімділігі туралы мәліметтер қажет. Осылайша, кез-келген доменді қараудың өз шекаралары бар және дерекқорды құрастыру кезінде домен шекарасында ақпараттық объектілерді және домен шекарасынан тыс ақпараттың дерексіз дерекқорын бөлу қажет.

Пәндік аумақ үлгісі — бұл кез-келген тәсілмен көрсетілетін тақырыптық саланың ресми білімі. Оқу пәнінің мәтіндік сипаттамалары (лауазымдық нұсқаулықтар, құжат айналымының сипаттамасы, бастапқы құжаттардың мысалдары, шығыс есептері және т.б.) осындай құрал ретінде қызмет етуі мүмкін. Деректер базасын дамытуда анағұрлым ақпараттылығы мен пайдалы болуы - арнайы графикалық белгілердің көмегімен орындалатын пәннің сипаттамасы. Ақпараттық жүйенің одан әрі дамуының табысы доменді модельдеудің қаншалықты жақсы екеніне байланысты.

Деректер қорын жобалау процесі доменнің ақпараттық құрылымының бейресми ауызша сипаттамасынан домендегі нысандардың формалдандырылған сипаттамасына модель тұрғысынан ауысу дәйектілігі.

Жалпы жағдайда дизайнның келесі кезеңдерін ажыратуға болады.

1. Доменнің ақпараттық нысандарын жүйелік талдау және ауызша сипаттау.

2. Доменнің концептуалды (ақпараттық) үлгісін жобалау домендегі нысандардың кейбір семантикалық модельдері бойынша ішінара ресми сипатталған сипаттамасы.

3. Деректерді қисынды немесе логикалық ДБ жобалау, яғни деректер қорының сипаттамасын қабылданған логикалық деректер үлгісі.

4. Дерекқордың физикалық дизайны, яғни тиімді жұмыс істеуін қамтамасыз ету үшін сыртқы медиадағы тиімді дерекқорды таңдау.

Осы кезеңдердің әрқайсысында белгілі бір деректер үлгісі әзірленеді. Екінші және үшінші кезеңдер арасында біздің жобамыз қандай стандартты СУБД қолданылатынын ескеру қажет болса, онда деректер базасын жасау процесі шартты түрде тиісті қадамдарды орындау реті арқылы көрсетілуі мүмкін (10.1-сурет).

Концептуалды (инфологиялық) жобалау — пәндік саланың семантикалық үлгісін құру, яғни абстракцияның ең жоғары деңгейінің ақпараттық үлгісі. Мұндай үлгі қандай да бір СУБД мен деректер үлгісіне бағытталмай жасалады. «Семантикалық үлгі», «тұжырымдамалық үлгі» және «инфологиялық үлгі» терминдері синонимдер болып табылады. Сонымен қатар, осы контексте сөздер бірдей қолданылуы мүмкін



10.1 сурет. Деректер қорларын жобалау кезеңдері

Пәндік аумақтың жүйелі талдауы, Концептуалды (ақпараттық) жобалауы, логикалық (дatalogиялық) жобалау, физиологиялық жобалау

«дерекқор үлгісі» және «домен үлгісі» (мысалы, «тұжырымдамалық дерекқор үлгісі» және «доменнің концептуалды үлгісі»), себебі мұндай үлгі бұл шындықтың әдісі және деректер базасы осы шындыққа арналған.

Дерекқордың тұжырымдамалық үлгісінің нақты түрі мен мазмұны осы мақсат үшін таңдалған ресми аппаратпен анықталады. Әдетте ER диаграммалары сияқты графикалық белгілер қолданылады.

Дерекқордың тұжырымдамалық үлгісі жиі қамтиды:

- ақпараттық объектілердің сипаттамасы немесе тақырыптық аймақтың түсінігі және олардың арасындағы қарым-қатынас;
- тұтастығы шектеулерінің сипаттамасы, яғни деректердің жарамды деректеріне қойылатын талаптар және олардың арасындағы қарым-қатынас.

Логикалық(дatalogиялық) жобалау — дерекқор схемасын белгілі бір деректер үлгісіне негізделе жасалады, мысалы, қатынас үлгісі. Реляциялық деректер үлгісі үшін логикалық үлгі - бұл бастапқы кілттерді көрсететін қарым-қатынас сызбаларының жиынтығы, сондай-ақ сыртқы кілттер болып табылатын қатынастар арасындағы «қатынастар» болып табылады. Тұжырымдамалық үлгіні логикалық үлгіге айналдыру әдетте ресми ережелерге сәйкес жүзеге асырылады. Бұл кезең автоматтандырылған болуы мүмкін.

Логикалық конструкция кезеңінде белгілі бір деректер үлгісінің ерекшелігі ескеріледі, бірақ нақты СУБД ерекшелігі айтылмайды.

Физикалық жобалау — дерекқорды басқару жүйесі үшін дерекқор схемасын жасалады. Белгілі бір ДҚБЖ-ның ерекшелігі атауға шектеу қоюы мүмкін

дерекқор нысандары, қолдау деректер түрлері бойынша шектеулер және т.б. Әрі қарай, физикалық жобалау кезінде нақты СУБД ерекшелігі физикалық жинақтаушыға байланысты шешімдерді таңдау (таңдау диск жады басқару әдістері, дерекқор файлының бөлімшесі, және құрылғы қол әдістері қамтиды деректерге), индекстерді құру және т.б. түрлі құралдар домен мен дерекқор модельдеу пайдалана отырып деректер базасын жобалау және дамыту, сондай-ақ тіпті нақты жүйесі шеңберінде әр түрлі мақсаттар үшін үлгілер бірқатар талап етеді. Әрі қарай, осы кезеңдердің әрқайсысын егжей-тегжейлі қарастырамыз.

10.2.

ПӘНДІК АУМАҚТЫ ТАЛДАУ

Зерттеу аймағын талдау кез-келген ақпараттық жүйені құрудан тұрады және оның дамуының бір бөлігі болып табылады. Бұл кезеңде пайдаланушылардың ақпаратқа қажеттіліктері анықталады, бұл өз кезегінде болашақ жүйе үшін дерекқордың құрылымы мен мазмұнын алдын ала анықтайды. Зерттеу нысаны нақты нысандардың жиынтығы болып табылады. Бұл нысандардың әрқайсысы белгілі бір қасиеттер жиынтығына (атрибуттарға) ие. Доменнің нысандары арасында әртүрлі маңызы бар сілтемелер болуы мүмкін. Ақпараттық жүйе үшін деректер базасын құру, пайдаланушы әртүрлі негізде ақпаратты ұйымдастыруға тырысады. Бұл, егер қажет болса, керекті деректер жиынтығын алу үшін - сипаттамалардың қажетті комбинациясы бар үлгіні алу үшін жасалады. Бұл деректер құрылымдалған жағдайда ғана мүмкін, яғни деректерді ұсыну әдістеріне қатысты келісімдерге сәйкес келеді.

Домендік үлгі зерттелетін пәннің құрылымын және / немесе жұмысын имитациялайды және осы салаға барабар болуы керек. Домендік модельдің ерекше рөлі болашақ ақпараттық жүйе үшін қалыптастыру кезеңінде қалыптасады. Доменді алдын-ала модельдеу сізге жобалау жұмыстарының уақытын және уақытын қысқартуға және тиімді және сапалы жобаны алуға мүмкіндік береді. Пәндік саланы үлгілеуді жүзеге асырмастан, стратегиялық мәселелерді шешуде көптеген қателерге жол бермеу ықтималдығы жоғары, бұл экономикалық шығындар мен жүйені кейіннен қайта құру үшін жоғары шығындарға алып келеді.

Пәндік аумақты зерттеу, жалпы алғанда, әзірленіп жатқан жүйе үшін де деректер базасы да қатысуы керек. Бұл жағдайда, деректер үлгісі жүйенің барлық нысандары анықталған кезде, олардың өзара әрекеттесуінің логикасы және берілген ақпарат ағындары ғана жасалуы мүмкін. Дерекқор - жұмыс істегенде жүйе пайдаланатын берілетін деректердің репозиторийі ретінде жұмыс атқарады. Деректер базасы жүйенің негізі деп айтуға болады, сондықтан оған өте байсалды түрде қарау керек. Деректер базасын құру кезінде көптеген қателіктер туындайды, оның құрылымы туралы ойластырылған болмауы және дизайнерлік нашар жұмыс сипаттамаларын зерттеу басталады.

Дизайн кезеңінде жүйе әзірлеушілері мен домен сарапшылары мен тұтынушылар арасында өзара түсіністікке қол жеткізу өте маңызды, өйткені әрбір жобаның өз көзқарасы бар. Бұл жұмыс нысандарын, жүйенің маңызды функцияларын, ақпараттық ағымдардың және олардың өзара жүйесін кезең-кезеңмен бөлу дейін төмендейді кезде. даму нақты мәселелер бойынша қатысушылардың тұрғысынан презентация түрінде келуі мүмкін жобада бірге жұмыс асқыну әкеледі, әр түрлі болуы мүмкін. Бұл жағдайда маңызды құрал - дизайн шешімдерінің бірыңғай тілдік көрінісі – үлгілеу тілі. Біз барлық жобаға қатысушыларға бір-бірін түсінуге мүмкіндік беретін белгілер жүйесін, процестерді, нысандарды, құбылыстарды және олардың өзара байланысын сипаттайтын ережелерді анықтауымыз керек («бірдей тілде сөйлеу»).

Үлгілеу тілі — бұл дизайндағы үлгілерді сипаттау үшін қолданылатын графикалық белгілер жиынтығы.

Нотация бұл үлгі пайдаланылатын графикалық нысандардың жиынтығы, және модельдеу тілінің синтаксисі. Ал үлгілеу тілі, бір жағынан, дизайнерлер екінші жағынан, пайдаланушыға шешімдер түсінікті жасауға тиіс - дизайнерлер келісілген жүйесін құрайтын бағдарламалық пакеттері түрінде іске асырылатын жобалық шешімдердің құралы өте формальды және біржақты анықтамасын беруге болады.

Кешенді көзқарас пен ортақ белгілерді пайдалану үлгілеу домені кезінде, сондай-ақ бағдарламалық жүйесін дамытудың кейінгі кезеңдерінде ғана емес, өте маңызды болып табылады.

Пәндік аумақты зерттеу онда жатқан процестерді тікелей бақылаудан, құжаттарды зерттеуден тұрады

жүйедегі айналымға, сондай-ақ осы процестерге қатысушыларға сұхбат беру болып табылады.

Домендік үлгілеуді әзірлеу кезінде белгілі бір шекаралар айқындалады, соның ішінде деректердің логикалық үлгісі жасалуы мүмкін, яғни қарастырылып жатқан тақырып доменінің доменінен шығып кетпейтін нысандарды үлгілеуге болады. Сонымен бірге алынған барлық үлгілерді талдауға және талдаудың үрдісін күрделендіруге болмайды.

Домендік зерттеулердің нәтижесі жүйелік талаптар, спецификациялар, ақпараттық ағымдар мен олардың сипаттамаларының тізбесі болуға тиіс. Бұл үшін DFD (деректер ағынының диаграммасы) үлгілерін, UML (бірыңғай модельдеу тілі), доменді сипаттаудың стандартты әдістері жиі пайдаланылады.

10.3. КОНЦЕПТУАЛДЫ ҮЛГІЛЕУ

Концептуалды үлгі — пәндік облыстың семантикалық құрылымы болып табылады, олардың арасында түсініктер мен қарым-қатынас, бұл, атап айтқанда, жиынтығы болып табылады. Тұжырымдамалық үлгісі болашақ деректер базасын бастапқы прототипі болып табылады, бірақ нақты СУБД сілтеме салынған. Домендік тұжырымдамалық модельді құру процесі креативті және ресми түрде қалыптасуы қиын. тұжырымдамалық моделін салу және оның ақпаратты логикалық қарым-қатынастарды түсіну, оның семантикасы, пәндік облыстың жақсы білуді талап етеді.

Концептуалды домендік үлгіні әзірлеу жүйені пайдаланушылардың ақпараттық қажеттіліктерін талдау негізінде құрылған. Біз тұжырымдамалық үлгі қасиеттері мен сипаттамалары, осы ұғымдардың жіктеу, түрлері, өнер белгілері жағдайларда, және оған процестердің заңдарына бірге, пәндік аймағын сипаттау үшін пайдаланылатын байланысты ұғымдардың тізімінен тұрады деп айтуға болады.

Домен үлгісіне келесі талаптар қойылады:

- тақырыптық аумақтың құрылымын анық сипаттайтын формализация;
- графикалық дисплей құралдарын пайдалану негізінде тұтынушылар мен әзірлеушілерге түсінікті болу;
- АЖ-дегі домен үлгісін физикалық іске асыру үшін құралдардың қолжетімділігін болжайтын іске асырылу;

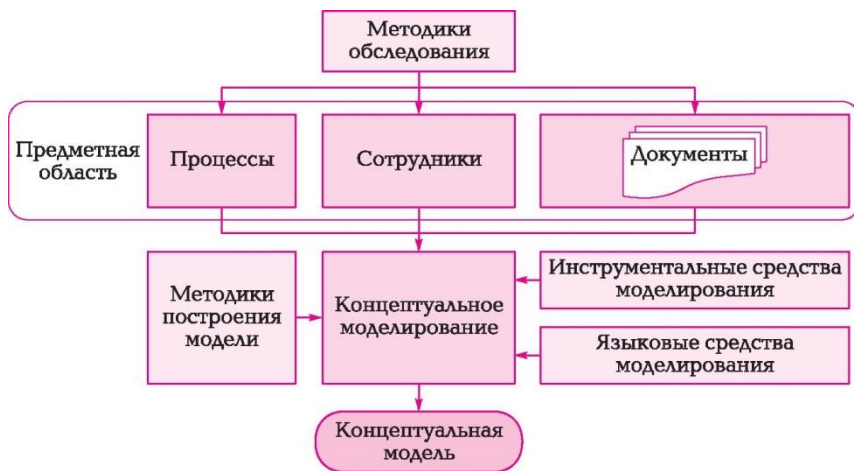
- модификациядағы қарапайымдылық (әртүрлі себептер бойынша жаңа нысандарды енгізу немесе оларды өзгерту қажет), үлгі бұрын белгіленген деректерді өзгертпестен жаңа деректерді енгізу мүмкіндігін қамтамасыз етуі керек);
- белгілі бір әдістер мен есептелген көрсеткіштер негізінде домен үлгісін іске асыру тиімділігін бағалауға мүмкіндік беру.

Домен үлгісін жеткіліктілігі негізгі критерийі әзірленген деректер базасын функционалдық толықтығы болып табылады.

Тұжырымдамалық үлгіде тек қана дерекқор мамандары ғана емес, оңай оқитын пәндік саланың ресми сипатталған сипаттамасы болуы керек. Және бұл сипаттаманың соншалықты кең болуы керек, дерекқор жобасының тереңдігін және дұрыстығын бағалауға болады, және, әрине, жоғарыда айтылғандай, ол белгілі бір СУБД-мен байланысты болмауы керек. СУБД таңдау - бұл бөлек тапсырма, дұрыс шешім қабылдау үшін СУБД-ны нақты бағдарсыз жобаны құру қажет.

1970 және 1980 жылдары, әдебиетте дерекқордың тұжырымдамалық дизайны «инфологиялық дизайн» және «инфологиялық үлгі» терминдерімен белгіленді. 10.2 суретте нәтижесінде бастапқы ақпаратты тұжырымдамалық үлгісін құру үшін кестесін ұсынады.

Зерттеу әдісі, пәндік аумақ, процесстер, қызметкерлер, құжаттар, үлгіні құру әдістемесі, концептуалды үлгілеу, үлгілеудің құралдық жабдыктары, үлгілеудің тілдік жабдыктары, концептуалды үлгі



10.2 сурет. Концепциялық модельді құрастыру кестесі

Дерекқордың тұжырымдамалық үлгісі көбінесе:

- ақпараттық нысандардың сипаттамасы немесе тақырыптық аймақтың түсінігі және олардың арасындағы қарым-қатынас;
- тұтастық шектеулерінің сипаттамасы, яғни деректердің жарамды деректеріне қойылатын талаптар және олардың арасындағы қарым-қатынас.

Тұжырымдамалық дизайн, ең алдымен, дерекқор үлгісіндегі доменнің семантикасын білдіруге тырысады. Дерекқор технологиясы саласындағы мамандар әрқашан үлгілерде семантиканы ұсыну мәселесіне қызығушылық танытты. 1970 жылдары семантикалық модельдер деп аталатын бірнеше түрлі деректер үлгісі жасалды. Барлық үлгілердің оң және теріс жағы болды.

1976 жылы Питер Чен «мән — байланыс» атты үлгіні ұсынған болатын (*entity—relationship*). Ол доменнің негізгі іскерлік ережелерін көрсететін нысандар мен қарым-қатынастарды қамтиды. Қазіргі уақытта «мән-байланыс» үлгісі үшін жалпы аты ER-моделі деп аталады. Кейінірек көптеген авторлар ұқсас үлгілердің өздерінің нұсқаларын әзірледі. Бірақ «мән-байланыс» үлгілерінің барлық нұсқалары бір идеядан келеді - графикалық кескін мәтін сипаттамасынан гөрі айқынырақ болып келеді. Қазіргі уақытта ER-үлгі доменді семантикалық құрылымдаудың нақты форматты стандартына айналды - дерекқорларды тұжырымдамалық үлгілеу стандарты жүзеге асырылды.

ER-үлгісінің семантикалық негізі келесі позициялар:

- нақты әлемнің бөлігі, яғни деректер базасында орналастырылуы тиіс өзара байланысты нысандар жиынтығы субъектілердің жиынтығы ретінде ұсынылуы мүмкін;
- Ұйым субъектілердің түрлеріне қарай жіктелуі мүмкін: әрбір субъектінің данасы (нысанды білдіретін) белгілі бір сыныпқа тағайындалуы мүмкін - әр данасы оларға тән қасиеттері бар және оларды басқа сыныптардың мәндерінен ажырата алатын субъектілердің түрі болып табылады.

Мұнда біз «мәнін» тұжырымдаманың ақпараттық сипатын және оның доменнің материалдық немесе мнималды нысандармен өзара байланысын тағы бір рет атап өтуіміз керек. Доменнің кез-келген нысаны қасиеттерге ие, олардың кейбіреулері қолданбалы міндет тұрғысынан елеулі болып бөлінеді. Бұл жағдайда, мысалы, пәндік доменді талдау және жүйелеу процесінде әдетте сыныптар ерекшеленеді.

Сынып — бұл атрибуттар жиынтығы түрінде көрсетілген қасиеттер жиынтығы бар нысандардың жиынтығы.

Бұл сыныптың барлық нысандар үшін қасиеттерінің жиынтығы бірдей болады, бірақ осы сипаттамаларын нақты құндылықтар, сондай-ақ осы құндылықтарды, әсіресе үйлесімі нысандарын іс жүзінде екіншісіне нысанның бір данасын ерекшелейтін нысанға әр түрлі болады.

Қарау деңгейіндегі доменнен реттеу тұжырымдамасын (есте нысан) ретінде қарастырылады, оның тұжырымдамалық үлгісі, нысанның, яғни «мәні» ұғымына сәйкес келеді; материалдық дүниенің бір бөлігі ретінде (адамның сана-сезіміне қарамастан) «үлгі субъекті» ұғымына сәйкес келеді; «Субъект типі» түсінігі нысандар сыныбына сәйкес келеді.

Болашақта, нысандар мен сынып тұжырымдамалық моделі-қарастырылмайды жеке жағдайларда, яғни, осы екі деңгейдегі тиісті ұғымдарды ажырата бермейді. Е нысаны мен мәні, нысанның меншік және тұлғаның меншік тұжырымдамасын басын мойнына алады.

ER-үлгісін визуализациялауға болатын стандартты графикалық белгілер ретінде, «мән-қатынас» диаграммасы (ER-диаграммасы) ұсынылды. Табиғат, атрибуттары және қарым-қатынастар: ER-диаграмма үш құрылымдық элементтері болып табылады.

Мән— бұл тақырып домені үшін маңызды болып табылатын ұқсас нысандардың класы. Әрбір ұйымда сингулярлық есіммен көрсетілген атау болуы керек. Ұйым атауы үлгіде бірегей болуы керек.

Ұқсас нысандар класы модельденген мәнді «айқын анықталатын нысан» ретінде анықтайды. Әрбір нысан объективтік құндылықтар жиынтығымен ерекшеленетін сияқты, субъект субъектілердің жекелеген даналарын ажыратуға мүмкіндік беретін осындай атрибуттар жиынтығы арқылы анықталуы керек.

Кәсіпорын нақты немесе дерексіз нысандардың (адамдар, оқиғалар, штаттар, нысандар және т.б.) мысалдар жиынтығы.

Мәндердің экземпляры — бұл осы ұйымның нақты өкілі болып табылады. Ұйымның әр данасы сол бір тұлғаның кез-келген басқа данасынан ерекшеленуі тиіс (бұл талап реляциялық кестелерде қайталанатын топтамалардың болмауы туралы талапқа ұқсас). Ұйым нақты немесе дерексіз нысандардың (адамдар, оқиғалар) мысалдар жиынтығы,

лар, заттар ж.т.б.). Мысалы, «Қызметкер» ұйымының өкілі «Иванов II қызметкері» болуы мүмкін. Кәсіпорын даналары бөлек болуы керек, яғни субъектілер осы ұйымның әрбір данасына бірегей қасиеттерге ие болуы керек. Мысалы, бірегей тұлға әрбір данасын анықтау үшін «Қызметкерлерге» байланысты олардың табиғаты әрдайым кәсіпорынның ішінде бірегей болады, ол атрибут «Қызметкерлердің саны» енгізіледі. Яғни, ұйымның бірегей идентификаторы атрибут, атрибуттардың тіркесімі, қарым-қатынастардың тіркесімі немесе сол бір ұйымның басқа даналарынан ұйымның кез келген данасын бір-бірінен ерекшеленетін қатынастар мен атрибуттардың тіркесімі бола алады.

Жоғарыда айтылғандай, домен нысаны-келісілген өкілдігін сипаттау үшін кейбір белгілер жүйесі (символдар, тілі) бар. Белгі Баркер - кең таралған. Біз бұл белгілерге сүйенеміз. Кәсіпорынның өзара қарым-қатынас диаграммасын құру үшін, біз реляциялық деректер базасының теориясынан кейбір анықтамаларды еске салуымыз керек.

Баркер белгілеріндегі диаграммада субъект тіктөртбұрышпен, кейде дөңгелек бұрыштармен ұсынылған (10.3-сурет). Әрбір ұйымның бір немесе бірнеше атрибуттары бар.

Мән атрибуттары — бұл сипаттың белгілі бір сипаты болып табылатын атақты сипат. Атрибуттың аты сингулярлық затпен (мүмкін, сын есімдермен) көрсетілуі керек. Қызметкер субъектінің атрибуттарының мысалдары «Қызметкерлер нөмірі», «Тегі», «Әкесінің аты», «Аты», «Позиция», «Жалақы» және т.б. сияқты атрибуттар болуы мүмкін. Атрибут субъектінің жағдайын анықтау, белгілеу, жіктеу, сандық бағалау немесе көрсету үшін арналған (10.3-сурет, б).



10.3 сурет. Баркердің белгілерінде субъектіні сәйкестендіру:

a — атрибутсыз; *б* — атрибутты көрсете отырып; *в* — атрибут пен оның типтерін көрсете отырып; # — нақты, * — міндетті, о — міндетті емес

Осылайша атрибут нақты немесе дерексіз нысандардың жиынтығына байланысты сипаттамалардың немесе *сипаттардың* белгілі бір түрі болып табылады. Мүліктің сипаты, сондай-ақ мүліктің және объектінің (объектінің) арасындағы қарым-қатынастың сипаты әртүрлі болуы мүмкін. Негізгі қасиеттерін қарастырамыз (атрибуттар).

Атрибут экземпляры — ұйымның белгілі бір данасының нақты сипаттамасы, төлсипаттың мәні (мысалы, «түс» атрибуты, ал «жасыл» - атрибут данасы).

Атрибут көп немесе жалғыз болуы мүмкін, яғни сипатты көрсететін төлсипат бір уақытта бірнеше мәнге немесе сәйкесінше бір ғана болуы мүмкін. Мысалы, қызметкердің бірнеше мамандықтары болуы мүмкін, бірақ жалғыз мән - бұл INN.

Атрибут қарапайым болуы мүмкін (қосымша тапсырмалар тұрғысынан одан әрі бөлуге жатпайды) немесе құрамдас - егер оның мәні қарапайым қасиеттердің мәндерінен құралған болса. Мысалы, «Туған жылы» сипаты қарапайым және «Мекенжай» сипаты күрделі болып табылады, себебі ол қарапайым қасиеттердің мәндерін қамтиды: Қала, Көше және Үй.

Атрибуттың егжей-тегжейлі дәрежесін қалай анықтау керек? Яғни, тапсырмалар құжаттарды тұтынушының мекен-жайы басып келеді, бірақ толық мекен-жайы, одан әрі талдау қажет етпеген жағдайда қала, көше, үй, пәтер, мысалы, біз қарапайым жылжымайтын мүлік үшін бүкіл мекенжайын алады, және ол толық сақталады және Пайдаланушы болады. Бұл деректерді дерекқордан толық таңбалар қатарынан ғана ала алады. Мұндай клиент қала болып табылады, сондай-мекенжайын құрайтын бөлшектер, талдау біздің проблемалар бар болса, біз жеке деректер элементі ретінде қаланы бөлектеу қажет, тек осы жағдайда, пайдаланушы, мысалы, барлық клиенттер үшін сұрау оны кіруге және орындауға болады, белгілі бір қалада, мысалы, Санкт-Петербуркте тұрады.

Дегенмен, пайдаланушы қажеттіліктері мен тапсырыс берушінің толық мекен-жайы болса, онда мекен-жай бойынша ақпарат қалған, сондай-ақ, мысалы, «жасыру-көбею үшін жоғары тұрғыда болады.» Бөлек саласында кездесетін деп аталатын болады, сақтау керек жұмыстары көп болады. Бұл жағдайда дерекқордағы әрбір клиент үшін «Қала», «Қысқартылған мекенжай» деген мағына беріледі.

Кейбір жағдайларда негізгі және алынған қасиеттер арасындағы айырмашылық пайдалы болып табылады. Мысалы, «Жеткізуші» жобамен берілген бөлшектердің санын қосу арқылы есептелетін «Жеткізілген бөлшектердің жалпы саны» сипатына ие болуы мүмкін. Егер ұйымның барлық даналары үшін мүліктің болуы міндетті болмаса, онда бұл шарт шартты деп аталады.

Мысалы, барлық қызметкерлердің «академиялық дәрежесі» болмайды.

Атрибуттардың мәндері тұрақты (статикалық) немесе динамикалық болуы мүмкін, яғни уақыт өзгереді. Мысалы, «ИНН» сипаты статикалық болып табылады және «Мекен жай» сипаты динамикалық болып табылады. Сипат динамикалық болса, анықталмайды, бірақ оның ағымдағы мәні әлі көрсетілмеген.

Мәннің алғашқы кілті — бұл атрибуттардың жинақталмаған жиынтығы, олардың жиынтығында ұйымның әрбір данасы үшін бірегей болып табылады. Кілттен кез-келген атрибутты жою оның бірегейлігін бұзуға әкеледі.

Бастапқы кілт ұйымның әр данасын бірегей түрде анықтауға арналған, яғни ол объектіні сәйкестендіруге мүмкіндік беретін сипаттамалардың жиынтығын білдіреді. Негізгі атрибуттар тізімінің жоғарғы жағында орналасқан және «#» белгісімен белгіленген (10.3, с сурет).

Мүмкін, осы пәндік салада нысанның қасиеттері біз үшін қызықтырмайды. Осыған байланысты, ER-модельде қасиеттері жоқ және тек идентификаторларымен ұсынылған нысандар болуы мүмкін.

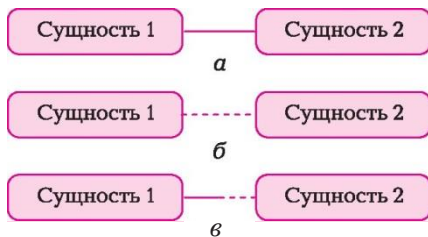
Атрибуттар негізгі және сипаттамалық болып бөлінеді. Негізгі төлсипат файлдары бастапқы кілттің бөлігі болып табылатын төлсипаттар болып табылады. Басқа атрибуттар сипаттама деп аталады.

Сипаттамалық атрибуттар міндетті немесе міндетті емес болуы мүмкін. Әрбір нысан үшін талап етілетін атрибуттар міндетті түрде нақты мәнге ие, міндетті емес - анықталмауы мүмкін. Міндетті және қосымша сипаттама атрибуттары тиісінше «*» және «o» белгілерімен белгіленеді.

Байланыс — басқа немесе өзіне басқа бір тұлғаның қатынасы. Әр сілтемеде екі байланыс тәсілдерінің бірі болуы мүмкін. Міндетті қосылу - бір тұлғаның кез келген сатыдағы тұтас сызықпен көрсетілген басқа тұлғаның кем дегенде бір (яғни, кем дегенде бір) данасымен байланысқан болса көрсетіледі (сурет 10.4, а.).

Міндетті емес байланыс бір субъектінің данасы басқа субъектінің бір немесе бірнеше даналарына байланысты болуы мүмкін немесе кез-келген данамен байланысты болмауы мүмкін сызық сызығымен белгіленеді (10.4-сурет, б). Байланыстың әртүрлі ұштардан өзгеше болуы мүмкін (10.4-сурет, с).

Әрбір ұйым әртүрлі типтегі байланыс түрлерін (біреуі бір, біреуден көпке, көпшілікке көп) басқа үлгідегі модельдерге (10.5-сурет) ие болуы мүмкін.



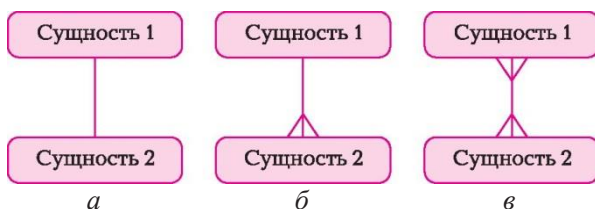
10.4 сурет. Байланыс үлгілігі:

a — міндетті; *б* — міндетті емес; *в* — аралас (нүктелі сызық жолдың ортасына дейін көрсетіледі)

Осы терминологияны пайдалану арқылы байланыс түрлерін келесідей анықтауға болады. Бір-бірімен қарым-қатынас бірінші инстанцияның бір данасы екінші субъектінің бір данасымен байланыстырылғанын білдіреді. Мұндай байланыс, ең алдымен, бір ұйымның екіге дұрыс емес екенін көрсетеді (кейде деректердің бір бөлігін «жіктеу» қажет болған жағдайда осы байланыс түрі қолданылады).

Біреуден көпке дейінгі қатынас бірінші инстанцияның әрбір данасы екінші субъектінің бірнеше даналарына байланысты екенін білдіреді. Көптеген-көп қатынастар дегеніміз, бірінші субъектінің әр данасы екінші субъектінің бірнеше даналарына және керісінше байланыстырылуы мүмкін. Байланыстың бұл түрі - бұл модельдің бастапқы кезеңдерінде қолайлы уақытша қосылу түрі. Болашақта, мұндай қарым-қатынас аралық ұйымды оқшаулау арқылы екіден көп қатынастарға ауыстырылуы керек.

Тәуелсіз мән жүйеде әрдайым қатысатын тәуелсіз деректерді ұсынады. Олар басқа субъектілермен байланысты болуы мүмкін немесе жоқ болуы мүмкін.



10.5 сурет. Баркердің белгілеріндегі әр түрлі байланыс түрлерін белгілеу:

a - «бір-біріне»; *б* - «біреуден көп»; *в* - «көпшілігі көп»



10.6 сурет. Баркердің белгілеріне байланысты субъектінің тағайындалуы

Тәуелді субъект жүйеде басқа субъектілерге байланысты деректерді ұсынады, сондықтан ол әрқашан басқа ұйымдармен байланыстырылуы керек.

Байланысты ұйым екі немесе одан да көп субъектілер арасындағы байланысқа байланысты деректерді ұсынады. Әдетте, бұл түрдегі көптеген көптеген қатынастарға арналған үлгіде пайда болады (10.6-сурет).

Егер ұйым данасы оның негізгі сипаттамалары арқылы анықталса, онда ұйым толықтай анықталған деп айтылады. Әйтпесе, *сәйкестендіру мәні байланысты* ұйым сызықша байланыс желісін (10.7 сурет) көрсетеді, бұл атрибуттары қолдану арқылы жүзеге асырылады.

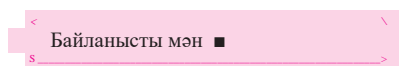
ER-диаграммада көрсетілген Кейбір тұтастығын шектеулер, басқалары табиғи тілде сипатталған.

Мысал ретінде белгілі бір өнімді тұрақты клиенттерге жеткізу және сату процесін бейнелейтін дерекқорды қарастырыңыз. Дерекқор автоматтандырылған ақпараттық жүйесіне арналған «Көтерме қойма» болып табылады.

«Өнім», «жеткізуші» және «сатып алушы» - талдау доменіне негізделген, шын мәнінде трек-нұсқаулықтарын оқшауланған болады.

«Тұтынушылар көптеген тауарларды сатып ала алады», «тауарларды көптеген сатып алушылар сатып ала алады» деген бірден-бір байланыс бар. «Тауарлар» және «жеткізушілер» арасындағы ұқсас қарым-қатынас. Мұндай байланыстарды типті (10.8 сурет) «көптеген көптеген» болып табылады.

Дегенмен, реляциялық деректер үлгісі көп-көпшілік қарым-қатынасты бірнеше бір-нан-көп қатынастарға ауыстыруды қажет етеді. Бұл қарым-қатынас тауарлар процесінің сату / сатып алу көрсетеді тұлғаның басқа түрін қосуға болады шешу үшін —



10.7 сурет. Баркердің белгілеуінде басқа біреудің сәйкестендіруін анықтау

10.8 сурет. ER-диаграммасының алғашқы нұсқасы

біз «Келіп түскен жол жүру құжаттары» және «Кіріс шот-фактурасы» деген қауымдастырылған субъектілерді енгіздік (10.9-сурет).

Нысандар арасында байланыс орнату. Бір сатып алушы бірнеше рет тауарларды сатып ала алады, сондықтан Сатып алушы нысандары мен Сату есебінен бірден көп қатынастар бар. тауарларды Әрбір атауы бірнеше рет объектілері арасындағы, нәтижесінде мәмілелер бойынша «тауарлар» айналысуға болады және «шот-фактура» байланыс «көптеген бір», бар. Сонымен қатар, сілтемелер «Табыс туралы заң» және «Тауарлар» арасында құрылады.

Субъектілердің атрибуттарын талдайық. Әрбір жеткізуші мен сатып алушы заңды тұлға болып табылады және аты, мекен-жайы, банктік деректемелері бар. Әрбір өнімде атауы, бағасы бар, өлшем бірлігі сипатталады. Әрбір шот-фактура бірегей нөмірі, берілген күні, мөлшерде және бағаның тауарлар тізімін, сондай-ақ шот-фактура жалпы сомасы бар. Сатып алушылар тауарды сатып алады, сонымен бірге сатып алынатын тауарлар саны мен бағасы туралы деректер енгізілетін шығын шот-фактураларын алады. Әрбір сатып алушы бірнеше шот-фактураларды ала алады. Әрбір шот-фактура бір сатып алушыға беріледі. Әрбір жүк жөнелтімінде кемінде бір өнім болуы керек («бос» жүкқұжат бола алмайды). Әрбір өнім өз кезегінде бірнеше жүк жөнелтімдерінде бірнеше тұтынушыға сатылуы мүмкін. «Өнім» және «Жеткізуші» арасындағы өзара қарым-қатынастарды анықтау үшін ұқсас логикалық тізбекті құруға болады. Сатып алушы бір болуы мүмкін уақытша және өнім беруші болып табылады, сондықтан осы екі нысан «Контрагенттің» бір мәнімен біріктіріледі.



10.9 сурет. ER-диаграммасының аралық нұсқасы

Дерекқорды құруда маңызды міндет - әрбір дананы бірегей түрде анықтайтын атрибуттарды анықтау, яғни бастапқы кілттерді идентификациялау. «Тауарлар» кестесі үшін тауарлардың атауы бастапқы кілт болып табылмайды, себебі әртүрлі тауарлардың атауы бірдей болуы мүмкін, сондықтан біз «Кодты» бастапқы кілт енгіземіз, ол арқылы біз мақаланың мақаласын түсінеміз. Сол сияқты, «Аты» да «Контрагенттер» кестесінде негізгі кілт болып табылмайды. Паспорттық нөмір, салық төлеушінің сәйкестендіру нөмірі немесе әр контракты бірегейлендіретін кез-келген басқа атрибут ретінде түсінуге болатын жеткізушілер мен сатып алушылар үшін бастапқы кодты енгіземіз. «Шығын шот-фактурасы» / «Кіріс алуы туралы» кестелер үшін «Нөмір» өрісі бастапқы кілт болып табылады, себебі нөмір әрбір шот-фактураны бірегей түрде анықтайды. Бастапқы кілт ретінде біз бір өрісті емес, кейбір өрістер жиынтығын таңдай алдық, бірақ қарапайым бастапқы кілттермен модельдеу процесін көрсету үшін пайдаланады.

Енді сіз мұның бәрін диаграммада жасай аласыз. Осылайша, нақтылаудан кейін диаграмма күрiште көрсетiлгендей көрiнедi (10.10 сурет).

Бұл диаграмма әзірленген жүйенің деректер ағындарының диаграммасында көрсетілген барлық шығыс деректерін алу мүмкіндігімен тексерілуі керек.

«Нысан» және «меншік» ұғымдары салыстырмалы болып табылады. Доменнің әрбір үлгісінде өзін өзі санау керек-



10.10 сурет. ER-диаграммасының нақты нұсқасы

нысан, ал бұл - басқа нысанның мүлкі осы тақырыптық аумақты қарастыру аспектісіне байланысты болады. Мысалы, кітапты есепке алудың ақпараттық жүйесі жасалсын. Кітаптар үшін баспагердің аты көрсетіледі. Баспагерлер туралы ешқандай ақпарат сақталмайды; Бұл негізде арнайы өңдеу жүргізілмейді. Бұл жағдайда «Баспа үйі» жеке объектісін бөліп шығару қажет емес, бірақ ол тиісті объектінің «Кітаптар» сипатын қарастыруы керек. Егер тақырыптық салада жариялаушылар туралы қосымша ақпарат көрсетілсе, мысалы, олардың мекен-жайы, телефон нөмірі, содан кейін «Баспа үйі» тәуелсіз нысан ретінде қарастырылуы керек.

Тұтастай алғанда, ER-модельдегі тәуелсіз нысан ретінде таңдалуы керек нәрселер туралы келесі нұсқауларды бере аласыз.

ER-модельіндегі тәуелсіз объект ретінде, субъектілерді бейнелеу қажет:

- олардың кез-келген қасиеттері бекітілген;
- біреуден астам қосылуға қатысады.

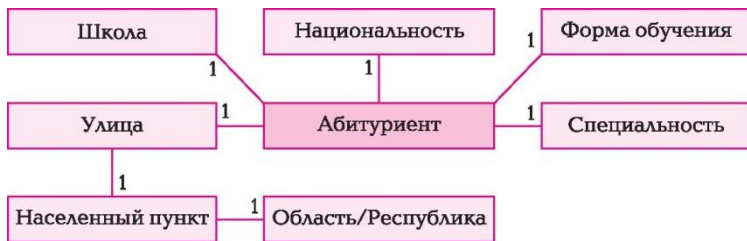
Егер күмән туындаса, тәуелсіз объект құру туралы шешім қабылдаған жөн, өйткені бұл болашақта модельдің аз өзгеруін талап етеді.

Есіңізде болсын, сандық сипаттамалар әрқашан нысанның қасиеттері және ешқашан тәуелсіз нысандар болып табылады. Мысалы, туған жылы тәуелсіз нысан ретінде қарастырыла алмайды.

ER-үлгісінде осы тақырыпта талқыланған барлық нәрсе (кіріс құжаттарында, шығыс құжаттарында және т.б.) көрсетілуі керек. Толық ER-модельді құрастырғаннан кейін сақталған көрсеткіштердің құрамын анықтау қажет. ER-модельінен логикалық модельге көшу тек сақталған индикаторлар үшін орындалуы керек.

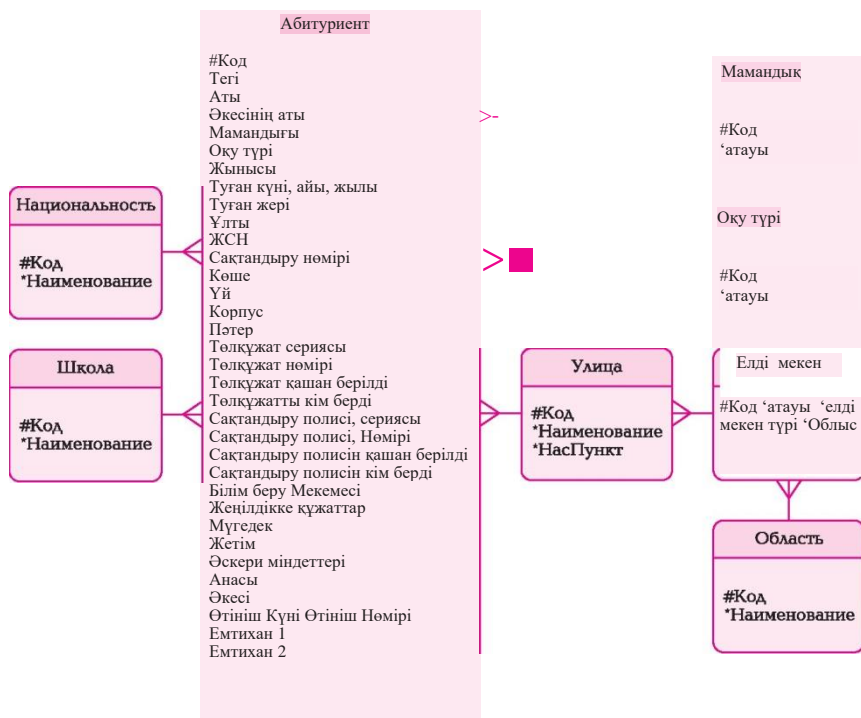
ER-диаграммасының құрылысының тағы бір мысалын қарастырайық. «Өтініш беруші» автоматтандырылған ақпараттық жүйе үшін деректер үлгісін құру қажет деп есептейік. Оны дамытудың мақсаты - талапкерлерді қабылдауды ұйымдастыру, кіріспе тесттерді өткізу үшін ақпараттарды дайындау және қабылдау қорытындысы бойынша қажетті есептерді қалыптастыру болып табылады. Жүйедегі бастапқы деректер абитуриенттердің, білім туралы құжаттардың, сертификаттардың және т.б. қосымшаларынан тұрады.

Автоматтандырылған ақпараттық жүйенің деректер моделі әзірленіп, міндетті түрде «Абитуриент» ұйымы болуы тиіс.



10.11 сурет. «Абитуриент» ақпараттық жүйесінің деректер базасын сипаттауға арналған «мән-сілтеме» диаграммасы (бастапқы нұсқасы)

Субъектінің атрибуттары: жынысы, туған күні, туған жері, ұлты, СТН, сақтандыру нөмірі, үй мекен-жайы (облыс / республика, қала, көше, үй), төлқұжат деректері, сақтандыру полисі деректері,



10.12 сурет. «Талапкер» деректер қорының ER-диаграммасының соңғы нұсқасы

жеңілдіктерге құқық беретін құжаттар, қосымша ақпарат (мүгедектер, жетім балалар, соғыс қатысушылары және т.б.), ата-аналар туралы мәліметтер (аты, жұмыс орны), сертификаттың орташа бағасы және т.б. Олардың ішінде мектеп, азаматтығы, білім беру түрі, мамандық, көше, қала және аймақ атрибуттары белгілі бір субъектілердің есімдері болып табылады. «Орналасқан жері» атрибуты толық мәні «көшесінің» сипатында анықталады және «Қолдану аясы» мәні атрибут «орны» сипатында анықталады.

10.11 суретте субъектілер арасындағы негізгі қатынастарды көрсетеді.

Келесі қадамда біз әрбір ұйымның атрибуттарын анықтаймыз және олардың түрлерін айқындаймыз (тиісті объектіде сипатталғандықтан, басқа субъектімен объектіні одан әрі анықтау үшін қолданылатын атрибуттар). Енді барлық осы диаграммаға қосуға болады (10.12 суретін қараңыз).

Доменді салыстыру кезінде ақпаратты мүмкіндігінше егжей-тегжейлі көрсетуге тырысыңыз, себебі ол болашақта дерекқор құрылымын жобалау кезінде көбірек шешім қабылдауға мүмкіндік береді. Мысалы, «Мекен-жайы», егер «Аты» ажырамас сипаттамасы, - ол ER-моделінің көрсетуі.

Тұжырымдамалық үлгісін құру «қолмен» орындады, және автоматтандырылған жобалау құралдарын пайдалана отырып, болуы мүмкін - CASE-құралдары болып табылады.

10.4. ДЕРЕКТЕР ҚОРЫНЫҢ ФИЗИКАЛЫҚ ҮЛГІСІ МЕН ЛОГИКАЛЫҚ ЖОБАЛАУЫ

Дерекқордың логикалық құрылымын жобалау кезінде түпнұсқа тұжырымдамалық үлгі нақты СУБД қолдайтын деректер үлгісіне айналады және көрсетілген доменнің алынған логикалық моделінің жеткіліктілігі тексеріледі.

Кез-келген тақырыптық аймақ үшін оны логикалық модельде көрсету үшін дизайн шешімдерінің көптеген нұсқалары бар. Дизайн әдіснамасы ең сәйкес дизайн шешімдерін таңдауды қамтамасыз етуі керек.

Логикалық үлгі құру кезеңі логикалық дизайн деп аталады. Конфигурациялық (инфологиялық) модельден логикалық (даталогиялық) күнге ауысқан кезде тұжырымдамалық модельде барлық ақпаратты қамтуы керек

деректер базасын жасау үшін қажетті тақырыптық аумақта жүзеге асырылады. Тұжырымдамалық үлгісін логикалық үлгіге айналдыру формальды ережелерге сәйкес жүзеге асырылады. Дерекқордың логикалық үлгісі деректер базасын жасақтайтын нақты СУБД-да жарамды ақпараттық бірліктер тұрғысынан жасалады.

СУБД тілінде дерекқордың логикалық құрылымының сипаттамасы *деректер схемасы* деп аталады.

Деректер базасының логикалық құрылымы, олардың арасында барлық ақпарат бірліктерін және сілтемелер анықтау болып табылады жобалау, олардың аттарын сұрайды; ақпараттық бірліктер түрін қолдана алады, егер, олардың түрін анықтау болып табылады. Сондай-ақ, кейбір сандық сипаттамаларды, мысалы, өрістің ұзындығын орнатады.

Кез келген деректерді басқару жүйесі оған қолайлы деректердің логикалық бірліктерімен жұмыс істейді. Сонымен қатар, көптеген деректер базасын деректер базасын құрылымына сандық және басқа да шектеулер жүктейді. Сондықтан, логикалық моделін құру өтпес бұрын, сіз, егжей-тегжейлі деректер базасын ерекшеліктерін зерттеу оны пайдалану мүмкіндігін және орындылығын қажет, қолданыстағы жобалау әдістерін таныс болуға, сондай-ақ қолданыстағы жобалау автоматтандыру құралдарын талдау болып табылады.

Логикалық жобалау деректер базасын логикалық құрылымын жобалау болып табылады да, ол СУБД ұсынған мәліметтердің физикалық ұйымдастыру мүмкіндігі әсер етеді. Сондықтан, деректерді физикалық ұйымдастыру сипаттамаларын білу логикалық құрылымын жобалау кезінде пайдалы болып табылады.

Дерекқор белгілі бір тақырыптық аймақты көрсетеді. Сондықтан, дерекқор жобалау процесі домен нысандарды алдын ала жіктеу, нысандарын және олардың арасындағы қарым-қатынастар туралы ақпаратты жүйелі ұсынуды көздейді. жобалық шешімдер туралы деректерді өңдеу қалаған сипаттамаларын әсер етеді. Сондықтан, тиісті ақпаратты дерекқорды жобалаудың бастапқы кезеңдерінде көрсету және талдау қажет.

Тұжырымдамалық үлгіде ақпараттық жүйедегі барлық ақпарат көрсетілуі керек. Бірақ бұл барлық деректер базасында сақталуы керек дегенді білдірмейді, ал тұжырымдамалық үлгіде бекітілген барлық субъектілер логикалық модельде айқын көрінуі керек. Логикалық модельді құрастырмас бұрын қандай ақпарат сақталатындығын деректер базасында анықтау керек.

Мысалы, тұжырымдамалық үлгіде есептеу индикаторлары көрсетілуі керек, бірақ дерекқорда сақталуы міндетті емес. Осыған байланысты, бірінші жобалық қадамдардың бірі дерекқордың құрамын, яғни, деректер базасында сақталуы керек көрсеткіштердің тізімін анықтау болып табылады.

Сонымен қатар, доменде бар сілтемелердің барлық түрлері нақты логикалық модельге тікелей салыстыру мүмкін емес. Осылайша, көптеген СУБД элементтер арасындағы элементтерден көптеген-көп қатынастарға тікелей қолдау көрсетпейді. Бұл жағдайда логикалық модельге қосымша көмекші элемент енгізіледі, бұл осы қатынастарды бейнелейді (осылайша, көп-көпшілік қатынасы осы жаңадан енгізілген элемент пен түпнұсқа элементтер арасындағы екіден көп қатынастарға бөлінеді). Бұған мысал келтірілген.

Кәсіптік қатынас диаграммасын салыстырмалы дерекқор схемасына түрлендірудің әдеттегі қадамдық процедурасын сипаттайық.

1. Әрбір мән кестеге айналады. Кәсіпорынның атауы кестенің аты болып табылады. Кәсіпорын түрі мысалдары тиісті кестенің қатарына сәйкес келеді.

2. Әр атрибут бір атаумен кестенің бағанына айналады; Деректерді ұсыну үшін неғұрлым дәл пішім таңдалуы мүмкін. Қосымша атрибуттарға сәйкес бағандар анықталмаған мәндер болуы мүмкін; міндетті атрибуттарға сәйкес келетін бағандар мүмкін емес.

3. Бірегей идентификатордың компоненттері кестенің негізгі кілті болады. Бірнеше мүмкін бірегей идентификаторлар болса, негізгі кілт үшін ең сипаттамалар таңдалады. Қосылымда бірегей идентификатор болса, сілтеменің басқа соңында орналасқан бірегей идентификатордың көшірмесі бастапқы кілттің бағандар санына қосылады. Осы бағандарды атау үшін, соңғы нүктенің атаулары және / немесе жұпталған нысан атаулары пайдаланылады.

4. Өзара қарым-қатынас сыртқы кілттерге айналады. Қосымша таңбалар анықталмаған мәндерге мүмкіндік беретін сыртқы кілт бағандарына сәйкес келеді; міндетті хабарламалар - белгісіз мәндерге жол бермейтін бағандар болып табылады. Егер А және В нысандарының арасында бір-бірімен қарым-қатынас болса, тиісті сыртқы кілт дизайнердің өтініші бойынша кестеде де, В-кестеде де жариялануы мүмкін.

5. А және В типті субъектілер арасындағы көпсалалы коммуникацияны қолдау үшін, екі қосымша міндетті С кестесі бар

олардың біреуі А субъектілерінің бірегей идентификаторларын, ал екіншісі В субъектісінің даналары үшін бірегей идентификаторларды таңдайды.

6. Индекстер бастапқы кілт үшін (бірегей индекс), шетелдік кілттерге және негізінен сұрауларды негіздейтін атрибуттарға арналған. Жоғарыда айтылғандай, индекстер мен басқа қосалқы деректер құрылымдарының анықтамасы деректердің логикалық, құрылымдық емес, физикалық сипатына жатады. Әрине, іс жүзінде бұл кезеңдер көбінесе уақыт өте жұмысқа келеді. Айта кету керек, барлық ықтимал және шетелдік кілттерге арналған SQL-бағдарланған СУБД индекстерінде әдетте жүйе автоматты түрде жасалады.

Логикалық үлгіні сақтау ортасына байлау үшін қысқа мерзімді физикалық модель деп аталатын физикалық қабаттың деректер моделі қолданылады. Бұл үлгі сақтау ортасында деректерді физикалық түрде қалай ұйымдастыру керектігін анықтайды. Физикалық қабатты модель СУБД ұсынған мүмкіндіктерді ескере отырып салынған. Дерекқордың физикалық құрылымының сипаттамасы *сақтау схемасы* деп аталады. Деректер базасын жасаудың тиісті кезеңі *физикалық дизайн* деп аталады. ДББЖ физикалық деректерді ұйымдастыру үшін әртүрлі мүмкіндіктері бар. Осыған байланысты физикалық дизайнның күрделілігі мен еңбегі және орындалатын қадамдардың жиынтығы нақты жүйелер үшін әртүрлі болады.

Физикалық деректер үлгісі белгілі бір ДҚБЖ көмегімен деректерді сипаттайды. Физикалық дизайн кезеңінде нақты деректер үлгісінің ерекшелігі және нақты ДҚБЖ ерекшелігі ескеріледі. Логикалық деректер үлгісін қалыптастыру сатысында қалыптасқан қатынастар кестелерге айналдырылады, атрибуттар кестелер бағанына айналады, негізгі атрибуттар үшін бірегей индекстер жасалады, домендер қолданылған дерекқорда қабылданған деректер түрлеріне айналады. Логикалық деректер үлгісіндегі шектеулер түрлі инструментальды құралдармен жүзеге асады, мысалы, индекстерді, тұтастығы шектеулерді, триггерлерді, сақталған процедураларды пайдалану. Сонымен қатар, логикалық модельдеу деңгейінде қабылданған шешімдер физикалық деректер үлгісі жасалатын белгілі бір шекараларды анықтайды. Сол сияқты, осы шекараларда әр түрлі шешімдер жасалуы мүмкін. Мысалы, логикалық деректер үлгісіндегі қатынастар кестелерге түрлендірілуі керек, бірақ әрбір кесте үшін деректерге қол жеткізу жылдамдығын арттыратын түрлі индекстерді қосымша жариялай алады. Көптеген жерде нақты СУБД-ға тәуелді болып келеді.

Дерекқордың физикалық дизайны сонымен қатар, өтінімнің барынша тиімді жұмыс істеуін қамтамасыз ету үшін сыртқы медиа бойынша деректер қорын тиімді бөлуді таңдауды қарастырады.

Физикалық үлгілеу кезеңінде жүйенің жұмыс істеуі үшін қажетті компьютерлік ресурстарға қойылатын талаптар, компьютердің түрі мен конфигурациясын таңдау, операциялық жүйенің түрі мен нұсқасы бағаланады. Таңдау индикаторларға байланысты:

- дерекқордағы деректердің шамамен алынған көлемі;
- деректер көлемінің өсу динамикасы;
- деректер сұрауларының сипаты (жеке жазбаларды алу және жаңарту, жазу топтарын өңдеу, жеке қатынастарды өңдеу немесе байланыстарды байланыстыру);
- сұраныстың түрі бойынша деректер сұрауларының қарқындылығы;
- жүйенің сұрау түрлеріне жауап беру уақытына қойылатын талаптар. СУБД мен бағдарламалық құралдарды таңдау дерекқордың жобасын әзірлеудегі ең маңызды сәттердің бірі болып табылады, себебі ол дерекқордың бүкіл процесіне және ақпараттық жүйені енгізуге түбегейлі әсер етеді. Теориялық тұрғыдан, бұл таңдауды жүзеге асыру кезінде ондаған факторларды ескеру керек. Бірақ іс жүзінде, әзірлеушілер тек өз түйсігі мен бірнеше маңызды критерийлерге ғана басшылық етеді, олар, атап айтқанда,:
- осы СУБД қолдайтын деректер үлгісінің типі, аталған бағдарламалық жасақтама құрылымына деректер үлгісінің барабарлығы;
- СУБД жұмыс сипаттамалары;
- ақпараттық жүйенің одан әрі дамуы үшін функционалдылық қоры;
- деректерді басқару персоналы үшін құралдармен жабдықталған СУБД дәрежесі;
- СУБД жұмысының ыңғайлылығы мен сенімділігі;
- СУБД құны және қосымша бағдарламалық қамтамасыз ету.

Жоғарыда айтылғандарды қорытындылай отырып, деректер базасын құрудың негізгі қадамдары мен міндеттері кесте түрінде ұсынылуы мүмкін (10.1-кесте). Әрбір саты белгілі бір жұмыс дәйектемесі, деректердің белгілі бір деңгейімен сипатталады.

Жобалау - деректер базасын құрудағы ең маңызды кезең, өйткені осы кезеңде маңызды стратегиялық шешімдер қабылданады, бұл тиімді ақпараттық жүйені құрудың бүкіл процесіне әсер етеді.

10.1 кесте. БД жобалау кезеңдері

Деңгей	Негізгі мақсаттар	Негізгі қадамдар	Негізгі ұғымдар
Концептуалды	Деректерге қойылатын талаптарды жинау, талдау және өндеу	Зерттеу тақырыбын зерттеу, оның ақпараттық құрылымын зерттеу. Доменнің барлық фрагменттерін анықтаңыз, олардың әрқайсысы реттелетін көрініспен, ақпараттық нысандармен және олардың арасындағы қарым-қатынастармен, ақпараттық объектілер бойынша процестермен сипатталады. Осы кезеңнің соңында дерекқор құрылымына инвариантты ER-диаграммасы тұжырымдамалық үлгісін аламыз	Мәні Атрибуттары Байланыс
Логикалық	Деректерге қойылатын талаптарды деректер құрылымына айналдыру	ER-диаграмма оларды қалыпқа келтірумен бірге көптеген кестелерге айналдырылады. Шығарылымда біз СУБД-ға бағдарланған дерекқор құрылымын және бағдарлама бағдарламасының ерекшеліктерін аламыз. Бұл кезеңде деректер қоры әртүрлі ДҚБЖ және үлгілерді салыстырмалы талдау үшін жиі үлгіленеді.	Кестелер Жазбалар Жазбалар арасындағы байланыс
Физикалық	Деректерді сақтау ерекшеліктерін, деректерге қол жеткізу әдістерін анықтау және т.б.	Таңдалған бағдарлама алаңына логикалық үлгіні бейімдеу. Индекстерді таңдау және құру. Жазу құралдарын ұйымдастыру және т.б.	Индекстер Деректер түрлері қол жетімділік әдістері

10.5 НОРМАЛИЗАЦИЯЛАУ ПРИНЦИПТЕРІНІҢ НЕГІЗІНДЕ

Реттелудің теориясы көмегімен салыстырмалы дерекқорларды жобалаудың классикалық технологиясы байланысты. **Нормаландыру** - деректерді дәйектілікпен және дұрыс өңдеуге мүмкіндік беретін пішінге деректерді қайтаруды және басқа да сәйкессіздіктерді жою арқылы деректерді қайта құру процесі.

Басқаша айтқанда, қалыпқа келтіру дерекқордың құрылымын минималды логикалық артықшылығын қамтамасыз ететін көзқарасқа әкеледі және жұмыс өнімділігін азайтуға немесе көбейтуге немесе дерекқордың нақты көлемін азайтуға немесе көбейтуге бағытталған емес. Нормаландырудың мақсаты оңтайлы дерекқор құрылымын алу болып табылады.

Нормализация үрдісінің жалпы мақсаты мынада:

- резервтеудің кейбір түрлерін жою;
- кейбір жаңартулардың ауытқуларын жою;
- нақты әлемнің жеткілікті «жоғары сапалы» көрінісі болып табылатын дерекқордың жобасын әзірлеу интуитивті болып табылады және келесі кеңейту үшін жақсы негіз бола алады;
- тұтастықтың қажетті шектеулерін қолдану рәсімін жеңілдету.

Реттелу әдісі реляциялық деректер үлгілері теориясына негізделген. Бастапқы нүкте доменді бір немесе бірнеше қатынастар түрінде ұсыну болып табылады және әр дизайнерлік қадамда «жақсартылған» қасиеттері бар кестелердің белгілі бір жиынтығы жасалады. Дизайн процесі - қарым-қатынас сұлбаларын қалыпқа келтіру процесі. Басқаша айтқанда, бұрынғы қалыпты пішіндегі қатынастың екі немесе одан да көп қатынастарға ыдырауы келесі қалыпты формалардың талаптарын қанағаттандырады.

Кестенің жіктелуі (бөлінуі) - деректердің артық болуын болдырмау үшін кестені бірнеше кестеге бөлу процесі болып табылады.

Деректердің артық болуын деректер базасында қамтылған деректердің қайталануы деп түсінуге болады. Бұл жағдайда қарапайым (артық емес) қайталану мен артық қайталануды ажыратамыз. Қажетсіз қайталану табиғи және рұқсат етілмеген болып бөлінеді. Қайталану

деректер дерекқор кестесінің бірнеше жазбаларында бірдей ақпарат қайталанатындығын көрсетеді, оны жою бір жерде дерекқордағы толық жоғалтуға әкелмейді.

Реляциялық дерекқорлар теориясында қалыпты пішіндердің келесі тізбегі:

- бірінші нормалық қалып (1нф, 1 NormalForm, 1NF);
- екінші нормалық қалып (2нф, 2NF);
- үшінші нормалық қалып (3нф, 3NF);
- Бойса-Кодттың нормалық қалыбы (BCNF);
- Төртінші нормалық қалып (4нф, 4NF);
- Бесінші нормалық қалып немесе, жобалау қосындысының нормалық қалыбы (5нф, 5NF, или PJ/NF).

Әрбір қалыпты пішінде шектеулердің белгілі бір жиынтығы бар, ал егер ол оған тән шектеулер жиынтығын қанағаттандыратын болса, қалыпты қалыпта болады. Әдетте практикада тек алғашқы үш қалыпты пішін табылды, олар төменде қарастырылады.

Кәдімгі нысандар әдісімен деректер базасын құру процесі кезең-кезеңмен жүреді. Дизайн осы нысандардың атрибуттарын анықтау, барлық нысандардың анықтау, деректер базасында қамтылған болуы тиіс, ол туралы ақпараттарда басталады. Барлық нысандардың атрибуттары бір үстелге көшіріледі, ол бастапқы болып табылады. Бұл кесте өзінің талаптарына сәйкес бірінші қалыпты пішінге дейін азайтылады. Кейінірек, үстел өз кезегінде, сондай-ақ басқа да кестелер айырбастауға болады екі немесе одан да көп үстелдер, ыдырайтын болып отыр. Осылайша, дәйекті түрде қалыпты формалардың талаптарына сәйкес келетін өзара байланысты кестелер жиынтығы жасалады. Іс жүзінде әдетте алғашқы үш қалыпты пішін қолданылады.

Мысалы, біз студенттер туралы мәліметтерді сақтауға арналған деректер базасын дайындаймыз: толық аты-жөні, туған жылы, топ, сынып мұғалімі, кодының нөмірі және мамандықтың атауы. Кесте қарапайым құрылымға ие, іс жүзінде кестеде әлдеқайда көбірек деректер болады. Бірақ деректер қалыпқа тәртібін көрсетуге жеткілікті болып табылады. Бұл жағдайда өрістердің түрі мен өлшемі үлкен рөл атқармайды, сондықтан біз тек өз аттарымен шектелеміз. Жасалған кесте бір-кесте дерекқор ретінде қаралатын болады.

Кесте өрістері бөлінбейтін (атом) ақпаратты қамтуы тиіс - 1 нф талаптар мынадай жағдайында кесте толтыру мақсатында жүзеге асырылады:

ТАӘ	Туған жылы	Мамандығы	Шифр	Топ	Сынып жетекшісі
Иванов Ф.И.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
Кириллова Е.Е.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
Потапов В.С.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
Дудко О.В.	1997	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
Таран О.С.	1998	Қысым металдарын өңдеу	150412	48о	Демина Е.Е.
Ильин Г.С.	1998	Компьютерлік желі	230111	44к	Павлова Н.И.
Федорова Д.С.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
Медведева Ж. А.	1997	Компьютерлік желі	230111	44к	Павлова Н.И.
Пушкина А. А.	1998	Компьютерлік желі	230111	44к	Павлова Н.И.

10.13 кесте. 1нф қалыбына сәйкес келетін кесте

Бастапқы кестені 1 нф түріне келтіреміз. Содан кейін «Студенттер» кестесінде 6 баған болады: толық аты, туған жылы, шифры, мамандығы, топ, сынып жетекшісі(10.13 сурет).

2 нф түрінде келесі талаптар қойылады:

- кесте 1 нф формасының талаптарына сай болуы керек;
- кез келген кілттік емес өріс негізгі өрістер арқылы бірегей түрде анықталуы керек.

Басқаша айтқанда, үстел нысаны егер 2 нф болса, жылы болып табылады және осы кесте 1 нф пішініне сәйкес келеді және әрбір кілт немесе атрибут негізгі кілтке толық тәуелді болса ғана жұмыс жасайды. Еске салайық, кілтсіз немесе кілт негізгі кілттің бөлігі болып табылмайтын қарым-қатынастың атрибуты деп аталады.

1 нф пішініне берілген кестенің жазбалары бірегей емес. 10.13 суретт 1 нф сәйкес нысанда, бірақ осыған байланысты бастапқы кілт болмауына пішіні 2 нф сәйкес келмейді.

Бір зерттеу тобында жеткілікті жоғары ықтималдығы біртекті болуы мүмкін, содан кейін кесте қайталанатын деректерді қамтитын болады. жазбалардың бірегейлігін қамтамасыз ету үшін кесте негізгі өрісіне енгізілген - студенттік билеттер санын. Бұл жағдайда, кілттің мәні кестеде әрбір жазбаны бірегей түрде анықтайды (10.14-сурет)

Осы кестенің жазбалары деректердің қайталанбайтын қайталануына ие, өйткені сынып жетекшісі әрбір топқа көрсетіледі және мамандықтың атауы әрбір мамандық шифры үшін жұмыс жасайды. Кестені бұзу арқылы қайталанудан құтылуға болады. Бұл жағдайда кестелер 3 нф пішініне сәйкес келеді. Форманың талаптары келесідей:

Ном.	ТАӘ	Туған жылы	Мамандығы	Шифр	Топ	Сынып жетекшісі
101	Иванов Ф.И.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
102	Кириллова Е.Е.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
103	Потапов В.С.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
104	Аудко О.В.	1997	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
105	Таран О.С.	1998	Қысым металдарын өңдеу	150412	48о	Демина Е.Е.
106	Ильин Г.С.	1998	Компьютерлік желі	230111	44к	Павлова Н.И.
107	Федорова ДС.	1998	Ақпараттық жүйелер	230401	35и	Попенко Б.С.
108	Медведева Ж. А.	1997	Компьютерлік желі	230111	44к	Павлова Н.И.
109	Пушкина А. А.	1998	Компьютерлік желі	230111	44к	Павлова Н.И.

10.14 кесте. 2нф қалыбына сәйкес келетін кесте

- кесте 2 нф пішінінің талаптарын қанағаттандыруы керек;
- кілтсіз өрістер бір-бірінен тәуелсіз.

10.14 кесте 3 нф пішініне сәйкес келмейді, өйткені «Сынып жетекшісі» өрісінің мәні санға байланысты және мамандықтың атауы шифрға және керісінше жұмыс жасайды. Кестені 3 нф пішіміне сәйкес келтіру үшін, оны бірнеше кестеге бөлінеді. Мамандықтар мен топтар туралы ақпарат бөлек кестелерде жасалуы тиіс, олардың әрқайсысында сіздің негізгі кілттіңізді анықтауыңыз керек.

3 нф пішіні азайғаннан кейін деректер базасында 10.15 суретте көрсетіледі.

«Студенттер» кестесінде енді мамандық пен топ бойынша толық ақпараттың орнына тек мамандық коды мен топтық код сақталады (10.16-сурет).

Қалыпты нысандардың талаптарына сәйкес әрдайым міндетті емес. Кестелердің саны көбейген сайын, дерекқордың құрылымы күрделі болады.



10.15 сурет. 3нф қалыбына келтірілгеннен кейінгі деректер қоры

Ном.	ТАӘ	Туған жылы	Мамандық коды	Топ коды
101	Иванов Ф.И.	1998	10	13
102	Кириллова Е.Е.	1998	10	13
103	Потапов В.С.	1998	10	13
104	Дудко О.В.	1997	10	13
105	Таран О.С.	1998	30	15
106	Ильин Г.С.	1998	20	14
107	Федорова Д.С.	1998	10	13
108	Медведева Ж.А.	1997	20	14
109	Пушкина А. А	1998	20	14

Код	Мамандық	Шифр	Топ коды	Топ	Сынып жетекшісі
10	Ақпараттық жүйелер	230401	13	35и	Попенко Б.С.
20	Металл қысымдарын өңдеу	150412	15	48о	Демина Е.Е.
30	Компьютерлік желі	230111	14	44к	Павлова Н.И.

10.16 сурет. 3нф қалыптағы кесте

деректерді өңдеу уақытын арттырады. Кейбір жағдайларда, құрылымды жеңілдету үшін деректердің ішінара қайталануына жол берілуге болады, ал олардың тұтастығын бұзбауға және келісімді сақтауға болады.

Нормаландыру идеялары дерекқорларды жобалау үшін өте пайдалы болғанына қарамастан, олар дерекқор жобасының сапасын жоғарылатудың әмбебап немесе толық құралы болмайды. Бұл деректер базасының құрылымында қалыпты жағдаймен жойылмайтын қателер мен кемшіліктердің әртүрлілігіне байланысты. Осы көзқарастарға қарамастан, қалыпқа келтіру теориясы реляциялық теория мен практиканың өте құнды жетістігі болып табылады, себебі ол ДБ жобасының сапасына ғылыми негізделген және жарамды критерийлерді және осы сапаны жақсартудың ресми әдістерін береді.

Нормалдау «жай мағынада» деген пікір бар, және кез келген құзыретті маман және өзі «қалыпты» қалыпқа келтірілген деректер базасын қалыпқа келтіру теориясын қолданудың қажетінсіз жобалайды. Дегенмен, қалыпқа келтіру - бұл кәсіби дизайнер өзінің мағынада, яғни қалыпқа келтіру қағидаттары - бұл ресми түрде қалыптасқан ақыл-ой болып табылады.

10.6. ДЕРЕКТЕР ҚОРЫНЫҢ КЕСТЕСІН ӨНДЕУДІҢ ЗАМАНАУИ ҚҰРАЛ ЖАБДЫҚТАРЫ

10.6.1. Негізгі анықтамалықтар

Деректер қоры, басқа ақпараттық жүйелер сияқты, жүйенің тұжырымдамасынан, жобалау алдындағы тексерулерден, жүйені жобалаудан, операциядан кейін, содан кейін - жүйені модернизациялаудан бастап, олардың өмірлік циклінің түрлі сатыларынан өтеді.

Үлкен жобаларды құру дизайнерлік автоматтандыру құралдарын пайдаланбай дерлік мүмкін емес.

1970 және 1980 жылдардағы ақпараттық жүйелерді жобалау мен дамытуды автоматтандыру болып табылады. Құрылымдық әдіснама кеңінен қолданылды - диаграммалар мен диаграммаларды қолданумен ақпараттық жүйелердің түрлі модельдерін сипаттау үшін графикалық құралдар пайдаланылды. Ақпараттық жүйелерді қолмен әзірлеумен осы графикалық модельдерді әзірлеу және қолдану өте қиын.

Бұл жағдайлар CASE-құралдар деп аталатын бағдарламалық құралдар мен технологиялық құралдардың пайда болу себептерінің бірі және ақпараттық жүйелерді құру және қолдау үшін CASE-технологиясын енгізу болды. Құрылымдық әдіснамаға қосымша, қазіргі заманғы CASE-құралдарының бірқатар объектілерге бағдарланған жобалау әдіснамасын пайдаланады.

CASE (ComputerAidedSoftwareEngineering) термині компьютер көмегімен бағдарламалық жасақтаманы әзірлеу ретінде ғана аударылады. Қазіргі уақытта бұл термин ақпараттық жүйенің дамуын автоматтандыруды білдіретін маңызды болды.

Қазіргі заманғы CASE құралдары көптеген қарапайым талдау және құжаттама құралдарынан толық көлемді автоматтандыру құралдарына дейінгі көптеген ақпараттық жүйелерді жобалау технологияларын қолдауға арналған кең ауқымды қамтиды. Оларды пайдалану жұмыстарды жылдамдатуға және оларды жүзеге асыру сапасын жақсартуға ғана емес, сондай-ақ дизайнерлер тобының ұжымдық жұмысын ұйымдастыруға мүмкіндік береді.

CASE-құралдары - ақпараттық жүйелерді құру және / немесе қызмет көрсетуді қолдайтын бағдарламалық құралдар, мысалы: талаптарды талдау және қалыптастыру, дерекқорды және қосымшаларды жобалау, кодты жасау,

тестілеу, сапаны қамтамасыз ету, конфигурация және жобаларды басқару болып табылады.

CASE-жүйесі нақты функционалдық мақсаты бар *CASE*-құралдарының жиынтығы ретінде анықталады және бір бағдарламалық жасақтама өнімінде орындалуы мүмкін.

CASE-технология - кешенді жүйелерді талдау, жобалау, әзірлеу және қызмет көрсету әдіснамаларының жиынтығы және бір-бірімен байланысты автоматтандыру құралдарының жиынтығы. *CASE*-технологиялар - бағдарламалық қамтамасыз ету және ақпараттық жүйелерді жобалау және әзірлеу процесін автоматтандыру, қағаз және қарындаш компьютерді ауыстырады, ол жүйе талдаушылар, әзірлеушілер және бағдарламашылар үшін құрал болып табылады. құрылымдық талдау әдістемесін, (және т.б. қиындық түсіну, пайдалану көп еңбек және құнын, жоба ерекшеліктер қолайсыздық өзгерістер) шектеулер бірқатар пайдаланған кезде. Басынан бастап, *CASE*-технологиялар бұл шектеулерді жеңу құралдарын талдау мен интеграциялауды автоматтандыру арқылы дамытты.

Дизайнды автоматтандыру құралдары пайдаланылатын тіл құралдарының жиынтығында және концептуалды модельді дерекқор үлгісіне түрлендіруге арналған алгоритмдерде ерекшеленеді. Бұл, өз кезегінде, қоршаған ортаға модель жасау әдісіне әсер етеді.

Қазіргі заманғы *CASE* құралдарының көпшілігі *ER*-модель формализміндегі деректерді сипаттауға арналған құралдарды қамтиды. *ER*-модель дерекқордың толық циклін немесе оның жеке кезеңдерін қолдайтын коммерциялық *CASE*-өнімдерінің маңызды санының негізі болып табылады. Алайда, олардың көпшілігі дамыған жүйесін домен тұжырымдамалық жобалау сатысында қолдау, сондай-ақ осындай кез келген нақты серверге немесе деректер базасына дерекқор схемасының ретінде таңдалған деректер базасын логикалық дерекқор жобалау кезеңінде, құралдарын салынған моделінің негізінде жүзеге асыруға мүмкіндік береді ғана емес. Бұл жағдайда субъектінің аймағын модельдеу графикалық диаграммаларды пайдалануға негізделген, ол құрамдастардың салыстырмалы түрде аз саны және ең бастысы осындай диаграммаларды құру технологиясы.

CASE-технологиясының және *CASE*-құралдардың пайда болуы алдында бағдарламалау әдіснамасы саласында зерттеулер жүргізілді. Бағдарламалау жоғары деңгейлі тілдерді, құрылымдық және модульдік бағдарламалау әдістерін, дизайн тілдері мен құралдарын әзірлеу және енгізу жүйелі тәсілдің ерекшеліктерін алды.

олардың қолдауы, ресми және бейресми тілдері, жүйелік талаптар мен сипаттамалардың сипаттамасы және т.б. Сонымен қатар, CASE-технологиясының пайда болуы келесі факторларға ықпал етті:

- модульдік және құрылымдық бағдарламалау саласындағы сарапшылар мен бағдарламашыларды оқыту;
- графикалық құралдарды тиімді пайдалануға және көптеген жобалық кезеңдерді автоматтандыруға мүмкіндік беретін компьютерлік өнімділікті кеңінен енгізу және үздіксіз өсу;
- жоба туралы қажетті ақпаратты қамтитын ортақ деректер базасын пайдалану арқылы жеке орындаушылардың күш-жігерін бірыңғай жобалау процесінде біріктіруге мүмкіндік беретін желілік технологияны енгізу. Құрылымдағы құралдарды пайдалану деректер базалары ақпараттық жүйелердің өмірлік циклінің әртүрлі сатыларына әсер етеді. Бұл белгілі бір дәрежеде зерттеу процесін алдын ала анықтайды және оның нәтижелерін көрсету құралын ұсынады.

CASE-индустриясы әртүрлі бағдарлардағы жүздеген компаниялар мен компанияларды біріктіреді. Шетелдегі барлық елеулі бағдарламалық жасақтамалардың барлығы дерлік CASE-құралдарымен жүзеге асырылады, ал таратылған пакеттердің жалпы саны 500 атаудан асады.

CASE-жүйелер мен құралдардың негізгі мақсаты бағдарламалық жасақтаманың дизайнын кодтаудан және одан кейінгі даму кезеңдерінен (тестілеуден, құжаттамадан және т.б.) бөліп, сонымен қатар бағдарламалық жасақтама жүйелерін немесе инженерлік техниканы (ағылшын инжинирингінен - жұмыс).

Бағдарламалардың дамуы жүйенің кейбір алдын ала нұсқасынан басталады. Осы прототип немесе ескірген үшін арнайы әзірленген әрекет болады және мұндай опция ретінде жүйесінің жаңа талаптарына сай емес ретінде. Соңғы жағдайда, оларды қайта пайдалану инженерлік жүйелерді одан әрі пайдалану мақсатында бағдарламалық қамтамасыз етудің білімін қалпына келтіру үшін пайдаланылады.

Абаттандыру, оның бағдарламалық кодын тексеру арқылы бағдарламалық жүйесінің бастапқы моделін құру үшін азайтады. Модельге ие бола отырып, оны жақсарту аласыз, содан кейін дамуға қайта ораласыз. Олар жиі жобалауға кіреді. RoundTripEngineering (RTE - осы түріне ең танымал қағидаттарының бірі «қайтару дизайн» принципі).

Қазіргі уақытта ең кең таралған болып табылады графикалық тілдерді пайдалануға мүмкіндік беретін жүйелер, объектіні көрсету үшін (тұжырымдамалық модель құрастыру), содан кейін автоматты түрде көшу тұжырымдамалық моделін деректер моделін ортасында таңдалған мақсатты DBMS. Осы типтегі CASE-жүйелерді қолдану тек ақпараттық жүйелердің дизайнерлерін ғана емес, сонымен қатар жүйенің клиенттерін де біріктіреді, сондықтан жүйенің тұжырымдамалық модельдеу сатысында пайдаланылатын белгілер, атап айтқанда, жеке тетіктерді барлық әзірлеушілер.

Тікелей жобалау (forward-энгинееринг) моделі негізінде салынған таңдалған мақсатты деректер базасын құрылымын алу процесі, және инженерлік кері - тікелей проекциясы (reverse-энгинееринг— реверс-инжиниринг) ажырататын моделі бар деректер негізінде алынған кезде. CASE-құралдары әдетте осы процестерді де қолдайды.

Графикалық модельдеу тілдерді пайдаланып салынған түрлі схемалар түрлі режимдерде пайдаланылуы мүмкін: кішірейту режимі, жобалау режимін және тіл режимін. Сонымен қатар деректерді модельдеу жүйелерін қолдануға да қолданылады. Сонымен қатар, CASE-құралдарының әдетте сіз әрбір режимі / жобалау кезеңдерінде үшін тиісті модельдеу механизмін таңдауға мүмкіндік береді, әр түрлі деңгейдегі ықтимал өкілдігі модельдері алдын ала көрінуі мүмкін.

Әдетте CASE-құралдарының бағдарламалық қамтамасыз өмірлік циклінің процестерін белгілі бір жиынтығын автоматтандыратын кез келген бағдарламалық құралын қамтиды. Төменде CASE-құралдардың негізгі ерекшеліктері бар.

Бірінші ерекшелігі - ақпаратты жүйелерді сипаттау және құжаттандыру үшін қуатты графикалық құралдар, әзірлеушімен ыңғайлы интерфейс және оның шығармашылық мүмкіндіктерін дамыту. CASE-технологиялар клиенттерге қоса алғанда, барлық жоба қатысушыларын, қарапайым және түсінікті құрылымымен жақын компоненттер алуға мүмкіндік береді, бір, қатаң анық және интуитивті графикалық тілін қамтамасыз етеді. Бұл бағдарламада, (көп сипаттамасы қарағанда пайдалану оңай) қарапайым схемаларын ұсынылған клиенттер даму процесіне қатысуға мүмкіндік береді, және әзірлеушілер - жүйелер талдаушылар, бағдарламашылар және дизайнерлер жұмысын бөлісуге, домен сарапшылардың байланысу үшін, басқару жобаны қорғауға көмектесе,

Тікелей дизайны бар (forward-engineering) - жүйеге өзгерістер енгізуді жеңілдету және алу, сондай-ақ алу процесі.

Келесі функция жобалық метадеректердің (репозиторий) арнайы ұйымдастырылған репозиторийін пайдалану мүмкіндігі. CASE-технологиясының негізі жоба туралы дерекқорды жобаға қатысты барлық ақпараттарды сақтау үшін пайдалану болып табылады, оны қолжетімділік құқықтарына сәйкес әзірлеушілер арасында бөлуге болады. Репозиторийдің мазмұны әртүрлі түрдегі ақпараттық нысандарды ғана емес, сонымен қатар олардың компоненттері арасындағы қатынастарды, сондай-ақ осы компоненттерді пайдалану немесе өңдеу ережелерін қамтиды. Репозиторий құрылымдық диаграммаларды, экран анықтамалары мен мәзірлерін, жобаның есептерін, деректер сипаттамаларын, өңдеу логикасын, деректер үлгілерін, бастапқы кодтарды, деректер элементтерін және т.б. сақтауы мүмкін.

Тағы бір айрықша ерекшелігі ақпараттық жүйелерді дамыту үдерісінің басқарылуын қамтамасыз ететін CASE-құралдарының жекелеген компоненттерін біріктіру болып табылады. Репозиторийдің негізінде CASE-құралдарының интеграциясы және жүйе ақпаратының әзірлеушілер арасында бөлінуі жүзеге асырылады. Сонымен қатар, репозиторийдің мүмкіндіктері интеграцияның бірнеше деңгейлерін қамтамасыз етеді: барлық құралдар үшін ортақ пайдаланушы интерфейсі, құралдар арасындағы деректерді беру, бір сатылы өмірлік циклді ұсыну жүйесі арқылы даму сатыларын біріктіру, әртүрлі платформалар арасында деректер мен құралдарды беру.

Ұжымдық даму мен жобаларды басқаруды қолдау CASE-құралдардың маңызды ерекшелігі болып табылады. CASE технологиясы жоба бойынша топтық жұмысты қолдайды, желіде жұмыс істей алады, оларды әзірлеу және / немесе модификациялау үшін жобаның кез келген фрагменттерін экспорттайды және импорттайды, сондай-ақ жоспарлау, мониторинг, басшылық және өзара әрекеттесу, яғни қажетті функциялар жобаларды дамыту және қолдау процесінде. Бұл функциялар репозиторийдің негізінде жүзеге асырылады. Атап айтқанда, репозиторий арқылы қауіпсіздік мониторингі (қол жеткізуді шектеу және артықшылықтар), нұсқа және өзгертуді бақылау және т.б.

Барлық жобалық құжаттама репозиторий негізінде автоматты түрде жасалады. CASE-технологиясының шексіз артықшылығы құжаттама әрдайым ағымдағы жағдайды көрсетеді, себебі жобаның кез-келген өзгерісі автоматты түрде репозиторийде көрсетіледі.

CASE-технологиясы ерте сатылардағы толықтығы мен дәйектілігі үшін автоматты түрде тексеру және жобаны бақылауды қамтамасыз етеді

тұтастай даму табысқа әсер ететін оқиғалар болып табылады. Барлық тәуелді жүйелер деректер базасын әзірлеу процесін құжаттандыру үшін құралдар әзірледі, автоматты есеп генераторлары толық дерекқор нысандарының сипаттамасы және графикалық нысанда және айтарлықтай жобаның басқаруды жеңілдетеді дайын стандартты баспа есептер түрінде, олардың қарым-қатынастары бар жоба деректер базасын ағымдағы жағдайы туралы баяндама дайындау мүмкіндік береді. ER-модель үшін бірыңғай жалпы қабылданған жүйе жоқ және әртүрлі CASE-жүйелері әр түрлі графикалық белгілерді қолданады, бірақ біреуін түсінгенде басқа белгілерді түсіну оңай.

10.6.2. CASE-технологиясының жіктелуі

Барлық заманауи CASE-құралдарын негізінен түрлері мен санаттары бойынша жіктеуге болады. Түрі бойынша жіктелу CASE-құралдарының функционалдық бағдарын көрсетеді. Санаттар бойынша жіктелуі орындалатын функцияларға қатысты интеграция дәрежесін анықтайды.

CASE-құралдарын пайдалану құрылатын жобалардың сапасын жақсарту алады, ал үлкен корпоративтік жүйелерді құру кезінде сөзсіз. Алайда, CASE құралдарын пайдалану дизайнерді тек жалпы сипатты ғана емес, сонымен қатар логикалық дизайн туралы мәліметтерден де босатпайды.

CASE-құралдарын жіктеген кезде келесі мүмкіндіктерді пайдаланыңыз:

- өмір циклінің кезеңдеріне бағдарлау;
- толықтық функционалдылығы;
- пайдаланатын үлгі түрі;
- СУБД тәуелсіздік деңгейі;
- Қолжетімді платформалар.

Ең жиі қолданылатын мүмкіндіктер үшін CASE-құралдарының жіктелуін қарастырыңыз.

Өмір циклінің кезеңдеріне бағдарлау бойынша CASE-құралдарының келесі негізгі түрлері бар:

- домендік үлгілерді құрастыруға және талдауға арналған талдау құралдары;
- жобаның ерекшеліктерін жасауды қамтамасыз ету үшін талдау және жобалау құралдары;
- деректер базасын басқару жүйесі үшін мәліметтерді модельдеу және дерекқор схемасын жасауды қамтамасыз ететін деректер базасын құрастыру құралдары;

- қолданбаны әзірлеу құралдары.

CASE-жүйесінің функционалдық толықтығы мен құралдарды шартты түрде келесі типтерге бөлуге болады:

- өмірлік циклдегі бір немесе бірнеше кезеңдерде нақты проблемаларды шешуге арналған жүйелер;
- ақпараттық жүйелердің бүкіл өмірлік циклын қолдайтын және жалпы репозиториймен байланысты интеграцияланған жүйелер.

CASE-жүйесі қолданылатын модельдер түріне қарай үш негізгі түрге бөлінеді: құрылымдық, объектілі-бағытталған және аралас.

Тарихи жолмен бірінші құрылымдық CASE-жүйелері құрылымдық-модульдік бағдарламалау, құрылымдық талдау және синтез әдістеріне негізделген.

90-жылдардың басынан бастап *нысаналы-бағытталған әдістер мен CASE-жүйелер* кеңінен қолданылады. Олар даму шарттарын қысқартуға, сондай-ақ ақпараттық жүйелердің жұмыс істеу сенімділігі мен тиімділігін арттыруға мүмкіндік береді.

Бірлескен құралдар құрылымдық және нысаналы-бағытталған әдістерді қолдайды.

ДББЖ-нен тәуелсіздік дәрежесі бойынша CASE-жүйесін келесі топтарға бөлуге болады:

- дербес жүйелер дербес дерекқор жүйесінің бөлігі болып табылмайтын дербес жүйелер түрінде жеткізіледі;
- СУБДда құрылған жүйелер, негізінен олар жататын ДББЖ базасының дерекқорын қолдайды (дерекқордың басқа форматтары қолдауға болады)).

Бүгінгі таңда ресейлік бағдарламалық жасақтама нарығында көптеген CASE-құралдары бар. Сонымен қатар, жүйе жүйенің ішкі тұтынушылары үшін жаңа, сондай-ақ жаңа нұсқалары мен модификацияларын үнемі жаңартып отырады. Одан әрі олар қысқаша қаралады.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Деректер қорын жобалаудың негізгі кезеңдерін көрсетіңіз.
2. Неліктен домен үлгісі салынған?
3. Концептуалды модель дегеніміз не?
4. Домен үлгісіне қойылатын талаптар қандай?
5. Концептуалды жобалау кезеңінде қандай негізгі ұғымдар пайдаланылады?

6. Концептуалдық жобалау кезеңінде қандай міндеттер шешілуде?
7. Дерекқордың тұжырымдамалық моделіне не жатады?
8. Тұжырымдамалық жобаның қадамдарын келтіріңіз.
9. ER-моделінің семантикалық негізі қандай?
10. Ұйым не ұйымның көшірмесі деп аталады?
11. Субъект атрибуты және атрибуттың данасы деп аталады?
12. Субъектілер арасындағы қарым-қатынас дегеніміз не?
13. Қайсысы біріктіріледі?
14. Атрибуттың егжей-тегжейлі дәрежесін қалай анықтау керек?
15. Байланыстың мағынасы неде және ER диаграммасында қалай көрсетіледі?
16. Кәсіпті толығымен анықтау деген нені білдіреді?
17. Қандай логикалық деректер моделі деп аталады?
18. Логикалық жобалау кезеңінде қандай негізгі ұғымдар пайдаланылады?
19. Логикалық жобалау кезеңінде қандай тапсырмалар орындалуда?
20. Логикалық дизайнның қадамдарын келтіріңіз.
21. Дерекқорды пайдалану үшін қажетті ресурстарды есептеуге қойылатын талаптарды бағалау қай кезеңде жүргізіледі?
22. Байланыстық қарым-қатынас диаграммасын салыстырмалы дерекқор схемасына түрлендірудің әдеттегі қадамдық процедурасын сипаттаңыз.
23. Физикалық деңгейдегі деректер моделі қандай?
24. Физикалық дизайн фазасында қандай негізгі ұғымдар пайдаланылады?
25. Дене жасақтамасының қадамдарын келтіріңіз.
26. Физикалық дизайн кезеңінде қандай тапсырмалар орындалуда?
27. Нормализацияның мақсаттары қандай?
28. Ақпараттың шамадан тыс қайталану қаупі қандай?
29. Қалыпты формалардың негізгі қасиеттері қандай?
30. $1nf$, $2nf$ және $3nf$ нысандарына қандай кесте шектеулері жатады?
31. Кәдімгі формалардың талаптарына сәйкес келмейтін кестелердің мысалдарын келтіріңіз.
32. CASE-құралдарына және CASE-технологиясына анықтама беріңіз.
33. Болашағы бар CASE-жүйесіне қойылатын талаптарды белгілеңіз.
34. CASE-құралдарын жіктеу белгілері қандай?

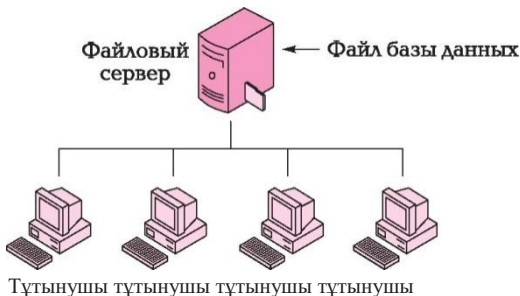
ДЕРЕКТЕР СӘУЛЕТІ

11.1 «ФАЙЛ-СЕРВЕР» СӘУЛЕТІ

Серверде сақталған «файл сервер» деректер қорын сәулет, клиент клиенттік компьютерде (11.1 сурет) сақталған барлық ақпараттық ресурстарды алу үшін файл командалық және бақылау механизмі серверіне қатынасады.

Файл-сервердің дерекқорына көптеген клиенттер арқылы желі арқылы кіруге болады. Дерекқордың өзі «файлдық сервер» желісінде бір көшірмеде сақталады. Әрбір клиент үшін, жұмыс кезінде жергілікті деректердің көшірмесі жасалады, оның көмегімен ол басқарады. Бірнеше пайдаланушыны сол кестеге немесе жазбаға бір мезгілде қол жеткізуге байланысты проблемалар бар. Бұл проблемалар (деректерді кірген сайын, олардың қол жетімділігін тексеріледі), өтініш салушылар дерекқор деректерді шешіледі.

Файл-сервердің архитектурасы маңызды кемшіліктерге ие. Олардың біреуі өнімді емес желі жүктемесі. Әрбір клиенттің сұрауы бойынша оның жергілікті көшірмесіндегі деректер сервердегі дерекқордан толығымен жаңартылады. Сұрау тек бір жазбаға қатысты болса да, дерекқордағы барлық жазбалар жаңартылады. Егер деректер базасында көп жазбалар, тіпті клиенттер желі шағын санына шындап сұрау өнімділігін әсер ететінін, өте мұқият жүктеледі. Желідегі үлкен көлемдегі артық ақпараттың айналымы нәтижесінде желінің жүктемесі күрт артып, бұл оның жылдамдығын және тұтастай алғанда ақпараттық жүйенің өнімділігін төмендетеді. Елеулі желілік трафик төмен жолақты сілтемелер астам «файл серверде» бойынша деректер базасына қашықтан қатынау ұйымдастыруға әсіресе,



11.1 сурет . «файл-сервер» сәулеті

Тағы бір кемшілігі - осы ұйымдағы деректердің тұтастығына аландаушылық клиенттік бағдарламаларға орналастырылады. Файлдық-серверлік жүйелер негізінде құрылған дәстүрлі құралдардың бірі жергілікті ДББЖ болып табылады. Мұндай жүйелер, әдетте, деректердің тұтастығы талаптарына сай келмейді, атап айтқанда, олар транзакцияларды өндеуді қолдамайды (құжаттармен аяқталған операциялар). Сондықтан, оларды пайдаланған кезде, деректердің тұтастығын қамтамасыз ету міндеті клиенттің қосымшаларына беріледі, бұл олардың асқинуына әкеледі. Егер олар мүжіят ойластырылмаса, қателерді барлық пайдаланушыларға әсер етуі мүмкін дерекқорға оңай енгізуге болады.

Сонымен қатар, файлдық сервердің архитектурасында дерекқорға бір пайдаланушыдан жасалған өзгертулер басқа пайдаланушыларға көрінбейді. Бір пайдаланушы жазбаны өндеп жатқанда, ол басқа клиенттер үшін блокталады. Жазбаларды блоктаумен байланысты жеке пайдаланушылардың жұмысын синхрондау қажет.

Ал басқа кемшіліктер - дерекқорды басқару түрлі компьютерлерден жүзеге асырылады, сондықтан қол жеткізуді бақылауды, құпиялылықты ұйымдастыру өте қиын, бұл деректер қорының тұтастығын сақтауды қиындатады.

Дегенмен, файлдық-серверлік сәулет қарапайымдылығы мен қол жетімділігі арқылы тартылады. Сондықтан, файлдық-серверлік ақпараттық жүйелер әлі де кішігірім жұмыс топтарына қызығушылық тудырады және, сонымен бірге, кәсіпорын ауқымында ақпараттық жүйелер ретінде жиі пайдаланылады.

Көбінесе, файлдық-серверлік дерекқорлар «жұмыс үстелі» деп аталады. Үстелдік ДББЖ салыстырмалы түрде кішігірім тапсырмалар үшін пайдаланылады (кішігірім

Көбінесе, файлдық-серверлік дерекқорлар «жұмыс үстелі» деп аталады. Үстелдік ДББЖ салыстырмалы түрде кішігірім тапсырмалар үшін пайдаланылады (кішігірім

өңделген деректердің саны, пайдаланушылар саны аз). Осыны ескере отырып, осы дерекқорлар салыстырмалы жеңілдетілген сәулет табылады, атап айтқанда, «файл сервер» режимінде жұмыс істейді, барлық ықтимал дерекқор функцияларды қолдамайды (мысалы, операция журналы болып табылмайтын, автоматты деректер базасын қалпына келтіру мүмкіндігі жоқ күйреудің кейін бар, және тағы басқалар). Дегенмен, осындай жүйелерде көптеген қосымшалар бар. Біріншіден, ол мемлекеттік (муниципалдық) мекемелері, білім беру, қызметтер көрсету, шағын және орта бизнес құрайды. одан туындайтын міндеттерді ерекшелігі деректер көлемі күрт жоғары емес болып табылады, жаңартулардың жиілігі тым жоғары емес, ұйым, әдетте, географиялық бір шағын ғимаратта орналасқан, пайдаланушылар саны бірінен 10 адамға дейін өзгеріп отырады. Мұндай жағдайларда, ақпараттық жүйелерді басқару жұмыс үстелі дерекқорын пайдалану толық ақталған болып табылады, және олар табысты қолданылды.

Бірінші деректер базасын бірі түрлі компаниялар әзірлеген деректер базасын үйлесімді бағдарламалық қамтамасыз ету жүйелері деп аталады. PLUS (компания Achten-Tate) - жүйенің бірінші таралған бұл түрі dBaseIII жүйесі болды. Жүйенің кең таралуы үшін әзірленген бағдарламалау тілі, қолданушыға қолайлы интерфейс, жаппай пайдаланушыға қол жетімді болды. Сонымен бірге, жүйенің түсіндіру режимінде жұмыс істеуі орындалу сатысында төмен өнімділікті тудырды. Clipper, FoxPro (Қатты FoxSoftware), FoxBase + (компания FoxSoftware), VisualFoxPro (Microsoft компаниясы) (фирма NantucketInc.): PLUS - Бұл dBaseIII жүйесіне жақын жаңа жүйелерін, құрастырушылар, пайда әкелді. Бұрын PARADOX дерекқоры (BorlandInternational) кеңінен пайдаланылды.

Соңғы жылдары MicrosoftOffice (Microsoft) бірнеше нұсқасына кіретін MicrosoftAccess дерекқорды басқару жүйесі өте кең таралған.

Ірі ұйымдар үшін жағдай түбегейлі өзгереді. Мұнда жоғарыда сипатталған себептерге байланысты файлдық-серверлік технологияларды қолдану жол берілмейді. Сондықтан қымбат серверлік ДББЖ пайдаланылады.

11.2. «КЛИЕНТ-СЕРВЕР» СӘУЛЕТІ

«Клиент-сервер» архитектурасы файлдық-серверлік қосымшалардың проблемаларын шешуге арналған,

оларды тиімді түрде жұмыс істейтін жерлерге орналастырыңыз. сәулет «клиент-сервер» ерекшеліктері Structured Query Language (StructuredQueryLanguage, SQL) үшін сұрау алған және іздеуге болады, сұрыптау және ақпаратты біріктірудің арнайы дерекқор серверлер болуы болып табылады.

Тұтастай алғанда, клиент пен сервер әр түрлі компьютерлерде жұмыс істейді деп болжанады. Сервер және клиенттік бағдарламалық қамтамасыз ету әртүрлі мақсаттарға ие.

Сервер - Бағдарлама ДҚБЖ функцияларды жүзеге асыру: деректер жазбаша анықтау, оқу, сыртқы қолдау тізбектерінің деректерді жою - тұжырымдамалық - ішкі деңгейлері және сұрау оңтайландыру, деректерді қорғау диспетчерлік.

Клиент - пайдаланушыға немесе жеткізуші деректер базасын, сыртқы немесе деректер базасына «кіріктірілген» жазылған бағдарлама. Клиент бағдарламасы деректерді сыртқы интерфейс арқылы ДББЖ компоненттеріне қол жеткізетін бағдарлама ретінде ұйымдастырылады.

«Клиент» және «серверлік» бөлігіне береді үшін сұрауға бөлу процесі:

- әртүрлі қосымшалар (клиент) бағдарламалары бір уақытта бірыңғай дерекқорды пайдаланады;
- ақпараттық қауіпсіздік, деректер тұтастығы, ресурстарды бөлісу басқару сияқты басқару функцияларын орталықтандыру;
- сұранысты параллель өндеуді қамтамасыз етіңіз;
- клиенттік компьютерлердің ресурстарын босату;
- мамандандырылған компьютерлер - дерекқор серверлерін пайдалану арқылы деректерді басқарудың тиімділігін арттыру.

«Клиент-сервер» платформасындағы дерекқор пайдаланушылардың көп саны бар жүйелерде (әдетте оннан астам) қолданылады. Дерекқордың контекстінде клиент пайдаланушы интерфейсін және бағдарлама логикасын басқарады, дерекқор бағдарламалары іске қосылған жұмыс станциясы ретінде әрекет етеді. Клиент бағдарламасы пайдаланушыдан сұрауды алады, синтаксистік тексереді және SQL тілінде немесе бағдарламаның логикасына сәйкес келетін басқа дерекқор тілінде дерекқорға сұрау жасайды. Содан кейін ол хабарды серверге жібереді, қабылданатын жауапты күтеді және қабылданған деректерді пайдаланушыға ұсыну үшін пішімдейді. Сұраулармен жұмыс істеуге бағытталған күшті сервер,

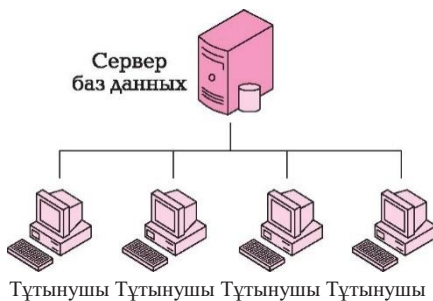
дереккорға сұрауларды қабылдайды және өңдейді, содан кейін нәтижелерді клиентке жібереді. Бұл өңдеу клиенттің құзыреттілігін тексеру, тұтастық талаптарын қамтамасыз ету, сұрауды орындау және деректерді жаңартуды қамтиды. Сонымен қатар, параллельді және қалпына келтіруді басқару қолдайды.

«Клиенттік-серверлік» жүйелерде қосымшаны жасай отырып, ол сервердің мүмкіндіктерін барынша кеңейтіп, желі бойынша жүктемені барынша азайта отырып, ол арқылы ең аз ақпаратты ғана жібереді.

Сипатталған сәулет екі деңгейлі архитектура болып табылады және ол «клиент» функционалды түрде жүктегендіктен «қалың клиент» деп аталады (11.2-сурет).

Қазіргі уақытта клиент-сервердің архитектурасы танымал және жұмыс топтары мен корпоративтік деңгейдегі ақпараттық жүйелерге арналған өтінімдерді ұйымдастыру әдісі ретінде кеңінен таратылды. Мұндай ұйым деректер базасының серверінің мүмкіндіктерін пайдалану арқылы қосымшалардың өнімділігін жақсартады, желідегі жүктемені төмендетеді және деректердің тұтастығын бақылауды қамтамасыз етеді. Ақпараттың қауіпсіздігін арттыру клиенттің барлық өтініштерін серверде орналасқан бірыңғай бағдарламамен өңдейтініне байланысты. Сервер дереккордың барлық пайдаланушылары үшін ортақ ережелерді орнатады, клиенттердің деректерге қол жетімділік режимдерін реттейді, атап айтқанда, әртүрлі пайдаланушылардың бір жазбасын өзгертуге тыйым салады.

Сондай-ақ, клиенттік қосымшалардың күрделілігі деректер базасының мониторингі мен оған қолжетімділікке қатысты кодтың болмауына байланысты азаяды.



11.2 сурет. «клиент-сервердің» екі деңгейлі сәулеті

Осыған сүйене отырып, сіз сервер мен клиент арасындағы функциялардың айырмашылығын келесідей сипаттай аласыз. Клиенттік қосымшаның функциялары:

- серверге сұраныстарды қалыптастыру және беру;
- серверден алынған сұраулардың нәтижелерін интерпретациялау;
- нәтижелерді пайдаланушыға қандай да бір пішінде ұсыну (пайдаланушы интерфейсі).

Серверлік бөлімнің функциялары:

- клиенттердің өтініштерінен өтініштер қабылдау;
- сұрауларды интерпретациялау;
- дерекқор сұрауларын оңтайландыру және орындау;
- нәтижелерді клиенттік қосымшаға жіберіңіз;
- қауіпсіздік жүйесі мен кіруді бақылау;
- деректер қорының тұтастығын басқару;
- көп функциялы жұмыс режимінің тұрақтылығын жүзеге асыру.

Сәулет «клиент-сервер» деп аталатын «өндірістік» деректер базасын іске қосады. Өнеркәсіптік олар, өйткені ол осы класс ақпараттық жүйелер ауқымды орта және ірі кәсіпорындардың, ұйымдардың, банктердің қамтамасыз ете алады СУБД деп аталады. MSSQLSERVER тиесілі өнеркәсіптік деректер базасын санаттағы үшін, Oracle, Informix, Sybase, DB2, InterBase басқалар жүзеге асады.

Әдетте, SQL-сервер қызметкерлердің жекелеген қызметкердің немесе топтың (әкімшілер SQL-сервер) арқылы беріледі. Олар деректер базасына физикалық сипаттамаларын басқару, оңтайландыру өндіруге, конфигурация және деректер базасын түрлі компоненттер отменить, жаңа деректер базасын құруға, қолданыстағы кәсіпорындарды өзгертуге, сондай-ақ әр түрлі пайдаланушыларға артықшылықтар (нақты деректер базасына қол жеткізу белгілі бір деңгейде рұқсаттар, SQL-сервер) беруге тиісті.

төмендегідей қысқаша сәулет «файл серверінің» салыстырғанда сәулет «клиент-сервер» негізгі артықшылықтары тұжырымдауға болады::

- қолданыстағы дерекқорларға көбірек қол жетімділік;
- жүйенің жалпы өнімділігі артады: клиенттер мен сервер әртүрлі компьютерлерде болғандықтан, олардың процессорлары параллельді қосымшаларды іске асыра алады. Сервермен компьютердің жұмысын теңшеу тек дерекқорда жұмыс істесе ғана жеңілдетіледі;
- аппараттық қамтамасыз ету құны төмендейді - үлкен сақтау құрылғысы бар жеткілікті қуатты компьютер қажет

тек сервер - дерекқорды сақтау және басқару үшін қуатты жұмыс станцияларына қажеттілік жоқ;

- желінің жүктелуі азаяды. Өтінімдер клиенттік компьютерлердегі операциялардың бір бөлігін орындайды және желі бойынша деректер қорына сұрау жібереді, бұл желі бойынша жіберілген деректердің көлемін едәуір азайтады;
- деректердің дәйектілік деңгейі артады. Сервер деректердің тұтастығын өз бетінше басқара алады, себебі оған тек барлық шектеулер анықталады және тексеріледі. Бұл жағдайда, әрбір өтінім өз тексерісін орындауға міндетті емес.

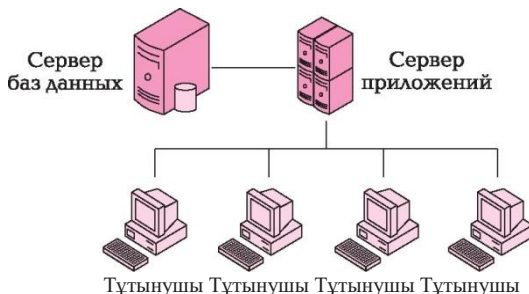
Кемшіліктері жоғары қаржы аппараттық және бағдарламалық қамтамасыз ету үшін шығындарды, сервер басқару қажеттігін, сондай-ақ әр түрлі жерлерде клиенттік компьютерлердің үлкен саны, барлық клиенттік компьютерлерде клиенттік қосымшаларды уақытылы жаңарту кейбір қиындықтар туғызады деп фактіні қамтиды Дегенмен, сәулет «клиент-сервер» функцияларын үлкен саны және сәулетке сәйкес құрастырылған дерекқор қазір іс жүзінде жақсы жұмыс істеді.

Дуплексті тұйықталу сәулеті «клиент-сервер» пайдаланушылар мен шатасқан логиканың әр түрлі күрделі ақпаратты қолдану кейбір проблемаларына әкелуі мүмкін. Бұл проблемаларды шешу көп деңгейлі сәулетті пайдалану болып табылады.

11.3. «КЛИЕНТ-СЕРВЕРДІҢ» ҮШ ДЕҢГЕЙЛІ СӘУЛЕТІ

Екі деңгейлі клиент-сервер сәулетін одан әрі кеңейту клиенттің функционалды бөлігін екі бөлікке бөледі - жұқа клиент пен қосымшалар сервері (11.3-сурет).

Жұқа клиент клиент-сервер немесе терминал сәулеті бар желілердегі компьютер немесе клиенттік бағдарлама, ол барлық өңдеу тапсырмаларын серверге тасымалдайды. «Жұқа» клиенттің мысалы веб-бағдарламалармен жұмыс істеу үшін пайдаланылатын браузері бар компьютер болуы мүмкін. Жұмыс станциясында «жұқа» клиент тек пайдаланушы интерфейсін басқарады, ал деректерді өңдеудің орташа деңгейі қалған логиканы бақылайды



11.3 сурет. «Клиент-сервердің» үшденгейлі сәулеті

Бұл архитектурада үшінші деңгей - дерекқор сервері.

Төменгі деңгейде пайдаланушылардың компьютерлерінде бағдарламаны орта деңгейде қоңырау шалу үшін бағдарламалау интерфейсі беретін функциялар мен логиканы қарауға арналған клиенттік қосымшалары бар.

Орташа деңгейде қосымшалар логикасын қолданатын қосымшалар сервері бар және оның деректерін өңдеу логикасы дерекқорда операцияларды орындайды, яғни бұл қабат пайдаланушылар мен таратылған дерекқорлар арасында деректермен алмасуды қамтамасыз етеді. Бағдарлама сервері барлық клиенттерге қол жетімді желі түйінде орналасқан.

Үшінші, жоғарғы, деңгей - қолданбалы серверден ақпарат алатын алыстағы мамандандырылған дерекқор сервері. Дерекқор сервері деректерді өңдеуге және файл операцияларына арналған.

Қысқаша айтқанда, төмендегідей үш деңгейлі клиент-сервердің архитектурасымен СУБД жұмысын сипаттай аласыз:

- файлдар жиынтығы түріндегі дерекқор арнайы компьютердің (желі сервері) қатты дискісінде орналасқан);
- СУБД желілік серверде де орналасқан;
- арнайы бағдарлама серверінде бағдарламалық қамтамасыз ету бар (бизнес логикасы);
- пайдаланушы интерфейсі іске асыратын клиенттік қосымша (жұқа клиент) біріктіріліп орнатылған. Қолданба қамтамасыз етуші дерекқордан пайдалану арқылы ол бағдарламаны дерекқорға қоңырау салады;

<p>Деректер қорының сервері</p> <p>Қосымша сервері</p>
--

- бағдарлама сервері пайдаланушы талаптарын талдайды және дерекқор сұрауларын жасайды. Байланыс үшін арнайы тіл SQL сұраулары пайдаланылады, яғни сұрау мәтіні ғана қосымшалар серверінен дерекқор серверіне;
- СУБД сервердегі деректер базасының физикалық құрылымы туралы барлық ақпаратты қамтиды;
- СУБД сервердегі деректерге қол жеткізуді бастайды, соның нәтижесінде сұраудың нәтижесі бағдарлама серверіне көшіріледі;
- бағдарлама сервері нәтижені клиенттік қосымшаға (пайдаланушыға) қайтарады;
- пайдаланушы интерфейсін пайдаланатын бағдарлама сұрауды орындаудың нәтижесін көрсетеді.
Үш деңгейлі сәулеттің артықшылықтары:
- дерекқор серверін бағдарлама серверіне жіберілген операциялардың бөлігінен босату;
- қажетсіз кодтан түсіру арқылы клиенттік қосымшалардың мөлшерін азайту;
- барлық клиенттердің бірыңғай мінез-құлқы;
- клиент конфигурациясын оңайлату - бағдарлама серверінің коды өзгерген кезде, клиенттік қосымшалардың әрекеті автоматты түрде өзгереді.

Үш деңгейлі сәулет екі деңгейлі клиент-сервер үлгісінің кемшіліктерін жояды. Бұл желідегі жүктемені теңестіруге мүмкіндік береді. «Клиент-сервер» жүйелерінің өсуімен үш деңгейлі ұйымға деген қажеттілік барған сайын айқын болады.

11.4. СЕРВЕРЛЕР КЛАСТЕРЛЕРІ

Ірі дерекқорлармен пайдаланушылардың айтарлықтай саны үшін үзіліссіз, сәтсіз, бәсекеге қабілетті операцияларды ұйымдастыру үшін кластерлік серверлер пайдаланылады.

Кластер - ортақ қосымшаларды орындау үшін бірлесе жұмыс істейтін бірнеше есептегіш жүйелердің жинағы және пайдаланушыға бірыңғай жүйе ретінде ұсынылады.

Клиент кластермен өзара әрекеттеседі, егер кластер бір сервер болса, физикалық бірнеше серверлердің тіркесімі болса да. Сервер тобы бір серверден гөрі сенімді және өнімдірек болады.

Кластер екі немесе одан да көп түйіндерден (серверлерден) тұрады, олардың әрқайсысы олардың кез келгенінде жұмыс істей алатын етіп конфигурацияланған. Бұл жағдайда бағдарламаның өзі виртуализацияланған, яғни кез келген түйіннен тәуелсіз болады. Міндетті шарт - сақтау жүйесінің барлық түйіндеріне ортақ болуы. Негізгі бағдарлама және оны пайдалану үшін қажетті барлық ресурстар, мысалы, файлдық ресурстар немесе желілік байланыс жалпы кластерлік топқа біріктіріледі. Кластерлік топтың ресурстарының біреуіне қол жетімсіз болған жағдайда, басқару қосымшасы басты бағдарламаның және барлық кластерлік топтың жұмысын басқа түйінге аударуды бастайды. Көбінесе, серверлер жергілікті желі арқылы топтастырылады.

Кластерлер бір компьютерге қарағанда есептеу уақытын қысқартуға мүмкіндік береді, бұл тапсырманы байланыстыру желісі бойынша деректермен алмасатын параллель жұмыс істейтін тармақтарға бөлу арқылы жүзеге асады. Қазіргі уақытта бөлінген есептеуіш жүйелер танымал болып келеді. Олар кластерлер ретінде қарастырылмайды, бірақ олардың қағидалары кластерлік технологияға негізінен ұқсас. Олар сондай-ақ тор жүйесі деп аталады. Бұл жүйелерде түйіндер жұмыс істеп тұрған кезде қосылады және ажыратылады, сондықтан міндет бөлек тәуелсіз процестерге бөлінуі керек, өйткені белгілі бір уақытта түйін жұмысына кепілдік беру мүмкін емес. Мұндай жүйе кластерлерге қарағанда, бір компьютерге ұқсамайды, бірақ есептеулерді жеңілдетілген құралы ретінде қызмет етеді. Бұл жағдайда конфигурация тұрақсыздығы көптеген түйіндермен өтеледі.

Жиі серверлер кластері программалық түрде ұйымдастырылады. Мұндай жағдайларда сервер кластері бірдей деректерге қызмет ететін процестер жиынтығын білдіретін логикалық тұжырымдама ретінде анықталады. Бұл жағдайда серверлер кластері логикалық байланысты қосылған, бірдей сұрауды өңдеуге және бір ресурс ретінде пайдаланылатын серверлер тобы ретінде анықталуы мүмкін. Бір ресурсқа серверлерді шоғырландыру бағдарламалық протокол деңгейінде орын алады.

Көптеген жағдайларда сервер кластерлері бөлек компьютерлерде жұмыс істейді. Бұл өнімділікті аппараттық ресурстарға жүктемені үлестіру және аппараттық деңгейде ақауларға төзімділікті қамтамасыз ету арқылы арттыруға мүмкіндік береді. Дегенмен, серверлер кластерін бағдарламалық жасақтама хаттамасының деңгейінде ұйымдастыру принципі бір компьютерде жұмыс істеуге мүмкіндік береді.

11.5 ДЕРЕКТЕР ҚОРЫНЫҢ НЫСАНДЫ СЕРВЕРІ

Дерекқор нысандары оның құрылымы мен деректері туралы барлық ақпаратты қамтиды. Дерекқор нысандары метадеректер деп аталады («деректер деректері»). Дерекқор әртүрлі нысандардан тұрады, мысалы, кестелер, домендер, сақталған процедуралар, триггерлер және т.б. Төменде серверлік дерекқор нысандарының анықтамалары берілген. Осы нысандардың кейбіреулері егжей-тегжейлі қаралатын болады.

Кесте (table) — реляциялық дерекқордың басты нысаны болып табылады. Кестелер деректер базасының барлық деректерін және метадеректерін сақтайды. Кесте - жолдардың ерікті санын қамтитын жазық екі өлшемді құрылым.

Домен (domain) - бағанның кейбір сипаттамаларын сипаттайтын дерекқор нысаны. Жасалатын кестенің бағандарын сипаттағанда немесе дерекқордағы бар кестенің бағандарының сипаттамаларын өзгерткен кезде доменге сілтеме жасай аласыз.

Индекс(index) - кестеден деректерді алуды жылдамдатуға, деректер кестесінен алынған нәтижелерді кестелерге (бастапқы, бірегей және шетелдік кілттер) байланыстыру үшін тапсырыс беруге арналған дерекқор нысаны. Әрбір индекс бір нақты кесте үшін жасалады. Индекс - реттелген жолдардың жиынтығы, олардың әрқайсысы индексті құрайтын өрістердің мәнін және осы өрістердің тиісті мәндерін қамтитын кесте жолына көрсеткішті қамтиды. Көрсеткіштерді активтендіруге / деактивтеуге, олардың сипаттамаларын жақсартуға арналған құралдар бар. Бастапқы, бірегей және сыртқы кілттер үшін индекстер автоматты түрде жасалады.

Генератор (generator) - бастапқы кілттің немесе кейде бірегей кілттің мәнін қалыптастыру үшін әдетте пайдаланылатын бірегей сандық мәнді алуға арналған дерекқордың дерекқоры.

Сақталған процедура (storedprocedure) - SQL тілінің процедуралық кеңейтілімінде жазылған бағдарлама (сонымен қатар сақталған процедуралар деп аталады және тілдерді іске қосады) және дерекқордағы деректермен әртүрлі әрекеттерді орындауға мүмкіндік беретін дерекқордың метадеректері аймағында сақталады. Сақталған процедураларға осы дерекқордың, пайдаланушы (клиент) бағдарламаларының сақталған процедуралары арқылы кіруге болады. Сақталған процедура сервер жағында орындалады, ол көптеген жағдайда желілік трафикті айтарлықтай төмендетеді және домендік проблемаларды шешу жылдамдығын арттырады.

Триггер (триггер) - сақталған рәсім сияқты, метадеректер аймағында сақталған және серверде орындалатын SQL тілінің процедуралық кеңейтілуімен жазылған бағдарлама. Алайда, триггерге тікелей қол жеткізу мүмкін емес. Ол операциялар және деректер базасында жұмысқа байланысты байланысты кестелер немесе жағдайда деректердің өзгеруіне байланысты оқиғаның фазалардың бірі басталған кезде автоматты түрде шақырылады. Іс-шаралар кестесі -, деректерді қосу кесте жолдарын өзгерту, жолдарды жою. Фазалар акцияның орындалуына дейін және «әрекеттен кейін» орындалады.

Пайдаланушының ерекше жағдайлары (ерекше жағдай) деректер базасымен жұмыс істеген кезде белгілі бір жағдай орын алғанда, пайдаланушыға хабарларды жасауға және шығаруға мүмкіндік беретін қатынас дерекқорының нысаны болып табылады. Бұл дерекқорда немесе дерекқорда деректерді өңдеудің кез-келген ерекше жағдайында қандай да бір бұзушылық болғанда орын алатын қате жағдай болуы мүмкін. Бұл ерекшеліктер тек сақталатын процедураларда және триггерлерде қолданыла алады.

Дерекқор оқиғалары (оқиғалар) сақталатын процедуралардан кейбір сақталған хабарларды және белгілі бір дерекқормен жұмыс істейтін барлық клиенттік бағдарламаларға триггерлер жіберуге мүмкіндік беретін нысандар болып табылады. Бұл құрал көптеген жағдайларда деректерді салыстыруды синхрондауды немесе клиенттік қосымшалардың өзара әрекеттесуі үшін күрделі әрекеттерді орындауға мүмкіндік береді, олар мезгілде дерекқормен бірге жұмыс істейді

Көрініс (көрініс, басқа атаулар - шолу, көрініс) - белгілі бір жиі күрделі крипторларға негізделген деректер базасының бір немесе бірнеше кестелерінен алынған деректердің нәтижесі. Өкілдік негізі кез-келген күрделіліктің SELECT нұсқауы болып табылады. Көрініс дерекқорда іс жүзінде сақталмаған виртуалды кестенің түрі болып табылады. Өкілдігі - пайдаланушыға осындай үлгідегі күрделі жағдайларды жасыру қажет болған кезде деректердің өте күрделі үлгісі болған жағдайда деректерді алудың ыңғайлы құралы. Сонымен қатар, пікірлер бұл мәндер осы пайдаланушының қарауына арналмаған пайдаланушыдан кестелердің кейбір бағандарын жасырудың пайдалы құралы болып табылады.

Пайдаланушы анықтайтын функциялар (UserDefinedFunctions, UDF) - кез келген бағдарламалау тілінде жазылған және дерекқордан тыс сақталған, бірақ осы дерекқорда сипатталған функциялар. SQL тілінің мүмкіндіктерін және тиісті бағдарламалау тілдерін кеңейту үшін қолдануға болады. Өте жиі

дерекқордан деректермен (немесе кіріс параметрлері ретінде тасымалданатын деректермен) өте күрделі әрекеттерді сипаттауға мүмкіндік береді).

БАҚЫЛАУ СҰРАҚТАРЫ

1. Қалай сәулет деректер базасы «файл сервер» жұмыс болып табылады?
2. Файл-сервердің архитектурасындағы деректермен жұмыс істеудің сенімділігі қалай қамтамасыз етіледі?
3. Файл сервер сәулетінің кемшіліктеріне не жатады?
4. «Клиент-сервер» сәулетінде ұйымдастырылған дерекқормен жұмыс қалай ұйымдастырылған?
5. «Клиент-сервер» архитектурасы файлдық-сервердің архитектурасымен салыстырғанда қандай артықшылықтарға ие? Қандай кемшіліктер бар?
6. Серверлер кластері дегеніміз не?
7. Бір компьютерде серверлер кластері ұйымдастырыла алады ма?
8. Сервер кластерін пайдаланудың артықшылықтары қандай?
9. Дерекқор нысандары көріністер деп аталады?
10. Сақталған рәсім дегеніміз не?
11. Триггер мен сақталатын рәсім арасындағы айырмашылық қандай?
12. Генератор не үшін пайдаланылады?
13. Қандай дерекқор нысандары пайдаланушылық ерекшеліктер деп аталады?
14. Нысандар дерекқормен жұмыс істеу кезінде оқиғалар деп аталады?
15. Пайдаланушы қандай функцияларды анықтайды?

SQL НЕГІЗДЕРІ

12.1. SQL ТІЛІНЕ ЕНГІЗУ

1970 жылдардың басында өткен ғасырда, арнайы тілдік SQL, содан кейін ол үшін зерттеулер-шумерлік зертханалық эксперименттік IBM корпорациясының бірінде реляциялық деректер базасын басқару жүйесі әзірленді (StructuredQueryLanguage - Structured Query Language). Стандартты 1986 жылы Американдық ұлттық стандарттар институты (ANSI) тілі SQL шығарды және 1987 жылы халықаралық ұйым (ISO) стандарттар халықаралық ретінде қабылданған.

Қазіргі уақытта, SQL тілі деректер базасымен жұмыс істеу үшін бірінші және әзірге жалғыз стандартты тілі болып табылады. SQL тілі көптеген есептеуіш платформалар үшін әзірленген көптеген ДББЖ түрлерімен қолдайды. Қазіргі уақытта барлық ірі ДҚБЖ әзірлеушілері SQL тілін пайдалануға бағытталған өнімдерін жасап жатыр.

SQL - ең алдымен, дерекқорда сақталған деректерді сипаттау, өзгерту және алу үшін жасалған ақпараттық және логикалық тіл. Бұл процедуралық емес тіл; Сонымен қатар, оны бағдарламалау тілі деп атауға болмайды. SQL тілі осылайша ұсыныстарын шағын жиынтығымен жинақы тілді жасау, логикалық байланысты кестелер түрінде ұсынылған деректермен операциялар бағытталған. SQL тілі құрылымының ең маңызды ерекшелігі деректер түпкілікті нәтижеге емес, осы емдеу процесіне назар болып табылады. SQL тілінің өзі деректерді, индекстерді қай жерде орналасқанын және тіпті нәтижелерді алу үшін операциялардың ең тиімді тізбектерін пайдалануды анықтайды, сондықтан осы мәліметтерді сұрауда дерекқорға көрсетудің қажеті жоқ. Сонымен қатар, тіл

SQL деректерді сұрау үшін және қолданбалы бағдарламалар жасау үшін пайдаланылуы мүмкін.

Функционалдылықты кеңейту үшін, көптеген әзірлеушілер қабылданған стандарттарға сәйкес стандартты SQL тіліне түрлі кеңейтімдерді қосады.

SQL тілін енгізу тиісті өндірушінің бағдарламалық жасақтамасы болып табылады.

Тілдің барлық нақты іске асырылуы бір-бірінен ерекшеленеді. Өндірушілердің өздерінің жүзеге асуы қазіргі ANSI стандарттарына портативті және ыңғайлылық тұрғысынан сәйкес келуін қамтамасыз ету. Дегенмен, әрбір SQL іске асыру нақты дерекқор серверінің талаптарына сай жақсартуларды қамтиды. SQL тіліне арналған бұл жақсартулар немесе кеңейтімдер стандартты пакетке толықтырулар болып табылатын және осы іске асыруда қол жетімді болатын қосымша пәрмендер мен опциялар.

SQL тілінде деректерді анықтау және өңдеуге арналған командалар ғана бар және есептеулердің бағытын басқару үшін ешқандай командалар болмайды. Мұндай міндеттерді бағдарламалау тілдері арқылы немесе интерактивті түрде - пайдаланушылардың өз әрекеттерімен шешу керек. SQL тілі екі жолмен қолданылуы мүмкін:

- 1) интерактивті жұмыс, ол пайдаланушыға бөлек SQL мәлімдемесін енгізеді;
- 2) процедуралық тілдерде бағдарламаларға SQL нұсқауларын енгізу.

SQL үйрену оңай. Себебі бұл процедуралық тіл емес, оған қандай ақпаратты алу керек екенін емес, оны қалай алуға болатынын көрсетіңіз. Басқаша айтқанда, SQL деректерге қол жеткізу әдістерінің сипаттамасын талап етпейді. SQL тілін кең ауқымды мамандар, соның ішінде деректер базасының әкімшілері, қолданбалы бағдарламашылар және бағдарламалау дағдылары жоқ басқа да көптеген пайдаланушылар пайдалана алады.

Орындалған іс-әрекеттер түріне қарай келесі әрекеттер бөлінеді:

- деректерді сәйкестендіру және деректер базасында өз орындарын табу;
- дерекқордан деректерді алу (оқу);
- дерекқорда деректерді қосу (жазу);
- дерекқордан деректерді жою;
- деректер базасының деректерін өзгерту (модификациялау).

SQL пәрмендерінің негізгі санаттары әртүрлі функцияларды орындауға арналған, соның ішінде дерекқор нысандарын құру және манипуляциялау, кестелерге деректерді бастапқы жүктеу, қолданыстағы ақпаратты жаңарту және жою, дерекқорға сұрауларды орындау, оған қол жеткізуді бақылау және оны жалпы басқару.

SQL тілінің негізгі пәрмендері:

- DDL — деректерді анықтау тілі;
- DML — деректерді манипуляциялау тілі;
- DQL — сұраныс тілі;
- DCL — деректерді басқару тілі;
- Деректерді әкімшілендіру пәрмендері;
- Транзакцияны басқару пәрмендері.

Деректерді анықтау тілі DDL (DataDefinitionLanguage) дерекқор нысандарының құрылымын жасауға және өзгертуге мүмкіндік береді, мысалы, кестелерді жасау және жою. DDL негізгі командалары төменде келтірілген:

- CREATE TABLE — кестені құру;
- ALTER TABLE — кестені өзгерту;
- DROP TABLE — кестені жою;
- CREATE INDEX — индексті құру;
- ALTER INDEX — индексті өзгерту;
- DROP INDEX — индексті жою.

Деректерді манипуляциялау тілі DML (DataManipulationLanguage) үш негізгі командалар арқылы салыстырмалы дерекқор нысандарындағы ақпаратты басқару үшін пайдаланылады:

- INSERT — жазбаны құру;
- UPDATE — жазбаны жаңалау;
- DELETE — жазбаны жою.

Кез-келген деректер үлгісі кейбір деректер базасын іске асыруда орындалуы мүмкін әрекеттер жиынтығын бір жағдайдан екіншісіне аудару үшін анықтайды.

Сұраныстар тілі DQL (DataQueryLanguage) дерекқордан жолдарды қайтаратын және бір немесе бірнеше кестелерден бір немесе бірнеше жолдарды немесе бағандарды таңдауға мүмкіндік беретін тек бір SELECTды қамтитын болса да, белгілі дерекқордың пайдаланушыларына белгілі. Деректер бойынша кез-келген операция деректерді таңдауды (select) қамтиды, яғни талап етілетін әрекеттер орындалатын мәліметтердің толық жиынтығынан және операцияның сипатын анықтайтын таңдалған деректер бойынша әрекетті қамтиды.

Таңдау шарты - деректер элементінің логикалық позициясы, оның мәні және деректер арасындағы қарым-қатынас қолданылуы мүмкін деректерді таңдаудың белгілі бір өлшемі.

Деректерді басқару тіліндегі DCL (DataControlLanguage) тілі дерекқордағы ақпаратқа қатынасты басқаруға мүмкіндік беретін деректерді басқару пәрмендерін қамтиды. Әдетте, олар деректерге қатынаумен байланысты объектілерді жасау үшін пайдаланылады, сондай-ақ пайдаланушылар арасында артықшылықтарды бөлуді басқару үшін қызмет етеді. Деректерді басқару пәрмендері келесідей:

- GRANT — қолжетімділік құқығын орнату;
- REVOKE — қол жетімділік құқығын жоққа шығару.

Бұл командалардың синтаксисі СУБД-ға байланысты. Қатынасты басқару процесін жеңілдету үшін, көптеген СУБД пайдаланушыларға топтарға топ немесе топ рөлдерін анықтау мүмкіндігін береді - пайдаланушыға берілген артықшылықтар жиынтығы.

Бұл тәсіл белгілі бір пайдаланушыға рөлді беруге немесе оған тағайындалған тапсырмаларға сәйкес құқықтардың жиынтығы бар белгілі бір пайдаланушылар тобына байланыстыруға мүмкіндік береді.

Деректерді басқару пәрмендерін пайдалану арқылы пайдаланушы дерекқормен әрекеттердің орындалуын бақылайды, дерекқор әрекеттерін талдайды, жүйенің өнімділігін талдайды және т.б. Деректер базасын басқару және дерекқорды басқару бірдей емес екенін атап өткен жөн. Дерекқорды басқару дерекқорды жалпы басқару болып табылады және барлық деңгейдегі командаларды пайдалануды қамтиды.

Транзакцияны басқару пәрмендері келесі пәрмендерді қамтиды:

- COMMIT — транзакцияны растау;
- ROLLBACK — транзакцияны жұмсау;
- SAVEPOINT — үзіліс нүктесін орнату (толық емес қайтару);
- SET TRANSACTION — транзакцияның басы.

SQL тілін сәтті меңгеру үшін SQL тілінің құрылымы мен түсініктемелерінің қысқаша сипаттамасын ұсыну қажет, олар әр түрлі тілдік құрылымдардың пішімін анықтау үшін қолданылады. SQL операторы сақталған сөздерден, сондай-ақ пайдаланушы анықтаған сөзден тұрады. Резервтелген сөздер SQL тілінің үнемі бөлігі болып табылады және тіркелген мәнге ие. Пайдаланушы анықтаған сөздерді пайдаланушы өзі белгілейді

(синтаксистік ережелерге сәйкес) және әр түрлі дерекқор нысандарының идентификаторларын немесе атауларын білдіреді. Өтініштегі сөздер де белгіленген синтаксистік ережелерге сәйкес орналастырылады.

SQL тіл идентификаторлары дерекқордағы нысандарды білдіреді және кестелер, көріністер, бағандар және басқа дерекқор нысандарының атаулары болып табылады. SQL стандарты әдепкі таңбалар жиынын анықтайды. Латын әліпбиінің (A-Z, a-z) жоғарғы және төменгі әріптері, сандар (0-9) және астын сызу (_). ID форматында келесі шектеулер белгіленеді:

- идентификатор ұзындығы 128 таңба болуы мүмкін;
- идентификатор хатпен басталуы тиіс;;
- ID идентификаторы бос орындардан тұруы мүмкін емес.

Көптеген тіл компоненттері регистрді ескермейді.

SQL тілінің сипаттамасы берілген тіл *метатіл* деп аталады. Синтаксистік анықтамалар, әдетте, Бэкуса-Наур формулалары (BNF) деп аталатын арнайы металингвистикалық рәміздердің көмегімен беріледі. Сақталған сөздерді жазу үшін бас әріптер қолданылады. Төменгі әріптер пайдаланушылық анықтайтын сөздерді жазу үшін пайдаланылады. BNF белгілерінде пайдаланылатын таңбалар мен олардың белгілері 12.1 кестеде көрсетілген.

Бұрын біз деректерді дерекқорда бірнеше түрлі түрлердің бірінде жинақталған жиынтық ақпарат ретінде анықтадық. Деректер түрлерін қолдана отырып, кестенің нақты бағанындағы деректерге, оның ішінде бөлінген жады көлеміне қатысты негізгі ережелер белгіленеді.

SQL тілінде стандарт бойынша анықталған алты скалярлы деректер түрлері бар (12.2-кесте).

12.1 кесте. БНФ металингвистикалық белгілері

	Анықтама бойынша тең
I	Бірнеше төмендетілген мәндердің біреуін таңдау қажеттілігі
<...>	Тілдің құрылымы метал тілінің көмегімен сипатталады
{...}	Тізімнен кейбір конструкцияларды міндетті іріктеу
[...]	Тізімнен кейбір конструкцияларды қосымша таңдау
[..n]	Құрылыстың қосымша қайталануы нөлден бірнеше есеге дейін

Деректер түрі	хабарландыру
Белгілік	CHAR VARCHAR
Биттік	BIT BIT VARYING
Нақты сандар	NUMERIC DECIMAL INTEGER SMALLINT
Шамамен алынған сандар	FLOAT REAL DOUBLE PRECISION
Күні/уақыты	DATE TIME TIMESTAMP
Интервал	INTERVAL

Таңбалар деректері СУБД жасаушыларымен анықталған таңбалар жинағына кіретін таңбалар тізбегінен тұрады. Таңбалар жиынтығы SQL тілінің әртүрлі диалектілеріне тән болатындықтан, таңба түрінің деректер мәндерінің бір бөлігі болуы мүмкін таңбалар нақты іске асыруға байланысты болады.

Таңба деректері түрімен бағанды анықтағанда, ұзындық параметрі берілген бағанда орналастырылуы мүмкін таңбалардың максималды санын көрсету үшін пайдаланылады. Таңба жолы тіркелген (CHAR) немесе айнымалы (VARCHAR) ұзындығы ретінде анықталуы мүмкін. Егер жол белгілі бір ұзындық мәндерімен анықталса, онда оған аздаған таңбаларды енгізген кезде, мән көрсетілген оң жаққа оң жаққа қосылатын кеңістіктермен толтырылады. Егер жол айнымалы мәндер мәнімен анықталса, онда дерекқорда азырақ таңбаларды енгізу арқылы сыртқы жадтың белгілі бір экономикасына қол жеткізуге мүмкіндік беретін енгізілген таңбалар ғана сақталады.

Бит жолдарын анықтау үшін бит деректер түрі пайдаланылады, яғни әрқайсысы 0 немесе 1 мәніне ие екілік сандардың (биттердің) кезектілігі болып табылады.

Нақты сандық деректер түрі нақты көрінісі бар сандарды анықтау үшін пайдаланылады, яғни сандар сандардан, қосымша ондық нүктеден және қосымша таңбалардан тұрады. Нақты сандық деректердің деректері бөлшек бөліктің дәлдігі мен ұзындығы бойынша анықталады. Дәлдік санның цифрлық ондық санының жалпы санын көрсетеді, ол бүтін бөлік ретінде де, бөлшек, бірақ ондық нүктенің өзін ескермей тұрады. Масштабта санның бөлшек ондық орындарының санын анықтайды.

NUMERIC және DECIMAL түрлері сандарды ондық санмен сақтауға арналған. Әдепкі бойынша бөлшек ұзындығы нөлге тең және әдепкі дәлдік орындалуына байланысты. Түр INTEGER (INT) үлкен оң немесе теріс бүтін сандарды сақтау үшін пайдаланылады. SMALLINT - кішкентай оң немесе теріс бүтін сандарды сақтау үшін; Бұл жағдайда сыртқы жақты тұтыну айтарлықтай азаяды.

Доңғалақты сандардың түрі компьютерде нақты көрсетіле алмайтын деректерді, атап айтқанда нақты нөмірлерді сипаттау үшін қолданылады. Дөңгелектелген сандар немесе өзгермелі нүктелер сандары ғылыми белгілерде ұсынылады, онда сан он (белгілі бір) күшпен көбейтілген мантисса арқылы жазылады.

«Күн / уақыт» деректер түрі белгілі бір дәлдікпен уақытша нүктелерді анықтау үшін пайдаланылады. SQL стандарты келесі пішімді қолдайды.

DATE деректер түрі YEAR (жыл), MONTH (ай) және DAY (күн) өрістерін қамтитын күнтізбелік күндерді сақтау үшін пайдаланылады. Деректер түрі TIME - HOUR (сағат), MINUTE (минут) және SECOND (секунд) жолдарын қамтитын уақытша мөртаңбаларды сақтауға арналған. Уақытты және уақытты бірлескен сақтау үшін TIMESTAMP деректер түрі.

INTERVAL түріндегі деректер уақыт кезеңдерін көрсету үшін пайдаланылады.

12.2. КЕСТЕЛЕРМЕН ЖҰМЫС. БҮТІНДІКТІ ШЕКТЕУ

12.2.1. Домендермен жұмыс

Домен - кестелер бағандарының сипаттамаларын сипаттауға мүмкіндік беретін реляциялық дерекқор нысаны.

Бағандарды сипаттағанда кез келген кестені жасағанда немесе өзгерткенде, барлық сипаттамаларын кестедегі бағанға көшіру үшін қолданыстағы доменге сілтеме жасай аласыз. Кейде дерекқорлардың бірнеше кестелерінде бірдей сипаттамалары бар бағандар бар. Бұл жағдайда ол доменді пайдалану арқылы алдын ала бағанның және оның мінез-құлық түрін сипаттау, содан кейін сол домен атауын әр бағанға хат-қоюға болады. Бағанды көшіру кезінде доменде сипатталған жеке сипаттамалар өзгертілуі және толықтырылуы мүмкін. Домендер

Ол сақталатын процедуралар мен триггерлер жергілікті айнымалы және параметрлерін сипаттамасында, сондай-ақ пайдалануға болады.

Доменнің негізгі сипаттамасы деректер түрі болып табылады. ішкі айнымалылар сақталатын процедуралар мен триггерлер сипаттамасында бағанының сипаттамаларын құру немесе өзгерту кезінде доменді жасау немесе өзгерту кезінде деректер түрлері анықталады. Деректер түрлері топтары немесе санаттарға аралас болуы мүмкін - сандық деректерді, жолды, күнді және уақыт деректерді, сондай-ақ екілік BLOB деректер түрі тобында ғана бір.

Домен келесі форматтағы CREATEDOMAIN операторымен анықталады:

```
CREATE DOMAIN <домен атауы>[AS] <деректер түрі>
[DEFAULT<мәндері>]
[NOTNULL] [CHECK(<шектеу шарттары>)]
[COLLATE<сорттау тәртібі>]
```

DEFAULT сөйлемі кесте жазбасын жасаған кезде доменмен байланыстырылған бағанға әдепкі күйге келтірілген өрнекті анықтайды. Бұл мән пайдаланушының қандай да бір жолмен өзгертпейінше жазбаның тиісті бағанында болады. Әдепкі мәндер литеральді мән (сан, жол немесе күн) ретінде көрсетілуі мүмкін.

CHECK сөйлемі доменмен байланыстырылған әрбір баған мәніне қойылатын талаптарды анықтайды. Бағандарға CHECK параметрі енгізген шектеулерге сәйкес келмейтін мәндер тағайындалуы мүмкін емес. Доменмен байланыстырылған өрістердің мәндеріне шектеудің форматы:

```
<доменді шектеу шарттаы>::= {
VALUE<оператор><мәні>
|VALUE [NOT] BETWEEN<мәні 1>AND<мәні 2>
| VALUE [NOT] LIKE <мәні>[ESCAPE <мәні>]
|VALUE [NOT] IN(<мәні1> [, <мәні2>...])
| VALUE IS [NOT] NULL |VALUE [NOT] CONTAINING <мәні>
|VALUE [NOT] STARTING [WITH] <мәні>
<доменді шектеу>
<доменді шектеу>OR< доменді шектеу >< доменді шектеу >AND<
доменді шектеу >
}
мұндағы
<оператор> = { = | < | > | <= | >= | !< | !> | <> | != } ;
```

VALUE <оператор> <мәні> - домен мәні <оператор> параметрімен анықталған қатынастардағы <мәні> параметрінде орналасқан. Басқаша айтқанда, VALUE доменмен байланысты бағанға тағайындалуы мүмкін барлық дұрыс мәндерді білдіреді;

Домен мәні <мәні1> және <2-мән> арасында болуы тиіс, соның ішінде - және <2-мән> <мәні1> АРАСЫНДАҒЫ

LIKE <1-мән> [ESCAPE <мәні2>] - үлгі ұқсастығын белгілейді, символы «%» кез келген ұзындығы кез келген мәнін көрсетеді, және символы «_» - кез келген жалғыз таңба. ESCAPE LIKE операторында қалыпты таңбалар ретінде «%» және «_» таңбалары болса, пайдаланылады. Бұл жағдайда <мән 2> таңбасын таңдаңыз, содан кейін қызметтік нышандар ерекше күйін жоғалтады және әдеттегідей іздеу жолын енгізеді. Мысалы, CHECK (VALUELIKE «%!%» ESCAPE «!») - кез келген «!» Таңбалары кейіптеледі «%»;

IN (<1 мән> [<мән 2> ...] - домендік мәні аталған параметрлердің бірімен сәйкес келуі тиіс;

IS [NOT] NULL бағанында бос емес басқа мән болуы керек;

<Мәні> құрамы - домен мәні <мәні> кіруін қамтуы тиіс, қандай орын жоқ мәселені;

Домен мәні параметр «мәнін» бастау керек - <мәні> [WITH] БАСТАУ.

Доменнің мәні сәйкес келетін шарттардың комбинациясы мүмкін. Бұл жағдайда жеке шарттарға AND немесе OR операторы қосылған.

Мысалы, осы доменге жазылған деректер «otd» таңбаларынан басталатын және «018» жолын қамтитын шартпен таңба түрінің DEPARTMENT жасаңыз:

ДОМЕНДІҢ БӨЛІМДЕРІН ОРНАТУ VARCHAR (10)

CHECK («Кімнен» және «018» мәні бар)

Домен анықтамасын өзгерту үшін операторды пайдаланыңыз:

ALTERDOMAIN<имя>

{ [SETDEFAULT{литерал | NULL | USER }]

```
| [DROP DEFAULT]
| [ADD [CONSTRAINT] CHECK (<доменді шектеу>)]
|[DROPCONSTRAINT]}
```

Оператор бұрын CREATEDOMAIN операторы арқылы анықталған домен параметрлерін өзгертуге мүмкіндік береді. Дегенмен, деректер түрін және NOTNULL анықтамасын өзгертуге болмайды, егер бағандар орнатылған деректер түріне сәйкес келмейтін мәндер болса немесе бос болса. Жасалған барлық өзгерістер осы доменді пайдалану арқылы анықталған барлық бағандар үшін ескеріледі. Бұл операторда:

- [SETDEFAULT] — ол операторда қалай жасалатынына ұқсас әдепкі мәндерді орнатады CREATEDOMAIN;
- [DROPDEFAULT] — әдепкі бойынша берілген ағымдағы әдепкі мәндерді тоқтату;
- [ADD [CONSTRAINT] CHECK(<доменді шектеу>)] — Доменмен байланыстырылған баған мәндеріне сәйкес келуі керек шарттар қосу. CHECK операторының ұсынысы бойынша жоғарыда қаралған шарттарды анықтауға болады CREATEDOMAIN;
- [DROPCONSTRAINT] — Домен үшін анықталған шарттарды CREATEDOMAIN үзіндісіндегі немесе бұрынғы ALTERDOMAIN мәлімдемелеріндегі CHECK параметрінде жойыңыз.

Домендерді пайдаланудың мысалдары кестелерді жасау операторларын зерттегеннен кейін қаралатын болады.

12.2.2. Кестелерді басқару

Бірыңғай дерекқор құрылымын жасағаннан кейін, дерекқор жобасын жасайтын қатынастарды көрсететін кестелерді жасай аласыз. Естеріңізге сала кетейік, кесте ақпаратты сақтау үшін негізгі объект болып табылады деректер базасында. Деректерді қамтитын жолдар мен бағандардан тұрады, дерекқордағы физикалық кеңістікті алады және тұрақты немесе уақытша болуы мүмкін. Реляциялық дерекқордағы баған деп аталатын өріс белгілі бір деректер түрін тағайындаған кестенің бөлігі болып табылады. Әр дерекқор кестесінде кемінде бір баған болуы керек. Деректер жолы - дерекқор кестесіндегі жазба, оған бір кесте жазбасынан деректерді қамтитын өрістер кіреді.

Дерекқор кестелерін жасамас бұрын, кестенің барлық бағандары мен әрбір бағанның сипаттамаларын анықтау арқылы ойлануыңыз керек

индекстер, басқа кестелерге қатысты тұтастық шектеулер. Алдын ала домендер кестелерде пайдаланылған жағдайда анықталуы керек. Жасалған кесте қосылатын дерекқорды ашу керек, яғни белсенді қосылым орнатылуы керек.

Дерекқор кестесін жасау CREATETABLE операторы арқылы орындалады. Кесте жасау операторының негізгі синтаксисі келесідей:

```
CREATETABLE<кесте атауы>  
({<баған анықтамасы>|<кесте шектеулерін анықтау>}  
[,...,{<баған анықтамасы > |< кесте шектеулерін анықтау >}])
```

Кестенің атауын үтірмен көрсетіп болған соң, жақшалардағы барлық сөйлемдер жеке кесте элементтерін немесе бағыныңқы мәндерді анықтайтын жақшаларда көрсетілуі керек:

- <кесте атауы> — құрылып жатқан кесте идентификаторы;
- <баған анықтамасы> — Кестедегі бір бағанның атауын, деректер түрін және параметрін көрсету. Баған атаулары идентификаторларға қатысты ережелерге сәйкес келуі және кестеде бірегей болуы керек;
- <кесте шектеулерінің анықтамасы> — кесте деңгейінде белгілі бір тұтастығы шектеулерін орнату.

<Кесте анықтамасы> тармағын пайдалану арқылы бағанның сипаттарын көрсетіңіз:

```
<баған анықтамасы> :: =  
<баған атауы>< деректер түрі>  
[<бағанды шектеу>][,...,<бағанды шектеу>]
```

Параметрлердің мақсаты мен қолдануын қарастырамыз:

- <баған атауы> - кесте бағанын атауын көрсететін идентификатор;
- <деректер түрі> - бағанның деректер түрі;
- <баған шектеулері> - Бұл тармақ бағанда анықталатын шектеулерді анықтайды. Сөйлемнің синтаксисі келесідей:

```
<баған шектеулері>;=  
[CONSTRAINT<сілтемелік бүтіндік атауы>]  
{[DEFAULT <мағынасы>]  
|[NULL|NOT NULL]}
```

```

[[PRIMARY KEY|UNIQUE]
|[FOREIGN KEY REFERENCES <басты кесте атауы>
|(<баған атауы> [,...,n])]
|[ON DELETE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}]
|[ON UPDATE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}]
]
|[(CHECK<баған шарттары>)]
}

```

Параметрлер мәндерін қарастырамыз.

CONSTRAINT— қосымша кілт сөзді, одан кейін баған мәндеріндегі шектеу атауын (<сілтеме тұтастығының атауы>). Аттар дерекқор ішінде бірегей болуы керек. Сілтеме тұтастығының атауы міндетті емес. Ол тұтастықты бұзу туралы жүйелік хабарламаларда бар және кестелер құрылымын өзгерту кезінде пайдаланылуы мүмкін. Егер бұл атау болмаса, жүйе атауы орнатылады. Атаусыз бүтіндігін жою үшін оның жүйелік атауын пайдалануыңыз керек.

DEFAULT— бағана үшін әдепкі мәнді анықтайды. Бұл мән бағанда ешқандай мән көрсетілмесе, жолды кірістірген кезде пайдаланылады.

NULL|NOTNULL— **NULL** мәндер бағанында сақтауға рұқсат беретін немесе қабылдамайтын негізгі сөздер. **NULL** кілт сөзі бұл бағанда **NULL** мәндері болуы мүмкін екенін көрсету үшін пайдаланылады. **NULL** мәні бос немесе нөлден ерекшеленеді - деректердің қолжетімсіз, қабылданбаған немесе жарамсыз екенін көрсету қажет болғанда қолданылады. Егер **NOTNULL** кілт сөзі көрсетілсе, осы бағандағы **NULL** мәнін қою әрекеті қабылданбайды. Егер **NULL** көрсетілсе, бағандағы **NULL** мәндері орналастырылады. Әдепкі бойынша, **SQL** стандарты **NULL** кілт сөзінің болуын болжайды.

PRIMARYKEY— бастапқы кілт анықтамасы. Бастапқы кілт бағанда орнатылған болса, **PRIMARYKEY** атрибуты бағанға тағайындалуы мүмкін. Егер бастапқы кілт бір бағанды қамтыса, біліктеме баған анықтамасына орналастырылады. Егер бастапқы кілтте бірнеше баған болса, іріктеу барлық бағандардың анықтамасынан кейін орналастырылады. Кез келген жағдайда, бастапқы кілт салынған өрістер бос болмауы керек, сондықтан **NOTNULL** спецификациясы көрсетіледі.

UNIQUE-атрибуты, яғни бағанда екі бірдей мән болмауы керек. Бағанда бірегей кілт салынған

(бағандар) кезде баған бастапқы кілт бөлігі емес, бірақ соған қарамастан, оның мағынасы әрқашан бірегей болуы керек. бастапқы кілт ата-ана мен бала кестеде арасындағы сілтемелер тұтастығын қамтамасыз ету үшін пайдаланылуы мүмкін осы атрибуты баған жариялады. Ата-аналық кестеге қосылу үшін шетел кілті бала кестесінде жасалады. Кестеде бірнеше UNIQUE тұтастығы шектеулерін жасауға болады.

FOREIGNKEY - сыртқы кілт, бағынышты кестеде сілтеме тұтастығын қамтамасыз ету үшін жасалады.

Сытқы кілт анықтамасы форматы:

FOREIGNKEY(<бағынышты кесте бағандарының тізімі>)

REFERENCES<басты кестенің атауы>

[<басты кесте бағандарының атауы>]

[ONDELETE {NOACTION|CASCADE|SETDEFAULT|SETNULL}]

[ONUPDATE {NOACTION|CASCADE|SETDEFAULT|SETNULL}]

Бағынысты кестенің бағандарының тізімі сыртқы кілтіне енгізілген өрістерді қамтиды. Негізгі кестенің бағандар тізімі кестелерді байланыстыру кілті болып табылатын өрістерді қамтиды (ата-аналық кестемен қосылым бастапқы кілтпен орындалса, тізім жоғалуы мүмкін).

ONDELETE, ONUPDATE қосымша параметрлер ата-аналық кестенің бастапқы кілті жойылған және өзгертілген кезде сервердің не істеу керектігін анықтайды:

- NOACTION — бала кестесінде бағынышты жазбалар болса, бас кестесіндегі тиісті жазбаларды жою немесе өзгертуге тыйым салынады;
- CASCADE— бас кестедегі жазбаны жойғанда, еншілес кестедегі барлық бағынышты жазбалар жойылады; жазбалар өзгертілген кезде, еншілес кестенің барлық бағынатын жазбаларындағы кілт өрісінің мәндері;
- SETDEFAULT— бас кестесіндегі жазбаларды жойғанда немесе өзгерткенде, бала кестесінің барлық жазбаларындағы кілт өрісі өріс анықтамасында көрсетілген әдепкі мән беріледі; Егер бұл мән бастапқы кілтте болмаса, ерекше жағдай туындайды; және сілтеме тұтастығын анықтау кезінде пайдаланылатын әдепкі мән, кейінірек бұл мән өзгертілсе, SETDEFAULT параметрімен тұтастық тұтастығы бұрынғы мәнді пайдаланады;
- SETNULL— бала кестесінің бағынышты жазбаларының негізгі өрісінде бос мән NULL деп орнатылады.

Кестенің бағандарына салынған шектеулер CHECK ережесі (<баған шарты>) арқылы анықталады, оның жалпы форматы төменде берілген:

```
<баған шарттары>::= {  
VALUE<оператор><мәні>  
|VALUE [NOT] BETWEEN <мән1 >AND <мән 2 >  
|VALUE [NOT] LIKE <мән>[ESCAPE <мән>] |VALUE [NOT] IN  
(<мән1>[, <мән2>...])  
| VALUE IS [NOT] NULL |VALUE [NOT] CONTAINING<мән>  
|VALUE [NOT] STARTING [WITH] <мән>  
}
```

Бұл жағдайда бағаның мәні бойынша шектеулер домен анықтамасындағылармен бірдей форматта сипатталады:

<оператор> = { = | < | > | <= | >= | !< | !> | <> | != };

VALUE<оператор><мәні> — домен мәні <оператор> арқылы анықталған қатынастарда <мәні> параметрінде орналасқан;

BETWEEN<мән1>AND<мән2> — Домен мәні арасында болуы керек <мән1> және <мән2> қоса алғанда;

LIKE<мән1>[ESCAPE<мән2>] — кез келген ұзындықтың кез келген мәнін көрсететін «%» белгісімен ұқсастық үлгісін анықтайды және «_» таңбасы кез келген жалғыз таңба болып табылады. ESCAPE LIKE операторында қалыпты таңбалар ретінде «%» және «_» таңбалары болса, пайдаланылады. Бұл жағдайда <мән 2> таңбасын таңдаңыз, содан кейін қызметтік нышандар ерекше күйін жоғалтады және әдеттегідей іздеу жолын енгізеді. Мысалы, CHECK (VALUELIKE «%!%» ESCAPE «!») - кез келген «!» Символы «%» дегенмен аяқталады;

IN (<1 мән> [<мән 2> ...] - домендік мәні аталған параметрлердің бірімен сәйкес келуі тиіс;

IS [NOT] NULL бағанында бос емес басқа мән болуы керек;

<Мәні> құрамы - домен мәні <мәні> кіруін қамтуы тиіс, қандай орын жоқ мәселені;

STARTING [WITH] <мән> - домен мәні «мән» параметрінен басталуы керек».

Доменнің мәні сәйкес келетін шарттар жиынтығы мүмкін. Бұл жағдайда жеке шарттарға AND немесе OR операторы қосылған.

Кестелерді жасағаннан кейін, дерекқордағы бар кестелерді сипаттау (бағандардың сипаттамалары, олардың тәртібі, әртүрлі кілттердің болуы немесе CHECK шектеулері) өзгеруі мүмкін. Жоғарыда келтірілген мысалдарды келтірейік.

DEPARTMENTS кестесін Номер және Атау өрістерімен анықтаңыз:
CREATE TABLE ОТДЕЛЫ (НомерINTEGER
NOT NULL,
Атауы VARCHAR(20))

Бөлімдер кестесінің өрісін Номер, Номер және өріс нөмірі үшін бастапқы кілтімен анықтаңыз:

CREATE TABLE БӨЛІМДЕР
(НөмірINTEGER NOT NULL PRIMARY KEY,
Атауы VARCHAR(20))

Сол кесте, бірақ екі өрістегі бастапқы кілтпен (кесте деңгейінде шектеулерді анықтау):

CREATE TABLE БӨЛІМДЕР (Нөмір
INTEGER NOT NULL,
Атауы VARCHAR(20) NOTNULL,
PRIMARYKEY(Нөмір, Атауы))

Енді домендік кестеде пайдалану мысалын келтіреміз. Доменді жасау Домен _ Нөмір мәні бар бүтін түрге ие сан 100-ден көп немесе тең:

CREATE DOMAIN Домен_НомерAS INTEGER CHECK (VA-
LUE>=100);

ҚЫЗМЕТКЕРЛЕР кестесінің анықтамасында НӨМІР өрісі ДОМЕН_НӨМІР доменімен байланысады:

CREATETABLEҚЫЗМЕТКЕРЛЕР (Номер Домен_Номер
NOTNULL,
ФИОVARCHAR(20),
PRIMARY KEY (Номер))

Екі кесте жасайық. Негізгі кесте Өрістермен өнімдер ТАУАРЛАР және баға, сондай-ақ өріс элементі бойынша бастапқы кілтпен:

CREATE TABLE ТАУАРЛАР

(Тауар VARCHAR (20) NOT NULL,
Баға INTEGER NOT NULL,
PRIMARY KEY (Тауар))

Бағынысты САТЫЛЫМ кестесінде Өріс өрісі үшін негізгі кілт және Өрістер кестесіне сілтеме тұтастығын қамтамасыз ету үшін Элемент өрісі үшін сыртқы болады. Негізгі кестедегі байланыс өрісі көрсетілмегендіктен, кестенің негізгі кілті байланыс үшін пайдаланылады:

```
CREATE TABLE ПРОДАЖИ (Нөмір INTEGER  
NOT NULL PRIMARY KEY,  
Күні DATE,  
Тауар VARCHAR (2 0) NOT NULL,  
FOREIGN KEY (Тауар) REFERENCES ТАУАРЛАР)
```

Негізгі және бағынышты кестелердің ортақ өрістерін анықтау дәл сәйкес болуы керек. Егер рәміздердің сұрыптау тәртібінде айырмашылықтар бар болса, байланыс бағандары іс жүзінде бірдей болмайды, бұл сілтеме тұтастығының бұзылуына әкеледі.

ТАУАРЛАРДЫҢ кестелері мен САТЫЛЫМ арасында байланыс орнатудың тағы бір мысалы.

Бағынысты кесте бастапқы кілтте мән өзгертілгенде және негізгі жазбадағы жазбаны жойғанда, сервердің әрекеттерін көрсетеді:

```
CREATE TABLE САТЫЛЫМ (Номер INTEGER  
NOT NULL PRIMARY KEY,  
Күні DATE,  
Тауар VARCHAR (2 0) NOT NULL,  
CONSTRAINT FK1 FOREIGN KEY (Тауар) REFERENCES  
ТОВАРЫОН UPDATE NO ACTION ON DELETE NO ACTION)
```

Енді ТАУАРЛАР кестесінде осы өнім сатылымы үшін САТЫЛЫМ кестесінде жазбалар бар болса, Тауар бағанындағы жоюдың немесе өзгертудің құлдырауы болады. Бұл мысалда сілтеме тұтастығына FK1 атауы берілді.

Деректермен толтырылған кестелер құрылымын өзгерту ең жиі дерекқордың ерекше жағдайына немесе кестеде бар деректердің жоғалуына әкелетін ең қауіпті әрекеттердің бірі болып табылады.

Қолданыстағы кестелердің құрылымын өзгерту үшін ALTER TABLE операторын пайдаланыңыз, оның оңайлатылған синтаксисі төменде келтірілген:

```
ALTERTABLE<кесте атауы>
{ [ADD<баған атауы><деректер түрі> [
NULL| NOTNULL] ]
| [DROP [COLUMN] <баған атауы>]}
```

Бір мәлімдемеде кестедегі өзгерістердің ерікті санын орындай аласыз.

Бар кестені өзгертуге арналған әртүрлі операциялар бір-бірінен үтірлермен бөлінеді. ALTER TABLE операторы рұқсат береді:

- Жаңа баған анықтамасын қосу;
- Кестедегі бағанды өшіру;
- кестенің немесе бір бағаның тұтастығы атрибуттарын жою;
- жаңа тұтастық атрибуттарын қосу.

Егер бағанның сипаттамаларын өзгерту, сондай-ақ бағанды жою, мүмкін болмаса, сәтсіз болуы мүмкін:

- баған PRIMARY KEY немесе UNIQUE атрибуттарын алады, бірақ бағандағы ескі мәндер бірегей деректерге қойылатын талаптарды бұзады;
- Кестелер арасындағы сілтеме тұтастығын бұзуға алып келген негізгі немесе шетелдік кілтдің бөлігі ретінде енгізілген қашықтағы баған;
- бағанда кесте деңгейінде тұтастық шектеулер тағайындалды;
- баға есептік бағандарда іздеу, триггерлер және өрнектердегі басқа дерекқор компоненттерінде пайдаланылады. Осылайша, егер қажет болса, атрибуттарды өзгерту
- бағанды немесе бағанды жою жағдайында алдымен кесте мен дерекқордың салдарын өзгерту немесе жою сияқты өзгерістерді неге әкелуі мүмкін екенін мұқият талдау қажет. Жаңа бағанды қосу:

```
ALTER TABLE <кесте атауы>ADD <баған анықтамасы>
```

Жаңа бүтіндік шегін қосу:

```
ALTER TABLE <кесте атауы>ADD [CONSTRAINT <шектеу атауы>^
<бүтіндік анықтамасы>
```

Бағанды өшіру:

```
ALTER TABLE <кесте атауы>DROP <баған атауы>[.....] бүтіндігін
```

шектеуді өшіру:

```
ALTER TABLE <кесте атауы>DROP <шектеу атауы>
```

Кесте оператор тарапынан жойылады:

```
DROP TABLE <кесте атауы>
```

Мысалы, бағанды келесі пәрмені бар кестеге қосуға болады:

```
ALTER TABLE KLIENT ADD ADRES VARCHAR(50)
```

Кестеге тұтастық шектеу қосу (бастапқы кілт) пәрмен арқылы орындалады:

```
ALTER TABLE KLIENT  
ADD CONSTRAINT PK_KLIENT  
PRIMARY KEY (ID)
```

Кестені өшіру мысалы:

```
DROP TABLE KLIENT
```

Өшіру негізгі кестелер үшін бұғатталуы мүмкін, оларға бағынатын кестелерде (қазіргі уақытта жойылмайды) осы кестелердің шетелдік кілтіне сілтемелер бар. Шындығында, негізгі кестені жою сілтеме тұтастығын бұзады. Сондықтан, барлық бағынды кестелердегі сілтеме тұтастығы шектеулерін алып тастаңыз немесе қажет болса алдымен бағынысты кестелерді жойыңыз, содан кейін негізгі кестені жойыңыз.

12.3. ДЕРЕКТЕРДІ ТАҢДАУ. SELECT ОПЕРАТОРЫ

SELECT мәлімдемесі DML сұрау тілінің DML бөлімшесіне сілтеме жасайды. Бұл ең күрделі және қуатты SQL операторларының бірі. Бұл оператор өте күрделі және дамыған құрылымға ие. Дерекқорлармен қандай да бір түрде байланысқан кез келген маманға белгілі болуы керек. Сондықтан осы оқу құралы оған үлкен көңіл бөледі.

12.1-суретте келесі талқылауға мысал ретінде пайдаланылатын дерекқор үлгісі көрсетілген.

Тауарлар кестесі үстелдермен байланысты, содан кейін тауарлық өріс бірден-көпге дейін. Осыған ұқсас, Клиенттер кестесі Клиент өрісіне арналған біріктірілген байланыспен Сатылым кестесіне байланысты. Сұранысты орындау нәтижелерін неғұрлым нақты көрсету үшін біз аяқталған кестелерді мысал ретінде пайдаланамыз (12.2-сурет).

Жолы	Жол түрі	Сипаты
Тауар	Жолы	Тауар атауы
ЕИ	Жолы	Өлшем бірлігі
Құны	Саны	Құны

a

Жолы	Жол түрі	Сипаты
Тұтынушы	Жолы	Тұтынушы
ЖСН	Жолы	Тұтынушының ЖСН
Қала	Жолы	Қала
Телефон	Жолы	Телефон

б

Жолы	Жол түрі	Сипаты
Нөмір	Саны	Құжат нөмірі
Күні	Күні	Құжат күні
Тауар	Жолы	Тауар атауы
Саны	Саны	Тауар саны
Тұтынушы	Жолы	Тұтынушы

в

12.1 сурет. Мысалдар үшін пайдаланылатын кестелер құрылымы:
a — ТАУАРЛАР кестесі; *б* — ТҰТЫНУШЫЛАР кестесі; *в* — САТЫЛЫМ кестесі

(«Таңдаңыз», ағылшын.) SELECT - шартты қанағаттандыратын дерекқорынан деректер жиынын (үлгі) қайтарады SQL тілі туралы есеп. Бұл деректерді бір немесе бірнеше дерекқор кестелерінен таңдауға және нәтижелерді қажетті пішінге түрлендіруге мүмкіндік береді. Бұл оператор реляциялық алгебра операторларына баламалы әрекеттерді орындай алады. Оның көмегімен әртүрлі кестелерден деректерді таңдау үшін күрделі және қиын жағдайлар жасай аласыз. Егер таңдау бірнеше кестеден жасалса, онда олар біріктіру әрекеті туралы айтады.

SELECT сұрауын жасаған кезде пайдаланушы деректердің қажетті жиынтығын (бағандар жиыны, жазбаларды таңдау критерийлері, топтау мәндері, жазбалардың шығысын ресімдеу және т.б.) сипаттайды. Алдымен, кестеден алынған барлық жазбалар шығарып алынады, содан кейін жиынтықтағы әрбір жазба үшін ол көрсетілген критерийге сәйкес келетіндігі тексеріледі. біріктіру, бірнеше кестелерден алынған өнім Кестенің бірінші компоненті жүзеге асырылады, және тек содан кейін қажетті таңдалған жазбалар нәтижесінде жинағынан болса. Тұтастай алғанда жалпы тәртіпте SELECT нұсқауында келесі синтаксис бар:

```

SELECT [ALL|DISTINCT ]{*| [бн атауы [AS жаңа_ атауы ] ] } [, .
. . n ]
FROM кесте атауы [ [AS] лақап аты] [, . . . n]
[WHERE <шарты>]
[GROUP BY баған атауы [, . . . n] ]
[HAVING <топты таңдау критерийлері>]
[ORDER BY атауы_баған [, . . . n] ]

```

SELECT мәлімдемесі сұраудың нәтижесіне қосылатын өрістерді (бағандарды) анықтайды. Тізімде олар үтірлермен бөлініп, сұраныстың нәтижесі ретінде ұсынылуы керек. Егер сіз өріс атауын қолдансаңыз,

Товар	ЕИ	Цена
Халва	кг	45
Сахар	кг	60
Мука пшеничная	кг	50
Конфеты "Мишка"	кг	150
Конфеты "Коровка"	кг	90
Мармелад "Кроха"	кг	85
Сосиски молочные	кг	260
Сардельки "Рошинские"	кг	300
Шпроты "Штурвал"	шт	56

a

Клиент	ИНН	Город	Телефон
ООО "Горизонт"	025500123152	Москва	4890605
ИП Привалов И.И.	025501025666	Санкт-Петербург	2560245
ООО "Ромашка"	025501244555	Москва	3658815
ООО "Перевал"	145889898622	Тверь	221588
ИП Иванов Ф.И.	025558000554	Москва	1155488
Таран О.С.	360224005454	Москва	1215648
Федорова Д.С.	500255510055	Санкт-Петербург	4449702
Лесовая В.Н.	212585832187	Москва	3021402
БМСТ	555878970025	Санкт-Петербург	4353822
Дремина Е.Е.	025578721058	Москва	6582209
ИП Газимова К. К.	025587978766	Москва	6521588

б

Номер	Клиент	Товар	Количество	Дата
451	Федорова Д.С.	конфеты "Коровка"	50	02.04.2013
156	ИП Газимова К.К.	Конфеты "Мишка"	30	02.04.2013
162	Федорова Д.С.	Халва	10	15.02.2013
200	Лесовая В.Н.	Мармелад "Кроха"	20	07.02.2013
25	Таран О.С.	Халва	5	04.01.2013
85	Лесовая В.Н.	Мука пшеничная	50	02.04.2013
112	ООО "Ромашка"	Халва	40	15.02.2013
254	Лесовая В.Н.	Сахар	50	01.02.2013
140	ИП Газимова К.К.	Конфеты "Коровка"	15	09.01.2013
144	Таран О.С.	Сосиски молочные	30	01.04.2013

в

12.2 сурет. Кестенің бастапқы жағдайы:

a — ТАУАРЛАР кестесі; *б* — ТҰТЫНУШЫЛАР кестесі; *в* — САТЫЛЫМ кестесі

бос орындарды немесе шектегіштерді қамтитын, тік жақшалармен қоршалуы керек. Барлық өрістерді * символы арқылы таңдай аласыз және өріс атауын емес, бірнеше атаулардан өрнекті пайдаланыңыз. Егер бірнеше кесте өңделсе, өрістер тізіміндегі әр түрлі кестелердегі сол өрістердің бар болуында толық өріс сипаттамасы пайдаланылады, яғни. Кесте атауы. Өріс атауы.

SELECT операторының элементтерін өңдеу келесі ретпен жүзеге асырылады:

- FROM — деректер таңдалатын дерекқор кестелерінің тізімі;
- WHERE — Белгіленген шарттарға сәйкес нысан жолдарын сүзгілеу;
- GROUP BY — көрсетілген бағанда бірдей мәнге ие жолдардың топтары жасалады;
- HAVING — нысан жолдарының топтары көрсетілген күйге сәйкес сүзіледі;
- SELECT — Шығарылған деректерде қандай бағандар болуы керек екенін анықтайды;
- ORDER BY — Операторлардың орындау нәтижелерінің тәртібі анықталды.

Сөйлемдегі сөйлемдер мен сөз тіркестерінің реті өзгертілмейді. Тек екі ELECT және FROM ұсыныстар талап етіледі, барлық қалғандарды алып тастауға болады. SELECT - жабық операция: кестеге сұраныстың нәтижесі - басқа кесте. Төмендегі мысалдарда көрсетілгендей, осы операторды жазудың көптеген нұсқалары бар.

Мысалы, осы операторды пайдаланып, барлық бағандардан және Клиенттер кестесіндегі барлық енгізулерден тұратын деректер жиынтығын жасай аласыз:

SELECT * FROM Тұтынушылар

Егер жұлдызшаның орнына, кестенің барлық бағандарын тізімдесе, сол деректер жиынтығын алуға болады:

SELECT Тұтынушы,ЖСН,Қала,Телефон
FROM Тұтынушылар

Сұраудың нәтижесі қайталанатын мәндер болуы мүмкін, себебі SELECT мәлімдемесі деректерді алуды орындау кезінде қайталанатын мәндерді алып тастамайды. DISTINCT предикаты таңдалған өрістерде қайталанатын жазбалары бар деректер блоктарын тастағыңыз келетін жағдайларда пайдаланылуы керек. SELECT нұсқауындағы өрістердің әрқайсысы үшін мәндер

бірегей болуы тиіс, сондықтан оларды қамтитын жазбалар шығару жиынтығына кіруі мүмкін. DISTINCT-ды пайдалануды шектеудің себебі - оны қолдану сұраныстардың орындалуын күрт төмендетуі мүмкін.

Алдыңғы сұрауды келесідей түзетеміз:

```
SELECT DISTINCT КЛИЕНТЫ.Қала  
FROM ТҰТЫНУШЫЛАР
```

Нәтижесінде біз қалалардың тізімін аламыз - клиенттердің орналасуы.

WHERE тармағы деректер жиынтығында қажетті жазбаларды ғана қамту үшін пайдаланылады. Бұл жағдайда SELECT операторының келесі пішімі бар:

```
SELECT {* | <мән1> [, <мән2> ...]}  
FROM <кесте1> [, <кесте2> ... ]  
WHERE <шарт>
```

WHERE кілт сөзі іздеу терминдерінің тізімін іздейді. SELECT операторы қайтаратын деректер WHERE тармағынан кейін көрсетілген іздеу шартына сәйкес келетін жазбаларды ғана қамтиды. Әрі қарай, әртүрлі кестелерден деректерді таңдау үшін өте күрделі жағдайларды қарастырамыз, бірақ қазір қарапайым талдау жасаймыз.

Іздеу шарттарының негізгі түрлері (немесе предикаттар) төменде берілген.

1. Салыстыру: бір өрнек есептеуді басқа біреудің есептеу нәтижесі немесе түрлі бағандардың мәндері салыстырылады.

2. Ауқым: бағанның мәнінің төмендеуі немесе өрнектің белгілі бір мәндер ауқымына баға беру нәтижесі тексеріледі.

3. Жиынға байланысты: бағанның мәніне тиесілі ме немесе өрнектің белгілі бір мәндер жиынын бағалау нәтижесі.

4. Үлгі сәйкестігі: жолдың мәні берілген үлгіге сәйкестігін тексереді.

5. NULL мәні: бағанда NULL (белгісіз мән) идентификаторы бар-жоғын тексереді.

Мысалы, ТҚТЫНУШЫЛАР кестесінен Санкт-Петербургтағы клиенттер тізімі сұратылып алынуы мүмкін:

```
SELECT *  
FROM ТҰТЫНУШЫЛАР  
WHERE ТҰТЫНУШЫЛАР.Қала = «Санкт-Петербург»
```

SQL тілінде келесі салыстыру операторларын қолдана аласыз:

=	Тең
<	кіші
>	Көп
<=	Кіші немесе тең
>=	Көп немесе тең
<>	Тең емес

Мысалы, сатылатын тауардың саны көп болатын тауарлардың барлық сатылымдарын көрсету 10:

```
SELECT *  
FROM ПРОДАЖИ WHERE саны>10
```

Неғұрлым күрделі предикаттарды AND, OR, немесе NOT логикалық операторларымен, сондай-ақ өрнек бағаланатын тәртіпті анықтау үшін пайдаланылатын жақшалар арқылы құрастыруға болады. Ережені есептеу жағдайында төмендегі ережелерге сәйкес жүзеге асырылады:

- өрнек солдан оңға қарай бағаланады;
- Алғашқы жақшадағы алдымен есептеледі;
- NOT операторлары AND және OR операторларына дейін орындалады;
- AND нұсқаулары OR орындалмай тұрғанда орындалады. Кез келген ықтимал белгісіздікті жою үшін жақшаларды қолдану ұсынылады. Міне, бірнеше күрделі іздеу шарттарын қолдану мысалдары. Келесі сұрау нәтижесі бойынша 50 немесе одан жоғары және 100-ден аз немесе тең баға бойынша тауарлардың тізімі:

```
SELECT Тауар, Құны FROM ТАУАРЛАР  
WHERE Құны>=50 AND Құны<=100
```

Сол сұрауды BETWEEN операторы арқылы орындауға болады. BETWEEN кілт сөзі оның ең аз және ең үлкен мәндері арқылы анықталған белгілі бір аралығындағы мәнді табу үшін пайдаланылады. Көрсетілген мәндер іздеу шартына қосылады:

```
SELECT Тауар, Құны FROM ТАУАРЛАР
WHERE Құны BETWEEN 50 And 100
```

Алынған үлгі бұрынғы мысалдағы үлгіге ұқсас болады.

Егер сіз BETWEEN емес теріс пайдалансаңыз, көрсетілген ауқым шегінен тыс мәндер алынады.

IN операторы белгілі бір мәнді көрсетілген мәндер тізімімен салыстыру үшін пайдаланылады және өрнекті бағалау нәтижесі берілген тізімдегі мәндердің біріне сәйкес келетінін тексереді. IN операторын пайдалану арқылы сол нәтижені OR операторын пайдалану жағдайында алуға болады, алайда IN сөзі жылдамдық болады. NOT IN ұсынылған тізімнен басқа кез келген мәндерді таңдау үшін пайдаланылады.

Мысалы, Мәскеуде немесе Санкт-Петербуркте өмір сүрмейтін клиенттерді тізімдеу үшін сұрауды орындау керек:

```
SELECT Тұтынушы, Қала FROM ТҰТЫНУШЫЛАР
WHERE Қала NOT IN («Мәскеу»,»Санкт-Петербург»)
```

LIKE операторымен өрнекті алмастырғыш таңбаларды пайдалануға мүмкіндік беретін берілген үлгімен салыстыруға болады:

- «%» таңбасы - таңбаның орнына еркін таңбалардың кез-келген саны ауыстырылуы мүмкін;
- «_» таңбасы жолдағы бір таңбаны ауыстырады.

Мысалы, келесі сұрау нәтижесі бойынша телефон нөміріндегі екінші санмен клиенттердің тізімі көрсетіледі — 4:

```
SELECT *
FROM ТҰТЫНУШЫЛАР
WHERE ТҰТЫНУШЫЛАР. Телефон LIKE '_4%'
```

Нәтижесінде, тұтынушы «Федорова Д.С.» 4449602 телефон арқылы табады.

Тегі мен аты бойынша сюжеттік «және» бар тұтынушыларды таңдау үшін «%» екі таңбаны пайдаланып сұрауды жасау керек:

```
SELECT *
FROM ТҰТЫНУШЫ
WHERE ТҰТЫНУШЫЛАР. тұтынушы LIKE '%ва%'
```

Іздеудің бұл түрі пайдаланушы қалаған нысан туралы барлық деректерді білмейді немесе есте сақтаған жағдайларда қолданылады.

IS NULL операторы ағымдық мәнді NULL мәнімен салыстыру үшін пайдаланылады - кез келген мәнің жоқтығын көрсететін ерекше мән (NULL - бос орын немесе нөл емес екенін атап көрсету маңызды). IS NOT NULL өрнегі өрістегі мәнің болуын тексеру үшін пайдаланылады.

Мысалы, телефон нөмірі жоқ тұтынушыларды табу қажет (Phone өрісі ешқандай мәнді қамтымайды):

```
SELECT Тұтынушы  
FROM ТҰТЫНУШЫЛАР  
WHERE Телефон IS NULL
```

Телефоны бар клиенттерді таңдау (Телефон өрісі кез келген мәнді қамтиды):

```
SELECT Тұтынушы FROM  
ТҰТЫНУШЫЛАР  
WHERE Телефон IS NOT NULL
```

Тұтастай алғанда, SQL сұрау деректерінің нәтиже жинағындағы жолдар ешқандай түрде реттелмейді. Дегенмен, оларды белгілі бір тәртіпте сұрыптауға болады. Осыны орындау үшін ORDER BY сөйлемі шығару кестесінің деректерін көрсетілген ретпен сұрыптайтын SELECT операторына орналастырылады. Сұрыптауды бірнеше өрістерде орындауға болады, бұл жағдайда олар үтірлермен бөлінген ORDER BY кілт сөзінен кейін тізімделеді.

Сұрыптау әдісі (өсуі немесе кемуі бойынша) ORDER BY параметрінде көрсетілетін кілт сөзді сұрыптау орындалатын өрістің атауымен анықтайды. Әдепкіде, өсу тәртібінде сұрыптау орындалады. Ол ASC кілт сөзімен айқын түрде беріледі.

Сұрыптауды кері тәртіпте орындау үшін, ол орындалатын өрістің атауынан кейін DESC кілт сөзін көрсету керек. ORDER BY тармағы таңдаулы жазбаларды кез келген бағанның немесе бағандардың тіркесімінің мәндерінің өсу немесе кему тәртібімен, осы бағандардың нәтижелер жиынында немесе жоқ екеніне қарамастан ұйымдастыруға мүмкіндік береді. ORDER BY сөйлемі әрдайым SELECT операторындағы соңғы элемент болуы керек. Мысалы, өнімді әліпбилік ретпен тізімдеу керек:

```
SELECT *  
FROM ТАУАРЛАР  
ORDER BY ТАУАРЛАР.Тауар
```

ORDER BY тармағында бірнеше элемент көрсетілуі мүмкін. Содан кейін сұрыптау критерийі - аталған барлық өрістерді «жабыстыру» нәтижесі. Мысалы, сатылым кестесін Өнім жәнеКкүн өрістері бойынша сұрыптауға болады:

```
SELECT *  
FROM САТЫЛЫМДАР  
ORDER BY САТЫЛЫМДАР.Тауар, САТЫЛЫМДАР.Күні
```

Келесі сұрауда сұрыптау біріншіден сатып алушының аты бойынша, содан кейін өнімнің атауы бойынша жүзеге асырылады, сайып келгенде, күні бойынша:

```
SELECT Тұтынушы, Тауар, Күні, СаныFROM  
Сатылымдар  
ORDER BY Тұтынушы, Тауар, Күні
```

Қосылыстар— Бұл екі немесе одан көп кестелер бір біріктірілген процесс. Ақпаратты бірнеше кестелерден немесе сұраулардан бірыңғай логикалық деректер жиынтығымен біріктіру мүмкіндігі кейінірек талқыланатын SQL кеңейтілген мүмкіндіктерін анықтайды.

Бұрын біз WHERE сөйлемін сұрау нәтижесінің шектеулерін енгізу құралы ретінде анықтадық. Бірақ, белгілі бір бағандардағы кестелер арасындағы байланысты орнату үшін шектеу шарты ретінде, бұл екі кестенің қосылуы болады. Федерацияда сол деректер түріне жататын және бірдей деректер түріне жататын өрістер қатыса алады, бірақ оларда бірдей атаудың болмауы керек. Екі кестеден алынған деректер блоктарын сәйкес өрістер көрсетілген өрістерде табылған кезде біріктіріледі.

Сатудың барлық кестелерін САТЫЛЫМ кестесінен және әрбір өнімнің бағасын ТАУАРЛАРДЫҢ кестесінен көрсету үшін таңдаңыз, мұндай сұрауды орындауға болады:

```
SELECT САТЫЛЫМДАР.*, ТАУАРЛАР.ҚҰНЫ  
FROM САТЫЛЫМДАР, ТАУАРЛАР  
WHERE САТЫЛЫМДАР.ТАУАР = ТАУАРЛАР.ТАУАР
```

Осы мәлімдемені орындаған кезде, САТЫЛЫМДАР кестесіндегі әрбір жазба үшін, ТАУАРЛАР кестесінде табуға болады, оның мәні Өріс өрісіндегі САТЫЛЫМДАР кестесінің ағымдық жазбадағы Элемент өрісіндегі мәнмен бірдей болады (12.3-сурет).

Номер	Клиент	Товар	Количество	Дата	Цена
140	ИП Газимова К.К.	конфеты "Коровка"	15	09.01.2013	90
45	Федорова Д.С.	Конфеты "Коровка"	50	02.04.2013	90
156	ИП Газимова К.К.	Конфеты "Мишка"	30	02.04.2013	150
200	Лесовая В.Н.	Мармелад "Кроха"	20	07.02.2013	85
85	Лесовая В.Н.	Мука пшеничная	50	02.04.2013	50
254	Лесовая В.Н.	Сахар	50	01.02.2013	60
144	Таран О.С.	Сосиски молочные	30	01.04.2013	260
25	Таран О.С.	Халва	5	04.01.2013	45
162	Федорова Д.С.	Халва	10	15.02.2013	45
112 000	"Ромашка"	Халва	40	15.02.2013	45

12.3 сурет. Кестенің ішкі қосылыстарының нәтижелері

Кестелерді іздеу күйінде қандай тәртіпте тізімдеу керек, яғни кестелердің қайсысында сол жақта, ал қайсысында оң жағында көрсетілетіні маңызды емес. Кестелерді қосу әдісі *ішкі байланыс* деп аталады.

Міне, кестелердің ішкі бірігуінің тағы бір мысалы бар. САТЫЛЫМДАР кестесінен тауарларды тұтыну туралы барлық жазбаларды және әр клиент үшін оның қаласын үстелден көрсетіңіз ТҰТЫНУШЫЛАР:

SELECT САТЫЛЫМДАР.*, ТҰТЫНУШЫЛАР

.Қала FROM САТЫЛЫМДАР, ТҰТЫНУШЫЛАР

WHERE ТҰТЫНУШЫЛАР.тұтынушы = ТҰТЫНУШЫЛАР.Тұтынушы

Мысалдан көріп тұрғанымыздай (12.4-сурет) деректердің жиынтығы САТЫЛЫМДАР кестесінде ешқандай жазбалар жоқ ТҰТЫНУШЫЛАР кестесінің жазбаларын қамтымайды.

Екі кестенің ішкі қосылымы кезінде, нәтиже жиынтығын қалыптастыру реті келесідей ұсынылуы мүмкін. САТЫЛЫМДАР сөзінен кейін көрсетілген бағандардан аралық деректер жиынтығы әрбір жазбаны бірінші кестеден әр жазбамен біріктіру арқылы құрастырылады.

Номер	Клиент	Товар	Количество	Дата	Город
140	ИП Газимова К.К.	конфеты "Коровка"	15	09.01.2013	Москва
156	ИП Газимова К.К.	Конфеты "Мишка"	30	02.04.2013	Москва
254	Лесовая В.Н.	Сахар	50	01.02.2013	Москва
200	Лесовая В.Н.	Мармелад "Кроха"	20	07.02.2013	Москва
85	Лесовая В.Н.	Мука пшеничная	50	02.04.2013	Москва
112 000	"Ромашка"	Халва	40	15.02.2013	Москва
25	Таран О.С.	Халва1	5	04.01.2013	Москва
144	Таран О.С.	Сосиски молочные	30	01.04.2013	Москва
162	Федорова Д.С.	Халва	10	15.02.2013	Санкт-
45	Федорова Д.С.	Конфеты "Коровка"	50	02.04.2013	Санкт-

12.4 сурет. ТҰТЫНУШЫЛАР және САТЫЛЫМДАР кестесінің

WHERE тармағында іздестіру шартын қанағаттандырмайтын барлық жазбалар.

Кестелерді SELECT сөзінен кейін және WHERE кейін іздеу күйінен кейін қайтарылған бағандар тізіміндегі кестелермен байланыстырудың жоғарыдағы мысалдарында, кесте аты бағанның атауына дейін жазылған, мысалы:

WHERE КЛИЕНТЫ. Тұтынушы= САТЫЛЫМДАР. Тұтынушы
Баған атауынан бұрын, әр түрлі кестелер бірдей бағандар болған жағдайда кестенің атауын көрсетуіңіз керек (бұл мысалдағы сияқты). Бағандарды анықтау үшін кесте атауын пайдалану ыңғайсыз, себебі бұл қиын және қиын оқиды. Әрбір үстелге қысқа атау - *бүркеншік атты* тағайындау әлдеқайда жақсы. Бүркеншік аттар нақты кесте атауынан FROM кілт сөзінен кейін бастапқы кестелер тізіміндегі бос орын арқылы бөлінеді. Мысалы, сұрау

SELECT САТЫЛЫМДАР.*, ТҰТЫНУШЫЛАР. Қала FROM САТЫЛЫМДАР, ТҰТЫНУШЫЛАР

WHERE ТҰТЫНУШЫЛАР.Тұтынушы= САТЫЛЫМДАР.Тұтынушы

Бүркеншік кестелерді енгізгеннен кейін ол әлдеқайда ықшам көрінеді:

SELECT П.*, К. Қала FROM САТЫЛЫМДАР П, ТҰТЫНУШЫЛАР К
WHERE К. Тұтынушы = П. Тұтынушы

Арифметикалық өрнектер нәтижелер жиынының есептелетін бағандарының мәндерін есептеу үшін пайдаланылады. Бұл жағдайда SELECT операторынан кейін есептелетін бағанның атауының орнына баған тізімінде өрнек көрсетіледі. Мәселен, келесі сұраныс САТЫЛЫМДАР кестесінен тауарларды сатуға арналған барлық жазбаларды қайтарады және әрбір тауар сатуға жіберілген тауарлардың жалпы құнын есептейді:

SELECT П.*, Т.Құны, П.Саны * Т.Құны FROM САТЫЛЫМДАР П,
ТАУАРЛАР Т WHERE П. Тауар = Т. Тауар

Бұл сұрақтың нәтижесін 12.5 суреттен көруге болады.

Есептелетін өрісті құру үшін, қосу, алу, көбейту және бөлудің арифметикалық амалдары, сондай-ақ SQL тілінің бекітілген функциялары қолданылады. Кез келген атауды көрсетуге болады

Номер	Клиент	[Товар	Количество	Дата	MULTIPLY
140	ИП Газимова К.К.	[Конфеты "Коровка"	15	09.01.2013	1350
45	Федорова Д.С.	Конфеты "Коровка"	50	02.04.2013	4500
156	ИП Газимова К.К.	Конфеты "Мишка"	30	02.04.2013	4500
200	Лесовая В.Н.	Мармелад "Кроха"	20	07.02.2013	1700
85	Лесовая В.Н.	Мука пшеничная	50	02.04.2013	2500
254	Лесовая В.Н.	Сахар	50	01.02.2013	3000
144	Таран О.С.	Сосиски молочные	30	01.04.2013	7800
162	Федорова Д.С.	Халва	10	15.02.2013	450
112	ООО "Ромашка"	Халва	40	15.02.2013	1800

12.5 сурет. Есептелетін жолды құру

кестенің бағанында, бірақ тек FROM сөйлемінде тізімделген кестенің немесе сұраудың баған атауын ғана пайдаланыңыз. Күрделі өрнектерді салу жақшаларды пайдаланады. Алынған кестенің баған атауларын айқын көрсете аласыз, олар үшін AS фразасы қолданылады. Келесі сұраныс сатылым жылы мен айы бар өнімдер тізімін көрсетеді:

```
SELECT T.Тауар, Year(n.Күні) AS Жыл, Month(n.Күні)
AS Ай
FROM ТАУАРЛАР T ,
САТЫЛЫМДАР P WHERE T .
Тауар=P.Тауар
```

Сұрау Year және Month айын белгілеу үшін кірістірілген функцияларды Жыл және Айды пайдаланады.

Агрегат (қорытынды) функциялары деректер жинағының барлық жазбаларында операциялардың жалпы мәндерін есептеуге арналған. Келесі функциялар жиынтыққа қатысты:

- COUNT (<өрнек>) — SQL-сұрауының шығыс жиынтығындағы жазбалардың санын анықтайды (деректердің нәтиже жинағының барлық жазбаларындағы өрнектің мәндерінің саны);
- SUM(<өрнек>) — өрнек мәндерін қорытады;
- AVG(<өрнек>) — бұл функция таңдалған жазбалардың белгілі бір өрісінде сақталатын мәндердің жиынтығының орташа мәнін есептеуге мүмкіндік береді. Бұл орташа арифметикалық мән, яғни олардың санына бөлінетін мәндердің сомасы;
- MAX (<өрнек>) — ең жоғарғы мәнді анықтайды;
- MIN(<өрнек>) — ең төменгі мәнді анықтайды.

Барлық осы функциялар бір мәнде жұмыс істейді кестенің бағаны (бірнеше кестенің бағандары) немесе арифметикалық өрнекпен бірге бір мәнді қайтарады. Кез келген функциялардың нәтижелерін есептеу кезінде барлық бос мәндер алдымен жойылады, содан кейін қажетті операция қолданылады

бағанның қалған ерекше мәндеріне ғана. COUNT (*) нұсқасы - COUNT функциясын пайдаланудың ерекше жағдайы, оның мақсаты бос, қайталанатын немесе кез келген басқа мәндер бар-жоқтығына қарамастан алынған кестенің барлық жолдарын санау. Келесі сұраудың нәтижесінде САТЫЛЫМДАР кестесіндегі жолдар саны есептеледі:

```
SELECT Count (*) AS Сатылым саны FROM  
САТЫЛЫМДАР
```

Егер сіз бірдей жазбалар тобынан біреуін қарастыруыңыз қажет болса, онда жақшадағы өрнек кілт сөзді қамтиды DISTINCT:

```
COUNT(DISTINCT Тұтынушы)
```

DISTINCT Бұл MIN және MAX функциялары үшін мағынасы жоқ, бірақ оны пайдалану SUM және AVG функцияларының нәтижелеріне әсер етуі мүмкін, сондықтан әр нақты жағдайда қатысуы керек екенін алдын-ала ойластырған жөн. Сонымен қатар, DISTINCT кілт сөзі кез келген сұрауда бірнеше рет көрсетілуі мүмкін.

Мысалы, келесі қоймада тауарды сатып алатын сатып алушылар санын есептейді:

```
SELECT COUNT(DISTINCT САТЫЛЫМДАР.КЛИЕНТ) AS  
Тұтынушылар саны  
FROM САТЫЛЫМДАР
```

Басқа мысалда бір күнде жіберілетін тауарлардың жалпы құны есептеледі:

```
SELECT SUM^ Саны*Т. Құны) AS Жалпы Бағасы FROM  
САТЫЛЫМДАР П,ТАУАРЛАР Т  
WHERE (П. тауар. Тауар) AND (П. Күні=' 02.04.2013')
```

COUNT, MIN, MAX функциялары сандық және сандық емес өрістерге де қолданылады. SUM және AVG функциялары сандық өрістер жағдайында ғана пайдаланылуы мүмкін.

Кейде сұраулар сұрауларда талап етіледі, яғни сіз барлық деректер жиынтығына емес, белгілі бір топтар үшін функцияны орындау нәтижесін аласыз (мысалы, белгілі бір күн үшін әрбір өнім үшін әрбір клиент үшін жиынтықты есептеңіз). Осы мақсатта SELECT операторы GROUP BY ережесін қолданады. GROUP BY қатысатын сұрауды шақырады

Сұрауды топтастыру, себебі ол SALES операциясының нәтижесі ретінде алынған деректерді топтастырады. Әрбір жеке топ үшін бір жиынтық жол жасалады. SALES сөйлемінен кейін топтағы мәндерді өзгертпейтін сол бағандар тізімге енгізіледі, яғни олар топтауға қатысады. Мысалы, SALES кестесіндегі тауарларға арналған сұранысты топтастыру арқылы SALES сөйлеміндегі күнді көрсете алмайсыз, себебі сатудың әртүрлі күндері әр өнімге сәйкес келуі мүмкін және сондықтан мұндай сұрау қате болады:

```
SELECT П.Тауар, П.Күні SUM(n. Саны)
FROM
САТЫЛЫМДАР П
GROUP BY П.Тауар
```

Сатылған тауарлардың жалпы санын алу үшін (12.6-сурет) сұрауды қолданыңыз:

```
SELECT П. Тауар, SUM^. Саны)
FROM
САТЫЛЫМДАР П
GROUP BY П. Тауар
```

Ал әрбір күн үшін сатылатын тауарлардың санын алу үшін, алдыңғы сұрауды келесідей түзету қажет:

```
SELECT П. тауар П. Күні, SUM^. Саны)
FROM САТЫЛЫМДАР П
GROUP BY П. Тауар, П. Күні
```

Басқаша айтқанда, SALES сөйлемі тізімінде тізілген барлық өріс атаулары, жиынтығы функциясында баған атауы пайдаланылған жағдайды қоспағанда, GROUP BY сөйлемінде болуы керек. Сонымен қатар GROUP BY сөйлемінде SALES сөйлемінде жоқ бағандар болуы мүмкін.

Сұраулардағы топтауды қолданудың тағы бір мысалы - әрбір клиенттің сатып алуларының қосалқы шоты:

Товар	SUM
Конфеты "Коровка"	65
Конфеты "Мишка"	30
Мармелад "Кроха"	20
Мука пшеничная	50
Сахар	50
Сосиски молочные	30
Халва	55

12.6 сурет. Топтық сұранысты орындау нәтижелері

```
SELECT Тұтынушы, COUNT (*)
FROM САТЫЛЫМДАР
GROUP BY Тұтынушы
```

WHERE сөйлемі GROUP BY-мен бірге пайдаланылса, онда ол алдымен өңделеді және іздеу шартын қанағаттандыратын жолдар ғана топтастырылады.

Мысалы, әр номинацияның тауары сатылатын соманы анықтау үшін (12.7-график):

Товар	SUM
Конфеты "Коровка"	5850
Конфеты "Мишка"	4500
Мармелад "Кроха"	1700
Мука пшеничная	2500
Сахар	3000
Сосиски молочные	7800
Халва	2475

12.7 сурет. Шартпен топтау сұрауын орындаудың нәтижесі

```
SELECT П. Тауар, SUM (П. Саны^ Кұны) FROM
САТЫЛЫМДАР П, ТАУАРЛАР Т WHERE Т.Тауар =
П.Тауар GROUP BY П. Тауар
```

Әрбір күн үшін сатып алушылардың санын анықтау үшін осы сұранысты орындау қажет:

```
SELECT Күні, COUNT(DISTINCT Тұтынушы) FROM
ПРОДАЖИ GROUP BY Күні
```

Егер нәтиже жиынынан алынған жолдардың нәтижелеріне шектеулер қою керек болса, онда HAVING сөйлемі GROUP BY сөйлемінен кейін пайдаланылады. Бұл шығу жиынтығын «сүзу» үшін қосымша мүмкіндік. Алынған функцияларды SALES сөйлемі тізімінде және HAVING сөйлемінде ғана қолдануға болатындығын ескеру керек, ал WHERE тармағында мұндай функцияларды көрсету мүмкін емес. Мысалы, келесі сұраныс 3000-нан астам сатылған тауарлардың тізімін көрсетуге мүмкіндік береді.:

```
SELECT П. Тауар, SUM (П. Саны^ Кұны)
FROM САТЫЛЫМДАР П,
ТАУАРЛАР Т WHERE Т.
Тауар = П. Тауар GROUP BY
П. Тауар
HAVING SUM (П. Саны^ Кұны) >3000
```

Шығарылған тауарлардың жалпы саны 10 бірліктен кем болатын тауарларды өткізу күнін көрсетіңіз, әр күнде бір мәміле жасалмады:

```

SELECT Күні,COUNT(*)
FROM
САТЫЛЫМДАРWHERE
Саны>=10 GROUP BY
Күні HAVING
COUNT(*)>1

```

Кейде салыстыру әрекеттерін пайдаланған кезде сұрауларда сұраныстың нәтижесін салыстыру қажет мән алдын ала анықталмайды және бірнеше мәндер болмайды немесе басқа SELECT операторын орындау арқылы алынады. Мұндай жағдайларда, ішкі сұраулар пайдаланылады (кірістірілген сұраулар).

Ішкі сұрау (салынған сұраулар) уақытша кестені жасау құралы болып табылады, оның мазмұны сыртқы оператормен шығарылады және өңделеді. Ішкі сұраудың мәтіні жақшада болуы керек. Ішкі сұраныстағы SELECT мәлімдемесі келесі пішінге ие:

```

SELECT . . .
FROM ...

```

WHERE <салыстырмалы мәні><оператор>(SELECT . . .) Ішкі сұранымдардың екі түрі бар.

1. *Скалярлық ішкі сұрау* бір мәнді қайтарады. Негізінде ол бір мәнді қажет ететін жерде қолданыла алады.
2. *Кестенің ішкі сұрауы мәндер жиынтығын* қайтарады, яғни. Кестенің бір немесе бірнеше бағандарының мәндері бірнеше жолға орналастырылады. Бұл үстелге рұқсат етілген жерде мүмкін.

Сұраулар бойынша бар:

- байланыстырылмаған — жоғары деңгейлі сұрауға сілтемелер болмайды, олар жоғары деңгейлі сұрау үшін бір рет есептеледі;
- байланыстырылған — негізгі сұраудағы өріс мәндеріне тәуелді шарттар бар; жоғарғы деңгейлі сұраудың әр жолына есептеледі.

Ішкі сұраудың негізгі немесе сыртқы синтаксисі бар, сондықтан ішкі сұрау да ішкі сұрауды қамтуы мүмкін. Тағы бір SELECT операторы басқа SELECT операторының корпусына ендірілген сияқты.

Сыртқы SELECT операторы бүкіл операцияның соңғы нәтижесінің мазмұнын анықтау үшін ішкі нұсқадағы нәтижені қолданады.

Ішкі сұраулар салыстыру операторынан кейін бірден орналастырылуы мүмкін (=, <,>, <=,> =, <>)

WHERE тармағы және сыртқы SELECT нұсқасының HAVING тармағы. Сонымен қатар, ішкі SELECT операторлары INSERT, UPDATE және DELETE операторларында (жазбаларды кірістіру, жаңарту және жою) пайдаланылуы мүмкін.

Ішкі сұранымдарға келесі ережелер мен шектеулер қолданылады:

- ORDER BY пайдалана алмайсыз, бірақ бұл тіркесім сыртқы сұрауда болуы мүмкін;
- SELECT сөйлеміндегі тізбеде олардан тұратын жеке бағандар немесе өрнектердің атаулары бар, егер қосымша сөз EXISTS кілт сөзі болса;
- әдепкі бойынша, ішкі сұраудағы баған атаулары FROM сөйлемінде аты көрсетілген кестеге сілтеме жасайды. Дегенмен, сыртқы сұраныстың FROM бөлімінде көрсетілген кестенің бағандарына сілтеме жасауға болады, олар үшін көрсетілген баған атаулары (яғни, кесте) немесе олардың бүркеншік аттары қолданылады;
- егер ішкі сұрау - салыстыру операциясына қатысатын екі операндтардың бірі болса, сұрау осы операцияның оң жағында көрсетілуі керек.

Келесі сұрау өнімнің максималды саны сатылған күндерді қайтарады.

SELECT Саны, Күні FROM
САТЫЛЫМДАР

WHERE Саны = (SELECT MAX(Саны)FROM САТЫЛЫМДАР)

Әлбетте, кірістірілген SELECT мәлімдемесі тізім емес, бір мәнді қайтаруы керек. Ішкі ішкі сұраныс өнімнің ең көп мөлшерін анықтайды. Сыртқы қосалқы сұрауда - тауардың саны максимумға тең болатын күн. Айта кету керек, сөйлемді тікелей қолдануға болмайды

WHERE Саны= Макс (Сан),

WHERE тармақтарында қорыту функцияларына рұқсат берілмегендіктен.

Алдыңғы мысалды қиындатады. Тауардың максималды саны қоймадан жіберілген күнді анықтаңыз және осы өнімді сатып алған әрбір клиент үшін біз қаланы көрсетеміз (12.8-сурет):

Количество	Дата	Клиент	Город
50	02.04.2013	Лесовая В.Н.	Москва
50	02.04.2013	Лесовая В.Н.	Москва
50	02.04.2013	Федорова Д.С.	Санкт-

12.8 сурет. Салынған сұраныстар нәтижелері

SELECT П. Саны, П.Күні, К.Тұтынушы, К.Қала FROM
САТЫЛЫМДАР П, ТҰТЫНУШЫЛАР К WHERE (П.Тұтынушылар =
К.Тұтынушылар)

AND

П.Саны =(SELECT MAX(Саны) FROM САТЫЛЫМДАР)

Алдыңғы мысалмен салыстырғанда, сұрауда SALES кестелері мен CLIENTS ішкі қосылымы бар.

Ішкі сұраныстың тағы бір мысалы - өнімнің жалпы орташа мәнінен асатын және ауытқу мөлшерін анықтайтын сатып алу күндерін анықтау (12.9-сурет):

SELECT Күні, Саны,

Саны-(SELECT AVG(Саны) FROM САТЫЛЫМДАР)

AS ауысу FROM САТЫЛЫМДАР WHERE Саны>

(SELECT AVG(Саны)FROM САТЫЛЫМДАР)

Орташа деңгейден ауытқу мөлшерін есептеу, және мерзімдері туралы ақпаратты іріктеу үшін сыртқы SELECT есепте пайдаланылатын жалпы сату барлық фактілер бойынша тауарлар санының орташа мәні болып табылады сұрау салынған, бұл мысалда, нәтижесі.

Жиі ұсыныстар WHERE немесе HAVINGсалыстырғанда тиіс мән, бір емес, білдіреді, бірақ бірнеше мағынасы - аралық уақытша кестесін қалыптастыру салынған сауалдар.

атап айтқанда, өз кезегінде көптеген қолданылады операциялар негізінде операцияларды сауалдар, жағыңыз:

Дата	Количество	Отклонение
02.04.2013	50	20
02.04.2013	50	20
15.02.2013	40	10
01.02.2013	50	20

12.9 сурет. Қосымшаның нәтижесі

[NOT] IN;
{ ALL | SOME | ANY };
[NOT] EXISTS;

IN операторы белгілі бір мәнді мәндер тізіміне салыстырып, салыстырылған тізімде мән бар ма немесе салыстырылатын мән тексерілген тізімнің элементі емес екенін тексеру үшін пайдаланылады.

Төмендегі мысал өнімнің ең көп лотын сатып алған тұтынушылардың сату тізімін көрсетеді:

```
SELECT П.*  
FROM САТАЛЫМДАР П WHERE П. Тұтынушы IN  
(SELECT П1. Тұтынушы FROM САТЫЛЫМДАР П1  
WHERE П1. Саны =  
(SELECT MAX (П2. Саны ) FROM САТЫЛЫМДАР П2))
```

Бұл сұрақтың логикасын түсіндірейік. Біріншіден, Саны бағандағы ең үлкен мән анықталады:

```
SELECT MAX (П2. Саны ) FROM ПРОДАЖИ П2
```

Келесі сұраудан кейін осы сатып алуды жасаған тұтынушылар анықталады:

```
SELECT П1. Тұтынушы FROM САТЫЛЫМДАР П1  
WHERE П1. Саны =  
(SELECT MAX (П2. Саны) FROM САТЫЛЫМДАР П2)
```

Содан кейін негізгі сұраныс SALES кестесінен табылған клиенттермен IN операторымен, яғни CUSTOMER өрісінің мәні кірістірілген сұрауларда алынған жиынтыққа тиесілі SALES кестесінен алынған деректерді ғана алынған деректер жиынына түсетін барлық жазбаларды таңдайды (12.10-сур.).

Номер	Клиент	Товар	Количество	Дата
451	Федорова Д.С.	конфеты "Коровка"	50	02.04.2013
162	Федорова Д.С.	Халва	10	15.02.2013
200	Лесовая В.Н.	Мармелад "Кроха"	20	07.02.2013
85	Лесовая В.Н.	Мука пшеничная	50	02.04.2013
254	Лесовая В.Н.	Сахар	50	01.02.2013

12.10 сурет. Тауарлардың барынша көп мөлшерін сатып алатын тұтынушылардың сатып алу тізімі

NOT IN пайдаланған кезде салыстыру мәні кірістірілген сұрауларда алынған жиынтыққа кірмейтін жолдарды таңдауға болады.

ANY және ALL деген кілт сөздерді сандардың бір бағанын қайтаратын ішкі есептермен пайдалануға болады.

Сәйкестендірілген мәннің және төменгі сұрау арқылы қайтарылатын мәндердің арақатынасы ALL және SOME (ANY):

- ALL дегеніміз, салыстырмалы мәннің төменгі сұрау арқылы қайтарылған барлық мәндермен сәйкес келетін коэффициенттің мәні болған жағдайда, яғни, төменгі сұраудың нәтижелендірілген бағанындағы барлық мәндер үшін дұрыс екендігін көрсетеді;
- SOME (немесе ANY) - іздеу шарты салыстырылған мәннің төменгі сұрау арқылы қайтарылған кем дегенде бір мәнмен дұрыс қатынасында болғанда, яғни ол төменгі сұраудың нәтижелендірілген бағанында кем дегенде біреуіне қанағаттандырылады. Егер төменгі сұрау нәтижесі бойынша бос мән алынды, онда ALL сөзі үшін салыстыру шарты орындалды деп есептеледі және кілт сөзді ANY - аяқталмаған. SOME кілт сөзі ANY сөзбен синоним болып табылады.

Жоғарыда келтірілген мысалдарды келтірейік. Тауар сатылатын бірліктердің саны орташа мәннен асатын тауарларды сату фактілерінің барлығын анықтаңыз:

```
SELECT * FROM САТЫЛЫМДАР
П1 WHERE П1. Саны > ALL
(SELECT AVG(П2. Саны)
FROM
САТЫЛЫМДАР П2
GROUP BY
П2.Тұтынушы )
```

Және HAVING сөйлемін және ALL кілт сөзін пайдаланудың тағы бір мысалы. Ең көп тауарларды сатып алған клиент туралы ақпаратты шығарып алу керек SELECT K.*

```
FROM
ТҰТЫНУШЫЛАР К
WHERE К.
Тұтынушы =
(SELECT П.Тұтынушы FROM САТЫЛЫМДАР П
GROUP BY П. Тұтынушы HAVING SUM^. Саны) >=
ALL (SELECT SUM (П1. Саны) FROM
САТЫЛЫМДАР П1 GROUP BY П1. Тұтынушы))
```


Сатылған тауарлардың бірліктерінің саны кем дегенде бір тауарды сатудың орташа мәнінен асатын тауарларды сату фактілерінің барлығын көрсетеді.:

```
SELECT * FROM САТЫЛЫМДАР
P1 WHERE P1.саны>SOME
(SELECT AVG(P2.саны)
FROM САТЫЛЫМДАР P2
GROUP BY P2.Тұтынушылар)
```

Құпия сөздер бар және тек қана сауалдар бірге пайдалану үшін арналған EXISTS EMEC. Оларды өңдеудің нәтижесі TRUE немесе FALSE логикалық мәні болып табылады.

Кілт сөз нәтижесі шынайы болса және үстел нәтижесінде сұранысты қайтару кем дегенде бір сызық бар. Нәтижесінде кесте сұранысы бос болса, нәтижесі өңдеу операция FALSE мәні болады бар. NOT EXISTS кілт сөзі EXISTS кілт сөзіне кері қайтарылған өңдеу ережелерін қолданады. EXISTS және NOT EXISTS кілт сөздері нәтижесінде алынған ішкі сұрау кестесінде жолдардың болуы үшін ғана тексерілгендіктен, бұл кестеде бағандардың ерікті саны болуы мүмкін. кілт сөзді EXISTS пайдалана отырып, сіз тауарларды сатып алу үшін кем дегенде бір уақыт бар, барлық клиенттер тізімін жасай аласыз:

```
SELECT *
FROM ТҰТЫНУШЫЛАР К
WHERE EXISTS (SELECT П.*
FROM САТЫЛЫМДАР П
WHERE К. Тұтынушы = П. Тұтынушы)
```

Бір сұранымда ешқандай сатып алулар жасаған клиенттердің тізімін алу үшін сөзді енгізу қажет NOT:

```
SELECT *
FROM ТҰТЫНУШЫЛАР К
WHERE NOT EXISTS (SELECT
П.*
FROM САТЫЛЫМДАР П
WHERE К. Тұтынушылар = П. Тұтынушы)
```

Егер сіз іздестіру шартында сіз ішкі сценарий қайтарылатын кестеден тек жазбаларды таңдағыңыз келсе, анықтасаңыз

тек бір мән, SINGULAR сөйлемі көрсетіледі. Алдыңғы сұрауды сәл өзгертіп, мүлде басқа нәтиже алайық:

```
SELECT *
```

```
FROM
```

```
ТҰТЫНУШЫЛАР
```

```
K WHERE
```

```
SINGULAR
```

```
(SELECT П.САТЫЛЫМДАР FROM САТЫЛЫМДАР П WHERE K.
```

```
Тұтынушы =
```

```
П.Тұтынушы)
```

Сұраудың нәтижесі, яғни, тек бір рет өнімнің тек бір түрін сатып барлық клиенттерінің тізімі, және сату кестеде осы клиент үшін бір ғана жазба бар.

Жоғарыда дерекқор кестесінің ішкі байланыстарын қарастырдық. Сіз қосылуға шартты қанағаттандыруы екі кестеден барлық жолдарды қамтиды келген кезде ішкі байланыс пайдаланылады. Бұл жағдайда, бірінші және екінші кестелер декарттық өнім желілерін, сондай-ақ қосылуға жағдайын сәйкес жазбаларды таңдалған нәтижесінде деректер жиынынан.

Қосылым кестелерінің тағы бір түрі бар - *сыртқы байланыс*. Бұл ерекшелікке сәйкес FROM тармағында анықталған:

```
SELECT ...
```

```
FROM <кесте1><байланыс түрі>JOIN <кесте2>
```

```
ON <байланыс шарттары>
```

Сыртқы байланыс ішкі байланыстың айырмашылығы сол кестелердің біреуінде шебері. Деректер жиынының нәтижесі басқа кестеден жазбалардың бос жиынымен біріктіруге болатын негізгі қосылым кестесінің жазбаларын қамтиды. Кестелердің қайсысы қосылыс түрін анықтайды.

Сол жақ сыртқы біріктіру (LEFT) - бұл бірінші кесте мастер болып табылатын байланыс, FROM сөйлеміндегі қосылым түрінің сол жағына. Тиісінше, екінші кестенің жалпы бағандарында бірдей мәндері жоқ жетекші (бірінші) кестенің жолдары деректердің нәтиже жиынына қосылады.

Оң жақ сыртқы біріктіру (RIGHT) екінші кесте шебер болып табылатын кезде, FROM тармағындағы қосылым түрінің оң жағына байланыс болып табылады. Алынған қатынас дұрыс кестенің барлық жолдарын қамтиды.

Сыртқы байланыс (FULL) - жетекші кестелер барлық кестелер болғанда. Деректер жиынтығы екі кестенің барлық жазбаларын қамтиды. Егер бірінші кестені жазудың қосылым шартын қанағаттандыратын екінші кестенің жазбалары болса, екі кестенің осындай жазбаларын қосудың барлық комбинациясы алынған деректер жинағына қосылады. Әйтпесе, бос жазбаға қосылған бірінші кестенің жазбасы деректердің нәтиже жинағына қосылады.

Екінші жағынан, қосылым жағдайын қанағаттандыратын екінші кестенің жазбалары үшін бірінші кестенің жазбалары бар болса, екі кестенің осындай жазбаларын қосудың қосындысы сұрануға қосылады. Әйтпесе, бос жазбаға қосылған екінші кестенің жазбасы алынған деректер жинағына қосылады.

Мұны мысалмен келтірейік. ТҰТЫНУШЫЛАР және САТЫЛЫМДАР кестелерінің сыртқы сол жаққа қосылуын жүзеге асыратын операторды орындау осындай деректер жиынтығын құруға әкеледі (12.11-сурет):

```
SELECT САТЫЛЫМДАР.*, ТҰТЫНУШЫЛАР.*
FROM ТҰТЫНУШЫЛАР LEFT JOIN
САТЫЛЫМДАР ON
САТЫЛЫМДАР.Тұтынушы =
ТҰТЫНУШЫЛАР.Тұтынушы
```

Нәтижеден көріп отырғандай, ТҰТЫНУШЫЛАР кестесінің кейбір жолдарында қосылым күйін қанағаттандыратын САТЫЛЫМДАР кестесінде жұпталған жазбалар жоқ. Сондықтан осы ТҰТЫНУШЫЛАР кесте деректері бос жазбалармен бірге көрсетіледі.

№	Клиент	Товар	Количество	Дата	Клиент 1	ИНН	Город	Телефон
	<null>	<null>	<null>	<null>	000 Т оризонт"	0255001231521	Москва	4890605
112	000 "Ромашка"	Халва	40	15.02.201	ИП Привалов	02550102566	Санкт-	2560245
	<null>	<null>	<null>	<null>	000 "Перевал."	14588989362	Тверь	221588
	<null>	<null >	<null >	<null >	ИП Иванов Ф.И.	02555800055	Москва	1155488
25	Таран О.С.	Халва	5	04.01.201	Таран О.С.	36022400545	Москва	1215648
144	Таран О.С.	Сосиски молочные	30	01.04.201	Таран О.С.	36022400545	Москва	1215648
45	Федорова Д.С.	Конфеты	50	02.04.201	Федорова Д.С.	50025551005	Санкт-	4449702
162	Федорова Д.С.	Халва	10	15.02.201	Федорова Д.С.	50025551005	Санкт-	4449702
200	Лесовая В.Н.	Мармелад "Кроха"	20	07.02.201	Лесовая В.Н.	21258583218	Москва	3021402
85	Лесовая В.Н.	Мука пшеничная	50	02.04.201	Лесовая В.Н.	21258583218	Москва	3021402
254	Лесовая В.Н.	Сахар	50	01.02.201	Лесовая В.Н.	21258583218	Москва	3021402
	<null>	<null>	<null>	<null>	БМСТ	55587897002	Санкт-	4353822
	<null>	<null>	<null>	<null>	Дремина Е.Е.	02557872105	Москва	6582209
156	ИП Газимова	Конфеты "Мишка"	30	02.04.201	ИП Газимова	02558797876	Москва	6521588
140	ИП Газимова	Конфеты	15	09.01.201	ИП Газимова	02558797876	Москва	6521588

12.11 сурет. Кестелердің сыртқы сол жаққа қосылу нәтижесі ТҰТЫНУШЫЛАР және САТЫЛЫМДАР

Сол сияқты, сыртқы сыртқы біріктіру жүзеге асырылады:

```
SELECT САТЫЛЫМДАР.*, ТҰТЫНУШЫЛАР.*  
FROM ТҰТЫНУШЫЛАР RIGHT JOIN  
САТЫЛЫМДАР ON САТЫЛЫМДАР.
```

Тұтынушылар = ТҰТЫНУШЫ.Тұтынушы
және толық сыртқы байланыс:

```
SELECT САТЫЛЫМДАР.*, ТҰТЫНУШЫЛАР.*  
FROM ТҰТЫНУШЫЛАР FULL JOIN  
САТЫЛЫМДАР ON САТЫЛЫМДАР.
```

Тұтынушы = ТҰТЫНУШЫЛАР. Тұтынушы

Кейде SELELECT операторларының орындалуынан кейін қайтарылған деректердің екі немесе одан да көп нәтиже жиынтығын біріктіру қажет. Мұндай біріктіру UNION операторы арқылы орындалады. Біріктірілетін деректер жиынтығы бірдей құрылымға ие болуы тиіс, яғни бірдей өріс құрамы (тип, өлшем). Егер алынған деректер жинақтарында бірдей жазбалар болса, олар бір жолға біріктірілген жиынтыққа жазылады:

<оператор SELECT>

UNION

<оператор SELECT>

SELECT операторын қолдану арқылы әртүрлі кестелердегі деректерді таңдау үшін өте күрделі шарттар жасай аласыз.

12.4. ДЕРЕКТЕРДІ ӨЗГЕРТУ. INSERT, UPDATE, DELETE ОПЕРАТОРЛАРЫ

Дерекқорды пайдаланушы деректерімен толтыру, бар деректерді өзгерту және жою үшін, DML ішкі кілттің SQL операторларын пайдаланыңыз. Операторлар бұл жасалуы тиіс дәл қалай көрсетпестен, дерекқор деректермен не істеу керектігін сұрайды. INSERT операторы жаңа жолдарды кестелерге немесе дерекқор көріністеріне қосу үшін пайдаланылады. Дерекқор кестелеріндегі деректерді өзгерту үшін UPDATE операторын пайдаланыңыз. Кестелердің жолдарын жою үшін DELETE операторын қолданыңыз.

Барлық іс-шаралар (жүгіру) контексінде дерек өзгеруінің кез келген мәміле бойынша жүзеге асырылады. Ол алдын-ала ала деректер және метадеректер, деректер базасында кез-келген операцияларды орындау кезінде қажетті сипаттамаларды немесе әдепкі операциямен клиент делінген SET операция жүйесі арқылы автоматты түрде қосылатын іске қосылды болады.

SQL тілі жазбалар тобында операцияларды орындауға бағытталған, бірақ кейбір жағдайларда операция жеке жазбада орындалуы мүмкін. Сондықтан, жазбаларды қосу, өзгерту және жою үшін операторлар, әдетте, жазу топтарындағы тиісті әрекеттерді шақырады.

Көріністің негізі болатын тұрақты кестеге немесе кестеге жаңа жолдарды қосу, INSERT мәлімдемесі арқылы жасалады. Оның синтаксисі келесідей:

```
INSERT INTO <НЫСАН>
[(баған1[, баған2 ...] ) ]
{VALUES (<мән>[,< мән2>...)] | <оператор SELECT>}
```

<Нысан> — бұл деректер базасының кестесі. Бағандардың тізімі қосылатын жазбаларда қандай мәндер тағайындалатын бағандарды көрсетеді. Бағандардың тізімі алынып тасталуы мүмкін. Бұл жағдайда объектінің барлық бағандары түсініледі және олар осы нысанда анықталған ретпен.

Мәндердің тізімдерін екі жолмен тағайындауға болады: VALUES сөзінен кейін мәнді нақты көрсете отырып және SELECT операторын пайдаланып мәндер тізімін құрыңыз.

Мәндердің тізімінің анық көрсеткіші жеке жазбаны қосу үшін және пішімі бар:

```
INSERT INTO <нысан>
[(столбец1[, баған2 ...] ) ]
VALUES (<мән> [,< мән 2>...])
```

Мәндер бағандарға бір және екіншісінің тәртіпте тағайындалады: бірінші бағанда бірінші мән, екінші баған екінші мән және т.с.с. Мысалы, кестеде ТАУАРЛАР жаңа жазбаны келесі оператор қосуға болады:

```
INSERT INTO ТАУАРЛАР (Тауар, ЕИ, Құны)
VALUES («Қант», «кг.», 100)
```

ТАУАРДЫ кестенің бағандары кестені жасаған кезде толық құрамда және тәртіпте тізімделгендіктен, оператор жеңілдетілуі мүмкін:

```
INSERT INTO ТАУАРЛАР
VALUES («Қант», «кг.», 100)
```

INSERT операторының екінші пішіні келесі пішім:

```
INSERT INTO <нысан>  
[(баған1[, баған2 ...] ) ]  
<оператор SELECT>
```

Бағандарға тағайындалған мәндер - SELECT операторы арқылы қайтарылған мәндер. SELECT сөйлемі әдетте жазбалардың жиынын қайтаратындықтан, осы пішіндегі INSERT мәлімдемесі нысанға жаңа жазбалардың санын қосады.

Мысалы, кез-келген дерекқорда САТЫЛЫМДАП кестесімен ұқсас өрістердің құрамы мен тәртібіне сәйкес САТЫЛЫМДАП кестесі анықталған:

```
CREATE TABLE САТЫЛЫМДАП (Нөмір INTEGER NOT  
NULL,  
Тұтынушы VARCHAR(20) NOT NULL COLLATE PXW_CYRL,  
Тауар VARCHAR(20) NOT NULL COLLATE PXW_CYRL, Саны  
INTEGER NOT NULL,  
Күні DATE NOT NULL,  
PRIMARY KEY (Нөмір)
```

Бұл кестеден алынған деректер күнделікті кез келген географиялық қашықтағы дерекқорға, мысалы, САТЫЛЫМДАП-ға қосылсын делік. Содан кейін САТЫЛЫМДАП кестесіне САТЫЛЫМ кестесіндегі жазбаларды күнделікті түсіру оператор арқылы жүзеге асырылады:

```
INSERT INTO САТЫЛЫМДАП_ГЛ SELECT *  
FROM САТЫЛЫМДАП  
WHERE Күні = < күні>,  
мұндағы <күні> — ағымдағы күннің мағынасы.
```

Қалыпты кестедегі немесе өзгермейтін көрініс үшін негіздегі кестеде бар жолдардың бағандарындағы деректерді өзгерту үшін UPDATE операторын пайдаланыңыз. Бір қоңырауға арналған оператор барлық жолдардағы немесе кестедегі немесе көріністегі жолдардың бөліктеріндегі деректерді өзгертуге мүмкіндік береді. Оның синтаксисі келесі пішіммен ұсынылған:

```
UPDATE <нысан>  
SET баған1=<мәні1> [, баған2=<мәні2>...] [WHERE <шарты>]
```

Аталған бағандардың әрқайсысын өзгерткенде, сәйкес мән тағайындалады. SELECT нұсқаулығында көрсетілгендей шартты қанағаттандыратын барлық жазбалар үшін өзгертулер жасалады. SET сөйлемі орындалатын өзгертулердің тізімін: өзгертуге болатын айнымалының аты және бағаның жаңа мәні теңдік белгісінен кейін анықталады. Мән INSERT операторы сияқты, күрделі күрделі өрнек болуы мүмкін. Үтірлер бойынша келесі SET тармақтары кестедегі басқа бағандардың мәндерін тізімдеуі мүмкін. Оператордағы WHERE тармағы қолданыстағы деректерді өзгерту әрекеті қолданылатын жолдардың жиынын анықтайды. Егер бұл сөйлем көрсетілмесе, кестенің барлық жолдары өзгереді:

```
UPDATE ПРОДАЖИ SET Құны=Құны*2 WHERE Тауар = «Ұн»
```

Өнім кестесіндегі оператор қызметінің нәтижесі ретінде «Баға» өрісіндегі «Тауар» өрісінің мәні «Ұн» болып табылатын барлық жазбаларда екі еселенген болады.

DELETE операторы нысаннан жазбалар тобын жою үшін пайдаланылады. Белгілі бір жағдайда тек бір ғана жазбаны жоюға болады. DELETE мәлімдемесі пішімі:

```
DELETE FROM <нысан>  
[WHERE <шарты>]
```

Іздеу шартына сәйкес келетін барлық жазбалар жойылады. WHERE сөйлемін жойсаңыз, барлық жазбалар нысаннан жойылады.

Төменде САТЫЛЫМДАР кесте қантты сататын фактілердің барлық жазбаларын жойатын мысал болып табылады:

```
DELETE САТЫЛЫМДАРWHERE Тауар= «Қант»
```

12.5. САҚТАЛАТЫН ІС ШАРАЛАР МЕН ТРИГГЕРЛЕР

12.5.1. Сақталатын іс шаралар мен тиггерлердің тілі

Сақталған процедуралар мен триггерлер - дерекқордың метадеректер аймағында сақталған бағдарламалар. Олар сервер жағында орындалады, ол көптеген жағдайларда үнемдеуге мүмкіндік

Сақталған процедураларға, триггерлерге және клиенттік қосымшаларға жалпыға қол жетімділікке қол жеткізуге болады. Триггерге тікелей қол жеткізуге болмайды - кесте (деректерді өзгерту) немесе дерекқор оқиғасы үшін белгілі бір оқиға болған кезде олар автоматты түрде шақырылады. Әрбір іс-шара кесте үшін екі кезеңі бар - тиісті оқиғаға (дейін) дейін және іс-шараға (кейін) пайда болған соң.

SQL тілінің кеңейтімі сақталатын процедуралар мен триггерлердегі деректерді өңдеуге арналған алгоритмдерді сипаттау үшін қолданылады. Бұл кеңейтім процедуралық SQL (PSQL) немесе сақталатын процедуралар тілі мен триггерлер деп аталады. Тіл әдеттегі тағайындалу операторларын, филиал операторларын және циклдік операторларды қамтиды. Триггерлерде нақты мәтінмен айналымыларын пайдалануға болады.

Сақталған рәсімдер мен триггерлердің тілі классикалық бағдарламалау тілдерінің барлық негізгі құрылымдарын қамтиды. Сонымен қатар, ол дерекқор кестелерінен (INSERT, UPDATE, DELETE және SELECT) бар деректерді қосу, өзгерту, жою және алу үшін бірнеше өзгертілген операторларды қамтиды. Сақталған процедуралардың тілінде ешқандай метадеректер өзгермейді.

Сақталған процедуралар мен триггерлерде белгілі бір оқиғалар (оқиғалар) туралы клиенттерге хабарлауға, кедендік ерекшеліктерді шығаруға мүмкіндік беретін құралдарды пайдалануға болады. Деректер қорын қосу әрекеттерін орындауға жол берілмейді, транзакцияны бастауға, транзакцияларды жасауға, құтқару жолдарын жасауға, құтқарушыларға оралу, транзакцияларды растауға немесе жоюға болмайды. Сақталған процедура мен триггер сақталатын процедура тікелей анықталған транзакция контекстінде орындалады немесе деректерді өңдеу операциясы дерекқорда орындалды, нәтижесінде триггер автоматты түрде іске қосылды. Егер дерекқор дерекқордан қосылып, дерекқордан ажыратылған кезде шақырылатын болса, онда ол үшін әдепкі транзакция басталады.

Сақталған процедуралар мен триггерлер жасау синтаксисінде тақырып пен денені таңдауға болады. Тақырыпта бағдарлама нысанының атауы, жергілікті айналымылардың сипаттамасы бар. Триггерлер үшін тақырып дерекқор оқиғасын және триггер автоматты түрде іске қосылатын кезеңді анықтайды. Сақталған рәсімнің тақырыбында кіріс және шығыс параметрлерін көрсетуге болады. Сақталған процедураның немесе триггердің құралы - бұл блок

бағдарлама орындалатын іс-әрекеттердің сипаттамасын қамтитын оператор. Операторлар блогы оператордың BEGIN және END жақшаларынан тұрады. Бағдарламалардың өзінде, дәйекті және кірістірілген блоктың еркін саны болуы мүмкін.

Барлық іс-шаралар үлгі негізделген, сақталатын процедуралар мен триггерлер деректер өңдеу және модификациялау ирек жақша BEGIN және END болып есептілігін блогында орындалады.

Триггерлер және сценарий шеңберінде сақталатын процедуралар жазған кезде, (SQL стандарттары үшін, нүктелі үтірмен) операторларының символы аяқтау пайдалануға қатысты белгісіздік болдырмау үшін, дерекқорға қажет бағдарламалық қамтамасыз ету нысандарын құру қатаң SQL мәлімдеме емес, сөйлейтін, оператор белгіленген мерзім, қолданылады. Бұл псевдо-оператормен триггер немесе сақталған процедура жасалмай тұрып, триггер немесе сақталған процедураның соңы болып табылатын символ пайда болады. нүктелі үтірмен - сол оператор жиынтығын пайдаланып бағдарлама нысанның мәтінін сипаттайтын кейін МЕРЗІМДІ терминатор мәні қалыпты нұсқасы қайтарылады.

Мысалы, триггер жасаған кезде, сіз төмендегі мәлімдемелерді орындауыңыз керек:

```
SET TERM ';
```

```
/* Кестенің бастапқы кілті мәнін қалыптастыру ТАУАРЛАР */
```

```
CREATE TRIGGER TBI_STAFF FOR ТОВАРЫ BEFORE
```

```
INSERT . . . [Тиггер мәтіні]
```

```
END ';
```

Блокта операторлар дәйекті түрде орындалады. SQL кеңейтімдерінде дерекқорда деректерге қол жеткізу операторлары (деректерді іріктеу, деректерді таңдау, өзгерістер, жою).

Бос орынға рұқсат етілген мәтіндегі кез келген орын, / * және * / таңбалар арасында орналасқан түсініктемелерді орналастыруға болады. Мұндай түсініктемелердің біреуі кез келген жолды қабылдай алады. Түсініктемелердің тағы бір түрі бар: қатарынан екі таңба минус (-). Бұл жағдайда түсініктеме мәтіні ағымдағы жолдың соңына дейін жалғасады.

Бір жергілікті айнымалыны сипаттау үшін DECLARE VARIABLE операторын пайдаланыңыз. Оның оңайлатылған синтаксисі келесі түрде ұсынылуы мүмкін:

```
DECLARE [VARIABLE] <жергілікті ауысым атауы > {  
<деректер түрі>  
| <домен атауы>  
| TYPE OF <домен атауы>  
}
```

Бір мәлімдемеде тек бір жергілікті айнымалы мәнді жариялай алады. Іске қосу және сақтау процедурасында әрбір айнымалы үшін жеке DECLARE VARIABLE операторын пайдаланып, жергілікті айнымалы мәндердің ерікті санын жариялай алады. Жергілікті айнымалы атау жергілікті бағдарлама нысанында сақталатын іс рәсімінің кіріс және шығыс параметрлерінің жергілікті айнымалы атауларының арасында бірегей болуы керек. Деректер түрі SQL түрінде пайдаланылатын деректердің кез келген түрі болуы мүмкін. Деректер түрінің орнына бұрын құрылған доменнің атауын көрсетуге болады.

12.5.2. Тиггерлермен жұмыс

Ұшыру мета-деректерді дерекқорында сақталады және сервер жағында орындалатын бағдарлама. Триггерге тікелей кіру мүмкін емес. Бір нақты кестеге (көрініске) қатысты бір немесе бірнеше оқиғалар немесе дерекқор оқиғаларының бірі болғанда, ол автоматты түрде шақырылады. Кесте оқиғасы пайда болған кезде іске қосылатын кесте немесе көрініс, сол кесте немесе көрініс үшін бір немесе бірнеше оқиғалар (деректерді қосу, өзгерту немесе жою) және осындай оқиғаның дәл бір кезеңі (оқиғадан бұрын немесе кейін) байланысты іске қосылады. Триггер транзакция контекстінде тиісті оқиғаға себеп болған бағдарлама аясында орындалады. Ерекшелік дерекқор оқиғаларына жауап беретін триггерлер болып табылады. Кейбіреулер үшін әдепкі транзакция басталады.

Кестеге арналған «оқиға фазасы» қатынасының алты нұсқасы бар (өкілдік):

- 1) Жаңа жолды қосқанға дейін (BEFORE INSERT);
- 2) Жаңа жолды қосқаннан соң (AFTER INSERT);
- 3) Жолды өзгерткенге дейін (BEFORE UPDATE);
- 4) Жолды өзгерткен соң (AFTER UPDATE);
- 5) Жолды өшіруге дейін (BEFORE DELETE);
- 6) Жолды өшірген соң (AFTER DELETE).

Дерекқор оқиғаларына байланысты триггер келесі оқиғаларда шақырылуы мүмкін:

- 1) дерекқорға (CONNECT) қосылса, триггер орындалмас бұрын, әдепкі бойынша операция автоматты түрде басталады;
- 2) дерекқордан (DISCONNECT) ажыратылса, триггер орындалмас бұрын әдепкі транзакция басталады;
- 3) транзакция басталған кезде (TRANSACTION START), триггер осы мәміленің контексінде орындалады;
- 4) мәміле расталған кезде (TRANSACTION COMMIT), триггер осы мәміленің контексінде орындалады;
- 5) мәміле бұзылған кезде (TRANSACTION ROLLBACK), триггер операция мәмілесі бойынша орындалады.

Бір кесте (көрініс) үшін, бір фаза және бір оқиға үшін, сондай-ақ бір фаза және бірнеше оқиғалар үшін автоматты түрде шақырылатын триггерлер жасауыңызға болады.

Егер бір кестеге (көрініске), бір оқиғаның және бір фазаның бірнеше триггерлері бар болса, оларды орындау тізбегін көрсете аласыз, осы тізбектегі триггер позициясын көрсете аласыз.

Триггер жасау үшін, синтаксисі төменде көрсетілген CREATE TRIGGER операторын пайдаланыңыз:

```
CREATE TRIGGER <триггер атауы>
[ACTIVE | INACTIVE]
{ ON <деректер қорының әрекеті>
| FOR {<кесте атауы> | <көрініс атауы>}
{BEFORE | AFTER}
<кесте әрекеті (көрініс)>
[OR <кесте әрекеті (көрініс)>] . . .
| {BEFORE | AFTER}
< кесте әрекеті (көрініс)>
[OR < кесте әрекеті (көрініс)>] . . .
ON {<кесте атауы> | <көрініс атауы>}
}
[POSITION <бүтін>]
AS <триггер ағзасы>;
< кесте әрекеті (көрініс)> ::=
{INSERT | UPDATE | DELETE}
<деректер қорының әрекеті> ::=
{ CONNECT |
DISCONNECT |
TRANSACTION START
```

| TRANSACTION COMMIT | TRANSACTION ROLLBACK }

Триггер белсенді (ACTIVE) немесе белсенді емес (IN-ACTIVE) болуы мүмкін. Егер триггер белсенді (әдепкі мән) болса, ол кестенің немесе дерекқордың сәйкес оқиғасы (лері) болғанда автоматты түрде шақырылады. Триггер белсенді емес болса, триггер пайда болмайды.

Триггердің дене құрылымы:

```
[<жергілікті үздіксіз хабарламасы>]
BEGIN
  <оператор>
  [<оператор> ... ]
END
```

Триггері органға жергілікті айнымалылар сипаттамасы кейін ирек жақша Қоса беріліп отырған есептілігіне BEGIN және END блок болуы тиіс.

Триггерлер үшін OLD және NEW нақты контекстік айнымалылар бар. Осы кілт сөздердің неғұрлым дұрыс атауы - баған атауы префикс. триггерлер және өзгерту (Баған атауының алдында жаңа және нүктесін орналасқан) кейін (осы кілт сөз ескі және нүкте Баған атауының алдында қойылады үшін) клиент оны өзгерту үшін кесте кез келген бағанының (қарау) мәнін кіре аласыз.

Триггерлер барлық түрлері үшін айнымалы контекстен OLD (баған префикс) айнымалы тек оқу болып табылады. Бұл қарамастан фазалық оқиғалардың, деректерді қосу кезінде туындаған триггерлер қол жетімді.

(Кейін) фаза оқиғалар үшін триггерлер жаңа контекст айнымалылар, сондай-ақ айнымалы тек оқу болып табылады. Деректерді жою оқиғасы үшін триггерлерде қол жетімді емес.

OLD мәні. Бағанның атауы - мүмкін болатын өзгерістерді жасамас бұрын жасалған бағанның күйіне сілтеме жасауға мүмкіндік береді.

Мән NEW. Бағанның атауы - мүмкін өзгерістерді жасағаннан кейін бағанның жағдайына сілтеме жасауға мүмкіндік береді.

Мысалы:

```
CREATE TRIGGER BU_TOVARY FOR
TOVARY ACTIVE
BEFORE UPDATE AS
```

```
BEGIN
IF (OLD.TOVAR<>NEW.TOVAR)
THEN UPDATE PRODAJA SET
TOVAR=NEW.TOVAR WHERE
TOVAR=OLD.TOVAR;
END
```

Осы мысалда триггер TOVAR бағанының TOVAR кестесінің жазбасындағы өзгерген болса, PRODAJA кестесіне сәйкес өзгертулер жасайды.

Бірнеше кесте (көрініс) оқиғалары немесе бірнеше дерекқор оқиғалары үшін бір триггер жасай алмайды.

Қолданыстағы триггер оператор ALTER TRIGGER пайдаланып тақырыбын және / немесе орган өзгерту үшін.

Оператор өз қызметін мемлекеттік (белсенді / белсенді емес), кестенің оқиғаны (лар) (өнімділігі) мен іс-шараның кезеңін, триггер орнын өзгерту және орындалатын іс-әрекеттерді іске қосу үшін шүріппесін өзгертуге болады.

Бар триггерлерді жою үшін операторды пайдаланады.

```
DROP TRIGGER <триггер атауы>
```

Трафиктер дерекқор деректерін өзгерткенде әртүрлі әрекеттерді орындау кезінде пайдалы құрал болып табылады. негізгі мақсаты деректер өзгерістерге қатысты хабарларды беру жасанды бастапқы кілт мәндері, дерекқор ақпаратты қалыптастыру автоматты орындау функциясын іске қосу үшін кейбір дерекқор ақпарат (өзгерту) қосқанда, деректер тұтастығын және декларативтік қолдау үшін нақты әдістер белгілі бір әрекеттерді орындау. Мәміленің дерекқор қосылымы, деректер базасынан ажыратып, іске қосу, растау немесе жою - іс-шара тұтас ретінде деректер базасымен байланысты кезінде триггерлер, сондай-ақ пайдалануға болады.

12.5.3. Сақталатын іс әрекеттермен жұмыс

Сақталған процедура, сондай-ақ триггер - дерекқордың метадеректер аймағында сақталған және сервер жағында іске қосылған бағдарлама. Триггерден айырмашылығы, сақталатын процедуралар, сақталған процедуралар, триггерлер және клиент бағдарламалары сақталған процедураға қол жеткізе алады. Реквѐсия ықсат етіледі -

рәсім өздігінен қолданылуы мүмкін. Сақталған процедуралар шалу бағдарламалары сияқты бір транзакция контексінде орындалады.

Сақталған процедуралардың екі түрі бар - орындалатын сақталатын процедуралар және сақталған процедуралар болып саналады.

Орындалатын сақталатын процедуралар дерекқорда деректерді өңдеуді немесе дерекқормен байланыстырылмайды. Бұл процедуралар енгізу параметрлерін ала алады және шығыс параметрлерін қайтарады. Орындалатын сақталатын процедураларға қол жеткізу SQL EXECUTE PROCEDURE операторын орындау кезінде орындалады.

Сақталған іріктеу процедуралары, әдетте, қабылданған жолдардың ерікті санын қайтара отырып, дерекқордан деректерді алуды жүзеге асырады. Таңдау процедуралары енгізу параметрлерін де ала алады. Әрбір келесі оқу сызығының мәні шығыс параметрлеріндегі қоңырау бағдарламасына қайтарылады. Бұл процедураның орындалуын уақытша тоқтатып, таңдалған деректерді қоңырау шалу бағдарламасына сақталған процедураға жіберу үшін SUSPEND операторын пайдаланыңыз. Сақталған іріктеу процедурасына қол жеткізу SELECT нұсқауы арқылы жүзеге асырылады.

Синтаксистік түрде, сақталатын процедураны сақтайтын процедураны жасау ешқандай айырмашылығы жоқ. Сақталған рәсімді жасау үшін, синтаксисі төменде көрсетілген CREATE PROCEDURE операторын пайдаланылады:

```
CREATE PROCEDURE <сақталатын шаралар атауы>
[(<кіріс параметрлер тізімі>)]
[RETURNS (<шығыс параметрлер тізімі>)]
AS
[<ауыспалы хабарламалар тізімі>]
BEGIN < операторлар блогы>END
```

Сақталған процедура шақырылатын бағдарламадан енгізу параметрлерін қабылдай алады. Параметрлер мәнмен қабылданады, яғни кіріс параметрлерінің мәндеріндегі кез келген өзгерістер шақырылған бағдарламада осы параметрлердің мәндеріне әсер етпейді. Енгізу параметрлерін әдепкі мән берілуі мүмкін. Әдепкі мәндер орнатылатын параметрлер тізімнің ең соңында болуы керек. Егер кіріс параметрі DEFAULT сөйлемінде әдепкі мән берілген доменге негізделген болса, жаңа әдепкі мән домен сипаттамасында көрсетілгенді қайта анықтайды.

Сақталған процедура шығыс параметрлерінің ерікті санын шақыру бағдарламасына қайтара алады. Параметр үшін домен атауын көрсетсеңіз, процедураның жергілікті айнымалысы болса, сол доменнің барлық сипаттамалары көшіріледі. Сақталған процедураның денесінде жергілікті айнымалылардың ерікті саны сипатталуы мүмкін.

Сақталған рәсімнің енгізу параметрлері тізімі сақталған процедураның атауынан кейін жазылады және жақшаға салынған:

<Кіріс параметрлері тізімі> :: = (<параметр сипаттамасы> [, <параметр сипаттамасы] ...)

Шығу параметрлері RETURNS ішінде сипатталған:

RETURNS (<шығыс параметрлерінің тізімі>)

Шығу параметрінің сипаттамасы кіріс параметрінің сипаттамасына сәйкес келеді.

Триггерлер мен рәсімдердің жергілікті параметрлері, сақталатын процедураларда ғана қолданылатын енгізу және шығару параметрлері *ішкі айнымалылар* болып табылады. Ішкі айнымалылардың атаулары пайдаланылатын дерекқор кестелерінің баған атаулары сияқты болуы мүмкін. Бұл атаулардың белгісіздігіне себеп болмайды. SELECT ішкі айнымалылар пайдаланған кезде кестеде бағандардың атаулары оларды шатастырмау ретінде емес, сондықтан, INSERT есептілігі UPDATE немесе ішкі айнымалылардың аттары DELETE әрқашан, нүкте алдында болуы тиіс. Барлық басқа жағдайларда, кез келген басқа да операторлар ішкі айнымалылар аттары қос нүктесіз әдеттегі жолмен жазылған.

Ішкі айнымалының мәнін тағайындау синтаксисі келесідей:

<айнымалы атауы> = <мағынасы>;

Өрнектің кез келген дұрыс SQL өрнегі болуы мүмкін.

EXIT мәлімдемесі триггердің кез-келген нүктесінен немесе сақталған процедураны түпкілікті END сөзіне өтуге мүмкіндік береді, яғни бағдарламаның орындалуын тоқтату:

EXIT [<белгі>]

SUSPEND мәлімдемесі сақталатын процедураның орындалуын уақытша тоқтатады (жиі кестенің көптеген жолдарын таңдайтын SELECT нұсқауы бар рәсім, мұндай рәсімге қол жеткізу ақ орындалады

оператор SELECT) және шығыс мәндерін қоңырау бағдарламасына жібереді.

Трафиктер мен сақталған процедуралар операторды пайдаланған кезде сақталатын процедураларды шақыруы мүмкін EXECUTE PROCEDURE:

```
EXECUTE PROCEDURE <шара атауы>
```

```
[(<параметр> [, <параметр>] . . . ) ]
```

```
[RETURNING_VALUES (<параметр>[,<параметр>]...)
```

```
|<параметр>[,<параметр >...]]
```

FOR SELECT DO операторы төмендегідей форматтары бар:

```
FOR
```

```
<оператор SELECT>
```

```
DO
```

```
<оператор>
```

FOR SELECT DO сөзінің алгоритмі келесідей. SELECT операторы орындалады және алынған нәтижелер жиынының әр жолы үшін DO сөзі орындалатын операнд орындалады. Бұл оператор жиі SUSPEND болып табылады, бұл шығыс параметрлерін шақыру қосымшасына қайтаруға әкеледі.

Бар сақталған процедураны өзгерту үшін ALTER PROCEDURE операторын қолданыңыз. Бар сақталған процедураны жою үшін DROP PROCEDURE операторын қолданыңыз. Сақталған процедураларды жасаудың кейбір мысалдары берілген.

Келесі рәсім тауар_кіріс параметрінің мазмұнымен анықталған белгілі бір өнімді сатуға арналған барлық фактілерді береді.

```
CREATE PROCEDURE PRODAJA_TOVARA (Товар_вход VARCHAR (20) )
```

```
RETURNS (күні_шығыс DATE, Тұтынушы_шығыс VARCHAR (2 0) , саны_шығыс INTEGER) AS
```

```
BEGIN
```

```
FOR SELECT Күні, Тұтынушы, Саны FROM Сатылымдар
```

```
WHERE Тауар =:Тауар_кіріс
```

```
INTO :Күні_шығыс, :Тұтынушы_шығыс, :Саны_шығыс DO SUSPEND;
```

```
END
```

Біріншіден, SELECT мәлімдемесі орындалады, ол сату күнін, сатып алушының атын және шығыс сомасын қайтаратын

Тауарлар үлгілері Тауар_шығыс параметрінің кіріс параметріндегі мәнмен бірдей мәнді қамтитын әр енгізу үшін тауарлар Көрсетілген мәндер шығыс параметрлеріне жазылады (Күні_шығыс, тұтынушы_шығыс, Номер_шығыс). Нәтижелердің әрбір жазбасын жазғаннан кейін, SUSPEND мәлімдемесі орындалады. Шығарылатын параметр параметрлерін шақыру бағдарламасына қайтарады және шығыс параметрлерінің келесі бөлігін шақыру бағдарламасынан сұрағанға дейін рәсімнің орындалуын тоқтатады. ол шақырушы қосымшаға кіру параметрлерін бірнеше мәндерді қайтаруға болады, өйткені бұл процедура таңдау рәсім болып табылады.

Тағы бір процедура Тұтынушы_кіріс тауарларды сатып алған барлық клиенттердің атауын осы өнімнің орташа сатып алу мөлшерінен асатын соманы қайтарады. Клиенттік атау клиенттің атауының орнына бос мән болса, «Деректер көрсетілмейді».

```
CREATE PROCEDURE Тұтынушы_тізім (Тауар_кіріс VAR-  
CHAR(20))
```

```
RETURNS (Тұтынушы_шығыс VARCHAR(20)) AS
```

```
DECLARE VARIABLE Орта_саны INTEGER;
```

```
BEGIN
```

```
SELECT AVG(саны)
```

```
FROM Сатылымдар
```

```
WHERE Тауар =:Тауар_кіріс
```

```
INTO :Орташа_саны;
```

```
FOR
```

```
SELECT
```

```
Тұтынушылар
```

```
FROM
```

```
Сатылымдар
```

```
WHERE Саны >: Орташа_саны INTO
```

```
:Тұтынушы_шығыс DO
```

```
BEGIN
```

```
IF (: Тұтынушы_шығыс IS NULL) THEN
```

```
Тұтынушы_шығыс = «деректер жоқ»;
```

```
SUSPEND;
```

```
END
```

```
END
```

Сақталған процедуралардың артықшылығы, олар бірінші кезекте серверлік жағынан орындалады, бұл көптеген жағдайларда желілік трафикті айтарлықтай төмендетеді; Екіншіден, бір рет жазылған және жақсы сақталған сақталған процедураны пайдалануға болады

Көптеген бағдарламалар - сақталатын процедуралар, триггерлер, клиент бағдарламалары.

Сақталған іріктеу процедурасы, әдетте, дерекқордан жеткілікті үлкен деректерді таңдау үшін пайдаланылады. Осындай жағдайларда деректерді іріктеу алгоритмі өте күрделі. Бұл рәсім, әдетте, SELECT мәлімдемесінің декларативті құралы тиісті деректерді кестелерден немесе көріністерден алудың барлық әрекеттерін орындау үшін жеткіліксіз болғанда қолданылады. Бұл процедура кіріс параметрлерін және қайтару параметрлерін қайтара алады.

Сақталған процедуралар қосымшалар мен дерекқорлардың тұтастығын жақсартады, ұжымдық операциялар мен есептеулердің өзектілігіне кепілдік береді. Осындай процедуралардың, сондай-ақ қауіпсіздіктің жақсаруы (деректерге тікелей қол жетімділік жоқ).

Алайда, қолданбалы логикамен сақталатын процедураларды шамадан тыс жүктеу серверді жүктеуі мүмкін екендігін есте сақтаңыз, бұл өнімділіктің жоғалуына әкеледі. Бұл проблема ірі ақпараттық жүйелерді дамытуда, әсіресе, көптеген клиенттер бір мезгілде серверге қол жеткізе алады. Сондықтан, көптеген жағдайларда, ымыралы шешімдер қабылдау қажет: серверлік жағынан қосымшалар логикасының бір бөлігін, ал кейбіреулері клиенттің тарапынан (супер логикасы бар жүйелер).

12.6. ИНДЕКСТЕРМЕН ЖҰМЫС

Индекс — Бұл нақты кестенің көрсетілген бағандарының мәндерін және мән деректерін қамтитын кестенің жолдарына сілтемелерді қамтитын дерекқор нысаны.

Индексі Осы кестедегі деректерді шығарып алу процесін жеделдету, және кейде деректер тапсырыс жеделдету көптеген жағдайларда беретін, белгілі бір кестеге пайдаланушы немесе жүйемен құрылған SELECT есепте ORDER BY құрылымы негізінде пайдаланушының өтініші бойынша қабылдады. Әрбір индекс жолында индексті және көрсеткішті сол баған мәндері бар кестедегі жолға жасайтын бағандардың мәні бар.

Индекстер болған жағдайда, көптеген жағдайларда деректерді индекстеудің болмауына қарағанда тезірек жасауға болады, себебі индекстегі мәндер реттеледі және индекстің өзі аз болады. мысал бағандар бағандар үшін, атап айтқанда, екі мән, бар, осы үшін құндылықтарды саны аз болуы бағандардың көрсеткіштері жасау емес,

еркек немесе әйел - баған TRUE және FALSE, немесе жеке тұлғаның жыныстық қатынасқа орнату жағдайда ғана болуы мүмкін маңызы логикалық деректер түрі. Мұндай индекстері ғана сыртқы жадында орын ала отырып, және операцияларды Сынама мен деректер сұрыптау үшін орындау кез келген пайданы қамтамасыз етпейді.

Бастапқы кілт, бірегей кілт және сыртқы кілт, ол автоматты түрде индекстер салады.

Маңызды ереже: Сіз деректерді іріктеу тырысқанда, осы дерекқор серверіне әкелуі, автоматты түрде бастапқы жүйенің қалыптасады индексі, бірегей немесе шетелдік кілтке сәйкес келетін құрылымы мен тәртібі туралы индекс жасау мүмкін емес.

Индекс бірегей ретінде жасалуы мүмкін (UNIQUE кілт сөзі). Бұл жағдайда, кесте екі түрлі жолдар болуына жол берілмейді бірдей баған мәні, бірегей индексі бір бөлігін бар.

Немесе осы мәндерді (кемуі) кемуі - индексі баған құндылықтарды бойынша сұрыпталған реті болады, оның құрамы (әдепкі көтерілмес) енгізілген. Кез келген уақытта, дерекқор индексі жұмыс істеу т (белсенді) белсенді жасалуы мүмкін. Яғни бірден индексі көрініс индексі, немесе енжар (әрекетсіз) құрайды Кесте бағандары өзгерістер Барлық кезде желілерінде ешқандай өзгеріс Тиісті кесте индекс мазмұнын қозғамайды.

Бар дерекқор кестесінің индексін жасау үшін, операторды пайдаланыңыз:

```
CREATE [UNIQUE] [ASC[ENDING] | DESC[ENDING]]  
INDEX <индекс атауы>ON <кесте>  
(<баған>[, <баған>] . . . )
```

Индекс атауы барлық дерекқор индекстерінің аттары арасында, сондай-ақ кесте деңгейіндегі шектеулер мен кесте деңгейіндегі шектеулер арасында бірегей болуы керек. Бастапқы, бірегей немесе шетелдік кілт шектеулерін көрсетіп жатқанда, CONSTRAINT тармағындағы шектеу атауын көрсетіңіз, жүйе сол атпен индексті құрастырады.

UNIQUE кілт сөзі бірегей индексті құруды анықтайды, бұл индекстің барлық бағандарының бірдей мәндері бар индексте екі жол бола алмайтынын көрсетеді. Бірегей индекс бөлігі болып табылатын бағандарда бос NULL мәні болмайды.

ASCENDING кілт сөзі (ASC қысқартылған нұсқасы) әріпсандарды құрайтын бағандардың мәндерін ұлғайту арқылы индекстік жазбаларға тапсырыс берілетіндігін білдіреді. Бұл параметр әдепкі бойынша қабылданады.

DESCENDING кілт сөзі (DESC үшін қысқа) индекстік жазбалар индексі бағандарының мәндерін азайту арқылы реттелгенін білдіреді.

Индекс BLOB деректер түріне ие бағандарды және массивтер болып табылатын кез-келген деректер түрінің бағандарын қамтуы мүмкін емес.

Мысалы. Егер SALES кестесіне тауарлардың атауын сақтайтын баған бойынша бағаларды көтеру және азайту қажет болса, төменде келтірілген мәлімдемелерді орындау арқылы екі индекс жасаңыз:

```
CREATE ASCENDING INDEX I_ASC ON САТЫЛЫМДАП (Тауарлар)  
CREATE DESCENDING INDEX I_DESC ON САТЫЛЫМДАП  
(Тауарлар)
```

Бастапқы құру кезінде индекс әдепкі бойынша белсенді болады - кестеде жаңадан қосылған барлық жолдар немесе негізгі кестенің индекстелген бағандарында жасалған өзгерістер индекс күйінде дереу көрсетіледі.

Кейбір жағдайларда индексті «уақытша өшіріп», оны белсенді емес етіп жасау пайдалы болуы мүмкін. Бұл кесте арқылы топтық операциялар деп аталатын кезде уақытты үнемдеуге болады, егер жолдан үлкен саны жолдан файлға жазылады, онда индекс жасалады немесе кесте өзгертілсе немесе жолдың үлкен саны кестеден жойылса. Бұл әрекетті бастау алдында индекс белсенді емес (INACTIVE) күйіне ауысады және операция аяқталғаннан кейін - белсенді (ACTIVE) күйіне қайта оралады. белсендірілген кезде осы жағдайда, индексі индексінің толық қалпына келтіріп жасады. жаңадан енгізілген барлық, өзгертілген немесе жойылған жолдар индексі жаңа мемлекет енгізілетін болады.

Индекс жағдайының өзгеруі оператордың көмегімен жүзеге асырылады:

```
ALTER INDEX <индекс атауы> {ACTIVE | INACTIVE}
```

ACTIVE кілт сөзі белсенді емес индексті белсенді күйге орнатады. INACTIVE кілт сөзі индекс белсенді емес екенін білдіреді. Индекс белсенді емес күйден белсенді күйге көшірілгеннен кейін, жүйе бүкіл индексті толығымен қайта жасайды.

Бұл оператор индекс құрылымын немесе оның тәртібін өзгерту үшін пайдаланылмайды. Егер сіз индекс құрылымына өзгертулер енгізуіңіз немесе тапсырысыңызды өзгертуіңіз қажет болса, бар индексті жоюыңыз керек (келесі бөлімді қараңыз), содан кейін сол атаумен және қажетті сипаттамалары бар индексті жасаңыз.

Пайдаланушы жасаған индексті жою үшін операторды пайдаланыңыз

DROP INDEX <индекс атауы>

Сондықтан негізгі, бірегей немесе шетелдік кілт үшін жүйе автоматты түрде жасалған индексті жоюға болмайды. Пайдаланушы жасаған индексті ғана жоюға болады.

12.7. ГЕНЕРАТОРЛАР

Генератор(generator) — бұл қарапайым дереккөз нысаны. Генератор деректер базасының серверінде сақталатын механизм болып табылады, ол бұрынғы генератормен шығарылған мәндермен ешқашан сәйкес келмейтін бірегей мәндерді береді. Бұл бүтін сандарды өте үлкен мәндерге сақтауға мүмкіндік береді. 8 байт бөлінді. Бұл жасанды бастапқы кілттердің мәндерін қалыптастыру үшін қолайлы құрал. Кез-келген дереккөз кестесінің әрбір жасанды бастапқы кілті үшін пайдаланушы өзінің жеке генераторын жасайды, оған барлық негізгі әрекеттердің бастапқы мәнінің мәндерін жасау үшін барлық әрекеттер орындалады.

Негізінде, генераторлар басқа мақсаттар үшін қайталанбайтын бүтін сандардың қатарын алу үшін пайдаланылуы мүмкін.

Генераторды құру үшін келесі нұсқауларды пайдаланыңыз: CREATE GENERATOR <генератор атауы>

Генератордың аты барлық дереккөз генераторларының атаулары арасында бірегей болуы керек.

Генератор жасалса, ол 0 мәніне орнатылады. Бұл генератордың аты пайдаланылатын GEN_ID функциясына кейінгі қоңыраулар осы мәнді көрсетілген мәнге өзгертеді. Әдетте, өсім бір, бірақ нөлден басқа кез келген басқа бүтін сандарды қолдануға болады (синтаксис ережелеріне сәйкес нөлге де қолдануға болады),

Дегенмен, бұл сандардың бірізділігін алуға мүмкіндік бермейді).

Генератордың мәні оператормен белгіленеді:

SET генераторы <генератордың аты> TO <басталу мәні>

Бірегей мәнді алу үшін, функцияны пайдаланыңыз

GEN_ID (Генератор атауы, кадам)

Генератордың бастапқы мәнін қалпына келтіруге немесе әртүрлі қоңырауларға арналған кадамды GEN_ID функциясына өзгерту ұсынылмайды. Мұндай жағдайларда генератор бірегей емес мәнді шығара алады және соның салдарынан, дерекқор кестесінде жаңа жазбаны сақтауға тырысқанда, ерекше жағдай жасалады.

Генератор SQL операторы арқылы жойылуы мүмкін DROP GENERATOR <генератор атауы>

Генератор барлық генераторларға сілтеме жасаған кезде сақталатын процедуралар мен дерекқордан жойылғаннан кейін ғана жойылуы керек.

Төмендегі мысалда жаңа жолды қосқанда генератордың қолданылуы суреттеледі. Дерекқорда САТЫЛЫМ бағанындағы бағанның генераторы анықталсын:

```
CREATE GENERATOR Ген_нөмір;  
SET GENERATOR Ген_нөмір TO 0;
```

Генераторға тікелей INSERT операторынан сілтеме келесідей жазылуы мүмкін:

```
INSERT INTO PRODAJA  
(Нөмір, күні, Саны, Тауар, Тұтынушы)  
VALUES (  
GEN_ID (Ген_номер, 1) ,  
'01.0 5.2 013"7,  
100, «Сахар»,  
«ООО Ромашка»)
```

Триггердің көмегі арқылы негізгі баған дерекқордағы жаңа жолды жазудан бұрын шақырылған бірегей мән беріледі деп түсінуге болады:

```
CREATE TRIGGER BI_PRODAJA FOR САТЫЛЫМ ACTIVE  
BEFORE INSERT
```

AS

BEGIN

NEW. Нөмір = GEN_ID (Ген_нөмір, 1)

END

Бірегей мәндерді жасау үшін сақталатын процедураны пайдалана аласыз:

```
CREATE PROCEDURE GEN_PROC RETURNS (N  
INTEGER)
```

AS

BEGIN

N= GEN_ID (Ген_Нөмір, 1)

SUSPEND

END

Генераторлардың маңызды ерекшелігі - олармен жұмыс істеу кез-келген мәміленің контекстінен тыс орындалады. Бұл дегеніміз, сіз бір мезгілде әртүрлі бәсекелес транзакциялардың бір генераторына қол жеткізгенде ешқашан құлыптау қақтығысы болмайды және әрбір параллель процесс бірегей жаңа сандық мәнге ие болады. Мән генераторға қол жеткізу уақытына байланысты.

12.8. SQL ТІЛІНІҢ АРТЫҚШЫЛЫҚТАРЫ

SQL тілі енді өте кең таралымға ие болды және іс жүзінде реляциялық дерекқордың стандартты тіліне айналды. Деректер базасын басқару жүйелерінің барлық ірі әзірлеушілері қазіргі уақытта өз өнімдерін SQL тілін пайдалана отырып жасайды. Бұл әзірлеушілер мен пайдаланушылардан үлкен инвестициялар жасайды. Қолдану архитектурасының бөлігі болды, көптеген ірі және беделді ұйымдардың стратегиялық таңдауы. Осылайша, SQL-ті пайдаланушыларға, бағдарламалар мен есептеу жүйелеріне ақпаратқа қол жеткізуге мүмкіндік беретін қуатты құрал ретінде ұсынуға болады. SQL тілі пайдаланушыға опциялармен қамтамасыз етеді:

- әртүрлі дерекқор нысандарын, олардың құрылымын толық сипаттайтын кестелерді жасаңыз;
- деректерді өңдеудің негізгі операцияларын орындау: кестелерден деректерді қосу, өзгерту және жою;
- деректерді түрлендіруді орындау өте күрделі және ауқымды қоса алғанда, түрлі сұрауларды орындау.

Айта кету керек, SQL тілі барлық жоғарыда көрсетілгендерді шешеді

пайдаланушының ең аз күш-жігері бар тапсырмалар, оның командаларының құрылымы мен синтаксисі өте оңай және қол жетімді зерттеу.

SQL тілінің негізгі ерекшеліктері:

- *стандарттылық* — бағдарламаларда SQL қолдану халықаралық ұйымдармен стандартталған;
- *нақты СУБД тәуелсіздігін* — барлық ортақ пайдаланылатын СУБД-ны пайдалану SQL, реляциялық деректер базасы ең аз өзгертулермен бір СУБД-на ауыстырылуы мүмкін;
- *бір компьютерлік жүйеден екіншісіне көшу мүмкіндігі*
- *тілдің реляционды негізі* — SQL - реляциялық дерекқорлардың тілі, сондықтан деректерді ұсынудың реляциялық үлгісі кең таралған кезде танымал болды. Реляциялық дерекқордың кесте құрылымы жақсы түсініледі, сондықтан SQL тілін үйрену оңай;
- *Интерактивті сұрауларды жасау қабілеті* - интерактивті режимде, сұраудың нәтижесін өте қысқа уақыт ішінде кешенді бағдарлама жазбастан алуға болады;
- *SQL дерекқорларына кіруге қажет қосымшаларда қолдану оңай.* Сол SQL мәлімдемесі интерактивті және бағдарламалық қол жеткізу үшін де пайдаланылады, сондықтан деректер базасына кіруді қамтитын бағдарламалардың бөлігі алдымен желіде тексеріліп, содан кейін бағдарламаға біріктірілуі мүмкін;
- *SQL-ні пайдалану арқылы деректерді басқа ұсыну,* пайдаланушы әртүрлі көріністерді көретін осындай деректер құрылымын көрсете аласыз. Сонымен қатар, дерекқордың әртүрлі бөліктеріндегі деректер біріктірілуі және бір қарапайым кестеде ұсынылуы мүмкін, яғни бұл ұсыныстарды дерекқорды қорғауды жетілдіруге және жекелеген пайдаланушылардың нақты талаптарына сәйкестендіруге жарамды екенін білдіреді;
- *деректер базасының құрылымын динамикалық өзгерту және кеңейту қабілеті* - SQL тілі сізге дерекқордың құрылымын басқаруға мүмкіндік береді, осылайша икемділікті қамтамасыз етеді. Дерекқор нысандарын жасай және жоя аласыз және олардың құрылымын орындау уақытында өзгерте аласыз. Бұл дерекқордың доменнің өзгеретін талаптарына сәйкестігі тұрғысынан өте маңызды;

- *клиент-сервердің архитектурасында жұмыс істеу мүмкіндігі* — SQL - клиент-сервер өзара әрекеттесудің ең жақсы құралдарының бірі, сонымен қатар, ол өзара әрекеттесетін клиенттік жүйе мен дерекқорды басқаратын серверлік жүйе арасындағы байланыс ретінде қызмет етеді, олардың әрқайсысы өз функцияларын орындауға.

SQL құрастыру қажетті теориялық негіздерді дамытуға ғана емес, табысты іске асырылған техникалық шешімдерді дайындауға да көмектесті. Жаңа нарықтарға арналған мамандандырылған тілді енгізу - (Online Transaction Processing OLTP) және онлайн-аналитикалық өңдеу жүйелерін немесе онлайн шешімдерді қолдау жүйелерін (OnLine Analytical Processing, OLAP).

БАҚЫЛАУ СҰРАҚТАРЫ

1. Дерекқорлармен жұмыс істеу тіліне қойылатын талаптар қандай?
2. SQL тілін іске асыру дегеніміз не?
3. Неліктен SQL тілі әзірлеушілер арасында танымал болды?
4. SQL-нде қандай санаттар бар?
5. Әрбір тіл санатына қандай командалар қолданылады?
6. SQL деректерінің қандай түрлері пайдаланылады?
7. Кестелерді жасау кезінде домендер қалай пайдаланылады?
8. Доменде қандай шектеулерді сипаттауға болады?
9. Кестені SQL-де жасау кезінде негізгі кілт қалай анықталады?
10. Сілтеме тұтастығы шектеуі деп аталады және ол SQL қалай жасалады?
11. SQL-кестелерді анықтаған кезде қарым-қатынас қалай жасалады?
12. ТАҢДАУ нұсқауын пішімін сипаттаңыз.
13. SELECT нұсқаулығында қандай функцияларды пайдалануға болады?
14. Сұраудың нәтижесін қалай топтауға болады?
15. Сұраудың нәтижесін қалай шектей аламын?
16. Сұраудың нәтижесін қалай сұрыптаймын?
17. Кестелердің ішкі бірлестігі не деп аталады?
18. Сұрауларда HAVING тармағы қашан қолданылады?

19. Неліктен ішкі сұраулар SINCULAR, EXISTS, ALL, SOME бөлімдерін пайдаланады?
20. Кестелердің сыртқы қосылымы дегеніміз не және сыртқы байланыс ішкі түрден қалай ерекшеленеді?
21. Генератор деген не?
22. Триггерлер не үшін қолданылады?
23. Қандай оқиғалар триггерді тудырады?
24. Сақталатын процедураларды және триггерлерді пайдаланудың артықшылықтары қандай?
25. Триггерлер мен сақталатын рәсім арасындағы негізгі айырмашылық қандай?
26. Индекстерді қашан жасау керек?
27. SQL арқылы кестеге жаңа жолдарды қалай қосуға болады?
28. SQL арқылы кестелердегі жолдарды қалай жоюға болады?
29. SQL кестесін пайдаланып, деректерді қалай өзгерте аламын?
30. SQL тілін қолданудың қандай артықшылықтары бар?

ДЕРЕКТЕР БҮТІНДІГІН ҚАМТАМАСЫЗ ЕТУ

13.1. ТРАНЗАКЦИЯ МЕХАНИЗМІ

ДББЖ серверлік деректер базасымен жұмыс істеу үшін деректер қорының тұтастығы мәселесінің айрықша маңыздылығын атап өтті. Дерекқордың тұтастығымен дерекқорда бар ақпараттың ішкі логикасына, құрылымына және барлық анық көрсетілген ережелерге, яғни оның мазмұнының дұрыс құрылымына және консистенциясына сәйкестігін түсінеміз. Адалдықты бұзу, мысалы, бағдарламалық қателер немесе техникалық ақаулар арқылы туындауы мүмкін, себебі бұл жағдайда жүйе қалыпты өндеуді немесе дұрыс деректерді беруді қамтамасыз ете алмайды. Дерекқордың шектеулері жағдайында тұтастығын қамтамасыз ету үшін, жеке операцияларды емес, транзакция механизмін қамтамасыз ету керек.

Транзакция — деректер әсер, деректерді манипуляциялау операциялар реттілігі тұрғысынан, бөлінбейтін болып табылады.

Пайдаланушы үшін мәміле «барлығы немесе ештеңе» негізінде жүзеге асырылады. операция толығымен жүзеге асырылады және басқа дәйекті мемлекетке біртұтас (дәйекті) мемлекет дерекқорды қояды және өзгерістер растайды (commit), немесе қандай да бір себептермен мынадай мәміле бір мүмкін емес немесе жүйенің кез келген бұзу, дерекқор бар болса, не алдын ала мәмілеге болған бастапқы күйіне қайтару, яғни (rollback) транзакция түгел жұмыс жасайды.

Бұл механизм келесідей жұмыс істейді. Әрбір транзакция басталған кезде, олар бірегей бірегей нөмірін алады. Сондай-ақ, операцияның түгендеу бет деп аталады, ішкі құрылымын көшірмесін бар (TIP), ол үшін ол іске қосылған сәтте басқа мәмілелер мәртебесі туралы ақпарат алады. Мәміле жолға әсер еткенде

кестеде (өзгерту, қосу немесе жою және т.б.) сызықтың жаңа нұсқасы жасалып, оны жасаған транзакцияның бірегей санымен белгіленеді. Енді басқа транзакция осы деректер жолын қайтаратын жазбаларды алу үшін сұрауды орындайтын кезде, сервер ең соңғы нұсқаны қайтарады.

Мәміленің тұжырымдамасы деректер қорының тұтастығы тұжырымдамасымен тікелей байланысты. Жиі мәміле әртүрлі кестелерге қатысты кестелерді қосу үшін әрекеттер жасалатын бірнеше кестелердегі операцияларды біріктіреді. Жазбаларды бір кестеден екіншісіне ауыстыруды қарастырайық.

Жазбаны бірінші кестеде жойылады, содан кейін екінші кестеде сақталған болса, онда компьютер салдарынан электрмен жабдықтау үзіліс мысалы орындамаған жағдайда, жылы, ол рекордтық өзінде жойылған, бірақ екінші кестеге ала алмады, бұл кезде мүмкін болады. Бірінші жазба екінші кестедегі сақталады, содан кейін бірінші кестеде жойылады болса, онда істен рекордтық екі кестелерде болатынына мүмкін.

Екі жағдайда да дерекқордың тұтастығы мен дәйектілігі бұзылған. Осындай жағдайдың алдын алу үшін бір кестеден жазбаны жою және оны басқа кестеге қою әрекеті бір операцияға біріктіріледі. Бұл транзакцияның орындалуы кез-келген нәтиже арқылы дерекқордың тұтастығы бұзылмайды.

Кейбір бақылау нүктелерін құрған кезде кірістірілген транзакцияларды пайдаланудың құралдары бар және мәміле транзакцияның басында емес, белгілі бір бақылау нүктесінде қайта оралуы мүмкін.

Мәмілелер ұзақ және қысқа болуы мүмкін. Дерекқормен жұмыс істегенде, клиент бағдарламалары ұзын және қысқа операцияны қолдана алады. *Ұзақ транзакция* өте ұзақ уақыт бойы белсенді болып келеді. Осы контексте деректер базасымен көптеген операциялар жасалуы мүмкін. Бастапқы сәттен бастап оны растау немесе қайтару кезіне дейін жасалған қысқа мәміле секундтың белсенді бөлшегі болып табылады. Әдетте, осындай мәміленің контекстінде дерекқорда шектеулі саны бар.

Ұзақ транзакциялар, әдетте, дерекқор деректерін оқу кезінде пайдаланылады. Дерекқор кестелеріне өзгертулер жасасаңыз, қысқа операциялар өте ыңғайлы.

Транзакциялық деректер базасын басқару жүйесіне қойылатын талаптар төменде сипатталған:

- *Атомдық* (Atomicity) жүйеде ешқандай мәміле жасалмайтындығына кепілдік береді. Оның барлық операциялары орындалады немесе жүзеге аспайды;
 - *Келісімділік* (Consistency) — табысты аяқталғанын растаған транзакция дерекқордың үйлесімділігін сақтайды;
 - *оқшаулағыштық* (Isolation) — мәміле барысында параллель мәмілелер оның нәтижесіне әсер етпеуі керек. Бұл жылжымайтын мүлік төменде қарастырылады;
 - *сенімділік* (Durability) — Жүйе жұмысы аяқталғаннан кейін сәтті аяқталған транзакциямен жасалған өзгерістер сақталуы тиіс. Басқаша айтқанда, егер пайдаланушы жүйеден транзакция аяқталғаны туралы растау алған болса, ол жасаған өзгерістер қандай да бір сәтсіздікке байланысты жойылмайтынына сенімді болуы мүмкін.
- Осы талаптардың жиынтығы ACID акронимімен белгіленеді.

13.2. ТРАНЗАКЦИЯ МЕН БҰҒАТТАУ. ПАРАЛЛЕЛЬ ҮРДІСТЕРДІ БАСҚАРУ

Көптеген пайдаланушылар дерекқормен жұмыс жасағанда, бірдей жазбаны бір мезгілде жаңартуды болдырмау үшін деректерге тікелей қол жеткізуді шектеу енгізу қажет. Бұл жағдайда сіз құлыптарды қолдануға жүгінесіз.

Бұғаттау— бұл кейбір деректерді өңдеу операцияларын орындауға уақытша шектеу. Бұғаттауға жеке кесте жазбасында, кестеде, тіпті бүкіл дерекқорда салық салынуы мүмкін. Сервер нысанындағы құлыптарды басқарады, оларды пайдалануды және қақтығыстарды шешуді басқаратын құлыптау менеджері деп аталады.

Операциялар ACID талаптарына сәйкестігін қамтамасыз ету үшін деректердің құлыптарын орнатады. Блоктауды пайдаланбастан, бірнеше мәмілелер бірдей деректерді өзгерте алады.

Бұғаттаудың екі түрі бар:

- 1) жазбаны бұғаттау - транзакция кестелердегі жолдарды осы жолдарға басқа транзакция туралы сұрау жойылатын етіп бұғаттайды;

2) Оқуды құлыптау - транзакция жолдарды бөгеді, осылайша басқа транзакциядан сұрау осы жолдардың жазбаларын құлыптау үшін қабылданбайды және оқу құлпы қабылданады. Дерекқорды параллель өңдеуге арналған шешімдер кестелердің жолдары бұғатталған және осы жолдарды өзгертетін кейінгі әрекеттер қабылданбайды және күту режиміне қойылады.

Егер бұғаттау жүйесі дерекқорды басқару жүйесінде іске асырылмаса, бір уақытта бірнеше пайдаланушылармен бірдей деректерді оқу және өзгерту бір мезгілде кіруге қатысты проблемаларға әкелуі мүмкін:

- соңғы өзгеріс мәселесі бірнеше пайдаланушылар бірдей жолды оның бастапқы мәніне байланысты өзгерткен кезде пайда болады; онда кейбір деректер жоғалып кетеді, себебі әрбір келесі транзакция бұрынғыға жасалған өзгертулерді қайта жазады. Бұл жағдайдың шығу жолы үнемі өзгереді;
- егер пайдаланушы логикалық тұрғыдан бұрын бірнеше деректерді өзгертуді қажет ететін күрделі деректерді өңдеу операцияларын орындаса, «лас» оқу мәселесі мүмкін. Деректерді модификациялау барысында басқа пайдаланушы оларды оқиды, ол логикалық түрде дұрыс емес ақпарат алуы мүмкін. Осындай проблемаларды болдырмау үшін барлық өзгерістерді аяқтағаннан кейін деректерді оқып шығу керек;
- қайталанбайтын оқулар мәселесі мәміле бойынша сол деректердің бірнеше оқылымының нәтижесі болып табылады. Бірінші транзакцияны орындау барысында басқа деректерге өзгерістер енгізуге болады, сондықтан оны қайта оқығанда, бірінші транзакция тұтастықтың немесе логикалық сәйкессіздіктің жоғалуына әкелетін деректердің басқа жиынтығын алады;
- бір транзакция кестеден деректерді таңдағаннан кейін, екіншісі бірінші аяқталғанша жолдарды жойып немесе жойғаннан кейін фантомды оқу мәселесі пайда болады. Кестеден тандалған мәндер дұрыс болмайды.

Осындай проблемаларды болдырмау үшін қатарлас операцияларды келісілген орындаудың кейбір процедураларын жасау керек. Бұл рәсім келесі ережелерге сай болуы керек:

- мәміле барысында пайдаланушы келісілген деректерді ғана көреді. Пайдаланушы сәйкессіз аралық деректерді көрмеуі керек;

- деректер базасында параллельді екі мәміле жасалса, СУБД транзакцияның нәтижелері 1-транзакция сияқты орындалатын болады, содан кейін транзакция 2 немесе керісінше, транзакция 2, және керісінше, транзакцияның орындалу принципін қолдауға кепілдік беріледі, онда мәміле 1 жүзеге асады.

Мәмілелерді дұрыс ұстау дерекқордың тұтастығын қамтамасыз ету үшін негіз болып табылады, сондай-ақ көп қолданушы жүйелердегі пайдаланушыларды оқшаулау үшін негіз болып табылады. Егер дерекқордың тұтастығын сақтау үшін міндетті талаптарды орындасаңыз, транзакцияларды оқшаулаудың бірнеше деңгейлері болуы мүмкін.

Мәмілелерді оқшаулау деңгейі — мәміледе келісілмеген деректерге рұқсат етілетін деңгейді, яғни транзакцияның басқа біреуінің оқшаулану дәрежесін көрсететін мән. Оқшауланудың жоғары деңгейі деректердің дұрыстығын арттырады, бірақ бір мезгілдегі операциялардың саны азайтылуы мүмкін. Екінші жағынан, оқшаулаудың төменгі деңгейі біркелкі операциялар жасауға мүмкіндік береді, бірақ деректердің дұрыстығын азайтады.

Егер дерекқордың тұтастығын сақтау үшін міндетті талаптарды орындасаңыз, транзакцияны оқшаулаудың келесі деңгейлері мүмкін.

Оқшауланудың ең жоғары деңгейі транзакцияның серияландыру хаттамасына сәйкес келеді. Бұл деңгей, SERIALIZABLE деп аталатын, толық транзакция оқшаулануын және параллельді транзакциялардың дұрыс өңдеуін қамтамасыз етеді.

Оқшауланудың келесі деңгейі - расталған оқу деңгейі деп аталады - REPEATABLE READ. Мұндай жетіспейтін жаңартулары сияқты проблемалар, аралық немесе қайшы деректер орын мүмкін емес, сондықтан, бұл деңгейде, операция, басқа операциялардың аралық немесе қорытынды нәтижелерін қол жоқ. Алайда, олардың операциялар орындау кезінде, сіз басқа мәміле базасында қосылған жолды көре аласыз. Сондықтан, бір транзакция кезінде орындалған бірдей сұрақ әртүрлі нәтижелерге әкелуі мүмкін, яғни галстук сызықтар мәселесі қалады. Алайда, егер мұндай мәселе маңызды болса, оны бір транзакцияда қайта орындауды қоспағанда, өңдеу алгоритмін өзгертуді алгоритмдік түрде шешу керек.

Оқшауланудың тағы бір деңгейі аяқталған оқумен байланысты, ол READ COMMITED деп аталады

Бұл оқшаулау деңгейінде мәміле басқа транзакциялардың аралық нәтижелеріне қол жеткізе алмайды, сондықтан жоғалған жаңартулар мен аралық деректердің проблемалары туындауы мүмкін емес. Алайда, басқа транзакцияларды орындау кезінде алынған соңғы деректер біздің операцияға қол жетімді болуы мүмкін. Бұл жағдайда, «лас» оқу (яғни СОММИТ командасының дерекқорында тіркелмеген деректерді бір пайдаланушының оқуы) жобасы жоқ.

Дегенмен, бір транзакцияны жүргізу барысында, екіншісі сәтті аяқталуы мүмкін және оған жасалған өзгертулер тіркелген. Нәтижесінде, бірінші транзакция басқа деректер жиынымен жұмыс істейді. Бұл проблема емес оқу мәселесі. Бұл оқшаулау деңгейінде транзакция басқа транзакциямен жаңартылған жолды жаңарта алмайды. Мұндай жаңартуларды орындауға тырыссаңыз, жетіспейтін жаңарту мәселесін болдырмау үшін транзакция автоматты түрде жойылады.

Соңында, оқшауланудың ең төменгі деңгейі расталмаған немесе «кір» деп аталады. Ол UNCOMMITTED READ КОМПАНИЯСЫ деп белгіленеді. Бұл оқшаулау деңгейінде ағымдағы транзакция аралық және келісілмеген деректерді көреді, сондай-ақ, хайуанаттар желісі де қол жетімді. Бірнеше транзакция бірдей жолды бір уақытта өзгертуге тырысса, соңғы нұсқасында жолда соңғы сәтті мәміле арқылы анықталатын мән болады. Дегенмен, оқшауланудың бұл деңгейінде де ДББЖ жетіспейтін жаңартуларды болдырмайды.

Мәмілелерді оқшаулауды қамтамасыз ету үшін ДББЖ олардың бірлескен іске асыруын реттеудің кез келген әдістерін қолдануы керек.

Жүйе транзакциялардың белгілі бір жиынтығын мезгілде орындаңыз. Өртүрлі мәмілелердің мәмілелерінің қатарлас немесе параллельде орындалатын мәміле жасау әдісі егер мәмілелерді бірлескен жүзеге асырудың нәтижесі бірдей мәмілелердің бірізді орындалуының нәтижесі болып табылса, серия деп аталады.

Транзакцияның сериализациясы — Бұл кейбір сериялық жоспарға сәйкес оларды іске асыру тетігі. Осындай механизмді қамтамасыз ету - мәмілелерді басқаруға жауапты деректер базасын басқару компонентінің негізгі функциясы. Мәмілені сериялауға қолдау көрсетілетін жүйе нақты оқшаулауды қамтамасыз етеді.

13.3. ДЕРЕКТЕРДІ ҚАЛПЫНА КЕЛТІРУ МЕН ТІРКЕУ ЖУРНАЛЫ

Журналға тіркеу өзгерістер - логикалық немесе физикалық сәтсіздік жағдайында дерекқорды алдын-ала келісілген күйге келтіру үшін қажетті ақпаратты сақтайтын ДББЖ функцияларының бірі. Өзгерістерді журналдандыру, яғни деректер базасының барлық өзгерістері туралы ақпаратты сыртқы жадында сақтау, мәмілелерді басқарумен тығыз байланысты.

Ең қарапайым жағдайларда, өзгерістерді тіркеу деректер базасында жасалған барлық өзгерістерді сыртқы жадыға дәйекті түрде жазуды білдіреді. Өзгерту журналы дерекқордағы барлық деректерді өзгерту әрекеттерінің, атап айтқанда, өзгертілген нысанның ескі және жаңа мәнін, объектіні өзгертетін транзакцияның жүйелік нөмірін қамтиды. Осылайша қалыптастырылған ақпарат, дерекқордағы өзгерістердің журналы болып табылады. Журнал мәміленің бастапқы және соңғы белгілерін және бақылау нүктесінің қабылдауын қамтиды. Жүйе бақылау нүктелерін мезгіл-мезгіл орнатады. Бұл процесті орындау барысында барлық тіркелмеген деректер сыртқы жаққа ауыстырылады және бақылау нүктесі журналға жазылады. Осыдан кейін, бақылау нүктесінің алдында жазылған мазмұн журналы жойылуы мүмкін.

Физикалық сәтсіздікке ұшыраған жағдайда, журнал да, дерекқордың өзі де зақымдалмаса, rollforward процесі орындалады. Журнал алдыңғы бақылау нүктесінен бастап алдыңғы бағытта сканерленеді. Барлық жазбалар журналды журналдың соңына дейін шығарылады. Журналдан алынатын ақпарат сыртқы жады деректерінің блоктарына енгізіледі, онда санның белгісі журналда жазылғаннан аз. Егер сканерлеу барысында сәтсіздік қайталанса, сканерлеу қайтадан басталады, бірақ шын мәнінде қалпына келтіру тоқтаған жерінен жалғасады. Деректер базасын қалпына келтіру қажет болған жағдайларды қарастырайық:

- Транзакцияны дереу кері қайтару транзакциямен (ROLLBACK командасымен) немесе СУБД жүйесімен басталуы мүмкін, транзакция кез-келген қате болған жағдайда транзакциялардың қайтарылуын бастауы мүмкін (мысалы, нөлге бөлу);
- «жұмсақ» жүйенің істен шығуы (бағдарламалық жасақтаманың апатты жағдайда істен шығуы) жүйелік жады

Бұл жағдайда апат кезінде туындаған барлық транзакцияларға әсер етіледі, барлық дерекқор буферінің мазмұны жоғалады. Дискіде сақталған деректер өзгеріссіз қалады. «Жұмсақ» ақаулық орын алуы мүмкін, мысалы, қалпына келтірілмейтін процессордың істен шығуының нәтижесінде;

- Жүйелердің «қатты» істен шығуына (аппараттық құралдың ақаулығына) сыртқы сақтау құрылғыларының зақымдалуы сипатталады. Мысалы, дискідегі дискілердің бұзылуы немесе авариялық электр қуатының істен шығу салдарынан қиындықтар туындауы мүмкін.

Барлық үш жағдайда қалпына келтіру транзакция журналында ұсынылған деректердің артықшылығы болып

13.4

ҚОЛДАНУ БАСҚАРУ ӘДІСІДЕРЕКТЕР ЖӘНЕ БАСҚАРУҒА АРНАЛҒАН АРТЫҚШЫЛЫҚТАР

Дерекқорлар, дерекқор файлдарында сақталған деректер сияқты, қорғалуы керек. Пайдаланушы басқару жүйесі кез-келген дерекқорда сақталатын деректердің қауіпсіздігін қамтамасыз етудің міндетті шарты болып табылады. SQL-де дерекқор пайдаланушылары үшін бірыңғай стандартты пәрмен жоқ - әр іске асыруда өз командалары бар, кейде ұқсас. Бірақ нақты іске асырылуына қарамастан, барлық негізгі қағидалар бірдей.

Дерекқордың логикалық құрылымын, кестелердің, тұтастық шектеулердің және басқа нысандардың сілтемелерін құрастырғаннан кейін, дерекқорға қол жеткізе алатын пайдаланушылар шеңберін анықтау қажет.

Әдетте, ДББЖ екі деңгейлі мәліметтерге қол жеткізуді шектеуді ұйымдастырады. Бірінші деңгейде, серверге қосылу үшін тіркелгі деп аталатын тіркелгіні жасауыңыз қажет, ол әлі дерекқорға қатынасуға рұқсат бермейді. Екінші деңгейде, есептік жазбаға негізделген әрбір дерекқор сервері үшін, дерекқорға қосылу үшін пайдаланушы жазбасын жасау керек. Пайдаланушы тіркелгісін қосуға, өзгертуге немесе жоюға болады. Осылайша, сервер екі деңгейлі деректерді қорғауды қамтамасыз етеді:

- 1) сервер деңгейіндегі аутентификация;
- 2) дерекқор деңгейінде сәйкестендіру

Түпнұсқалық растама аяқталғаннан кейін және тіркелгі идентификаторы алынғаннан кейін, пайдаланушы тіркелген және серверге кіру рұқсаты ретінде қарастырылады. Пайдаланушы тіркелгісі оған дерекқор нысандарымен жұмыс істеу құқығын бермейді. Рұқсаттар GRANT командасымен орнатылады. Пайдаланушы GRANT командасымен берілген құқық басқа пайдаланушыға өзімен иеленетін немесе әлсіз құқықтар жиынтығын бере алады. Кестемен жұмыс істеу немесе презентация жасау үшін пайдаланушы SELECT, INSERT, UPDATE, DELETE немесе REFERENCES командасын орындау құқығына ие болуы керек. Барлық осы командаларға бір уақытта артықшылықтар орнату үшін ALL артықшылықты пайдалануға болады. Сақталған процедураны қолданбаға шақыру үшін пайдаланушы немесе нысан EXECUTE пәрменін орындауға құқылы болуы керек.

Пайдаланушыға жаңа құқықтар тағайындалмас бұрын, оның ескі адамдарын қабылдау қажет. Бұл REVOKE пәрменін жасауға мүмкіндік береді.

Басқа пайдаланушылардың берілген пайдаланушысына берген артықшылықтары REVOKE Операторына әсер етпейді. Сондықтан, егер басқа пайдаланушы пайдаланушыға жойылатын артықшылықты берсе, онда көрсетілген пайдаланушыға сәйкес кестеге кіру құқығы жалғасады.

Әдетте, шоттармен жұмыс істеу кезінде үлкенірек нысандар - рөлдер қолданылады.

Рөл - кейбір құқықтар бар және пайдаланушы тіркелгілері бар дерекқор нысаны. Рөлмен жұмыс жеке пайдаланушыларға қарағанда әлдеқайда ыңғайлы. Әрбір есепке қарағанда бір рөлдің құқықтарын жеке-дара айқындаған жөн. Рөл сізге сол топтағы бірдей функцияларды орындайтын пайдаланушыларды топтастыруға мүмкіндік береді. Рольді құру үшін операторды пайдаланыңыз:

CREATE ROLE <рөл атауы>

Бір тіркелгі бірден бірнеше рөлден тұруы мүмкін. Бірақ бір сессия барысында клиент тек бір рөлде жұмыс істей алады. Бұл жағдайға байланысты бір пайдаланушы әр түрлі қатынас құқықтарына ие болған жағдайларда ғана қолайлы болуы мүмкін. Роль құқықтарымен жұмыс істейтін пайдаланушы өз құқықтарын мұра етеді.

Әрбір СУБД тиісті рұқсаты бар пайдаланушыларға дерекқорға қатынаса алатындығын қамтамасыз ететін механизмді қолдауы керек. SQL тілі қамтиды

GRANT және REVOKE операторлары деректер базасында кестелерді қорғауды ұйымдастырады. Қорғау механизмі қолданушы идентификаторларын, иелену құқығын және оларға берілген артықшылықтарды пайдалануға негізделген.

13.5. ДЕРЕКТЕРДІҢ САҚТЫҚ КӨШІРМЕСІ

Сақтық көшірме (backup copy) - деректердің көшірмесін жасау немесе бүлінген жағдайда түпнұсқа немесе жаңа орынға деректерді қалпына келтіруге арналған ортаны құру процесі.

Дерекқорға зиян келтірудің себептері көп, олардың ішінде:

- серверлік компьютердің ақаулары мен ақаулары, әсіресе дискілер, диск контроллері, RAID контроллерінің компьютерлік жады және кәштері;
- Бір немесе бірнеше тұтынушылардың пайдаланушылар дерекқорына қате қосылу (пайдаланушылар);
- файловое копирование или другой файловый доступ к базе данных при запущенном сервере;
- дерекқормен жұмыс істеу кезінде бос дискілік кеңістіктің болмауы.

Сақтық көшірме деректерді жоғалтудан қорғай алады және бағдарламалық жасақтама мен аппараттық құралдың сәтсіздіктері, зақымдану және т.б. нәтижесінде деректерді жоғалтудан қорғаудың ең сенімді жолы болып табылады.

Сақтық көшірме - қалыпты файл көшірмесі емес. Дерекқорды резервтік көшіру - клиенттің қолжетімділік режимінде арнайы утилитамен орындалатын дерекқордан алынған ақпаратты оқу.

Сақтық көшірменің кейбір ерекшеліктері бар:

- деректер қорын көшіру қалыпты тұтынушылар бағдарламалардың жұмысымен бір мезгілде орындалуы мүмкін;
- дерекқордың көшірмесі утилитаның резервтік көшірмесі басталған кезде дерекқорда болған деректерді қамтиды. Клиенттік бағдарламалар бойынша бір уақытта жасалған резервтік көшірмелердің барлық өзгертулері сақтық көшірмеге қосылмайды;
- сақтық көшірме жасау барысында, әрбір енгізу дерекқордағы барлық кестелерден оқылады. Нақты емес жазбалардың немесе олардың үзінділерінің нұсқалары жойылады. Қалған жазбалар оңтайландырылған;

- сақтық көшірме жасау кезінде индексті қайта құрастырады, бұл осы индекстерді қолданатын операциялардың орындалуын жақсартады.

Сақтық көшірме дерекқорды сақтық көшірме құралында сақтайды. Мұнда тек қана резервтік көшірме жасау жеткіліксіз екенін еске түсірген жөн, кейде дерекқордың қалпына келтірілуін тексеру қажет, себебі қалыпты себептер бойынша дерекқор қалпына келтірілмеген жағдайлар бар.

Деректерді сақтаудың максималды сенімділігін қамтамасыз ету үшін, сіз сыртқы медиаға үнемі сақтық көшірмелеңіз.

БАҚЫЛАУ СҰРАҚТАРЫ

1. Транзакция дегеніміз не?
2. Бірнеше транзакциялардың бірлескен жұмысының мәселелерін келтіріңіз.
3. Дерекқордың тұтастығы дегеніміз не?
4. «Жуыр» оқуы деген не?
5. Параллельді транзакциялардың келісілген орындалу тәртібі қаншалықты қанағаттандырылады?
6. Мәмілені оқшаулау деңгейі қандай?
7. Мәмілелердің оқшаулануының әрбір деңгейін сипаттаңыз.
8. Транзакция журналына қандай ақпарат жазылады.
9. Мәмілелерді пайдалану қажеттілігін көрсететін мысал келтіріңіз.
10. Мәмілелер механизмін түсіндіріңіз.
11. Қисынды сәтсіздікке ұшыраған жағдайда не болады?
12. Физикалық сәтсіздік жағдайында не болады?
13. Мәмілелерді пайдалану мақсаттары қандай?
14. Дерекқорды қай кезде қалпына келтіру қажет?

ДЕРЕКТЕР ҚОРЫН БАСҚАРУДЫҢ ЗАМАНАУИ ЖҮЙЕСІ

14.1. ЗАМАНАУИ СУБД ҚОЙЫЛАТЫН ТАЛАПТАР

Қазіргі уақытта дерекқорды басқару жүйелеріне өте жоғары талаптар қойылған.

Пайдаланушылардың және СУБД әкімшіліктерінің тұрғысынан ең маңызды талаптардың бірі - *жоғары өнімділік*, яғни пайдаланушы сұрауларын жедел өңдеу және транзакцияларды орындау мүмкіндігі. Бұл жағдайда сұраныстарды оңтайландыру және индекстерді пайдалану құралдары маңызды рөл атқарады.

Қазіргі заманғы деректер қорын *қауіпсіздіктің* ең қажет етілетін ерекшеліктерін бірі орындау қатар кіріктірілген қолдаудың, яғни мәселенің аясында анықталған алдын ала белгіленген ережелерге сәйкес оқу және редакциялау деректерге пайдаланушы қол жеткізуін қамтамасыз етуге қабілеті, сондай-ақ сақтау кестелер сенімді қорғау ережелер. Бұл талапты орындау жолдары деректерді шифрлаудан аудитке дейін және оның нәтижелерін талдаудан өзгеше болуы мүмкін.

Масштабтау — функционалдығын және өнімділігін арттыра отырып, жүктемені ұлғайтумен және оның тиісті аппараттық жаңартуларымен, мысалы, СУБД көлемін кеңейтумен, процессорлар мен аппараттық серверлердің санын көбейтуге мүмкіндік береді. СУБД-нің масштабталуы және өндірісі өзара байланысты, себебі масштабталмай, өнімділіктің өсуі шектеулі. Ал егер СУБД пайдаланушылар қосымша ресурстарды қосу арқылы өсіп келе жатқан қажеттіліктерін қамтамасыз етпесе, бұл өте маңызды мәселе болады. Осындай шектеулер (деректердің максималды саны, кестеде жазылғандар саны, пайдаланушылардың саны) көп немесе аз болатынына тән

ал егер олар орын алса, өнімділіктің төмендеуі, қателердің пайда болуы, қол жеткізуден бас тарту немесе сұрау салудың орындалуы. Сондықтан жаңа ресурстарды қосуды қолдай отырып, мұндай шектеулерді еңсеру маңызды, мысалы, бірнеше компьютерден кластерлерді құру.

Мәмілені дұрыс өңдеу - қазіргі уақытта әзірлеушілер мен әкімшілер Oracle және IBM, сондай-ақ Microsoft SQL Server компанияларының мәмілелерді дұрыс өңдеу құралдарын жоғары бағалайды.

Деректер базасының құрылымы мен сұраныстың дизайнын жасау үшін мамандандырылған құралдарды - *деректерді модельдеу құралдарын* қолдануға болады. Мұндай құралдар СУБД өндірушілерімен де, тәуелсіз жеткізушілермен де жасалуы мүмкін. СУБД-ны осындай құралдарды өндірушілердің қолдауы, сондай-ақ олардың СУБД бағдарламалық жасақтамасының базасында қол жетімділігі қазіргі заманғы дерекқорды басқару жүйелерін құру болып табылады.

Көптеген заманауи СУБД жеткізу пакетінде әкімшілік құралдарды қамтиды. Бұдан басқа, үшінші тараптың СУБД-ны басқаруға болады. СУБД неғұрлым танымал және оның неғұрлым икемді болуы оның *сатушылар серіктестерімен жұмыс істеу саясаты* болып табылады, сондықтан, әдетте, осы СУБД басқару құралдарын нарықта қол жетімді етеді. Осы тұрғыдан көшбасшылар Oracle және Microsoft болып табылады.

Жақында заманауи дерекқордың пайдаланушылары XML-ті қолдауды талап етеді, өйткені XML - құжаттарды жасау және түрлі қосымшалар арасында деректермен алмасу үшін стандарт болып табылады.

Әртүрлі платформаларды қолдау гетерогенді ортада жұмыс істейтін қолданбалы әзірлеушілер үшін маңызды және әдетте АТ инфрақұрылымы өте әртүрлі ірі компаниялар үшін маңызды болып табылады.

14.2. ЗАМАНАУИ СУБД СИПАТЫ

Informix динамикалық сервері. Ақпараттық дерекқордың бірінші нұсқасы (Informix on unix) 1981 жылы жарық көрді. Бұл ДББЖ қазіргі заманғы нұсқалары (заманауи атауы - Informix Dynamic Server) транзакцияның жоғары жылдамдығымен, басқарудың сенімділігі мен қарапайымдылығымен ерекшеленеді.

Ең жиі Informix Dynamic Server негізінен ірі бөлшек сауда желілері мен телекоммуникациялық компаниялар, ірі кәсіпкерлік пайдаланылады - деректер базасын негізінде кейбір өте танымал биллинг жүйелерін құрылды.

Енді Informix Dynamic Server IBM-ке тиесілі. Осы деректер базасын соңғы нұсқасы жоғары өнімділік, ауқымдылығы, қолжетімділігін, қолдау идентификатор дамуын, және кластерлер үшін қолдау, жүктемені теңгеру, бір мезгілде өңдейтін арқылы, (жеке яшы м ндеріне дейін) қауіпсіздік белгілер негізінде деректер қол бақылау, XML қолдау құралдары бар SOA шешімдерін жасау. Дерекқор сервері клиент қосымшалар құру үшін қолдау құралдарының Blade-сервер, нысанды-бағытталған құралдары туралы Informix 4GL нысандарын құру үшін құралдар, соның ішінде IBM Data Studio құралдар, бірге жеткізіледі.

Informix Dynamic Server жоспарларын одан әрі дамыту, қолдайды идентификатор құралдарды жақсарды өнімділігін арттыру, конфигурациясын жеңілдету үшін дерекқор құралдарының әкімшілігімен байланысты түрлі міндеттерді автоматтандыруды қамтиды.

IBM DB2. DB2 алғашқы нұсқасы 1983 жылы IBM жасаған және MVC мейнфреймов реляциялық үлгісі Е.Ф. Кодд деректерінің авторы әзірленді SQL тілі, қолдау бірінші ДҚБЖ болды. DB2-дің заманауи нұсқасы нысанды-реляциялық DBMS болып табылады және әр түрлі аппараттық платформаларда Linux, UNIX және Windows амалдық жүйелерін, сондай-ақ әр түрлі IBM операциялық жүйелерін қолдайды. Ол деректерді қысуды, ықтимал мәселелерді болжауды және XML қолдауын қолдайды.

Informix Dynamic Server және DB2 ірі кәсіпорындар үшін арналған және сол компанияның IBM тиесілі болғанына қарамастан, олар жақсы шешімдер әр түрлі тапсырмаларды орындау үшін әзірленген фактісі байланысты өмір сүруі жатыр.

Informix Dynamic Server өнімділігі жоғары транзакциялық өңдеуді қажет шешімдерді жасау үшін, ең алдымен, жасалған болса, DB2, пайдалану арқылы шешіледі негізгі мәселелері - деректерді сақтау құру және техникалық қызмет көрсету болып табылады және күрделі сұрауларды өңдеу.

DB2 қауіпсіздік мүмкіндіктері пайдаланушыларымен өнімділігі мен масштабталуымен бірге жоғары бағаланады.

Microsoft SQL Server. Microsoft SQL Server 1990 жылы Microsoft мен Sybase арасындағы бірлескен жобаның нәтижесі ретінде құрылды. Бірнеше жылдан кейін бұл компаниялар бірігіп құрылған код негізінде өздерінің ДҚБЖ-ны әзірлеуді бастады және осы өнімнің 7.0 нұсқасы Sybase қатысуынсыз құрылған. Осы СУБД нұсқалары тек Microsoft операциялық жүйелеріне арналған.

SQL Server соңғы нұсқаларының басты ерекшеліктері - OLAP және аналитикалық деректерді өңдеу құралдары, геокеңістіктік деректерді сақтау құралдары, сондай-ақ басқару құралдарын оңай пайдалану, репликалау және кластерді қолдау.

Microsoft SQL Server негізгі тұтынушылары - шағын компаниялар үшін деректер базасын тұжырымы, сондай-ақ табысты қолданылған дегенмен ол, орта және ірі кәсіпорындар болып.

MySQL. MySQL-тың бірінші нұсқасы коммерциялық ДҚБЖ-ны тым қымбат деп санайтын әзірлеушілер тарапынан 1995 жылы шығарылды. Open source дерекқоры ретінде, MySQL негізсіз танымал болды - оның тарату жиынтығы күнделікті 50 мыңға дейін.

MySQL сервері еркін пайдалануға болады. Ерекше жағдай, бұл ДББЖ коммерциялық өнімнің бір бөлігі болған жағдайда, MySQL-ге лицензия беру керек. Өнімге техникалық қолдау ақ зарядталады.

MySQL көптеген интернет-шешімдерде, телекоммуникациялық компанияларға арналған қосымшалар, сондай-ақ басқа да іскери қосымшаларда пайдаланылады.

Бұл СУБД ерекшелігі - сенімділік, жоғары өнімділік және пайдаланудың қарапайымдылығы. Оның архитектурасы белгілі бір тапсырманы шешу үшін қажет емес функцияларды тастауға мүмкіндік береді, осылайша қолданба өнімділігін жақсартады - сауалнама статистикасы бұл дерекқордың пайдаланушылардың 80% өз мүмкіндіктерінің 30% -ын ғана пайдаланады.

Қазіргі уақытта MySQL ДББЖ Sun Microsystems компаниясына тиесілі, ол осы деректер базасын Sun серверлерінде және Sun бағдарламалық шешімдерімен.

Oracle Database. Oracle дерекқорының ең алғашқы нұсқасы 1979 жылы құрылды және сол уақытта коммерциялық СУБД-ны құрды, ол іскери қосымшаларды әзірлеушілерге деректерді сақтауға арналған өз шешімдерін жасауға және әр түрлі платформаларда жұмыс істейтін әмбебап шешімдерге көшуге мүмкіндік берді

1980 жылдардың ортасында. Oracle корпорациясы нарықтың көшбасшысына айналды және бүгінгі күнге дейін жетекші позицияны сақтап отыр.

Бұл дерекқордың ең соңғы нұсқалары жоғары сенімділік, қолжетімділік, қауіпсіздік және өнімділік, ыңғайлы басқару құралдары арқылы сипатталады. Бұл ДҚБЖ, ең алдымен, ірі кәсіпорындарға, сондай-ақ, транзакцияларды өңдеу және деректерді сақтау маңызды болып табылады, оның ішінде шағын және орта бизнеске арналған. Сондай-ақ, Oracle әзірлеушілерінің NET және Java технологияларын қолдана отырып, Oracle дерекқорына негізделген веб-қосымшаларды жасау үшін Application Express құралының қолжетімділігін атап өтеміз.

PostgreSQL. PostgreSQL дерекқор, сондай-ақ Oracle дерекқоры компания, ардагерлер жатқызуға болады - өнімнің алғашқы нұсқасы, кейінірек қазір PostgreSQL ретінде белгілі қандай болды, 1980 жылы пайда болды. 1996 жылдан бастап PostgreSQL ашық дерекқор болып табылады.

PostgreSQL-тың негізгі мақсаты қауіпсіздік пен сенімділіктің жоғары деңгейін талап ететін ірі кәсіпорындарға арналған тапсырмаларды орындау болып табылады. Бұл ДҚБЖ көптеген елдердегі мемлекеттік органдарда, сондай-ақ деректердің үлкен көлемін өңдеуге және транзакциялардың сенімді түрде жүргізілуіне (мысалы, генетикалық зерттеулер, геоақпараттық технологиялар, қаржы секторына арналған өтініштер) қажет болатын салаларда қолданылады.

PostgreSQL дерекқоры әзірлеушілер қауымдастығымен жасалған және кең ауқымды функционалдылықты қамтамасыз ететін ядро және қосымша модульдерден тұрады. Алайда, модульдер үлкен санының пайдалану сонымен қатар теңшеу дерекқорында осы қиын жасауға болады, және Сонымен қатар, осындай модульдер арасында әлі күнге дейін деректер талдау жоқ құралы болып табылады. Дегенмен, қосылатын модульдер процедуралық тілдер үшін қол жетімді, бұл серверлік кодты жасауға және ДҚБЖ қосымша функционалдығын қосуға мүмкіндік береді.

Ашық бастапқы ДҚБЖ арасында PostgreSQL - ең ауқымды (ол 32-ке дейін жұмыс істейді, ал MySQL масштабталуы 12-ке шектеледі). Қазіргі уақытта осы ДҚБЖ-ны одан әрі дамыту жоспарлары - өнімділікті арттыру, кластерлерді қолдау құралдарын жетілдіру және сәтсіздіктерден қалпына келтіру.

Sybase Adaptive Server Enterprise. Sybase СУБД бастапқыда Microsoft пен Sybase арасындағы бірлескен жобаның нәтижесі болды, ол 1980-жылдары басталды. Болашақта бұл компания

осы бірлескен өнімді одан әрі дамыту үшін және 1990 жылдың ортасында бөлек жобалар жасалды. Sybase SQL Server сервердің адаптивті серверіне өзгертілді.

Adaptive Server Enterprise сенімділік, оңтайлы шығындар және жоғары өнімділікке ие. Соңғы жылдары осы өнімді дамытуда қауіпсіздікті қамтамасыз ету және өнімділікті арттыру құралдарына ерекше назар аударылды. Бұл дерекқордың пайдаланушылары мен әкімшілері бүгінгі күні ең жоғары бағаланған қауіпсіздік, өнімділік және ауқымдылық.

Firebird (FirebirdSQL). Linux, Microsoft Windows және әртүрлі Unix-платформаларда жұмыс істейтін Compact, кросс-платформалы, тегін деректер базасын басқару жүйесі.

Ықшамдылық (тарату 5 Мб), жоғары өнімділікті, және сақталатын процедуралар мен триггерлер қуатты тілі қолдау: Firebird артықшылықтары көп-нұсқасы архитектурасын (оқу пайдаланушылар жазуды блоктауға емес, жедел және аналитикалық сұраулардың параллель өңдеу) атап өтуге болады ретінде. Firebird 2001 жылдан бастап әр түрлі өнеркәсіптік қосымшалар (қойма және бизнес, қаржы және мемлекеттік секторлар) пайдаланылады Ол Interbase 6.0 тегін нұсқасы ретінде Borland корпорациясы 25 шілде 2000 шығарды негізделген бастапқы кодында мультиплатформалы деректер базасын басқару жүйесін коммерциялық тәуелсіз жоба әзірлеушілер болып табылады .

Кемшіліктер арасында: кэш, жарты мәтін индекстерінің сұрау нәтижелерінің болмауы, дерекқордың ішкі фрагментациясының өсуімен айтарлықтай құлдырау.

БАҚЫЛАУ СУРАҚТАРЫ

1. СУБД-ны дамытудың әр кезеңінің ерекшеліктері қандай?
2. Дерекқор технологиясын әзірлеуде дербес компьютерлердің пайда болуы қандай рөл атқарды?
3. Деректерді өңдеу технологиясының дамуының заманауи кезеңі нені сипаттайды?
4. Дерекқорды дамытудың болашақ бағыттары қандай?
Қазіргі заманғы СУБД-ға қойылатын негізгі талаптар қандай?
6. Ең танымал заманауи СУБДсипаттаңыз?

І Бөлім. Инфокоммуникациялық жүйелер мен желілер

1. *Висандул Б.Д.* Основы компьютерных сетей : учеб. пособие / [Б.Д. Висан-дул и др.] ; под ред. Л. Г. Гагариной — М. : ИД «ФОРУМ» : ИНФРА. — М., 2007.

2. *Кондратенко С.В.* Основы локальных сетей. Курс лекций : учеб. пособие для вузов / С. В.Кондратенко, Ю.В. Новиков. — М. : ИНТУИТ.РУ, 2005.

3. *Максимов Н. В.* Компьютерные сети : учеб. пособие для студ. учреждений сред. проф. образования / Н. В. Максимов, И. И. Попов. — М. : ФОРУМ, 2008.

4. *Нанс Бэрри.* Компьютерные сети : пер. с англ. / Бэрри Нанс. — М. : БИНОМ, 1996.

5. *Норенков И. П.* Телекоммуникационные технологии и сети / И. П. Норенков, В. А. Трудоношин. — М. : Изд-во МГТУ им. Н.Э. Баумана, 1998.

6. *Новиков Ю.В.* Аппаратура локальных сетей : функции, выбор, разработка / Ю. В. Новиков, Д. Г. Карпенко ; под общ. ред. Ю. В. Новикова. — М. : ЭКОМ, 1998.

7. *Олифер В.* Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. — СПб. : Питер, 2006.

8. *Олифер В.* Основы компьютерных сетей / В. Олифер, Н. Олифер. — СПб. : Питер, 2009.

9. *Палмер М.* Проектирование и внедрение компьютерных сетей. Учебный курс / М. Палмер, Р. Б. Синклер. — 2-е изд.— СПб. : БХВ — Санкт-Петербург, 2003.

10. *Спортак М.* Компьютерные сети. Кн. 1. Всеобъемлющее руководство по устройству, работе и проектированию. Энциклопедия пользователя / М.Спортак, Ф.Паппас, Э. Резинг. — М. : ДиаСофт, 1998.

11. *Храмцов П.Б.* Администрирование сети и сервисов Internet / П.Б.Храмцов. — Центр Информационных Технологий, 1997.

12. *Фролов А.В.* Локальные сети персональных компьютеров. Использование протоколов IPX, SPX, NETBIOS / А. В.Фролов, Г. В. Фролов. — 2-е изд. — М. : Диалог, 1995.

13. *Шатт Стэн.* Мир компьютерных сетей : пер. с англ. / Стэн Шатт — Киев : ВНУ-Киев, 1996.

14. *Douglas E. Comer,* Internetworking with TCP/IP, Vol. 1, 2: Principles, Protocols, and Architecture (Third Edition) / E.Douglas. — Prentice-Hall, 1995.

II Бөлім. Деректер қорын өңдеу мен қорғау технологиясы

1. *Вирт Н.* Алгоритмы и структуры данных : пер. с англ. / Н.Вирт. — М. : Мир, 1989.
2. *Грофф Дж.* SQL: полное руководство : пер. с англ. / Дж.Грофф, П. Вайнберг. — 2-е изд. — К. : BHV, 2001.
3. *Дейт К.Дж.* Введение в системы баз данных : пер. с англ. / К. Дж. Дейт. — 7-е изд. — М. : Вильяме, 2001.
4. *Диго С. М.* Проектирование и использование баз данных : учебник / С.М.Диго — М. : Финансы и статистика, 1995.
5. *Дунаев С.Б.* Доступ к базам данных и техника работы в сети. Практические приемы современного программирования / С.Б. Дунаев. — М. : ДИАЛОГ-МИФИ, 1999.
6. *Каменнова М.* Управление электронными документами: технологии и решения / М. Каменнова // Открытые системы. — 1995. — № 4.
7. *Карпова Т. С.* Базы данных: модели, разработка, реализация / Т. С. Карпов. — СПб. : Питер, 2001.
8. *Ким Вон.* Технология объектно-ориентированных баз данных / Вон Ким // Открытые системы. — 1994. — № 4.
9. *Когаловский М. Р.* Абстракции и модели в системах баз данных / М.Р. Когаловский // Открытые системы. — 1998. — № 4 — 5.
10. *Кузнецов С.Д.* Основы современных баз данных / С.Д.Кузнецов // URL: www.citfo-rum.ru, 2002.
11. *Малкольм Г.* Программирование для Microsoft SQL Server 2000 с использованием XML : пер. с англ. / Г. Малкольм. — М. : Издательско-торговый дом «Русская редакция», 2002.
12. *Фаронов В.В.* Delphi 4. Руководство разработчика баз данных / В.В. Фаронов, П.В.Шумаков. — М. : Нолидж, 1999.
13. *Фокс Дж.* Программное обеспечение и его разработка / Дж. Фокс. — М. : Мир, 1985.
14. *Цикритзис Д.* Модели данных / Д. Цикритзис, Ф.Лоховски. — М. : Финансы и статистика, 1985.
15. *Шпеник М.* Руководство администратора баз данных Microsoft SQL Server 2000 : пер. с англ. / М. Шпеник, О. Следж. — М. : Вильяме, 2001.

Алғысөз	4
---------------	---

I БӨЛІМ

ИНФОКОММУНИКАЦИЯЛЫҚ ЖҮЙЕЛЕР МЕН ЖЕЛЛЕР

1 Тарау. Компьютерлік желінің құрылымы және элементтері, негізгі түсініктері.....	9
1.1. Негізгі түсініктер мен анықтамалар.	
Компьютерлік желінің жіктелуі	9
1.2. Компьютерлік желі топологиясы	14
1.3. Байланыстың кабельдік түрлері	16
2 Тарау. Компьютерде деректер алмасу және алмасу технологиясы	
желілері.....	21
2.1. Деректерді физикалық тарату	21
2.2. Желілік өзара қарым қатынас түрлері	24
2.3. OSI желі үлгісі.....	27
2.4. Хаттамалар мен стандарттар.....	36
2.4.1. Хаттамалардың жалпы сипаты	36
2.4.2. OSI хаттамалар стегі	39
2.4.3. TCP/IP хаттамалар стегі	40
2.4.4. IPX/SPX стегі	43
2.4.5. NetBIOS/SMB стегі	45
2.5. Өзекшелер мен желі мекен жайы.....	46
2.5.1. Желідегі мекен жай туралы жалпы түсінік	46
2.5.2. Физикалық мекен жай	48
2.5.3. IP-мекен жай	50
2.5.4. DHCP өзектерін реттеудің динамикалық хаттамасы	54
2.5.5. Домендік атаулар жүйесі	56
2.5.6. URL — әмбебап идентификатор ресурстары.....	62
3 Тарау. Жергілікті желінің қор технологиясы.....	65
3.1. Token Ring желі	65
3.2. Талшықты оптикалық интерфейс FDDI.....	67
3.3. Технологиясы Ethernet	68
3.3.1. Бірінші технология Ethernet	69

3.3.2. Fast Ethernet.....	71
3.3.3. Жоғары жылдамдықты технологии Gigabit Ethernet ...	73
4 Тарау. Компьютерлік желінің техникалық құралдары.....	79
4.1. Желі адаптеры	79
4.2. Концентраторлер	82
4.3. Репитерлер	84
4.4. Желі мосты.....	85
4.5. Желі коммутаторы.....	85
4.6. Бағыттаушылар және желі шлюзы	87
5 Тарау. Сымсыз технология.....	90
5.1. Байланыстың сымсыз жүйесі	90
5.2. Сымсыз байланыс жүйесі	92
5.3. Сымсыз байланыс жүйесін ұйымдастыру құралдары	
5.3.1. Wi-Fi-адаптерлері.....	94
5.3.2. Сымсыз байланыс нүктелері	96
5.3.3. Антенналар	96
5.3.4. Wi-Fi-бағыттаушылары	97
5.3.5. Wi-Fi ADSL-модемдері.....	98
6 Тарау. Желілік бағдарламалық қамсыздандыру. Қызметтер мен қызмет көрсету	102
6.1. Көп деңгейлі сәулет «клиент-сервер».....	102
6.2. Желілік қызмет пен желілік қызмет көрсету	104
6.3. Желілік операциялық жүйе	106
6.4. Бұлтты қызмет пен оның мүмкіндіктері	111
6.5. Желілік қауіпсіздік қызметі.....	115
6.5.1. Прокси-сервер	115
6.5.2. Брандмауэр	116

II БӨЛІМ

ТАЛАПТАРДЫ ДАМУҒА ЖӘНЕ ҚОРҒАУ ТЕХНОЛОГИЯСЫ

7 тарау. Деректер базасының теориясы, деректер қоймалары,, Білім негіздері	121
7.1. Деректер қоры мен ақпараттық жүйе. Негізгі анықтамалар	121
7.2. Деректер қорын басқару жүйесі. СУБД негізгі функциялары	127
7.3. Деректер қорының сәулеті. Физикалық және логикалық тәуелсіздік	133
8 Тарау. Деректер үлгісі.....	137
8.1. Деректер үлгісінің түсінігі.....	137
8.2. Теоретико-графикалық деректер үлгісі	138

8.3.	Реляционды үлгі	141
8.4.	Постреляциялық деректер үлгісі.....	143
8.5.	Көп өлшемді деректер үлгісі	143
8.6.	Нысанды-сипаттық үлгі	147
9	тарау. Реляционды деректер үлгісі.....	151
9.1.	Реляционды деректер үлгісінің ерекшеліктері	151
9.1.1.	Негізгі түсініктер мен компоненттер	151
9.1.2.	Қатынастар көрінісі	155
9.2.	Негізгі реляционды алгебра.....	156
9.3.	Индекстеу.....	164
9.4.	Кестелерді байланыстыру. Сілтемелік бүтіндік түсінігі	168
9.5.	Реляциялық қатынаста тұтастығын сақтау принциптері деректер қоры	172
9.6.	Реляциялық деректер моделінің артықшылығы мен кемшіліктері	174
10	Тарау. Концептуалды, логикалық құру принциптері және физикалық деректер үлгісі.....	177
10.1.	Дерекқорды жобалау міндеттері	177
10.2.	Пәндік аумақты талдау.....	181
10.3.	Концептуалды үлгілеу.....	183
10.4.	Логикалық дизайн және физикалық үлгі Деректер қоры.....	196
10.5.	Принциптерге негізделген деректер базасын құру нормалау.....	202
10.6.	Қазіргі заманғы деректер базасын құру құралдары	207
10.6.1.	Негізгі анықтамалары	207
10.6.2.	CASE-технологияны жіктеу	212
11	Тарау. Деректер сәулеті.....	215
11.1.	«Файл-сервер» сәулеті.....	215
11.2.	«Клиент-сервер» сәулеті	217
11.3.	«Клиент-сервер» үшденгейлі сәулеті.....	221
11.4.	Серверлер кластері	223
11.5.	Деректер қоры серверінің нысандары	225
12	Тарау. SQL негіздері.....	228
12.1.	SQL тіліне енгізу.....	228
12.2.	Кестелермен жұмыс. Бүтіндігін шектеу	234
12.2.1.	Домендермен жұмыс.....	234
12.2.2.	Кестелерді басқару.....	237
12.3.	Деректерді таңдау. SELECT операторы	245
12.4.	Деректерді өзгерту. INSERT, UPDATE, DELETE операторлары... 268	
12.5.	Сақталатын іс шаралар мен триггерлер.....	271
12.5.1.	Сақталатын іс шаралар мен триггерлер тілі.....	271

12.5.2. Триггерлермен жұмыс	274
12.5.3. Сақталатын өнімдермен жұмыс	277
12.6. Индекстермен жұмыс	282
12.7. Генераторлар	285
12.8. SQL тілдерінің артықшылықтары	287
13 Тарау. Деректер бүтіндігін қамтамасыз ету	291
13.1. Транзакция механизмі	291
13.2. Транзакциямен бұғаттау. Параллель процесстерді басқару	293
13.3. Деректерді қалпына келтіру мен реттеу журналы	297
13.4. Деректерге қол жеткізу және басқаруды басқару әдістері артықшылықтар	298
13.5. Деректерді сақтақы көшіру	300
14 Тарау. Деректер қорын басқарудың заманауи жүйесі	302
14.1. Заманауи СУБД қойылатын талаптар	302
14.2. Заманауи СУБД сипаттары	303
Әдебиеттер тізімі	308

Оқу баспасы

Федорова Галина Николаевна
Деректер қорын эзірлеу және басқару

Оқулық

Редактор *О.Ю.Румянцева, А.С. Кожаскіева*
Компьютерлік жабдықтау: *Е. Ю. Назарова* Корректор *Л. В. Гаврилина*

Баспа . № 101116888. Баспаға қол қойылған күні 05.06.2015. Формат 60^90/16.
Гарнитура «Балтика». Кеңсе қағазы. № 1. Офсеттік қолтанба. Баспа шарттары 20,0.
Тираж 1 500 дана. Тапсырыс №
«Академия» Баспа орталығы» ААҚ . www.academia-moscow.ru129085, Москва, Мира
даңғылы, 101В, бет 1.
Тел./факс: (495) 648-0507, 616-00-29.

Санитарлық-эпидемиологиялық қорытынды № РОСС RU. АЕ51. Н 16679 от 25.05.2015.

Баспаның электронды тасымалдаушысымен баспаға шықты.
«Тверь полиграфиялық комбинаты» ААҚ, 170024, Тверь қаласы, Ленин даңғылы, 5.
Телефон: (4822) 44-52-03, 44-50-34. Телефон/факс: (4822) 44-42-15
Home page — www.tverpk.ru. Электронная почта (E-mail) — sales@tverpk.ru