

004

C22

САРИНОВА А. Ж.

АВТОМАТТАНДЫРЫЛҒАН ЖҮЙЕЛЕР

БАҚЫЛАУДЫҢ
БАЗАЛЫҚ БЕРІЛГЕНДЕРІ
ЖОБАЛАУЫ



004
С2А

Саринова А. Ж.

**АВТОМАТТАНДЫРЫЛҒАН
ЖҮЙЕЛЕР БАҚЫЛАУДЫҢ
БАЗАЛЫҚ БЕРІЛГЕНДЕРІ
ЖОБАЛАУЫ**

Оқу-әдістемелік құралы



ЭВЕРО
Алматы, 2017

004. 053/04

ӘОЖ 681.5(075)

КБЖ 32.965я73

С 22

С. Торайғыров атындағы Павлодар мемлекеттік университетінің
физика, математика және ақпараттық технологиялары факультетінің
оқу-әдістемелік кеңесімен
баспаға ұсынылды

Пікірсарапшылар:

С. И. Деревягин – техника ғылымдарының кандидаты, ассоцияланған
профессоры (доценті);

Д. И. Исмоилов – физика-математика ғылымдарының докторы,
профессоры.

С 22 Саринова А. Ж.

Автоматтандырылған жүйелер бақылаудың базалық берілгендері
жобалауы : оқу-әдістемелік құралы / құраст. А. Ж. Саринова. – Алматы:

Эверо, 2017. – 89 б.

ISBN 978-601-327-087-6

Оқу-әдістемелік құралында «Автоматтандырылған жүйелер бақылаудың
базалық берілгендері жобалауы» пәнінің теориялық шолу SQL бағдаламалау
тілінің негізінде баяндалған. Әрбір тақырыбы қоса беру тәжірибелік
тапсырмалармен, студенттердің кәсіби компетенцияларын дамуына
бағытталған.

Оқу-әдістемелік құрал автоматизация және бақылау мамандығының
студенттеріне арналған.

ӘОЖ 681.5(075)

КБЖ 32.965я73

ISBN 978-601-327-087-6

С. Торайғыров
атындағы ПМУ-дің
академик С.Еейсембаев

атындағы ғылыми

ҮІБНОУ
КІТАПХАНАСЫ

© Саринова А. Ж., 2017

© С. Торайғыров атындағы ПМУ, 2017

© Эверо, 2017

Материалдың дұрыс болуына, грамматикалық және орфографиялық қателерге авторлар
мен құрастырушылар жауапты

Кіріспе

Мәліметтер қоры ақпараттық технологияда үлкен роль атқарады және үлкен көлемді ақпараттарды автоматты түрде өңдеудің маңызды құралы болып табылады. Мәліметтер қоры мәліметтермен жұмыс істейтін қарапайым және ыңғайлы құрал және өндірістің әр түрлі салаларында қолданылады. Мәліметтер қорының мүмкіндіктері мәліметтер қорының реляциялық жүйесі теориялық тұрғыдан жасалынып және жүзеге асырылғаннан кейін едәуір артты.

Объектілі-бағытталған мәліметтер қоры мен білім қорының жүйесі кең көлемде таралымға ие болып отыр. Объектілі-бағытталған мәліметтер қоры реляциялық мәліметтер қорын басқару жүйелері шеше алмайтын қиын есептерді шеше алады. Білім қорларындағы мәліметтер қорының реляциялық жүйелерінде мәліметтерден аса жоғарғы деңгейдегі ақпараттарды алуға мүмкіндік беретін предикаттарды есептеу логикасы қолданылады.

Есептеу желілерінің дамуы үлестірілген мәліметтер қорының ролін арттырып отыр. Клиент/сервер технологиясы мәліметтерге орталықтандырылған қол жеткізу және басқарумен үлестірілген мәліметтерді өңдеуді білдіреді.

Мәліметтер қорының жалпы теориясы, негізгі анықтамалар, мәліметтер қорының файлдық құрылымды мәліметтерден айырмашылығы мен артықшылықтары мәліметтер қорларын практика жүзінде қолдану кезінде қалыптасқан мәліметтер қорының әралуан модельдері, мәліметтер қорларын концептуальды жобалаудың принциптері мен бейнелеу деңгейлері беріледі. Теориялық негізін реляциялық алгебра, реляциялық есептеу құрайтын, қазіргі уақытта ең көп тараған реляциялық модель жеке қарастырылады.

SQL тілі – қазіргі заманғы көптеген реляциялық МҚБЖ-ның стандартты тілі болып табылады. SQL дегеніміз - Structured Queries Language сөздерінің қысқартылған түрі, ол сұраныстардың құрылымдалған тілі деп аударылады. SQL тілі реляциялық түрдегі мәліметтер қорларымен жұмыс істеуге арналған. Ол реляциялық МҚБЖ пайдаланушыларға ұсынатын барлық негізгі мүмкіндіктерді жүзеге асырады. SQL тілі командалар (операторлар) жүйелерінен тұрады.

І Ақпараттық жүйелер және мәліметтер қоры

1.1 Деректер базасының архитектурасы

Деректерді интеграциялау концепциясына негізделген қазіргі ақпараттық жүйелер үлкен көлемді және құрылымы жағынан күрделі болып келеді. Олар көптеген пайдаланушылардың әртүрлі деңгейдегі талаптарын қанағаттандыруы тиіс. Кеңінен түсіндірер болсақ, деректер базасы дегеніміз – шынайы өмірдегі белгілі бір сала бойынша бар объектілер жайлы мағұлматтар жиынтығы. Шынайы өмірдің бір бөлігін пәндік аймақ деп айтуға болады. Бұл бөлік басқарудағы ұйымдастыру мәселелерін зерттеу үшін және соңында оның жұмыстарын автоматтандыру үшін қажет. Мысалы, кәсіпорын, жоғарғы оқу орындары және т.б. Деректер базасын құрғанда пайдаланушы әртүрлі белгілері бойынша реттеп, одан көптеген басқа да белгілердің жинақталуы бойынша таңдау жүргізе алады. Мұндай жұмысты деректер құрылымы тәртіптелген жағдайда ғана жүргізуге болады. Қажетті деректер базасын жылдам жасауға рұқсат ететін әртүрлі программалар жиыны бар: Microsoft Access, Microsoft FoxPro, Paradox, dBase, InterBase, Oracle. Олардың ішінде Paradox пен dBase-да деректер базасы папкалар сияқты, ал әрбір кесте жеке файл сияқты анықталады, ал Access, InterBase және SQL-серверлерінің көпшілігі деректер базасынан тұратын бір үлкен файлды қолданады.

Деректер базасы – белгілі бір пәндік аумаққа жататын құрылымдық түрі бар деректердің атауы бар жинақ. Деректер базасы үшін ақпаратты өңдеу және рұқсат алу тәсілдері бойынша, сонымен қатар деректер базасының орналасқан орны бойынша қосымшалар архитектурасы бірнеше түрлерге бөлінеді. Жергілікті архитектура – программа және деректер базасы бір компьютерде орналасқан.

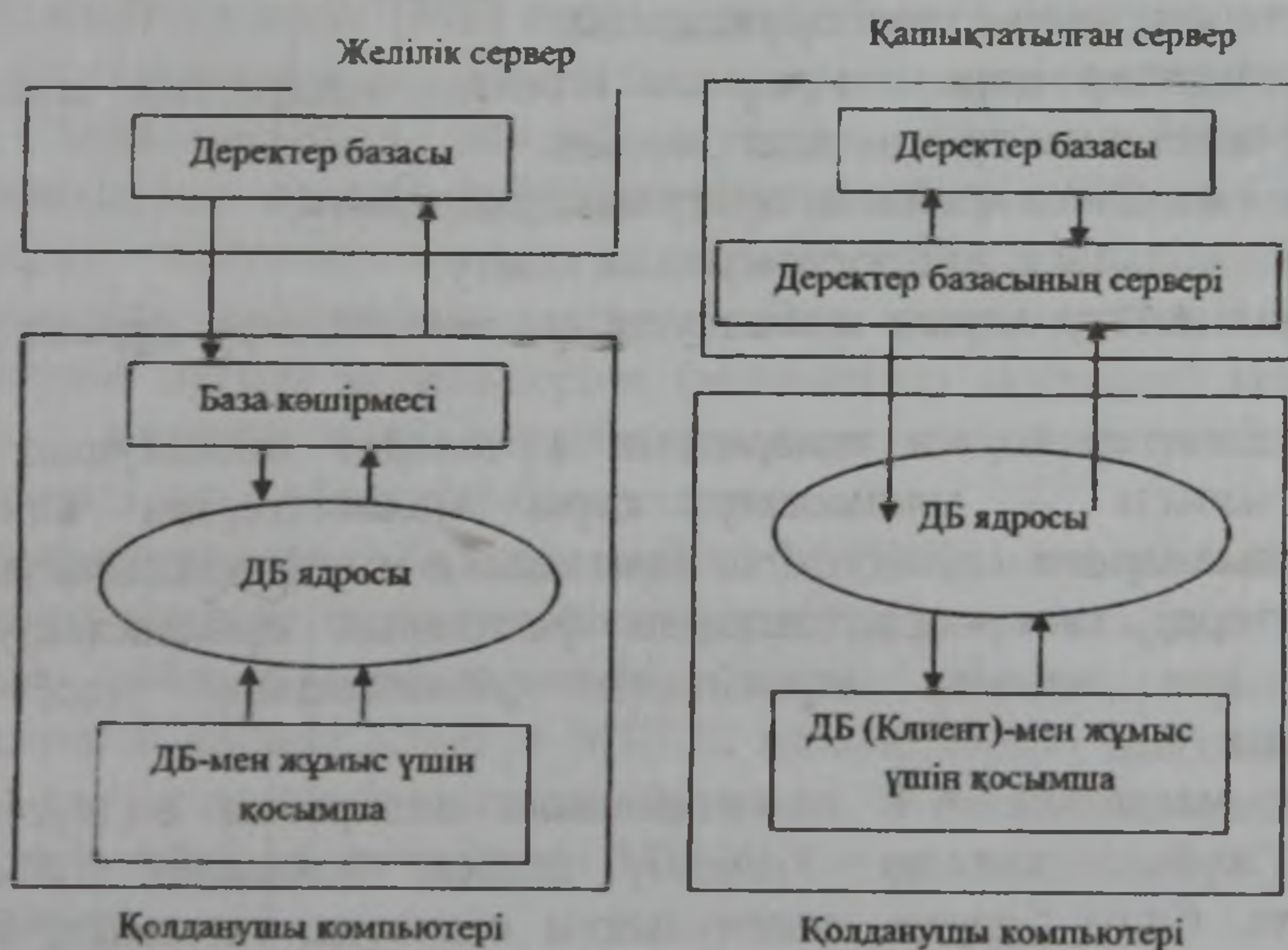
Файл-серверлік архитектура – деректер базасы сервер деп аталатын желідегі негізгі (орталық) болып бөлінген компьютерде орналасады. Ал клиенттік программалар орнатылған дербес компьютерлер оған жергілікті желі бойынша қосылған. Бұл архитектура бірнеше пайдаланушыларға бір деректер базасымен бір уақытта жұмыс істеу мүмкіндігін береді.

Клиент-серверлік архитектура – деректер базасы серверде сақталады және мұнда пайдаланушылар сұраныстарын өңдейтін деректер базасын басқару жүйесі (ДББЖ) жұмыс істейді. Бұл архитектураларда пайдаланушылар деректер базасымен қойылған тапсырмаларды орындайтын және алынған нәтижені қайтаратын

ДББЖ арқылы жұмыс істейді. Табылған деректер клиентке желі бойынша серверден клиентке қарай жіберіледі. Клиент – сервер архитектурасының ерекшелігі SQL – құрылымдық сұраныстар тілін пайдалану болып табылады. Бұл архитектура бірнеше пайдаланушыларға бір уақытта бір деректер базасымен жұмыс істеуге рұқсат етеді.

Таратылған архитектура – деректер базасы бірнеше серверлерде сақталады және олардың әрбіреуінде ДББЖ көшірмелері болады. Сонымен деректер базасы және клиенттік қосымшаларға ұқсас архитектурада қосымшалар серверлері және компоненттер қолданылады. Қосымшалар серверлері пайдаланушылар сұраныстарын өңдеуге және желідегі компьютерлер арасындағы жүктемені біркелкі бөлуге рұқсат етеді. Компоненттер – бұл бөлінген серверде күрделі есептеулерді орындау үшін қолданылатын программалар. Интернет-архитектура – деректер базасы және ДББЖ бір компьютерде орналасады, ал оларға рұқсат алу желілік протокол стандарты бойынша браузерлер арқылы жүзеге асады.

«Файл-сервер» және «Клиент сервер» архитектураларының сұлбасы төменде 1.1-суретте көрсетілген.



1.1-сурет – «Файл-сервер» және «Клиент сервер» архитектураларының сұлбасы

1.2 Ақпараттық жүйелердің архитектурасы

Автоматтандырылған ақпараттық жүйелер (АЖ) деп техникалық құралдарды, атап айтқанда ЭЕМ қолданатын жүйелерді атайды. Қолданыстағы АЖ-лердің көпшілігі автоматтандырылған болып келеді, сол себептен оларды қысқаша АЖ-лар деп атайтын боламыз.

Ақпараттық жүйе деп ақпаратты өңдеуге арналған есептеу кешені аталады. Ақпараттық жүйе келесілерден:

- компьютерлік құрал-жабдықтан;
- программалық қамтудан;
- мәліметтерден;
- қызмет көрсетушілер құрамы (пайдаланушылар, әкімші) құралады.

Жергілікті ақпараттық жүйе деп бір компьютерде орнатылған және сонда жұмыс істейтін жүйе аталады.

Клиент/сервер типіндегі компьютерлік желіде жұмыс істейтін ақпараттық жүйелер барған сайын кең таралуда. Бұл жүйелер мәліметтерді орталықтандырылған, сондай-ақ үлестірілген түрде өңделуді қамтамасыз етеді. Бұл жағдайда жалпы мәліметтер серверлерде орналастырылады, ал клиенттік компьютерлерде жеке клиенттердің мәліметтері орналасады.

Мәліметтер қорымен жұмыс істейтін ақпараттық жүйелерді программалық қамту төмендегілерден:

- жалпыға арналған программалық қамту;
- қолданбалы программалық қамту;
- мәліметтер қорын жобалауды автоматтандыру құралдарынан құралады.

Мәліметтер қорын ақпараттық жүйелерде қолданудың үлкен артықшылығы – мәліметтер қоры мәліметтердің клиенттік бағдарламалардан тәуелсіздігін қамтамасыз етеді, пайдаланушылар мәліметтерді тасымалдағыштарда физикалық орналастыру мен оларға қол жеткізу мүмкіндігін ұйымдастыру мәселесінен құтылады.

Кең мағынада АЖ анықтамасына ақпаратты өңдеудің кез келген жүйесі жатады. Қолдану саласы бойынша АЖ-ларды өндірісте, білім беруде, денсаулықты сақтауда, ғылымда, әскери істе, әлеуметтік салада, сауда-саттықта және өзге де салаларда қолданылатын жүйелерге бөлуге болады. Максаттық қызметі бойынша АЖ-ларды шартты түрде келесі негізгі категорияларға

бөлуге болады: басқарушы, ақпараттық-анықтамалық, шешімдерді қабылдауды қолдау.

Кейбір жағдайларда АЖ ұғымы қандайда бір қолданбалы мәселені шешуге пайдаланылатын аппараттық-программалық құралдардың жиынтығы деген өте тар түсініктеме түрде қолданылады. Ұйымдастыруда, мысалы, өздеріне сәйкес келесі міндеттер жүктелген ақпараттық жүйелер болуы мүмкін: мамандар мен материалдық-техникалық құралдардың есебін жүргізу, жеткізушілер мен тапсырыс берушілермен есеп айыру, бухгалтерлік есеп және т.б.

Мәліметтер банкі АЖ-ның бір түрі болып табылады, мұнда бір немесе бірнеше мәліметтер қорына ұйымдастырылған ақпаратты орталықтандырылған түрде сақтау мен жинаудың қызметтері жүзеге асырылған.

Мәліметтер банкі (МБ) жалпы жағдайда келесі компоненттерден: мәліметтер қорынан (бірнеше қорынан), мәліметтер қорын басқару жүйесінен, мәліметтер сөздігінен, әкімшіден, есептеу жүйесі мен қызмет көрсетуші персоналдан тұрады. Аталған компоненттер мен солармен байланысты маңызды түсініктерді қысқаша түрде қарастырып кетейік.

Мәліметтер қоры (МБ) есептеу жүйесінің жадында сақталатын және объектілер мен қарастырылып отырған пәндік салада олардың өзара байланысының жағдайын бейнелейтін, арнайы түрде ұйымдастырылған мәліметтердің жиынтығы болып табылады.

Қорда сақталатын мәліметтердің логикалық құрылымы мәліметтерді кейіптеудің моделі деп аталады. Мәліметтерді кейіптеудің негізгі модельдеріне (мәліметтер моделіне) келесілер жатады: иерархиялық, желілік, реляциялық, постреляциялық, көпөлшемді және объектілі-бағытталған.

Мәліметтер қорын басқару жүйесі (МҚБЖ) дегеніміз – бұл МҚ-ны құруға, енгізуге және оны көптеген пайдаланушылардың бірлесе отырып пайдалануына арналған тілдік және программалық құралдардың кешені. Әдетте МҚБЖ мәліметтердің қолданылатын моделі бойынша ажыратылады. Мысалы, реляциялық мәліметтер қорын пайдалануға негізделген МҚБЖ-лар реляциялық МҚБЖ-лар деп аталады.

Алғашқы МҚБЖ-ларға келесілер жатады: IMS (IBM, 1968 ж.), IDMS (Cullinet, 1971 ж.), ADABAS (Software AG, 1969 ж.) және

ИНЭС (КСРО АҒ ВНИИСИ, 1976 ж.). Қазіргі заманғы мәліметтер қорын басқару жүйелері мындап саналады.

Қосымша дегеніміз қолданбалы міндеттер үшін ақпаратты өңдеуді автоматтандыруды қамтамасыз ететін программа немесе программалар кешені болып табылады. Біз МҚ-н пайдаланатын қосымшаларды қарастырып отырмыз. Қосымшалар МҚБЖ ортасында немесе одан тыс ортада – МҚ-ға қол жеткізу құралдарын, мысалы, Delphi немесе С++ Builder пайдаланатын программалау жүйесінің көмегімен құрылуы мүмкін. МҚБЖ ортасында жасалған қосымшалар көбінесе МҚБЖ қосымшалары деп, ал МҚБЖ-дан тыс ортада жасалған қосымшалар сыртқы қосымшалар деп аталады.

Мәліметтер қорымен жұмыс істеу үшін көбінесе МҚБЖ құралдарын қолданған жеткілікті болады және өздерін құруға программалауды қажет ететін қосымшаларды пайдаланудың қажеті болмайды. Қосымшалар негізінен біліктілігі жоқ пайдаланушыларға МҚ-мен жұмыс істеу ыңғайлы болуы үшін немесе МҚБЖ интерфейсі пайдаланушыларды қанағаттандырмаған жағдайларда жасалады.

Мәліметтер сөздігі (МС) мәліметтердің құрылымы туралы, МҚ файлдарының өзара әрекеттері туралы, мәліметтердің типтері мен оларды ұсынудың форматтары туралы, мәліметтердің пайдаланушылардың меншігіне жатуы, қорғау және қол жеткізуді шектеу кодтары және т.б. туралы ақпаратты орталықтандырылған түрде сақтауға арналған МҚ ішкі жүйесі болып табылады.

Функционалды түрде МС барлық МБ-ларда болады, алайда осы қызметтерді атқаратын компонент әрдайым дәл сондай атқа ие бола бермейді. Көбінесе МС қызметін МҚБЖ орындайды және олар жүйедегі негізгі менюден шақыртылып немесе соның утилиталары арқылы жүзеге асырылады.

Мәліметтер қорының әкімшісі (МҚӘ) дегеніміз – МҚ-ға қатысты талаптарды даярлау ісіне, оны жобалауға, құруға, тиімді түрде пайдалануға және ілесіп жүруге жауап беретін тұлға немесе тұлғалар тобы. Пайдалану барысында МҚӘ әдетте ақпараттық жүйенің жұмысын қадағалайды, рұқсат етілмеген қол жеткізуден қорғауды қамтамасыз етеді, МҚ-да сақталатын ақпараттың көптігін, қарама-қайшылықсыздығын, сақтанымдылығы мен дәйектілігін бақылайды. Бір пайдаланушыға арналған ақпараттық жүйелерде МҚӘ қызметтері, әдетте тікелей МҚ қосымшаларымен жұмыс істейтін тұлғаларға жүктеледі.

МҚӘ есептеу желісінде, әдетте, желінің әкімгерімен өзара әрекеттеседі. Соңғысының міндеттеріне желінің аппараттық-программалық байланыс құралдарының жұмысын бақылау, желіні қайта құру, тоқтап қалулар мен істен шығулардан кейінгі программалық қамтуды қалпына келтіру, алдын-алу шаралары мен қол жеткізудің бөлініп шектелуін қамтамасыз ету ісі жатады.

Есептеу жүйесі (ЕЖ) өзара байланысқан және келісілген түрде әрекет ететін ЭЕМ-дердің немесе процессорлар мен өзге де құрылғылардың жиынтығы болып табылады, ол ақпаратты қабылдау, өңдеу мен оны пайдаланушыларға ұсыну процестерін автоматтандыруды қамтамасыз етеді. МБ-ның негізгі қызметтеріне мәліметтерді сақтау мен өңдеу ісі жататын болғандықтан, қолданылатын ЕЖ орталық процессорлардың (ОП) оңтайлы қуаттылығымен қатар тікелей қол жеткізудің жеткілікті көлемдегі жедел және сыртқы жады болуы тиіс.

Қызмет көрсетуші персонал техникалық және программалық құралдарды жұмысқа қабілетті қалпында ұстап отыру қызметін орындайды. Ол жоспар бойынша, сондай-ақ қажетінше алдын-алу, регламенттік, қалпына келтіру және өзге де жұмыстарды атқарады.

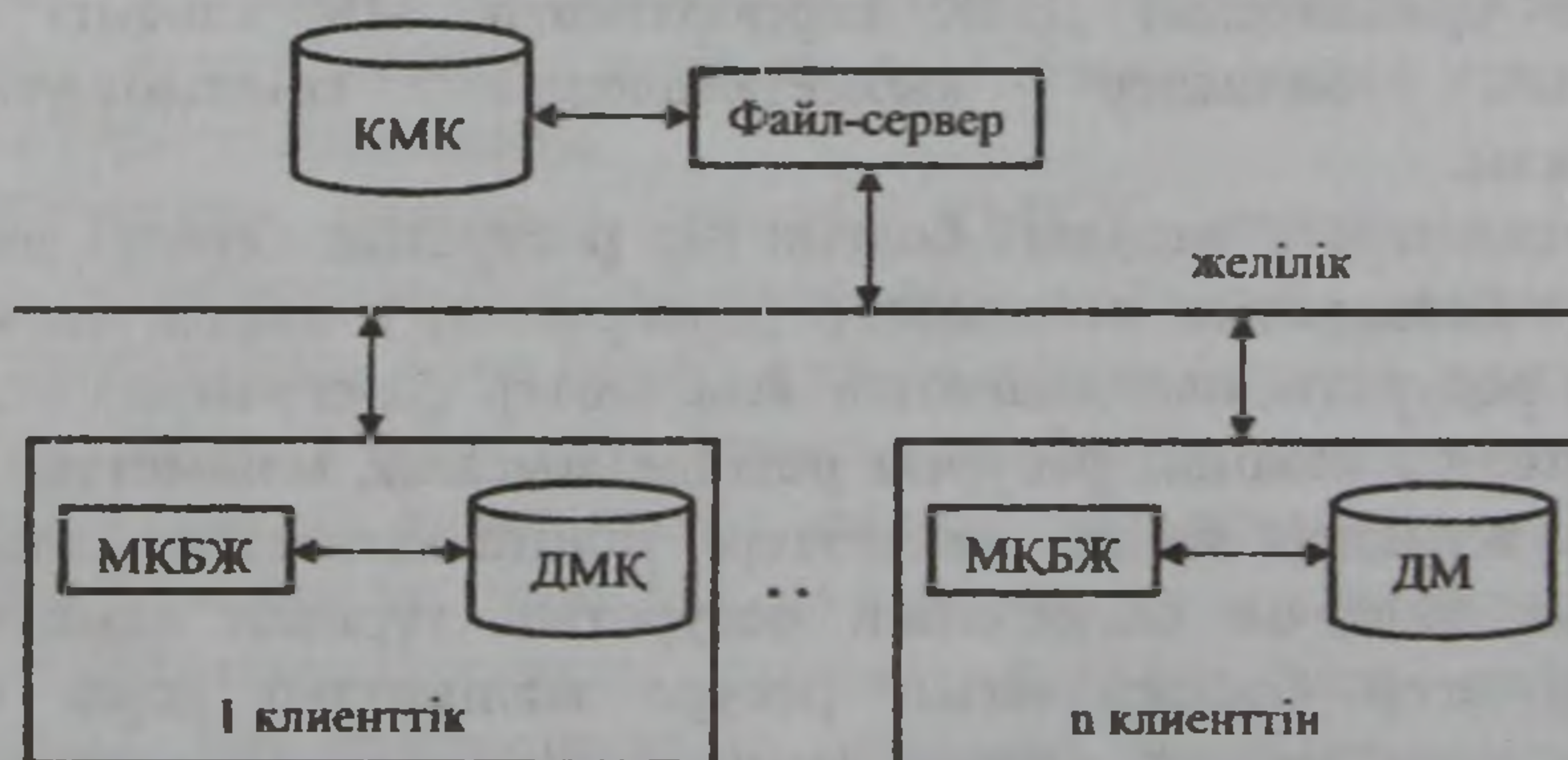
Ақпараттық жүйенің архитектурасы. Ақпараттық жүйенің (АЖ) тиімді қызмет етуі көп жағдайда оның архитектурасына байланысты. Қазіргі уақытта клиент-сервер архитектурасының болашағы зор болып отыр. Оның жеткілікті түрде таралған нұсқасында компьютерлік желі мен үлестірілген мәліметтер қоры болады, соңғысында корпоративтік мәліметтер қоры (КМҚ) мен дербес мәліметтер қоры (ДМҚ) орналасады. КМҚ компьютер-серверде орналасады, ДМҚ корпоративтік МҚ клиенті болып табылатын бөлімдер қызметкерлердің компьютерлерінде орналасады.

Компьютерлік желідегі белгілі бір ресурстың сервері деп осы ресурсты басқаратын компьютер (программа) аталады, ал клиент деп осы ресурсты пайдаланатын компьютер (программа) аталады. Компьютерлік желінің ресурсы ретінде мысалы, мәліметтер қоры, файлдық жүйелер, баспа қызметтері, пошта қызметтері алынады. Сервердің типі ол басқаратын ресурстың түрімен анықталады. Мысалы, егер басқарылатын ресурс мәліметтер қоры болып табылса, онда сәйкес сервер мәліметтер қорының сервері деп аталады. Ақпараттық жүйені клиент-сервер архитектурасы бойынша ұйымдастырудың артықшылығы пайдаланушылардың

дербес ақпаратпен жеке түрде жұмыс істеуімен орталықтандырылған түрде сақтау, қызмет көрсету және жалпы корпоративтік ақпаратқа ұжымдық түрде қол жеткізуді сәтті түрде үйлестіру болып табылады. Клиент-сервер архитектурасында оны жүзеге асырудың түрлі нұсқаларына жол беріледі.

Тарихи тұрғыдан алғашқы болып файл-серверін қолданумен үлестірілген АЖ-лар пайда болды (1.2-сурет). Мұндай АЖ-ларда пайдаланушылардың сұраныстары бойынша мәліметтер қорының файлдары дербес компьютерлерге (ДК) беріледі, ол жерде олар өңдеуден өткізіледі. Архитектураның бұл нұсқасының кемшілігі өңдеуден өткізілетін мәліметтерді берудің жоғары қарқындылығы. Оның үстіне көбінесе артық мәліметтер беріледі: пайдаланушыға мәліметтер қорындағы жазбалардың қаншасы қажет екендігіне тәуелсіз мәліметтер қорының файлдары тұтастай беріледі.

Мәліметтер қорының серверін пайдалану арқылы клиент-сервер архитектурасы бойынша құрылған үлестірілген АЖ-ның құрылымы. Мұндай архитектура болғанда мәліметтер қорының сервері мәліметтерді өңдеуден өткізудің басым бөлігінің орындалуын қамтамасыз етеді. Пайдаланушы немесе қосымша құрастыратын сұраныстар МҚ серверіне SQL тілінің нұсқаулықтары түрінде келіп түседі. Мәліметтер қорының сервері керекті мәліметтерді іздестіріп, іріктеп алады, содан кейін олар пайдаланушының компьютеріне беріледі. Бұл тәсілдің алдыңғымен салыстырғандағы артықшылығы берілетін мәліметтер көлемінің едәуір мөлшерде азаюы.



1.2-сурет – Файл-сервері бар АЖ құрылымы

Клиент-сервер архитектурасы бойынша үлестірілген МК-ны құрудың негізгі нұсқалары «Клиент-сервер архитектурасы» бөлімінде қарастырылатын болады. Дербес МК-лар мен олармен жұмыс істейтін қосымшаларды құру және басқару үшін Microsoft фирмасының Access және Visual FoxPro, Borland фирмасының Paradox тәрізді МКБЖ-лары қолданылады.

Корпоративтік МК, мысалы Microsoft SQL Server немесе Oracle Server тәрізді МК серверінің басқаруымен құрылып, қолданылады, қызмет етеді.

Шешілетін міндеттерді ұйымдастыру мөлшері мен ерекшеліктеріне қарай ақпараттық жүйе келесі конфигурациялардың біріне ие болуы мүмкін (1.3-сурет):



1.3-сурет – Мәліметтер қоры сервері бар АЖ құрылымы

- корпоративтік және дербес қорлардан тұратын компьютер-сервер;
- ДМК бар компьютер-сервер мен дербес компьютерлер;
- ДМК бар бірнеше компьютер-серверлер мен дербес компьютерлер.

Клиент-сервер архитектурасын пайдалану біріншіден, кәсіпорынның даму барысында, екіншіден, ақпараттық жүйенің өзінің дамуы барысында кәсіпорынның ақпараттық жүйесін біртіндеп ұлғайтуға мүмкіндік береді.

Жалпы МҚ-ны корпоративтік МҚ мен дербес МҚ-ларға бөлу орталықтандырылған нұсқасымен салыстырғанда МҚ-ны жобалаудың күрделілігін төмендетуге, сол арқылы жобалау кезінде кателер жіберіп алудың ықтималдығы мен жобалаудың бағасын төмендетуге мүмкіндік береді.

МҚ-ларды ақпараттық жүйелерде қолданудың аса маңызды артықшылығы – мәліметтердің қолданбалы программаларға тәуелсіздігін қамтамасыз ету. Бұл пайдаланушыларға физикалық деңгейде мәліметтерді кейіптеу мәселелерімен мәліметтерді жадыда орналастыру, оларға қол жеткізу әдістері және т.б. айналыспауға мүмкіндік береді.

Мұндай тәуелсіздікке МҚБЖ-мен қолдау көрсетілетін МҚ-да логикалық (пайдаланушылық) және физикалық деңгейлерде мәліметтерді көп деңгейлік кейіптеу арқылы қол жеткізіледі. МҚБЖ мен мәліметтерді кейіптеудің логикалық деңгейінің болуы арқасында МҚ-ның тұғырнамалық моделін (түсініктік) оның ЭЕМ жадында физикалық кейіптеуден бөліп алу қамтамасыз етіледі.

Жергілікті ақпараттық жүйелер.

Ақпараттық жүйенің функционалдык бөлімдері бір немесе бірнеше компьютерлерде орналасуы мүмкін. Біркомпьютерде АЖ-ны ұйымдастыру нұсқаларын қарастырайық. Сәйкес АЖ әдетте жергілікті немесе бір пайдаланушылық деп аталады (алайда соңғысы қатаң шарт болып табылмайды, себебі бір компьютерде кезекпен бірнеше пайдаланушы жұмыс істеуі мүмкін).

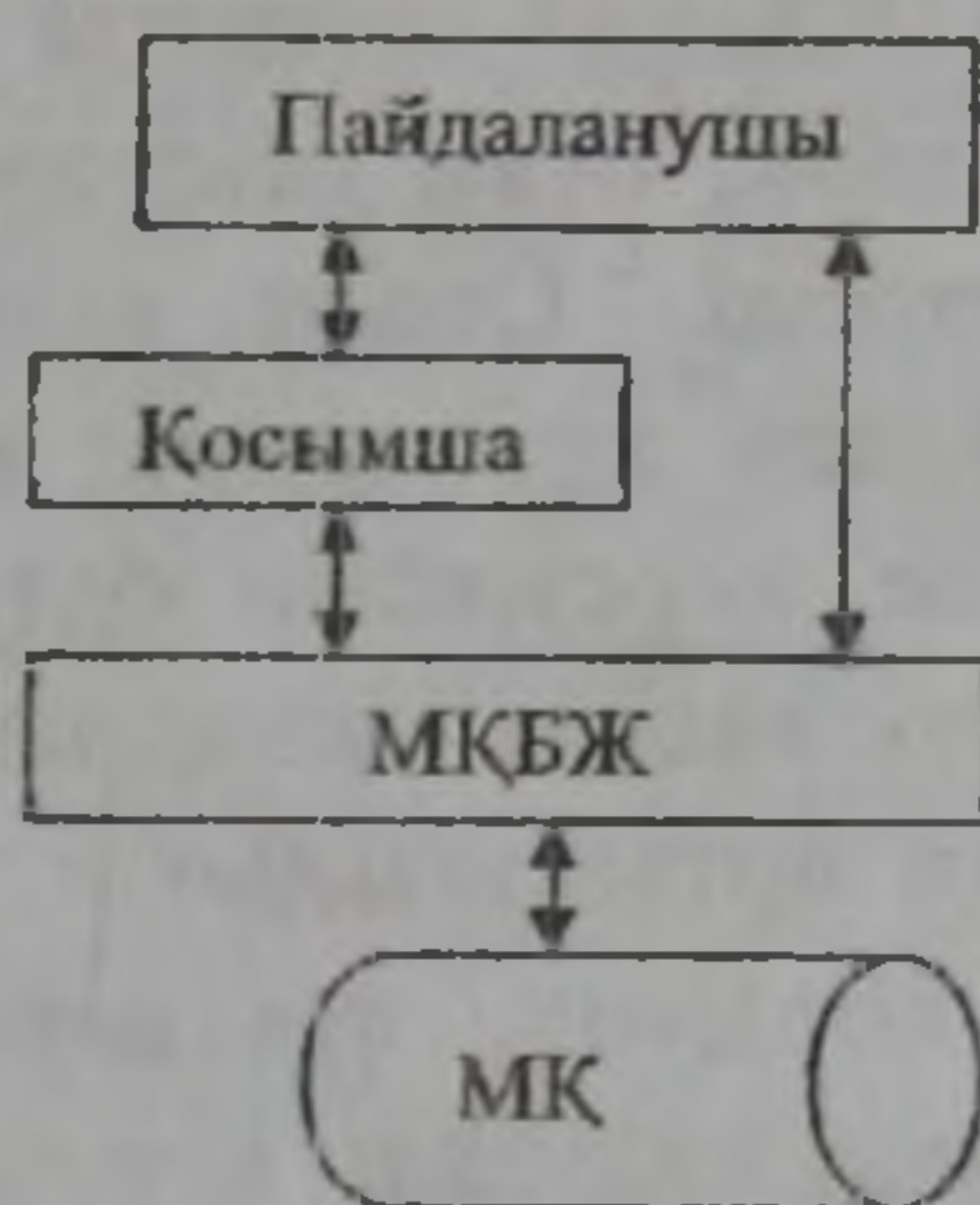
Қандайда бір операциялық жүйенің (ЭЖ) ортасында бір компьютерде жергілікті АЖ-ның қызметін ұйымдастыру программалық құралдарды қолданудың келесі нұсқалары арқылы мүмкін болады:

- «толық» МҚБЖ;
- қосымша және «кесілген» (ядро) МҚБЖ;
- тәуелсіз қосымша.

Алғашқы тәсіл әдетте компьютердің дискілік жадында бүкіл МҚБЖ орналасқан жағдайда қолданылады және ол көбінесе қосымшаны аяғына дейін өңдеуден өткізу үшін қолданылады (1.4-сурет).

Пайдаланушының МҚБЖ-мен өзара әрекеттесуі тікелей МҚБЖ-ның пайдаланушылық интерфейсі арқылы немесе қосымшаның көмегімен орындалады.

Қосымша интерпретациялау режимінде орындалады.

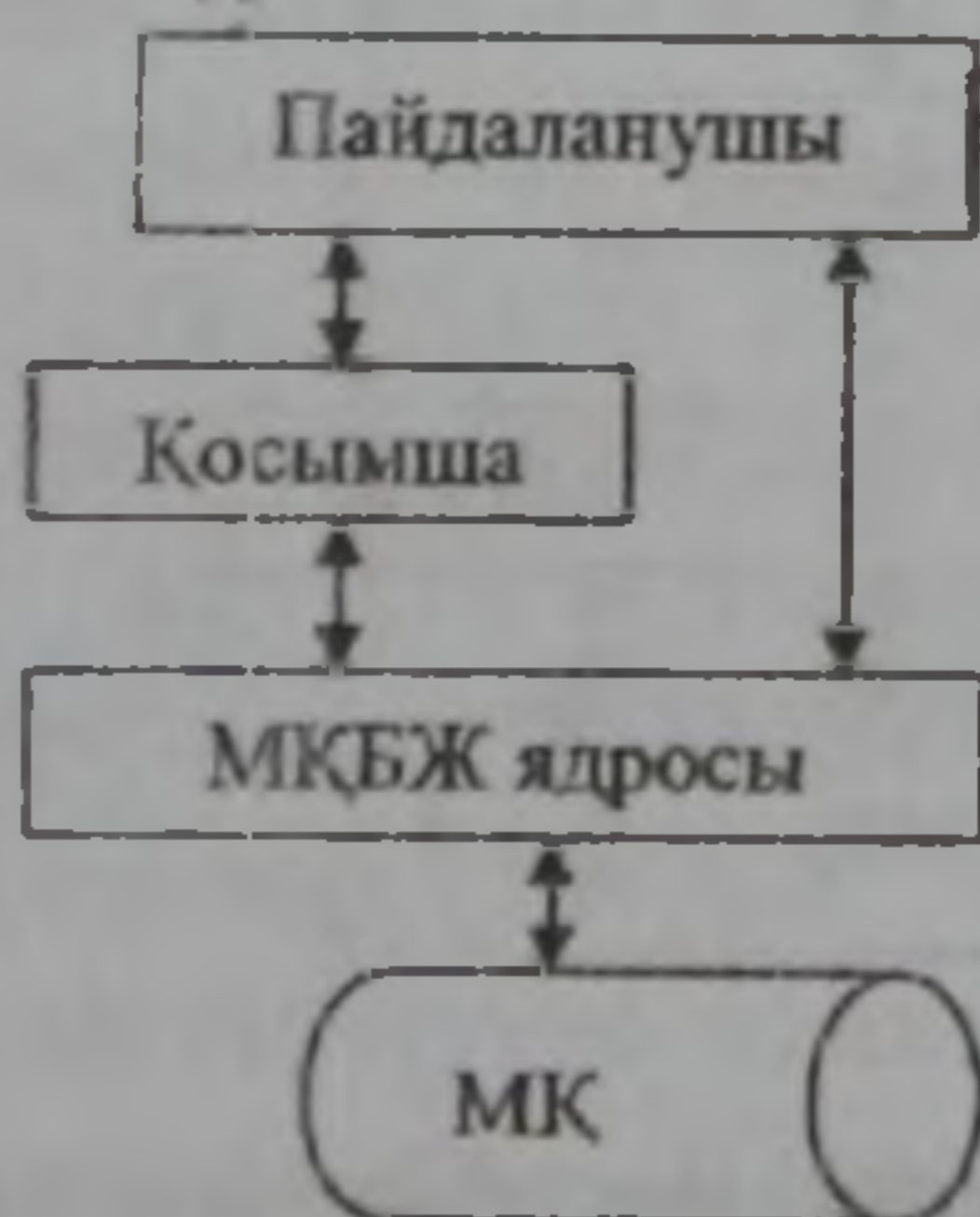


1.4-сурет – МКБЖ қолданбасын пайдалану.

Сұлбаның басты артықшылығы – дамытылған сәйкес жасау құралдары мен сервистік құралдар болған жағдайда МК мен қосымшаны жасау мен оларға ілесе жүрудің қарапайымдылығы. Бұл жүйенің кемшілігі – МКБЖ программасын сақтауға жұмсалатын дискілік жадының шығындары.

МКБЖ ядросы бар қосымша (1.5-сурет) келесі мақсаттарға жету үшін қолданылады:

- МКБЖ алып жатқан қатты дискідегі және жедел жадыдағы кеңістіктің көлемін азайту;
- қосымшаның жұмыс жылдамдығын арттыру;
- қосымшаны пайдаланушы тарапынан модификациялаудан қорғау (әдетте ядро қосымшаларды жасау құралдарын қамтымайды).



1.5-сурет – МКБЖ қосымшасы мен ядросын қолдану.

Бұл тәсілдің мысалы FoxBase+ жүйесінің FoxRun модулін пайдалану болып табылады. Қазіргі заманғы МҚБЖ-дың арасынан Microsoft Access-ті атап кетейік, оған Microsoft Access Developer's Toolkit қосымша пакеті кіреді. Оның көмегімен дискеталарда тасымалданатын, өзінде әзірлеу құралдары жоқ Microsoft Access-тың «қысқартылған» (run-time) версиясын құруға болады.

МҚБЖ-ның толық версиясын қолданумен салыстырғанда МҚБЖ ядросын қолданудың артықшылығы: компьютер жадының ресурстарын аз мөлшерде қолдану, қосымша жұмысының үдеуі және қосымшаны модификациялаудан қорғау мүмкіндігі. Негізгі кемшіліктерге МҚБЖ ядросын сақтауға қажетті дискілік жадының әлі де елеулі көлемі, және қосымшалардың жұмыс істеу жылдамдығының жеткілікті дәрежеде жоғары еместігі жатады (қосымшаны орындау бұрынғысынша интерпретациялау арқылы жасалады).

АЖ-ны ұйымдастырудың үшінші тәсіліне бастапқы программа алдымен компиляцияланады – орындалатын машиналық командалардың тізбегіне түрленіп өзгертіледі. Соның нәтижесінде өзінің қызмет етуі үшін не бүкіл МҚБЖ-ны, не оның ядросын қажет етпейтін орындауға дайын тәуелсіз программа пайда болады (1.6-сурет). Айта кететін нәрсе, мәліметтерді сақтау мен өңдеудің негізгі қызметтерін орындау тұрғысынан алып қарағанда бұл программа МҚБЖ-ның немесе оның ядросымен басқару арқылы орындалатын қосымшадан аздап қана ерекшеленеді.



1.6-сурет – Тәуелсіз қосымшаны пайдалану.

Бұл нұсқаның алдыңғы екеуімен салыстырғандағы негізгі артықшылығы – компьютердің сыртқы және жедел жадының

үнемделуі, қосымшаны орындаудың үдеуі және қолданбаны модификациялаудан толықтай қорғалуы (дизассемблерлеу жағдайы мен өзінің кодын қою және сол тәрізді жағдайлар есепке алынбайды). Қемшіліктеріне қосымшаларды аяғына дейін өңдеудің көп еңбекті қажет ететіндігін және МҚ-ны күтімге алу жөніндегі МҚБЖ-ның стандартты құралдарын пайдалану мүмкіндігінің болмауын жатқызуға болады. Клиент-сервер архитектурасының модельдері. МҚ-мен жұмыс істейтін үлестірілген АЖ-ларды құрған кезде клиент-сервер архитектурасы кең түрде қолданылады. МҚ-ны басқару процестерінің өзара әрекеттесуінің негізгі сұлбаларына сипаттама беру үшін OSI ашық жүйелер архитектурасының Эталондық моделін пайдаланайық. Бұл модельге сәйкес МҚ-ны басқару қызметі қолданбалы деңгейге жатады.

Жоғарғы екі деңгейге тоқталып кетейік: қолданбалы және өкілдік, оларға жасаушы мен пайдаланушы тарапынан аса үлкен көңіл бөлінеді. Қалған қызметтерді алғашқыкеуін жүзеге асыруға қажетті байланысқан қызметтер деп санайық. Бұл ретте МҚБЖ термині ретінде МҚ-дағы ақпаратты пайдаланатын барлық программалық жүйелердің кең түрдегі түсінігін ұстанатын боламыз.

Пайдаланушымен интерфейсті қолдайтын программа ретінде жалпы жағдайда МҚБЖ келесі негізгі қызметтерді жүзеге асырады:

- қордағы мәліметтерді басқару;
- қолданбалы программалардың көмегімен ақпаратты өңдеу;
- пайдаланушы үшін ыңғайлы болатын түрде ақпаратты бейнелеу.

Егер де жүйе бір ЭЕМ-де орналасса, онда барлық қызметтер бір программаға жинақталған және 1-бөлімде қарастырылған сұлбаға ұқсас түрдегі сұлба бойынша шақыртылады.

МҚБЖ-ны желіде орналастыру кезінде қызметтерді тораптар бойынша үлестірудің түрлі нұсқалары болады. Өздерінің арасында МҚБЖ қызметтері үлестірілетін тораптардың санына байланысты екі буындық, үш буындық және т.б. модельдерге бөлуге болады.

Екі буындық модельдер МҚБЖ қызметтерін желінің екі торабының арасында үлестірілуіне сәйкес келеді. Пайдаланушыға жақын және ақпаратты кейіптеу мәселелерімен міндетті түрде айналысатын компьютерді компьютер-клиент деп атайық. Компьютер-сервер мен компьютер-клиенттің арасында қызметтерді үлестірудің аса типтік нұсқалары (7-сурет) келесілер болып

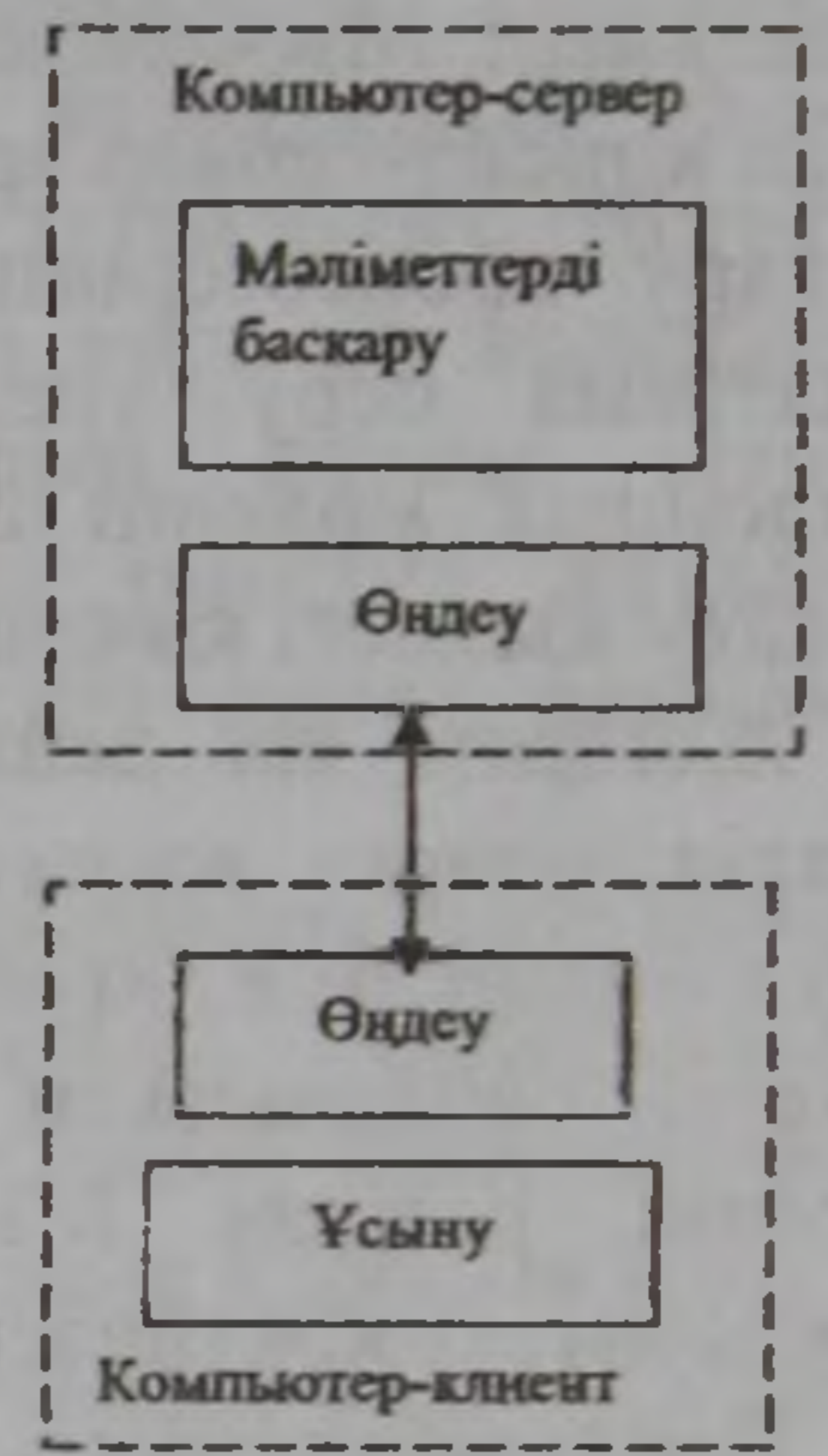
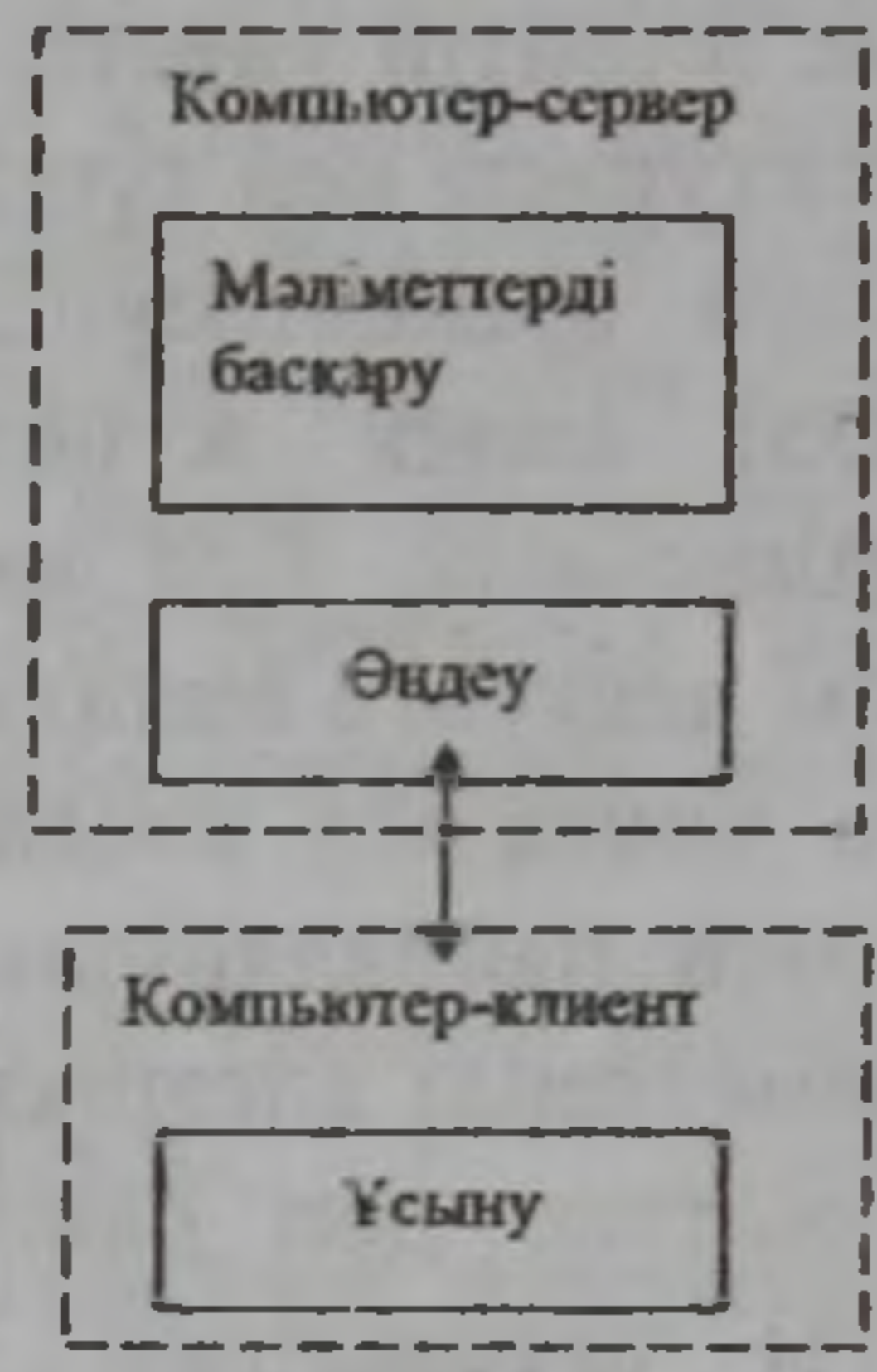
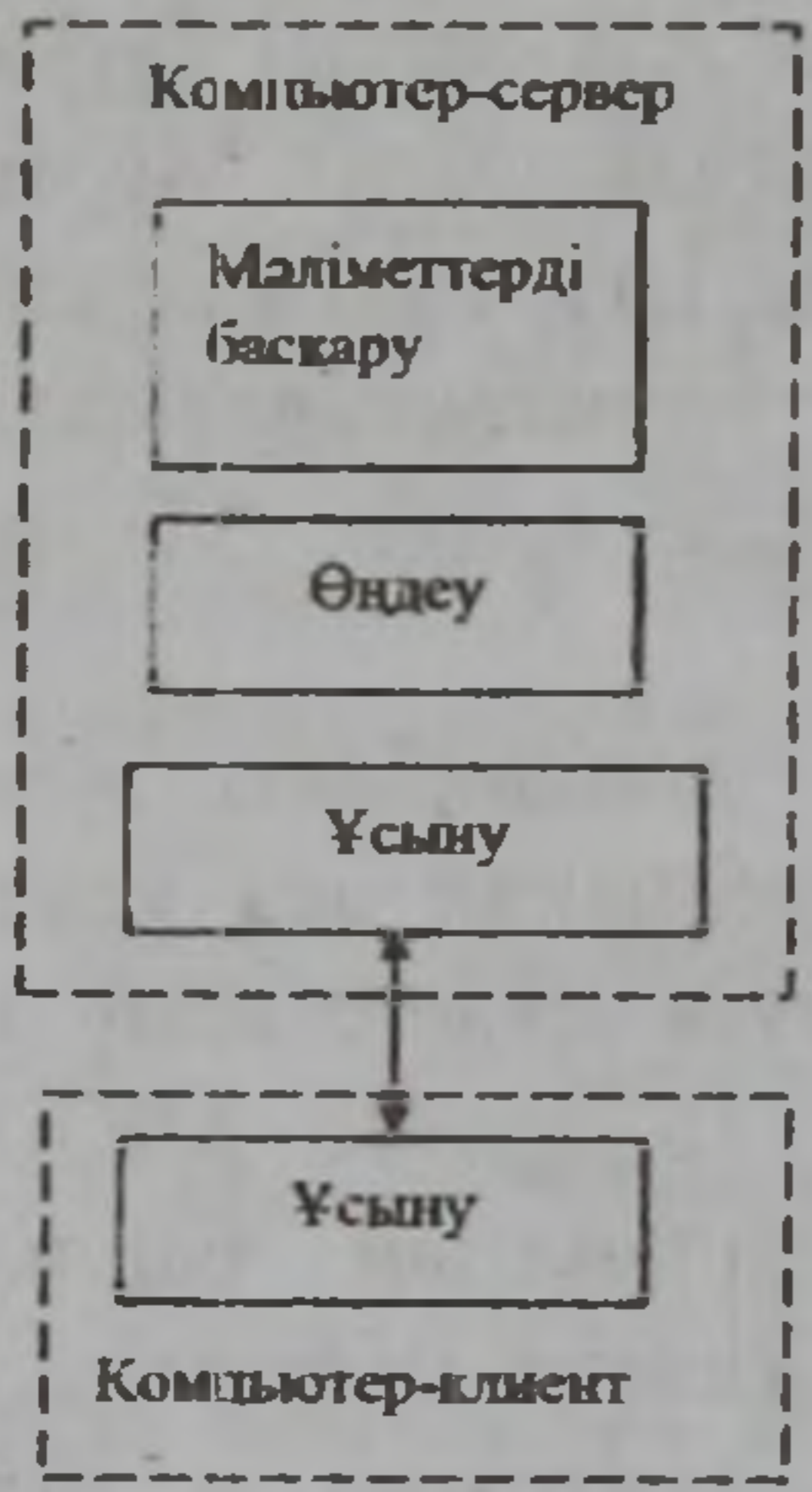
табылады: үлестірілген түрдегі ұсыну; қашықтан ұсыну; үлестірілген қызмет; мәліметтерге қашықтан қол жеткізу; үлестірілген МҚ.

Үлестірілген түрдегі ұсыну

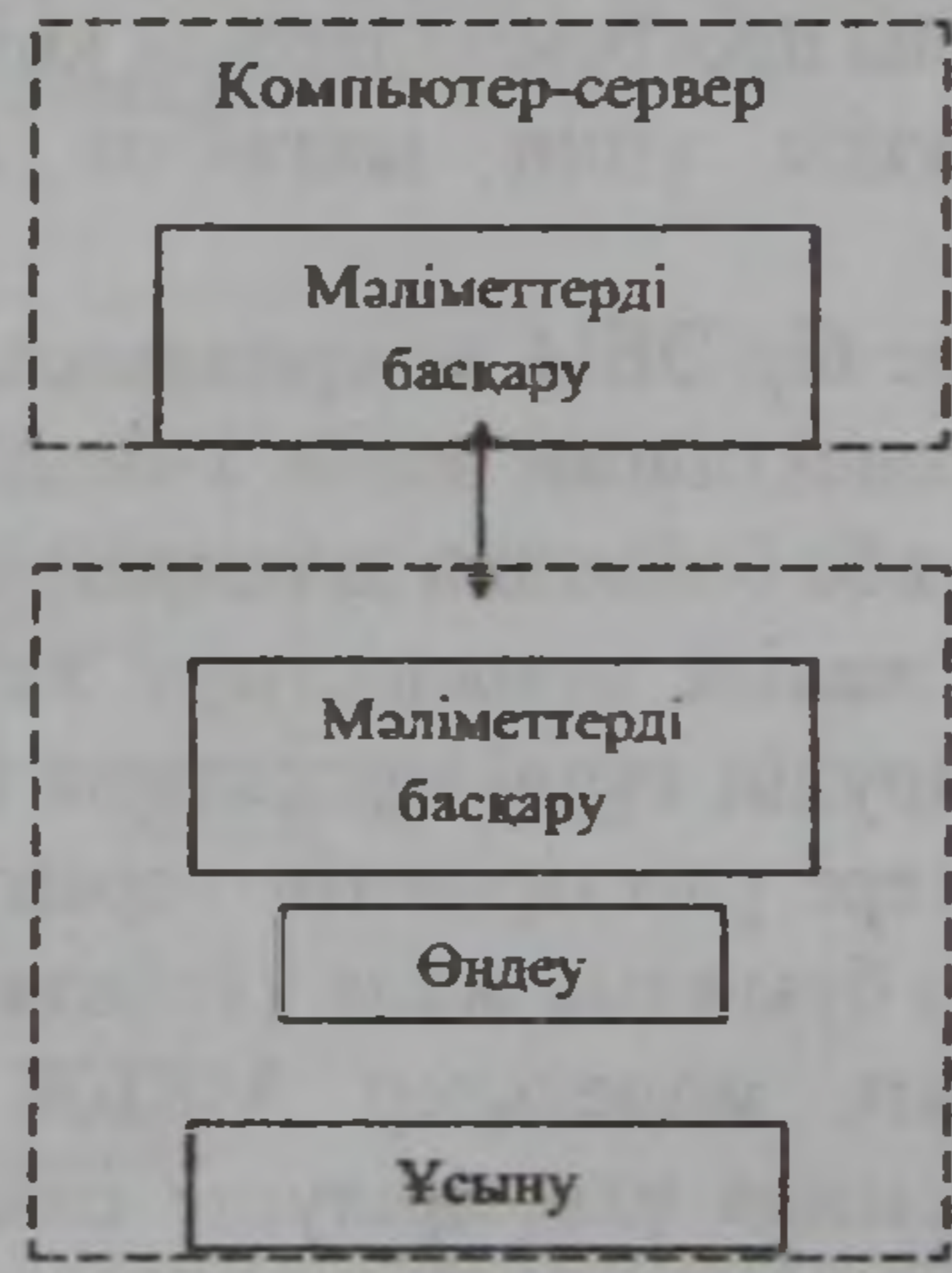
Қашықтан ұсыну

Үлестірілген қызмет

а



б



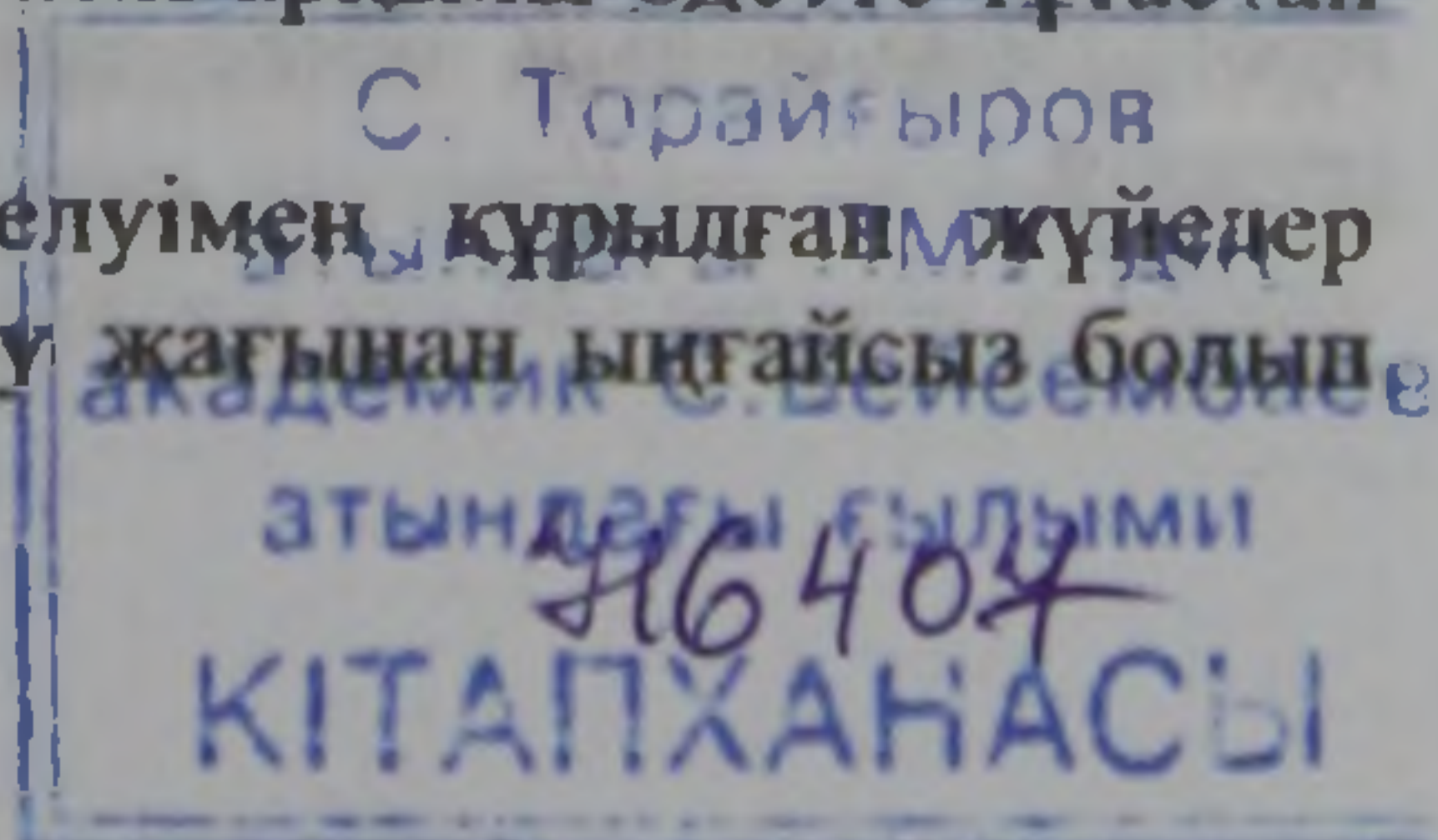
7-сурет – Екібуындық модельдер

Клиент-сервер архитектурасы бар жүйелерде қызметтерін үлестірудің аталған әдістері әр түрлі нұсқаларды көрсетеді: қуатты сервер – бұл жағдайда барлық жұмыс сонда орындалады, қуатты клиент-қызметтердің басым бөлігі жұмыс станциясында орындалады, ал сервер желі арқылы соған келіп түсіп жатқан SQL-шақыртуларды өңдеуден өткізеді. Мәліметтерге қашықтан қол жеткізу мен қашықтан ұсыну модельдерінде компьютер-клиент пен компьютер-сервердің арасында қызметтер қатаң түрде үлестіріледі. Басқа модельдерде бір мезгілде екі компьютерде мәліметтерді басқару (үлестірілген МҚ моделі), ақпаратты өңдеуден өткізу (үлестірілген қызмет моделі), ақпаратты ұсыну (үлестірілген ұсыну моделі) қызметтерінің біреуі орындалады. Алдымен аса кең тараған түрлері ретінде мәліметтерге қол жеткізу мен қашықтан ұсыну (МҚ серверін) модельдерін қарастырып кетейік. Мәліметтерге қашықтан қол жеткізу моделінде (Remote Data Access – RDA), ақпаратты ұсыну қызметтері мен қолданбалы өңдеудің логикасын жүзеге асыратын программалар біріктірілген компьютер-клиенте орындалады.

Мәліметтерді басқару сервисіне жүгіну ісі операторлардың көмегімен SQL тілінің операторлары арқылы немесе API (Application Programming Interface – қолданбалы программалау интерфейсі) арнайы кітапхананың қызметтерін шақырту арқылы орындалады. RDA-моделінің негізгі артықшылығы – SQL-интерфейстері бар дайын МҚБЖ-лардың көптеп болуы және клиенттік бөлік программаларының жылдам түрде құрылуын қамтамасыз ететін аспаптық құралдардың болуы. Жасау құралдары көбінесе MS Windows-те пайдаланушының графикалық интерфейсі, ODBC интерфейсінің стандартты мен кодты автоматты түрде генерациялау құралдарын қолдайды. Жасау құралдарының басым көпшілігі төртінші буынның тілдерін пайдаланады.

RDA-моделінің кемшіліктері біріншіден, барлық логика қосымшада шоғырланып, өңдеуден өткізілетін мәліметтер қашықтағы торапта орналасқандықтан мәліметтерді беру жүйесі жоғары дәрежеде жүктемеленгендігі болып табылады. Қосымшалардың жұмыс істеуі кезінде жеделі арқылы әдетте тұтастай МҚ-лар беріледі.

Екіншіден, RDA моделіне негізделуімен құрылған жүйелер өздерін жасау, өзгерту және ілесе жүру жағынан ыңғайсыз болып



келеді. Оның негізгі себебі – алынатын қосымшаларда қолданбалы қызметтер мен ұсыну қызметтері өзара тығыз байланысқан. Сол себептен жүйенің қызметі шамалы ғана өзгергеннің өзінде оның бүкіл қолданбалы бөлігін өзгерту қажет болады, бұл жүйені жасау мен модификациялауды қиындатады.

МҚ серверінің моделінің (DataBase Server – DBS) алдыңғы модельден айырмашылығы – компьютер-клиенттің қызметтері ақпаратты кейіптеу қызметтерімен шектеледі, ал қолданбалы қызметтер компьютер-серверде орналасқан қосымшамен қамтамасыз етіледі. Бұл модель RDA-модельге карағанда аса технологиялық болып табылады және Ingress мен Oracle тәрізді МҚБЖ-ларда қолданылады. Бұл жағдайда қосымшалар сақталатын процедуралар түрінде орындалады.

Процедуралар әдетте МҚ сөздігінде сақталады және оларды бірнеше клиенттер бөліседі. Жалпы жағдайда сақталатын процедуралар компиляция мен интерпретациялау режимдерінде орындалады.

DBS моделінің артықшылығы жасау, ілесе жүру және модификациялау кезеңдерінде қосымшаларды орталықтандырылған түрде жақсы әкімшілік ету мүмкіндігі, сондай-ақ есептеу және коммуникациялық ресурстарды тиімді пайдалану болып табылады. Соңғысына программаларды ұжымдық пайдалану режимінде орындау ісінің мәліметтерді желімен жіберуге едәуір аз шығындарды қажет ететінімен қол жеткізіледі.

DBS моделі кемшіліктерінің бірі – сақталатын процедураларды жасау құралдарын шектеумен байланысты. Негізгі шектеу – операторлардың сақталатын процедураларды нақты МҚБЖ-ға күштілеп байланыстыруы. Сақталатын процедураларды жазу тілі SQL тілінің процедуралық кеңейтілуі болып табылады және өзінің мағыналық құралдары мен қызметтік мүмкіндіктері бойынша С және Pascal тәрізді үшінші буындағы дәстүрлі тілдермен бәсекелесе алмайды. Сонымен қатар көптеген МҚБЖ-ларда сақталатын процедураларды бағлап реттеу мен тестілеудің қанағаттанарлық құралдары жоқ, бұл олардың механизмін қауіпті құралға айналдырады – реттелмеген программалар МҚ-ның қателесуіне, жүйенің жұмысы кезінде серверлік және клиенттік программалардың текке тұрып қалуына және т.б. әкеп соқтыруы мүмкін.

DBS-моделінің тағы бір кемшілігі – ЭЕМ-ның есептеу ресурстарын пайдалану тиімділігінің төмен болуы, себебі компьютер-сервердің программаларына қатысты сұраныстардың кіру ағынын басқару ісін ұйымдастыруды, сондай-ақ процедуралардың өзге компьютер-серверлерге ауысуды қамтамасыз ету мүмкін емес.

Үлестірілген түрде ұсыну моделінде қуатты компьютер-сервер бар, ал жүйенің клиенттік бөлігі төмендеп кеткен. Клиенттік бөліктің қызметі жай ғана ақпаратты монитордың экранында көрсету және негізгі компьютермен жергілікті желі арқылы байланысы болып табылады.

Осы тектес МҚБЖ-лар бұлайша айтқанда X-терминалдардың жұмысын қолдайтын желілерде орын алуы мүмкін. Оларда негізгі компьютер (хост-машина) бірнеше X-терминалдарға қызмет көрсету үшін жеткілікті мөлшердегі қуаттылыққа ие болуы тиіс. X-терминалдың да жеткілікті дәрежеде жылдам жұмыс істейтін процессоры және жеткілікті мөлшердегі жедел жады болуы тиіс (дискілік жинақтағыштары жоқ). Көбінесе X-терминалдарды RISC-компьютерлер (restricted [reduced] instruction set computer)-командалардың қысқартылған жинағынан тұратын компьютерлер негізінде құрылады. Бүкіл программалық қамтамасыз ету хост-машинада орналасқан. Ұсынуды басқару қызметтерімен желілік қызметтерді орындайтын X-терминалды программалық қамтамасыз ету X-терминал қосылған кезде желі арқылы серверден жүктеледі.

Үлестірілген түрде ұсыну моделі кіші, орташа және үлкен ЭЕМ-дарда жұмыс істеген алғашқы буындағы МҚБЖ-ларға ие болған болатын. X-терминалдар рөлінде дисплейлік станциялар мен абоненттік пункттер (жергілікті және қашықтағы) қызмет етті. Бұл жағдайда ақпаратты ұсыну қызметтерінің негізгі бөлігін МҚБЖ-лардың өздері орындады, ал пайдаланушының терминалдарында бейнелерді соңғы деңгейде құрастыру ісі шеткі құрылғыларда орындалды.

Үлестірілген ұсыну моделі бойынша МҚ пайдаланушыларына гетерогендік (біртекті емес) ортада қызмет көрсету жүйелері құрылған. Мұндай жүйелердің серверлік бөлігі әдетте қандай да бірегейлендірілген интерфейсті қамтамасыз етеді, ал клиенттік бөліктері шеткі құрал-жабдықтың ерекшелігін немесе ақпаратты ұсынудың бір форматын өзге бір форматқа түрлендіру ерекшелігін ескеру қызметтерін орындайды. Үлестірілген ұсыну моделі есептеу

ресурстарын басқарудың орталықтандырылған сызбасын жүзеге асырады. Осыдан оның басты артықшылықтары шығады – жүйеде қызмет көрсету мен соған қол жеткізуді басқару ісінің карапайымдылығы және салыстырмалы түрде арзан (шеткі терминалдардың бағасы онша жоғары болмағандықтан) болуы. Модельдің кемшіліктері – орталық тораптың сенімділігі онша жоғары болмауынан жүйенің осалдығы, сондай-ақ клиенттердің саны көп болғанда сервердің өнімділігіне қатысты қойылатын жоғары талаптар.

Үлестірілген қызмет моделінде мәліметтерді өңдеуден өткізудің логикасы екі торап бойынша таратылған. Мұндай модель өздерінде қолданбалы қызметтердің жалпы бөлігі компьютер-серверде, ал ақпаратты өңдеудің арнайы қызметтері компьютер-клиентте орындалатын АЖ-ларда болуы мүмкін. Жалпы сипаттағы қызметтерге мәліметтердің тұтастығын стандартты түрде, мысалы, сақталатын процедуралар түрінде қамтамасыз ету ісі жатуы мүмкін, ал қалған қолданбалы қызметтер арнайы қолданбалы өңдеуді орындайды. Осыған ұқсас модельге сонымен қатар бірнеше біртекті емес МҚ-лардағы ақпаратты қолданатын АЖ-лар ие болады.

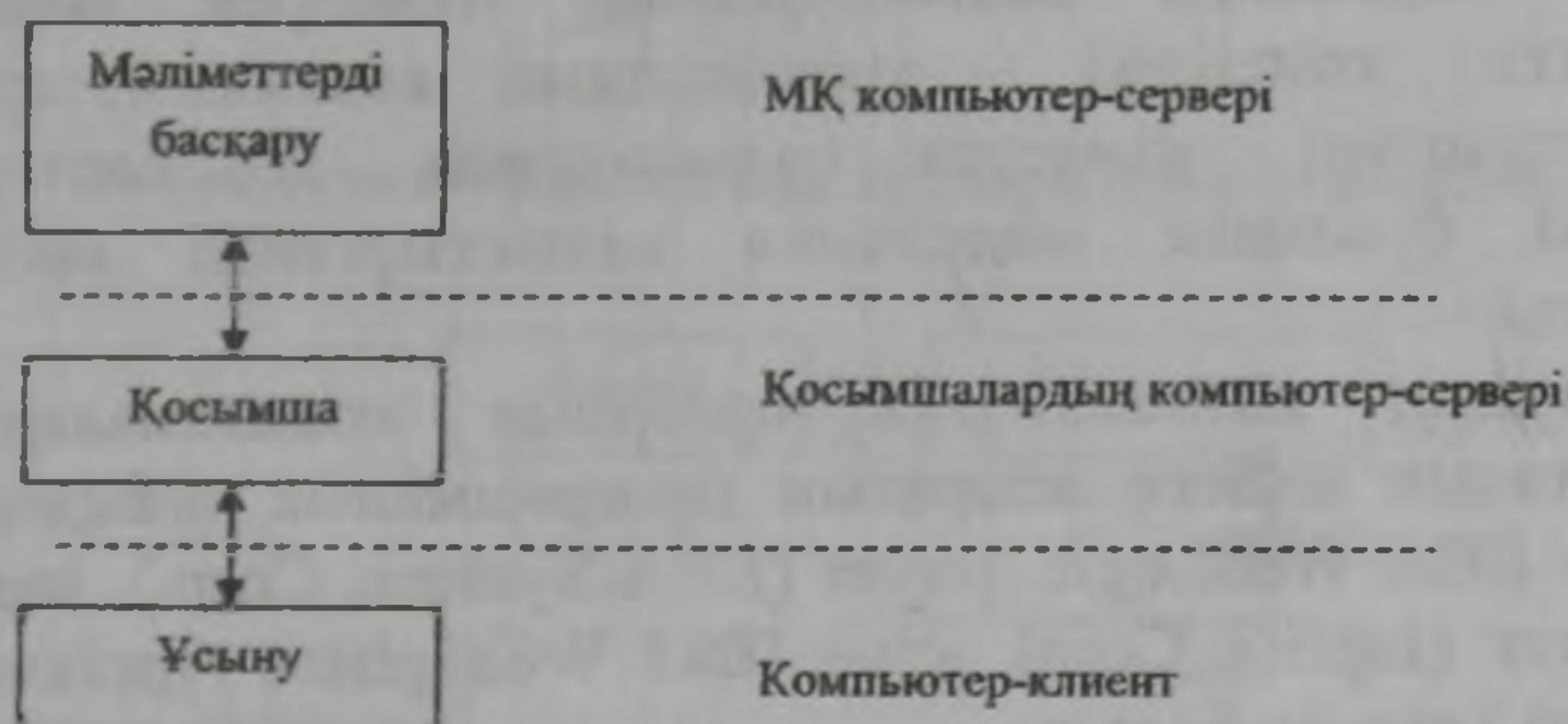
Үлестірілген МҚ моделі қуатты компьютер-клиентті пайдалануды қажет етеді, бұл ретте мәліметтер компьютер-клиентте және компьютер-серверде сақталады. Екі мәліметтер қорының өзара байланысы екі түрде болуы мүмкін: а) біртұтас МҚ-ның жекелеген бөліктері жергілікті және қашықтағы қорларда сақталады; б) жергілікті және қашықтағы МҚ өзара үйлестірілген көшірмелер болып табылады.

Үлестірілген МҚ моделінің артықшылығы – соның негізінде құрылатын АЖ-лардың икемділігі, олар компьютер-клиентке жергілікті және қашықтағы МҚ-ларды өңдеуден өткізуге мүмкіндік береді. Көшірмелердің сәйкестігін үйлестіру механизмдері болған жағдайда жүйе тұтастай алғанда жоғары дәрежедегі өміршендігіне ие болады, себебі клиент пен сервердің өзара қосылысының үзілуі жүйенің күйреуіне әкелмейді, оның жұмысы қосылыс қайта жасалғаннан кейін қалпына келуі мүмкін. Моделдің кемшіліктеріне компьютер-клиенттерде саны көп бірдей қосымшаларды орындау кезіндегі артық шығындарды жатқызуға болады.

Қызметтерді үлестірудің үш буындық моделі қосымшаның үш қызметінің әрқайсысы жекелеген компьютерде орындалатын типтік нұсқа болып табылады. Қосымшалардың қызметтерін

компьютерлердің көптеген санына үлестірудің нұсқалары болуы мүмкін, алайда олар сирек қолданылатын болғандықтан, қарастырылмайды.

Біз қарастырып отырған модель қосымшалар сервері немесе AS-модель (Application Server) деп аталады және 1.8-суретте көрсетілген.



1.8-сурет – Үшбуынды модель

Ақырғы пайдаланушымен сұхбат жүргізуге жауап беретін үш буындық AS-модельге сәйкес, аталған процесс, әдетте, ақпаратты ұсыну қызметтерін орындайды және қосымшаның компонентімен DBS моделіндегіге ұқсас түрде әрекеттеседі. Қосымшаның компоненті жеке компьютерде орналасып өз кезегінде RDA моделіне ұқсас түрде мәліметтерді басқару компонентімен байланысты болады.

AS-моделінің орталық буыны қосымшалар сервері болып табылады. Қосымшалар серверінде бірнеше қолданбалы қызметтер орындалады, олардың әрқайсысы оны қажет ететін программалардың барлығына қызмет көрсету қызметі ретінде рәсімделген. Қосымшалар серверлерінің бірнешеуі болуы мүмкін, олардың әрқайсысы өз қызметін ұсынады. Қосымшалар серверінен қызмет көрсетуін сұраған кез келген программа ол үшін клиент болып табылады. Клиенттерден серверлерге келіп түскен сұраныстар кезекке тұрады, олар ол жерден қандай да бір пәнге байланысты, мысалы, басымдылықтары бойынша іріктеліп алынады.

Ұсыну қызметтерін орындайтын және қосымшалар сервері үшін клиент болып табылатын компонент бұл модельге әдеттегіден

кеңірек түсіндіріледі. Ол ақырға пайдаланушы бар интерфейсті ұйымдастыру үшін қызмет етуі, құрылғылардан, мысалы, сигнал бергіштерден мәліметтерді қабылдауды қамтамасыз етуі, немесе еркін түрдегі программа болуы мүмкін.

AS-модельдің артықшылығы – қосымшаның қызметтері өзара тәуелсіз үш бөлікке бөлінгендіктен, оның икемділігі мен әмбебаптығы болып табылады. Көптеген жағдайларда бұл модель екі буындық модельмен салыстырғанда тиімдірек болады. Модельдің негізгі кемшілігі – қосымшаның компоненттерінің арасында ақпаратпен алмасуға жұмсалатын компьютердің ресурстары екі буындық модельмен салыстырғанда көбірек жұмсалатындығы.

Қосымшалардың компьютер-серверлерінде қосымшалардың қызмет ету ортасын жүзеге асыратын программалық өнімдердің мысалы ретінде BEA WebLogic Server (BEA Systems Corp.), Inprise Application Server (Inprise Corp) және IBM WebSphere Application Server (IBM Corp.) атауға болады.

1.3 Мәліметтер қоры және Интернет

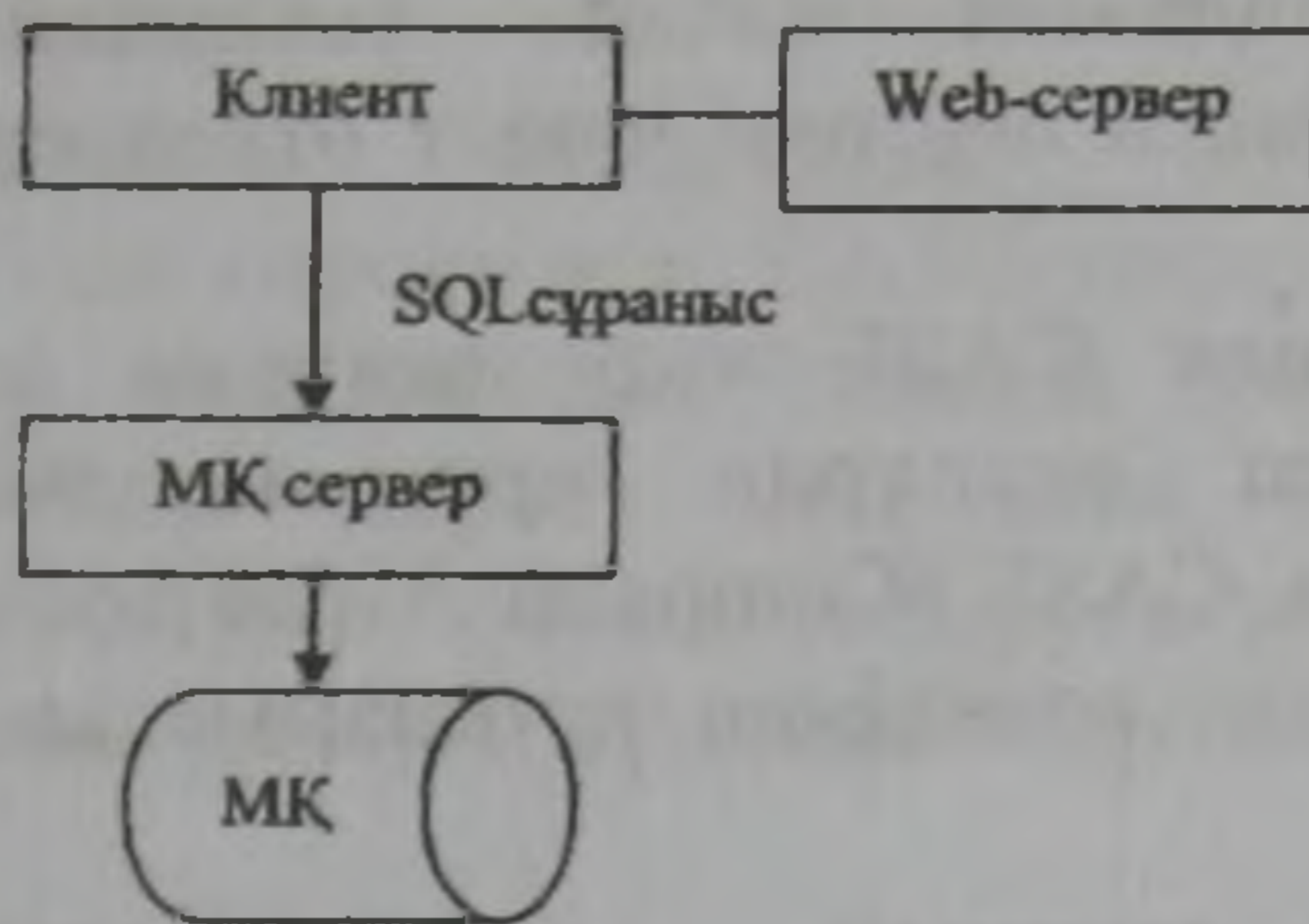
Интернет пен Интернеттегі ақпараттар көзі мәліметтер қорлары болған жағдайда Web-серверлер МҚБЖ компоненттерінің өзара әсері орын алады. Мәліметтер қорларын пайдаланатын программалық компоненттердің жұмыс істеуін ұйымдастырудың үш түрін ажыратады:

- a) клиент жағында;
- b) Web-сервер жағында;
- c) Қосымшалар сервері жағында.

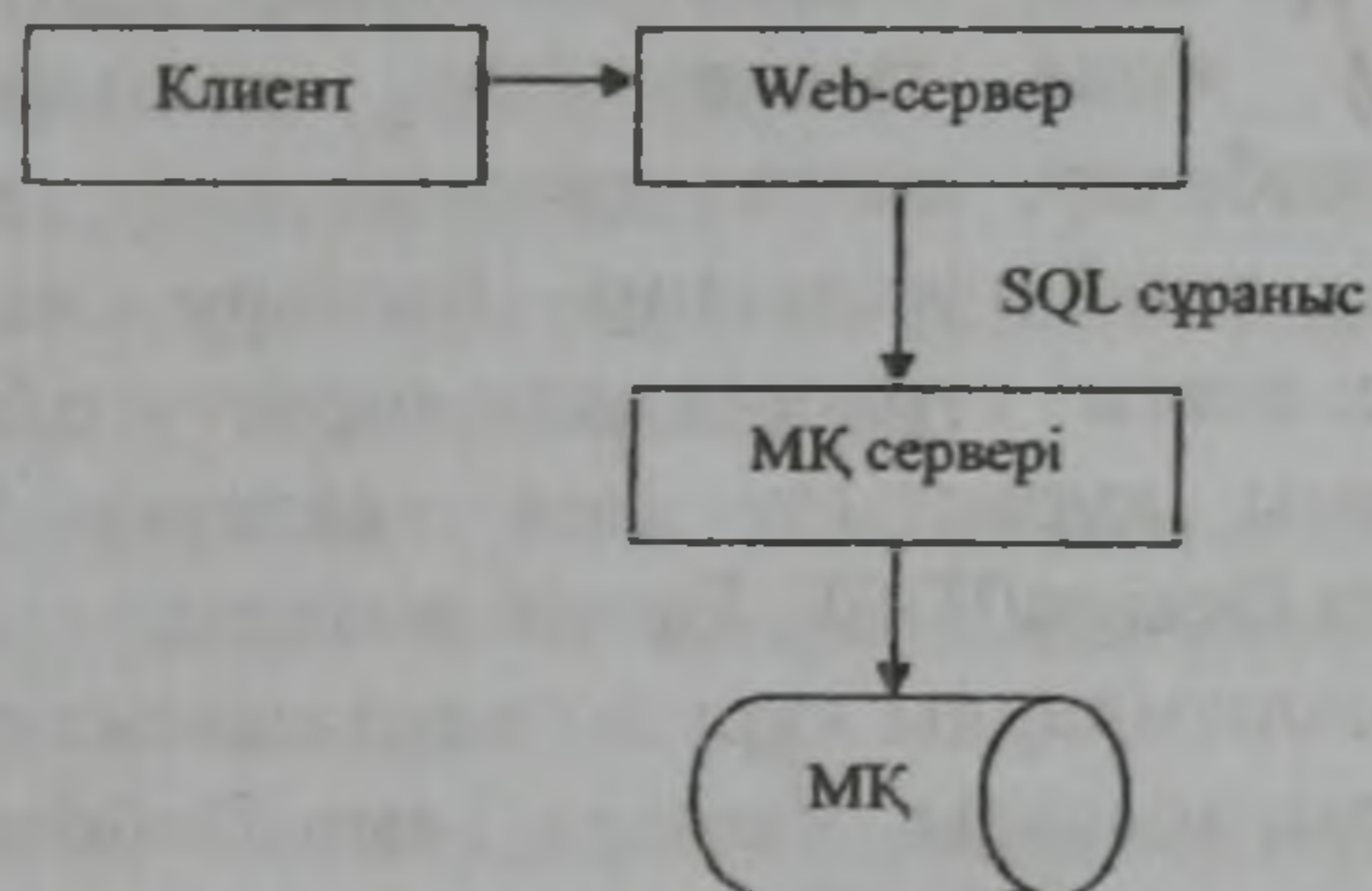
Клиент жағындағы мәліметтер қорына қол жеткізу кезінде Java тілі клиент пен сервердің өзара әсерінің негізгі құралы болады. Бұл жағдайда HTML тілінің мүмкіндіктерін кеңейтетін JavaScript, Jscript, VBScript сценарийлер тілдері пайдалануы мүмкін. Өзара әсерлер схемалары келесі 1.9-суретте көрсетілген.

МҚ серверіне жүгіну үшін Java – программаларының ішінен жұмыс принципі ODBC ұқсас JDBC хаттамасы пайдаланылады.

Web-сервер жағындағы МҚ қол жеткізу SQL тілінде МҚ өзара әсер ететін Web-сервердің сыртқы программаларын шақыру арқылы жүзеге асырылады, ол келесі 1.10-суретте келтірілген.

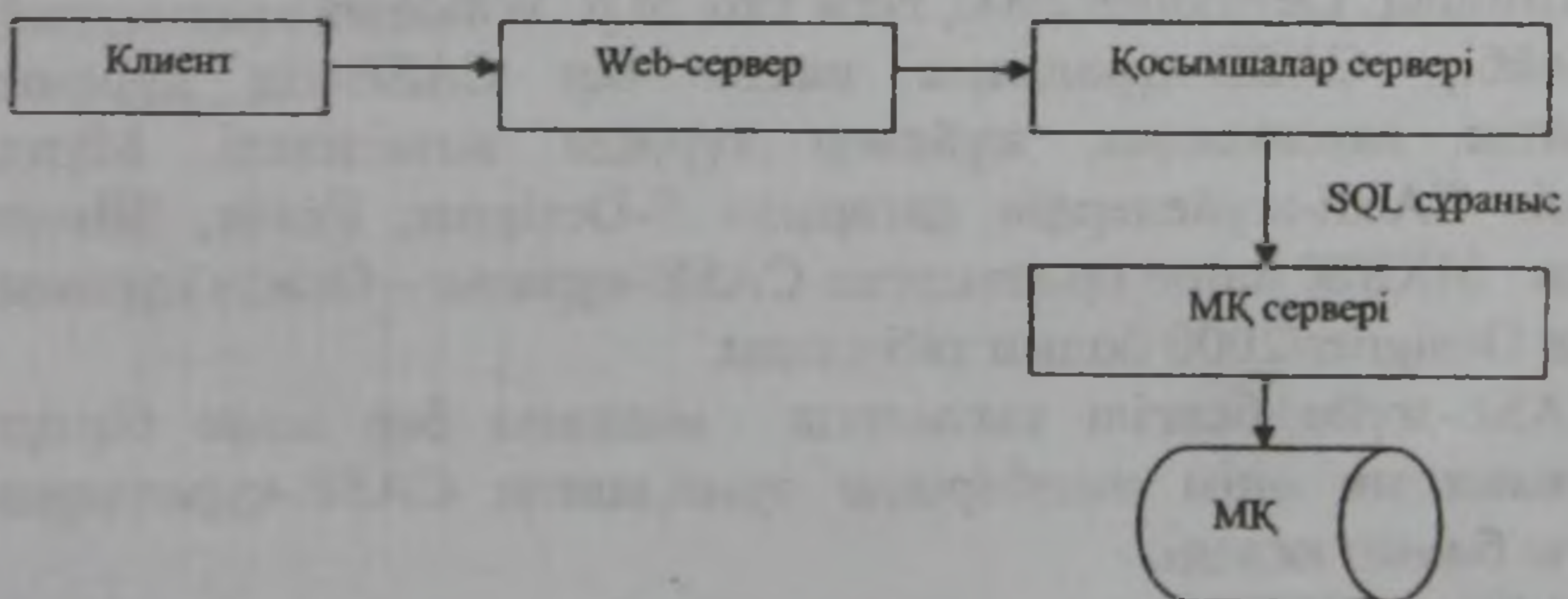


1.9-сурет – Клиент пен сервердің өзара әсерлер схемасы



1.10-сурет – Web-сервер жағынан MQ-на қол жеткізу

Келесі 1.11- суретте қосымшалар сервері жағындағы мәліметтер қорына қол жеткізу мүмкіндігі көрсетілген.



1.11-сурет – Қосымшалар сервері жағынан мәліметтер қорына қол жеткізу

Таратылған қосымшаларды жасаудың негізгі тілі – Java тілі болып табылады.

70-80 жылдардан бері колданылып келе жатқан әзірленген жүйені және қабылданып жатқан техникалық шешімдерді сипаттаудың ресмиленген әдістері уақыт өткен сайын көп еңбекті кажет етеді.

Соның нәтижесінде CASE-технологиясын жүзеге асыратын CASE-құралдары деп аталатын программалық-технологиялық құралдар пайда болды. CASE (Computer Aided Software Engineering) терминін компьютердің көмегімен программалық камтуды жасау деп аударуға болады.

CASE-құралдары дегеніміз – бұл ақпараттың жүйелерді құрастыру және оларға ілесе жүру үдерістерін қолдайтын программалық құралдар, оларға келесілер жатады: талаптарды талдаудан өткізу және тұжырымдау, мәліметтер қоры мен қосымшаларды жобалау, кодты генерациялау, тестілеу, сапаны қамтамасыз ету, конфигурацияны басқару және т.б. CASE-құралдарын келесі негізгі түрлерін атап көрсетуге болады:

- пәндік саланы құрастыру мен талдауға арналған талдау құралдары. Оларға Design/IDEF, Vpwin жатады;

- жобалық арналымдарды құруды қамтамасыз ететін талдау мен жобалау құралдары, мысалы, Vantage Team Builder, Silvertun, PRO-IV;

- мәліметтер қорын модельдеп, сұлбаларды жасауға мүмкіндік беретін мәліметтер қорын жобалау құралдары. Оларға Erwin, S-designot, DataBase Designer жатады;

- қосымшаларды жасау құралдары, мысалы, Uniface, JAM, PowerBuilder, Developer/2000, New Ero, SQL Windows.

Кейбір CASE-құралдары нақты бір CASE-тің құрамына кірмейтін автономдық жүйелер түрінде жеткізіледі. Мұндай тәуелсіз CASE-жүйелердің қатарына S-Designer, Erwin, Silvertun жатады. МҚБЖ ішіне орнатылған CASE-құралы – Oracle құрамына кіретін Designer/2000 болып табылады.

CASE-жүйе белгілі қызметтік мақсаты бар және біртұтас программалық өнім шеңберінде орындалған CASE-құралдардың жинағы болып келеді.

CASE-жүйелері мен құралдарының негізгі мақсаты – программалық қамтамасыз етудің жобалануын, оны кодтау мен кейінгі жасау кезендерінен бөліп алу, сондай-ақ программалық

жүйелерді құрудың бүкіл үдерісін автоматтандыру болып табылады.

CASE-технология ақпараттық жүйелерді жобалаудың әдіснамасы ретінде көрсетіледі. CASE-технологиясы дегеніміз -- бұл сонымен қатар тақырыптық саланы үлгілеуге, оның үлгісін ақпараттық жүйені әзірлеу мен оған ілесе жүрудің барлық кезеңдерінде талдаудан өткізіп отыруға және пайдаланушыларға арнап қосымшаны жасауға мүмкіндік беретін саймандық құралдар.

Бақылау сұрақтары

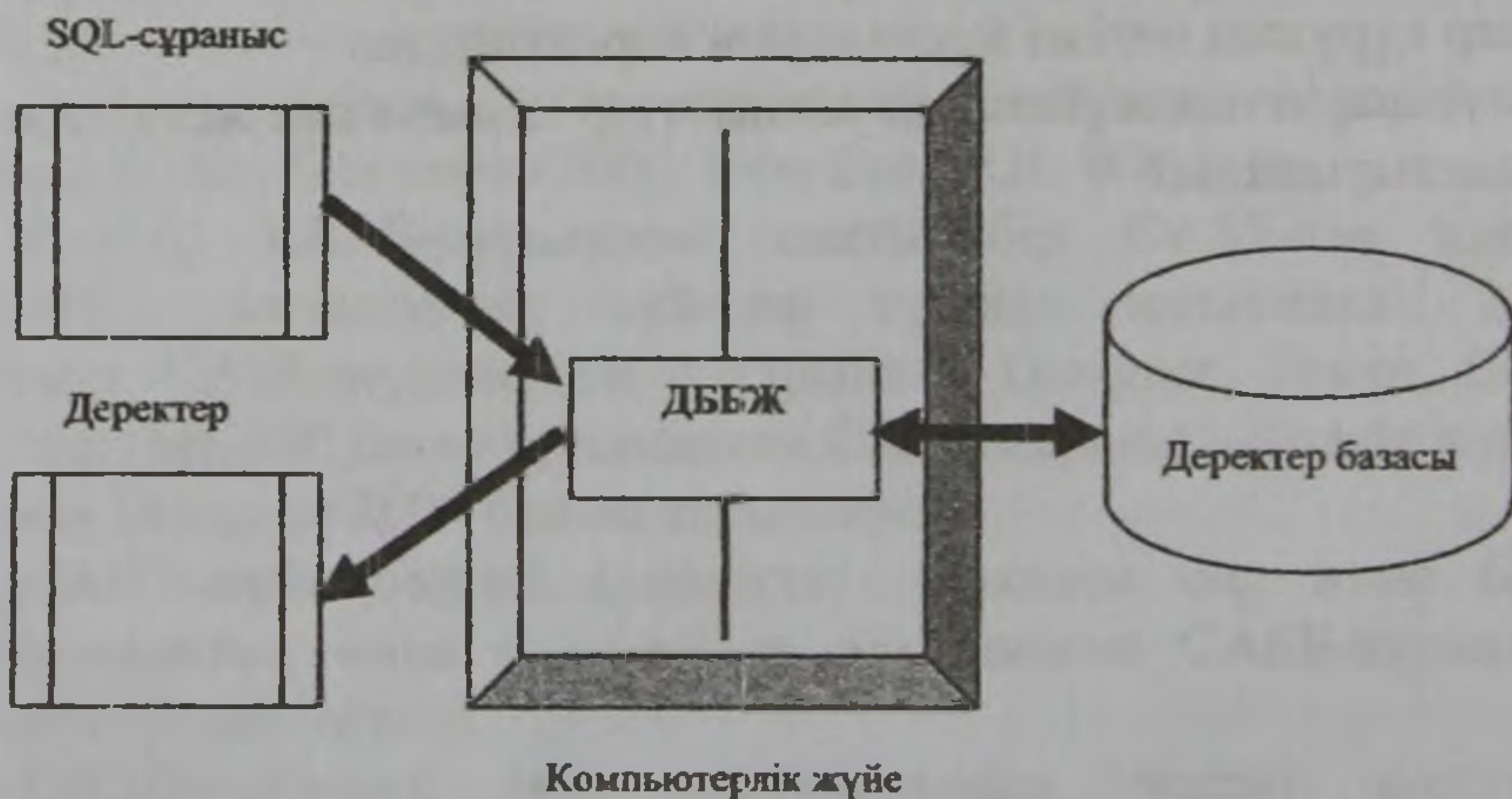
- 1 МҚ әкімшісінің қызметіне не жатады?
- 2 МҚ әкімшісі нені орындайды?
- 3 Ақпараттық жүйе дегеніміз не?
- 4 Мәліметтер қоры серверінде не орналасады?
- 5 Ақпараттық жүйеге не кірмейді?
- 6 Қандай ақпараттық жүйе жергілікті деп аталады?
- 7 МҚ-мен жұмыс істейтін, ақпараттық жүйелерді программалық қамтамасыз етуге не жатады?
- 8 Екі буынды модельде компьютер-сервер мен компьютер-клиент арасындағы қызметтер бөлінуінің нұсқаларын атап көрсетіңіздер.
- 9 Мәліметтер қорын қолданылатын жергілікті ақпараттық жүйелер құрудың негізгі нұсқаларын көрсетіңіздер.
- 10 Ақпараттық жүйелерде мәліметтер қорына қол жеткізу қалай ұйымдастырылады?

2 Деректер базасының стандартты SQL тілі

2.1 SQL тілі

Қазіргі заманғы компьютерлік өнеркәсіпте SQL тілі ең маңызды тенденция болып отыр. Соңғы жылдары SQL тілі деректер базасының жалғыз тілі болды. Бүгінгі таңда SQL дербес компьютерлерде жұмыс істейтін ДББЖ-нің жүзден астамын қолдайды. SQL тілі деректер базасын басқару жүйелерінің архитектурасындағы маңызды буын болып табылады.

SQL (Structured Query Language) - бұл құрылымдық сұраныстар тілінің қысқартылған атауы. Ол реляциялық деректер базасында мәліметтерді құру және өңдеу құралдарын береді. Компьютерлік технологиялардан ерекшелігінен тәуелсіздігі және SQL-ді реляциялық мәліметтер базасының технологиясы облысындағы өндіріс жетекшілерімен қолдауы оны деректер базасының стандартты тіліне айландырды. SQL тілі деректер базасының реляциялық типімен ғана жұмыс істейді. Төменде SQL жұмысының сұлбасы бейнеленген (2.1-сурет). Есептегіш жүйеде маңызды ақпарат сақталған деректер базасы болады. Егер есептегіш жүйе кәсіпорынның кадрлар бөліміне қатысты болса, онда деректер базасында кәсіпорында қызмет ететін адамдар жайлы ақпарат (аты-жөні, туған жылы, телефоны, қызметі және т.б.) сақталуы мүмкін.



2.1-сурет – SQL жұмысының сұлбасы

Деректер базасын басқаратын компьютерлік программа *деректер базасын басқару жүйесі* (ДББЖ) деп аталады. Егер қолданушыға деректер базасындағы мәліметтерді оқу қажет болса, онда оларды SQL көмегімен ДББЖ-нен сұрау керек. ДББЖ сұраныстарды өңдейді, қажет болған мәліметтерді тауып оларды қолданушыға жібереді. Мәліметтерді сұрау және нәтиже алу процесі деректер базасында сұраныстар деп аталады, яғни *сұраныстардың құрылымдық тілі*.

Реляциялық ДБ пайда болғанға дейін көптеген ДББЖ үшін құрылған барлық мәліметтерді басқару тілдері, файлдардың логикалық жазулары түрінде берілген мәліметтермен жүргізілетін операцияларға бағытталған. Әрине, бұл қолданушыдан мәліметтердің сақталуын, қандай мәліметтер қажеттігін, олардың қайда орналасқанын және қалай алу жолдарын білуді талап етеді.

Деректер базасында қолданылатын SQL тілі логикалық байланысқан кесте-қатынастар жиынтығы түрінде берілген мәліметтермен операциялар жүргізуге бағытталған. Осы тілдің құрылымының ерекшелігі өңдеу процедурасына емес, мәліметтерді өңдеудің соңғы нәтижесіне бағытталуында. SQL тілі мәліметтердің, индекстердің қайда орналасқанын, және нәтиже алу үшін қандай операциялар тізбегін пайдалану керектігін өзі анықтайды, сондықтан бұларды ДБ сұраныстарында көрсетудің қажеті жоқ.

Реляциялы ДБ-сы теориясының пайда болуы екі кластарға бөлуге болатын, сұраныстар тілінің қатарын өндіруге түрткі болды:

- қатынастарға қолданылатын, сұраныстарды арнайы операторлармен бейнелеуге мүмкіндік беретін алгебралық тілдер;
- предикаттарды санау тілдері, берілген қатынастар жиынтығынан жаңа қатынасты анықтайтын, ережелер жиынтығы. Яғни, ДБ-ғы сұранысқа жауап ретінде алынатын қатынастарды анықтау әдісі.
- қазір бұл тіл ДББЖ-сі қолданушыға ұсынатын барлық функциональды мүмкіндіктерді іске асыру үшін қолданылады.

Атап айтқанда:

- деректерді ұйымдастыру. SQL қолданушыға мәліметтерді ұсыну құрылымын өзгерту, сонымен қатар деректер базасының элементтері арасындағы қатынастарды орнату мүмкіндігін береді;

- деректерді оқу. SQL қолданушыға немесе қосымшаға деректер базасынан онда сақталған мәліметтерді оқу және оларды қолдану мүмкіндігін береді;
- деректерді өңдеу. SQL қолданушыға немесе қосымшаға деректер базасын өзгерту, яғни оған жаңа мәліметтерді қосу, сонымен қатар онда бар мәліметтерді жаңарту немесе өшіру мүмкіндігін береді;
- рұқсат алуды басқару. SQL көмегімен қолданушыға мәліметтерді оқу және өзгерту мүмкіндіктерін шектеуге және оларды бекітілмеген рұқсаттан қорғауға болады;
- деректерді біріктіріп қолдану. SQL бір-біріне кедергі жасамас үшін қарама-қарсы жұмыс істеп отырған қолданушыға деректерді біріктіріп қолдануды үйлестіреді;
- деректердің бүтіндігі. SQL келісілмеген өзгертулерден немесе жүйенің қабыл алмауын бұзғандардан қорғай отырып, деректер базасының бүтіндігін қамтамасыз етуге рұқсат етеді;
- сондықтан SQL ДББЖ-мен әрекеттесу үшін жеткілікті түрдегі қуатты тіл болып табылады. Сонымен қатар, SQL тілі бүгінгі таңда реляциялық деректер базасымен жұмыс істеуге арналған жалғыз стандартты тіл болып отыр.

SQL тілінің жетістіктері. SQL – деректерді басқаруға арналған әмбебап программалық құрал. SQL тілінің жетістіктерінің негізгі ерекшеліктерін атап өтейік:

- нақты ДББЖ-нен тәуелсіздігі;
- бір есептеуіш жүйеден басқасына ауысатындығы;
- стандарттардың барысы;
- IBM компаниясымен мақұлдау;
- Microsoft компаниясы жағынан қолдау;
- реляциялық негіз;
- ағылшын тілін еске түсіретін жоғарғы деңгейлі құрылым;
- арнайы интерактивті сұраныстарды орындау мүмкіндігі;
- деректер базасына программалық рұқсат алуды қамсыздандыру;
- деректерді әртүрлі ұсыну мүмкіндігі;
- деректер базасымен жұмыс істеуге арналған толық құнды тіл;
- деректерді динамикалық анықтау мүмкіндігі;
- клиент-сервер архитектурасын қолдау.

Жоғарыда аталған факторлардың барлығы дербес компьютерлерде деректерді басқаруға арналған SQL тілінің стандартты құрал болуының себептері болып табылады.

SQL -командаларының типтері. SQL -тілінің командалары бірнеше топқа бөлінеді. Командалардың негізгі типтері келесідей:

- *DDL(Data Definition Language)*- мәліметтердің анықталу тілі. Бұл топтың командалары мәліметтер қорының объектілерінің құрылымын құру және өзгерту(мысалы, кестелерді құру және жою үшін) үшін пайдаланылады.

- *DML(Data Manipulation Language)*- мәліметтерді басқару (манипуляция) тілі. DMLкомандалары мәліметтер қоры объектілеріндегі ақпараттарды басқару үшін пайдаланылады.

- *DCL (Data Control Language)*-мәліметтерді басқару тілі. Сәйкес командалар мәліметтер қорында сақталған ақпаратқа кіруді басқаруға арналған.

- *DQL (Data Query Language)*-тілі. Бұл жиі қолданылатын командалар, мәліметтер қорына сұранымды құруға арналған.(Сұраным дегеніміз сәйкес ақпаратты алу мақсатында мәліметтер қорына хабарласу.

- Мәліметтер қорын администрациялау командалары әрекеттердің орындалуына бақылауды жүзеге асыру және өндірілетін операцияларыды анализдеуге арналған.

SQL мәліметтерінің типі. Стандартты SQL- де пайдаланылатын мәліметтер типтерін келесі топтарға бөлуге болады:

- жолдық типтер;
- сандық типтер;
- дата мен уақытты көрсетуге арналған типтар.

Бұя типтердің нақтырақ сипаттамасы.

Жолдық типтер. SQL-де екі жолдық типтер анықталған:

- белгіленген ұзындықты символдық жолдар;
- айнымалы ұзындықты символдық жолдар.

Белгіленген ұзындықты символдық жолдар. Белгіленген ұзындықты символдық жолдар түрінде сақталған мәліметтер, өріске енгізілген жолдың нақты өлшемінен тәуелсіз, жадының бір ғана көлемін алады. ANSI-SQL-92-ге сәйкес белгіленген ұзындықты символдық жолдарды хабарлау түрі келесідей: Character(n).

Мұндағы n бұл хабарлау қатысты болатын өріс өлшемін анықтайтын жол ұзындығы.

Белгіленген ұзындықты жолдарды пайдаланғанда, әдетте, бос жолдар пробелдармен толтырылады. Мысалы, егер өлшемі 10 тең өріс беріліп, ал оған 3 символдан тұратын жол енгізілсе, онда 7 символ пробелдармен толтырылады.

Айнымалы ұзындықты символдық жолдар. Айнымалы ұзындықты жолдың ұзындығы барлық мәліметтер үшін тұрақты емес, ол мәліметтер қорының кестесі өрісінде сақталған жолдың нақты өлшеміне тәуелді. Айнымалы ұзындықты жолды хабарлау келесі түрде болады: `Varchar(n)`, n -максималды мүмкіндікті жолды анықтайтын сан.

`Character` типінен айырмашылығы `Varchar` пайдалану дискідегі кеңістікті үнемді шығындауды қамтамасыз етеді. Хабарландыруда көрсетілген жол өлшеміне қарамастан, өріс, енгізілген ақпаратты сақтауға қанша орын қажет болса, сонша алады. Мысалы, егер `Varchar(10)` өрісі хабарланып және оған 3 символдан тұратын жол енгізілсе, онда бұл жолды сақтау үшін белгіленген ұзындықты жол жағдайындағы сияқты 10 емес, 3 байт қана пайдаланылады.

Сандық типтер. Сандық типтер былай бөлінеді:

- бүтін санды типтер;
- белгіленген нүктелі заттық типтер;
- жүзуші нүктелі заттық типтер;
- белгіленген және айнымалы ұзындықты екілік жолдар;

Бүтін санды типтер. ANSI SQL стандартымен екі бүтін санды типтер орнатылады:

- `INTEGER-4` байтты пайдаланатын таңбалы бүтін сан. 2147483648-ден 2147483647-ге дейінгі диапазондағы сандарды бере алады;
- `SMALLINT-2` байтты пайдаланатын таңбалы қысқа бүтін сан. 32768-ден 32647-ге дейінгі диапазондағы бүтін сандарды бере алады.

Белгіленген нүктелі заттық типтер. Белгіленген нүктелі заттық типтер бөлшек сандарды нақты беру үшін арналған. Көп жағдайда бұл типтер қателіктер жіберілмейтін екілік формада жүзгімелі нүктелі заттық сандарды беруде пайдаланылады. (Мысалы, ақшалай шамадағы мәндерді сақтауда). Белгіленген

нүктелі заттық типтер шын мәнісінде ондық нүкте түрінде бейнеленетін бүтін санды типтер болып табылады.

Белгіленген үтірлі типті хабарлау синтаксисі келесідей: DECIMAL (n.m), мұндағы n-дәлділік; m-масштаб

Дәлділік -ол сандық мәннің жалпы ұзындығы, масштаб-ол ондық нүктеден оң жақта орналасқан белгілер саны.

Жүзгімелі нүктелі заттық типтер. Жүзгімелі нүктелі типтер әдетте ғылыми және инженерлік есептеулерде пайдаланылады. Бұл типтерді пайдалануда қандай да бір санды жүзгімелі нүктелі екілік формаға өзгерту кезінде мәліметтер қорына енгізуде үнемі қандай да бір қателік жіберіледі. Бұл қателік өте аз болса да, кейбір жағдайларда кешірілмейді және үлкен қателік әкелуі мүмкін, мысалы, үлкен санды мәндерді сомалауда. Сондықтан, жүзгімелі нүктелі типтер ақшалай шамалы мәндерді сақтауға қолданылмайды.

Көп жағдайда екі жүзгімелі нүктелі заттық типтер пайдаланылады:

- FLOAT-бірлік дәлділікті сандар;
- DOUBLE-екілік дәлділікті сандар.

Екілік жолдар. Екілік жолдар салыстырмалы сирек қолданылады. Әдетте мұндай типті өрістер жалаулар немесе екілік маскалар ретінде қолданылады.

Символдық жолдар сияқты, екілік жолдар белгіленген және айнымалы ұзындықты болады. Белгіленген ұзындықты екілік жолдар келесі жолмен хабарланады: BIT (n), мұндағы n-байтпен берілген жол ұзындығы.

Айнымалы ұзындықты жолды хабарлау түрі мынадай: BIT VARYING (n), мұндағы n- байтпен берілген жолдың максималды ұзындығы.

Дата мен уақытты берілген типтер. Аты айтып тұрғандай бұл типтер дата мен уақытқа қатысты ақпараттарды сақтау үшін пайдаланылады.

SQL-стандартында дата мен уақыт туралы ақпаратты сақтауға арналған келесі мәліметтер типтері анықталған:

- DATE-датаны сақтау үшін пайдаланылады;
- TIME-уақытты сақтау үшін пайдаланылады;
- TIME STAMP-дата мен уақытты сақтайды;
- INTERVAL-екі дата мен уақыт моментіндегі уақыт аралығын сақтайды.

2.2 Мәліметтер қорының объектілерін басқару

Мәліметтер қорының объектісі дегеніміз мәліметтер қорында анықталған және ақпаратты сақтауда немесе ақпаратқа хабарласуда пайдаланылатын кез-келген объект. Мәліметтер қорының объектісінің мысалдары ретінде кестелер, ұсыныстар мен индекстер қызмет етеді.

Мәліметтер қорының объектілерін басқару үшін SQLтілінің DDL командаларының жиындары пайдаланылады.

Кестелерді құру, түрлендіру және жою. Кесте, ақпаратты реляционды мәліметтер қорында сақтаудың негізгі объектісі болып келеді. Кестені құруда міндетті түрде кестедегі өріс аты, және өрістерге сәйкес мәліметтер типі көрсетіледі. Сондай-ақ, кестені құру кезінде өрістер үшін үндемей қабылданатын шекаралық шарттар мен мәндер ескертілуі мүмкін.

Шектік шарттар- мәліметтер қорының кестесінің өрісінде шама мәнін шектейтін ереже. *Үндемей қабылданатын мәндер*- жаңа жазу қосқанда, егер пайдаланушы осы өрістің мәнін көрсетпесе, мәліметтер қорының кестесінің өрісіне автоматты түрде енгізілетін шама.

CREATE TABLE операторы. Кестені құру үшін CREATE TABLE операторы пайдаланылады. Бұл оператордың синтаксисі келесі түрде болады:

CREATE TABLE кестенің атауы (Бағаның атауы_1 мәліметтердің типі. Бағаның атауы_2 мәліметтердің типі. Бағаның атауы_N мәліметтердің типі)

Мысал, Table_students кестесін құру операторы мына түрде болады:

```
CREATE TABLE STUDENTS
SNUM INTEGER.
SIMA CHAR (25).
SFAM CHAR (25).
SOTCH CHAR (25).
SROZH DATE.
SADR CHAR (25).
STEL VARCHAR (25).
```

ALTER TRABLE операторы. Құрылған кестені ALTER TABLE операторын пайдаланып түрлендірге болады. Осы оператор көмегімен кестеге жаңа өрістерді қосуға және жоюға, өрістер

мәліметтерінің типін өзгертуге, шектеулерді қосып немесе жоюға болады.

Жалпы түрде ALTER TABLE операторының синтаксисі былай болады:

ALTER TABLE кестенің атауы [MODIFY][Бағаның атауы мәліметтердің типі]

[ADD][Бағаның атауы мәліметтердің тип]

[DROP][Бағаның атауы мәліметтердің тип]

ALTER TABLE операторымен орындалатын әрекет кесте атынан кейін көрсетіліп негізгі сөзбен анықталады:

- MODIFY- өріс анықтамасын береді;

- ADD- кестеге жаңа өріс қосады;

- DROP-кестеден өрісті жояды.

Өрістің мәліметтерінің типін өзгерту үшін ALTER TABLE операторының келесі синтаксисі пайдаланылады:

ALTER TABLE имя_таблицы ADD (бағаның атауы мәліметтердің тип)

Мысалы, STUDENTS кестесіне студенттің электронды почтасынан тұратын өрісті қосу үшін келесі операторды пайдаланған жөн:

ALTER TABLE STUDENTS ADD (E-mail CHARACTER (25))

Егер бар өрістің мәліметтерінің типін өзгерту қажет болса, ALTER TABLE операторымен бірге MODIFY негізгі сөзін пайдаланған жөн:

ALTER TABLE кестенің атауы MODIFY(Бағаның атауы мәліметтердің тип)

Мысалы, STUDENTS кестесіне E-mail өрісін енгізгеннен кейін CHARACTER типінің пайдаланылуының тиімсіздігі анықталады-студенттердің көбінде электронды почта жоқ, сондықтан дискілік кеңістіктің бөлігі пайдаланылмай, бос қалады. Бұл өріс үшін VARCHAR мәліметтер типін қолданған дұрыс. Мәліметтер типін өзгерту үшін ALTER TABLE операторын пайдаланамыз:

ALTER TABLE STUDENTS MODIFY(E-mail VARCHAR(25))

Бар өрісті өшіру үшін ALTER TABLE операторын DROP негізгі сөзбен бірге шақыруды орындаумен жүзеге асырылады:

ALTER TABLE кестенің атауы DROP (Бағаның атауы)

DROP TABLE операторы. Кестелерді жою үшін DROP TABLE операторы пайдаланылады. Бұл оператордың синтаксисі мына түрде болады:

DROP TABLE кестенің атауы [RESTRICT | CASCADE]

Егер DROP TABLE операторын шақыру кезінде RESTRICT негізгі сөзі пайдаланылып, және жойылатын кестеге қандай да бір ұсыныс немесе шектеу сілтелінсе, онда кестені жою операторы орындалуы кезінде кесте туралы хабарлама шығарылады (генерацияланады). Егер де CASCADE негізгі сөзін пайдаланса, онда кестені жою орындалады және кестемен бірге оған сілтелінетін ұсыныстар мен шектеулер де жойылады.

2.3 Индекстарды құру және жою. Ұсыныстармен жұмыс

Қазіргі кезде ANSI стандарты индекстарды қолдамайды. Сонда да индекстар барлық мәліметтер қорында кең қолданылады, сондықтан олармен жұмысты назардан тыс қалдыруға болмайды. Индексті құру операторының синтаксисі пайдаланылатын SQL таратуына қарай өзгешеленуі мүмкін. Индексті құру командасының келесі синтаксистік формасы жиі кездеседі:

CREATE INDEX индекстің атауы

ON кестенің атауы (өрістің атауы_1.[өрістің атауы_2...])

Қарапайым индекс құру. Қарапайым индекс индекстардың өте қарапайым және сонымен қатар кең таралған түрі болып табылады. Қарапайым индекс кестенің бір ғана өрісінен (бағанынан) тұрады. Сондықтан оны көбінде *бірбағандық индекс* деп атайды.

Қарапайым индекс құру командасының типтік синтаксисі келесідей:

CREATE INDEX индекстің атауы

ON кестенің атауы (бағаның атауы)

Мысалы, **СТУДЕНТТЕР** кестесі үшін келесі оператор көмегімен, студенттер фамилиясынан тұратын өріс бойынша индекс құруға болар еді:

CREATE INDEX NAME_IDX

ON КЕСТЕ (ТЕГІ)

Керемет индекстер (уникальные). Керемет индекстер кестеге қайталанатын мәндердің енгізілуін болдырмайды. Сонымен,

керемет индекстер өнімділігін жоғарылату мақсатында ғана емес, мәліметтердің тұтастығын қолдау үшін де пайдаланылады.

Керемет индекстерді құру операторының типтік синтаксисі келесі түрде болады:

CREATE UNIQUE INDEX индекстің атауы

ON кестенің атауы (бағаның атауы)

Мысалы, **ДОЛЖНОСТЬ** кестесі үшін келесі команданың көмегімен "Должность" өрісі бойынша керемет индекс құруға болады:

CREATE UNIQUE INDEX POST_IDX

ON Lдолжности (Должность)

Құрамдас индекстер. Құрамдас деп екі немесе одан да көп өрістер бойынша құрылған индекстерді айтады. Құрылған индексті карағанда, құрылған индексте өрістердің тізбектелу реті мәліметті іздеу жылдамдығына әсер ететінін ескеру қажет. Жалпы жағдайда индексте өрістерді шектеу мәндерінің азаю ретімен орналастырған жөн.

Құрылған индексті беру синтаксисі жалпы түрде келесідей:

CREATE INDEX индекстің атауы

ON кестенің атауы (бағаның атауы 1. бағаның атауы 2...)

Біздің мысалымызда **СТУДЕНТТЕР** кестесінің "ЖӨНІ" және "АТЫ" өрістері үшін құрылған индекстарды құрудың мағынасы бар. Мұндай индекс құру операторының түрі келесідей:

CREATE INDEX FULLNAME_IDX

ON КЕСТЕ1 (аты, жөні)

Индекстарды жою.

Индекстарды жою ешқандай қиындық туғызбайды. Жою үшін индекс атын білу қажет. Индексты жою операторының синтаксисі келесі түрде болады:

DROP INDEX индекстің атауы

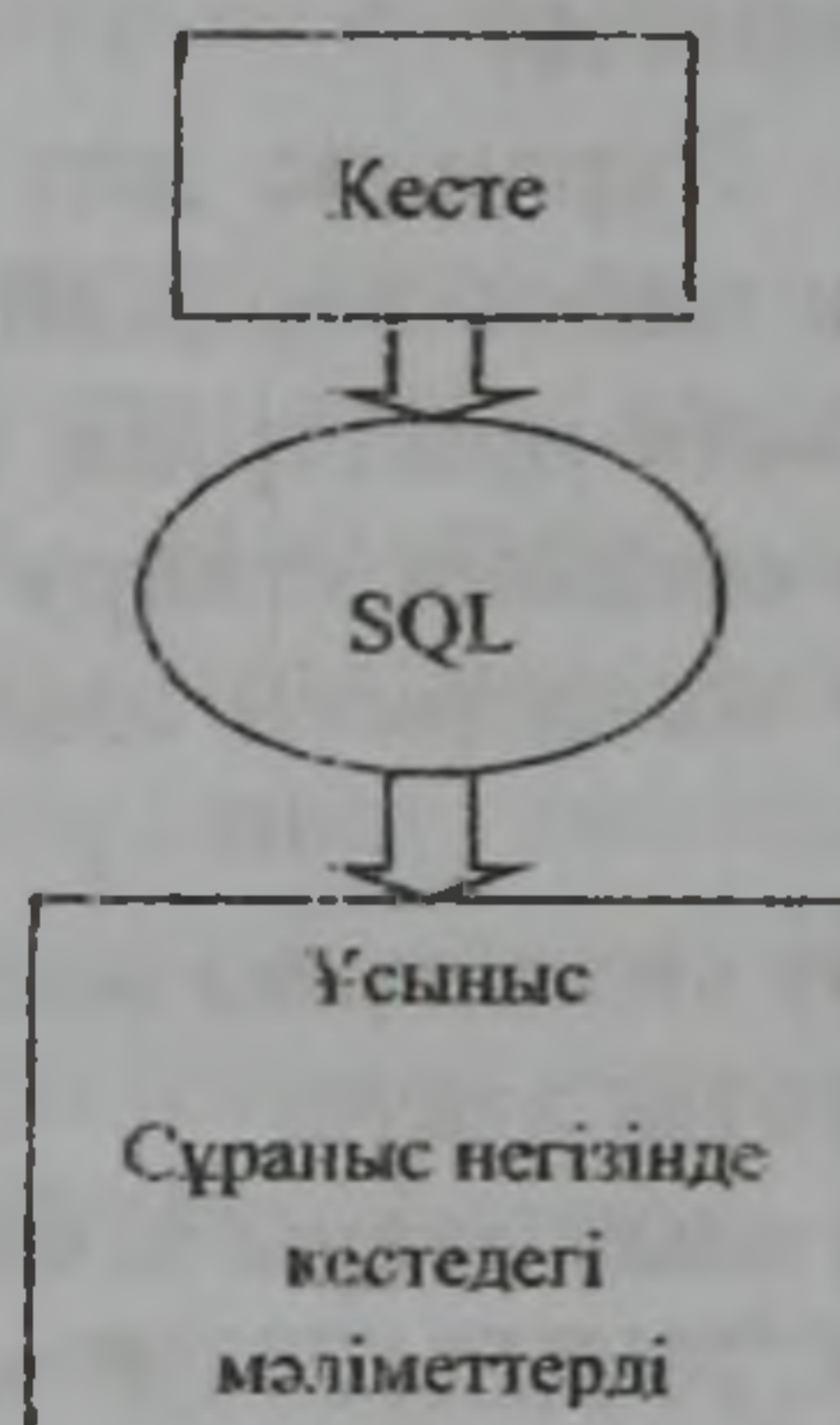
Индексті жою индекстелген өрістерге еш әсер етпейді. Жоюдан кейін индекс қайтадан құрылуы мүмкін.

Ұсыныстармен жұмыс. Ұсыныс (**VIEW**) мәліметтер қорының объектісі. Онымен жұмыс қарапайым кестемен жұмыстан өзгешеленбейді. Ұсыныстардың кестелерден айырмашылығы келесіде. Мәліметтер қорының қарапайым кестелері мәліметтерден тұрады. Ұсыныстар мәліметтерден тұрмайды, ал олардың мазмұны басқа кестелерден тандайынады (немесе басқа ұсыныстардан).

Ұсыныстар тұжырымдалатын кестелер (немесе ұсыныстар) базалық кестелер (немесе базалық ұсыныстар) деп атау қабылданған.

Шындығында ұсыныстар әрбір хабарласқан сайын орындалатын сұраныс болып табылады. Әрбір уақыт мезетінде осы сұраныстың орындалу нәтижесі ұсыныс мазмұны болып табылады. Ұсыныстың базалық кестесіндегі мәліметтерді өзгертсе ұсыныс мазмұны да өзгереді.

Ұсыныс мазмұнын өзгертсе, осы ұсынысты құруда негізге алынған кесте мазмұны да өзгереді. Төменде ұсыныстың тұжырымдалу процесінің схемасы 2.2-суретте берілген:



2.2-сурет – Ұсыныстың тұжырымдалу схемасы.

Ұсынысты пайдалану кесетені пайдаланудан өзгешеленеді. Ұсыныс-тардан мәліметтерді таңдау қарапайым кестедегі сияқты орындалады. Сондай-ақ ұсыныстың мәліметтермен басқару операцияларына рұқсат етіледі, бірақ ол мұнда кейбір шектеулер болады.

Ұсыныстар кестелерден айырмашылығы дискілік кеңістіктен орын алмайды (немесе дәлірек айтсақ, ұсыныс алатын дискілік кеңістік өте кішкентай-тек сұранысты сақтауға қажеттілері).

Ұсыныстың қолданылу облыстары.

Ұсыныстар көбінде екі жағдайда қолданылады:

- мәліметтерді қорғау мақсатында,
- қорытынды мәліметтерді тұжырымдау үшін.

Бірінші жағдайда ұсыныстар бүкіл кестеден емес, тек кейбір өрістеріндегі ақпараттарды беру үшін қолданылады.

Келесі мысалды қарастырайық. Мысалы, "СОТРУДНИКИ" кестесінің "Рейтинг" өрісінде сақталатын қызметкерлердің рейтингтері туралы ақпарат құпиялы болып есептелсін, және оларға кіру хұқы тек ұжым жетекшілерінде ғана болсын. Бірақ осы кестеде сақталатын ақпарат бөләгә кадрлар бөлімінің жұмысшыларына қажет-қызметкерлердің аты туралы және жұмысқа қабылдау күні туралы мәліметтер. Бұл жағдайда бір кестеге кіру рұқсатын шексіздеу үшін кадрлар бөлімінің қызметкерлерінде кіруге рұқсат болуға тиісті ақпараттарды ғана тандап ұсынысты пайдаланған ыңғайлы. Сонда олар өз қызметтік міндетін толық көлемде орындай алады және құпиялы ақпаратқа кіру рұқсаты болмайды.

Ұсыныстар өрістерге ғана емес, сондай-ақ кесте жазуларына кіру рұқсатына шек қою үшін пайдаланылады. Ол үшін ұсынысты құруда негізге алынатын мәліметтерді тандау сұранысында сәйкес шектеулік шарттарды көрсетсе болғаны. Мысалы, жоғарыда көрсетілген кадрлар бөлімінің жұмысшыларының мысалында ұсыныс құруда ұсыныстан қандай да міндетті алып тұратын жұмысшыларды шығарып тастау шартын беруге болады.

Сондай-ақ ұсыныстар есептерді тұжырымдауда қорытынды нәтижелерді тұжырымдау үшін де пайдаланылады. Өзгермелі ақпаратты кесет негізінде тұжырымдалатын есептерді жиі баспаға шығару қажет болған жағдайда, ұсыныстарды пайдаланған ыңғайлы.

Ұсыны, сұраныс негізінде құрылуы мүмкін болғандықтан ақпаратты базалық кестелер қатарынан алатын және қажеттілікпен топтайтын ұсыныс құруға болады, ал есепті шығарар кезде осы ұсынысқа қарапайым кесте сияқты хабарласуға болады. Бұл жағдайда есепті шығарған сайын күрделі SQL-сұраныс құрудың қажеті жоқ. Сондай-ақ, бұл жағдайда логиканың бөлігі мәліметтер қорының сервер жағына шығарылып тасталады, өйткені есептің тұжырымдалуы клиенттік қосымшадан тәуелді болмайды.

Ұсыныстарды құру. Ұсыныстарды құру үшін CREATE VIEW операторы пайдаланылады. Ұсыныс бір немесе бірнеше ұсыныстар негізінде құрылуы мүмкін. Ұсынысты құрудың операторының типтік синтаксисінің түрі келесідей:

```
CREATE VIEW имя_представления AS  
{оператор выборки данных}
```

Ұсыныс құрылған соң онымен ұсыныс аты ретінде берілген аты бар қарапайым кесте сияқты жұмыс істеуге болады. Қандай да бір өзгешелігі топтау сөйлемі бар ұсыныстар. Мұндай ұсыныстар үшін мәліметтер тандауда ешқандай шектеу жоқ, бірақ оларға мәліметтерді басқару операторын қолданбайды.

Ұсыныстарды жою. Ұсыныстарды жою DROP VIEW операторының көмегімен орындалады, оны шақырғанда RESTRICT және CASCADE параметрлері көрсетілуі мүмкін. Бұл параметрлер басқа ұсыныстар және/немесе шектеулер сілтелетін ұсыныстарды жою кезіндегі әрекеттерді анықтайды. RESTRICT вариантын падаланғанда, бұл жағдайда кәте туралы хабарлама шығарылады, және жою орындалмайды. Егер CASCADE режимі пайдаланылса, онда DROP VIEW операторының орындалуы, базалық ұсыныстардың және шектеулердің жойылуына әкеледі.

DROP VIEW операторының типтік синтаксисі келесі түрде болады:

DROP VIEW имя_представления [RESTRICT | CASCADE]

Сақталатын процедуралар, оларды құру, орындау және жою. Сақталатын процедуралар (Stored Procedure)- олар байланысқан SQL-операторларының топтары. Сақталатын процедураларды пайдалану мәліметтер қорымен жұмыстың қосымша икемділігін қамтамасыз етеді, өйткені сақталатын процедураларды орындау SQL жеке операторлар тізбегінен әлдеқайда қарапайымдау.

Сақталатын процедуралар орындалудың жоғары жылдамдығын қамтамасыз ететін мәліметтер қорында компиляцияланған түрде сақталады.

Сақталатын процедураларға кіретін параметрлер ала-алады, қосымшаға мәндерді қайтарады және қосымшадан шақырылады.

Сақталатын процедураларды пайдаланудың негізгі артықшылықтары келесіде қорытындыланады:

- сақталатын процедуралар логика бөлігін мәліметтер қорының серверіне шығаруға мүмкіндік береді. ол ақпараттық жүйенің мәліметтер қорының клиенттік бөлігінен тәуелділігін әлсіретеді;
- сақталатын процедуралар жобаның модульдігін қамтамасыз етеді. олар бір мәліметтер қорына хабарланатын клиенттік қосымшалар үшін жалпы болады, ол кодтың қайталануын болдырмайды және қосымша өлшемін кішірейтеді;

- сақталатын процедуралар қосымшалардың бәрге жүруін оңайлатады: процедуралар жаңартқан кезде өзгерулер автоматты түрде барлық қосымшаларда бейнеленеді;
- сақталатын процедуралар ақпараттық жүйенің жұмысының тиімділігін арттырады: олар клиентпен емес, желілік трафикті төмендететін сервермен орындалады;
- сақталатын процедуралардың орындалу жылдамдығы жеке sql операторлар тізбегінен жоғары. ол сақталатын процедуралардың серверде компиляцияланған түрде сақталуына байланысты.

Сақталатын процедуралардың екі түрін бөледі:

- таңдау процедуралары қосымшалар оларды мәліметтерді таңдау операторында кесте немесе ұсыныстар орнына пайдалануы мүмкін;
- орындалатын процедуралар олар арнайы операторлар пайдаланылып шақырылады. орындалатын процедура шақырылатын программаның нәтижелерін қайтармауы мүмкін.

Сақталатын процедураларды құру.

Сақталатын процедураларды құру үшін CREATE PROCEDURE операторы пайдаланылады.

CREATE PROCEDURE операторы мәліметтер қорынан жаңа сақталатын процедураны анықтайды. Процедуралар тілі SQL таратуларынан тәуелді, бірақ ереже бойынша SQL барлық инструкцияларын және мыналарды қосады:

- шартты операторлар;
- цикл операторларының әртүрлі түрлерін;
- ерекше жағдайларды өңдеу мүмкіндіктерін.

Сақталатын процедуралар тақырыптан және денесінен тұрады.

Процедураның тақырыбы:

- мәліметтер қорындағы кесте мен процедура аттарының ішінде ерекше болуы тиіс процедура атынан;
- процедура шақырылатын программадан қабылдайтын кіретін параметрлер және олардың мәліметтері типтерінің тізімінен;
- шығатын параметрлер және олардың мәліметтері типтерінің тізімінен тұрады, егер процедура мәндерді шақырылатын программаға қайтарса.

Процедура денесі мыналардан тұрады:

- жергілікті айнымалылар және олардың мәліметтері типтерінің тізімінен (егер олар процедура кодында пайдаланылса);
- BEGIN және END кілттік сөздерінің арасында қорытылған процедуралар және триггерлер тіліндегі инструкция блогынан.

Сақталатын процедураларды орындау.

Сақталатын процедураны орындауға жіберетін оператор процедураға тәуелді болады. Таңдау процедурасы оларға SELECT - мәліметтерді таңдау операторының көмегімен хабарласқанда орындалады.

Орындалатын процедураны шақыру үшін арнайы EXECUTE операторын пайдаланған жөн.

Сақталатын процедураларды жою. Сақталатын процедураларды жою үшін DROP PROCEDURE операторы пайдаланылады. Бұл оператор синтаксисі келесі түрде болады:

DROP PROCEDURE сақталатын процедураларын аты

Триггерлер, құру және жою операторлары. Триггерлер сақталатын процедуралардың бір түрі. Бірақ сақталатын процедуралардан айырмашылығы триггерлердің орындалуы SQL операторын шақыру нәтижесінде емес, мәліметтер қорына өзгерістер енгізетін мәліметтерді басқару операторларының бірінің орындалуы кезінде орындалады. Мұнда триггерлер мәліметтерді басқару операторларының орындалуына дейін де орындалады.

Триггердер қорда мәліметтердің сілтемелік тұтастығын қамтамасыз ету үшін пайдаланылады. Олардың келесі мүмкіндіктері бар:

- пайдаланушы кесте өрістеріне рұқсат етілген мәндерді енгізгеніне кепілдік беру үшін енгізілетін мәліметтерді бақылау мүмкіндігі;
- триггерлермен байланысқан кестелерді пайдаланатын қосымшаларда триггерлердегі өзгерту көрінетін бірге жүретін қосымшаларды оңайлату;
- кестелердің өзгертулерін автоматты құжаттау қосымша өзгерістер журналын кестеде өзгеріс болған сайын орындалатын триггерлер көмегімен басқару.

Триггерлерді құру. Триггерлерді құру үшін CREATE TABLE операторы пайдаланылады. Сақталатын процедуралар сияқты триггерлер де тақырып пен денесінен тұрады. Триггер тақырыбы мынадан тұрады:

- триггер аты, мәліметтер қоры ішінде керемет;
- триггер байланысқан кесте аты;
- триггер қашан орындалатынын анықтайтын инструкциялар;

Триггер денесі мыналардан тұрады:

- жергілікті айнымалылар және мәліметтердің типтерінің тізімінен (егер олар триггер кодында пайдаланылмаса);
- begin және end кілттік сөздерінің арасында қорытылған процедуралар мен триггерлер тіліндегі инструкция блогынан. блоктың ішінде басқа блок болуы мүмкін.

Сонымен триггер мен сақталатын процедураның айырмашылығы тек тақырыбында.

Триггер кестемен байланысқан. Кестеге бөлінген артықшылықтары бар кесте иесі мен кез-келген пайдаланушы онымен байланысқан триггерлерді автоматты түрде орындауға жұқы бар.

Триггерлерді жою.

Триггерлерді жою үшін DROP TRIGGER операторы пайдаланылды. Бұл оператордың жалпы түрдегі синтаксисі төмендегідей:

DROP TRIGGER триггердің атауы

Мәліметтерді басқару, кестедегі берілгендерді өзгерту. Мәліметтер қорында сақталатын мәліметтерді басқару үшін, командалардың жеке типтері ретінде бөлінетін және мәліметтерді басқару тілі (DML-Data Manipulation Language) деп аталатын SQL операторларының типтері пайдаланылады. DML операторларының көмегімен қолданушы кестеге жаңа мәліметтерді жүктеп, бар мәліметтерді түрлендіруге және жоюға болады:

SQL тілінде тек қана 3 негізгі DML операторы анықталған: INSERT; UPDATE; DELETE.

Кестеге жаңа ақпаратты қосу. Мәліметтер қорының кестесіне жаңа ақпаратты қосу процесі әдетте мәліметтерді жүктеу деп аталады. Мәліметтерді жүктеу үшін INSERT операторы пайдаланылады.

Кестеге жаңа жазу қосу. Кестеге жаңа жазу қосу үшін INSERT операторының келесі синтаксистік формасы пайдаланылады:

INSERT INTO кестенің атауы

VALUES (мәні1. мәні2..... мәні_N)

INSERT операторының бұл формасын пайдаланғанда VALUES тізімінде мәндер саны кесте өрістерінің санына тең болуы керек.

Және де VALUES тізімінде көрсетілген әдбір мәннің типі, өріс мәліметтерінің типімен сәйкес болуы керек.

Символдық типтер мен даталарға қатысты мәндер апостроф ішінде тұруы тиіс. Мәндер тізімінде NULL мәні де пайдаланылуы мүмкін.

Мысал қарастырайық. МАМАНДЫҚ кестесі келесі оператор пайдаланылып құрылған:

```
CREATE TABLE КЕҢСЕЛЕР (  
Кеңсенің коды INTEGER NOT NULL PRIMARY KEY  
Кеңсе VAR CHAR (50) NOT NULL UNIQUE.  
Разрядтылығы INTEGER NOT NULL).  
Төлемақы DECIMAL (7.2) NOT NULL).
```

Осы кестеге жаңа жазу қосу үшін келесі INSERT операторын пайдаланған жөн:

```
INSERT INTO Кеңселер  
VALUES (12 'Бағдарламашы'.12.2000.00)  
Кестенің жеке өрістеріне мәліметтер енгізу
```

Кестеге мәліметтер қосқанда барлық өрістерін емес, кейбіреулерін ғана толтыруға болады. Бұл жағдайда INSERT операторының келесі синтаксистік формасы пайдаланылады:

```
INSERT INTO кестенің атауы (бағаның атауы1. бағаның  
атауы2... бағаның атауы_N)  
VALUES (мәні1. мәні2... мәні_N)
```

Мысалы, СТУДЕНТЫ кестесіне жаңа студент туралы ақпарат қосқанда, студенттің толық аты туралы ақпаратты көрсетсе болғаны. Бұл жағдайда келесі операторды пайдалануға болады:

```
INSERT INTO Студенты (Номер_зач_книжки.Имя.  
Фамилия.Отчество)  
VALUES (234.'Абылгазина'. 'Айнур'. 'Кадыровна')
```

Бұл оператор орындалғанда басқа барлық өрістерге NULL мәні енгізіледі. Кесте атынан кейін жақша ішінде көрсетілмейтін өрістердің NOT NULL шектеуі болмайды, әйтпесе INSERT операторын орындау талабы табыссыз аяқталады.

Басқа кестедегі мәліметтерді кестеге енгізу

Кейде бір кестедегі ақпаратты екінші кестеге тасымалдау қажет болады. Бұл типтегі операцияларды INSERT операторымен SELECT мәліметтерді таңдау операторының комбинациясының көмегімен орындауға болады.

INSERT және SELECT операторын біріктіріп, басқа кестеге сұраныс орындалуы нәтижесінде алынатын мәліметтерді кестеге қоюға болады. Бұл жағдайда INSERT операторының синтаксисі келесі түрде болады:

```
INSERT INTO кестенің атауы (бағаның атауы1. бағаның атауы2... бағаның атауы1_N)
FROM кестенің атауы
WHERE шарт.
```

Бұл операторда VALUES сөйлемінің орнына SELECT операторы пайдаланылады. Бұл оператордың синтаксисінің қысқаша түсініктемесі. SELECT сөзінен кейін мәндері таңдамаға қосылатын өрістер тізімі көрсетіледі (егер SELECT кейін * символын көрсетсе, таңдамаға барлық өрістер қосылады). FROM сөйлемі мәліметтерді таңдау орындалатын кестенің атын көрсету үшін пайдаланылады. WHERE сөйлемі міндетті емес және таңдамаға қосылатын мәліметтерге шектеулерді қосу үшін пайдаланылады.

Кесте атынан кейін INSERT операторында жақша ішінде көрсетілетін өрістердің саны таңдамаға қосылатын өріс санына тең болуы керек. Өрістердің сәйкестігі, олардың тізбектелу ретімен анықталады: SELECT операторының тізіміндегі 1-ші өрісі сәйкес келеді және т.с.с.

Кестеде сақталатын мәліметтерді өзгерту

Кестеге енгізіліп қойылған мәліметтерді өзгерту үшін UPDATE операторы пайдаланылады. Бұл оператор кестеге жаңа жазу қоспайды, бар мәліметтерді жаңаға ауыстырады. UPDATE операторы бір өріске де, бірнеше өрістерге де қолданылуы мүмкін. Өзгертілетін жазулардың саны қолданушы қажеттілігіне тәуелді. UPDATE көмегімен бір немесе бірнеше жазуларды өзгертуге болады.

Кестенің бір өрісіндегі мәліметтерді түрлендіру

Кестенің бір ғана өрісіндегі мәліметтерді өзгерту үшін UPDATE операторының қарапайым формасы пайдаланылады:

```
UPDATE кестенің атауы
SET имя_поля=значение
[WHERE шарт]
```

UPDATE операторының синтаксистік элементтерінің мағынасы мынадай: UPDATE кілттік сөзінен кейін мәліметтер түрлендірілетін

кестенің аты көрсетіледі, SET кілттік сөзінен кейін аты берілген өріске жаңа мәді меншіктеу орындалады. Міндетті емес WHERE сөйлемінің көмегімен берілген шарт, түрлендірілетін жазулар санын анықтайды .

Мысалы, СТУДЕНТЫ кестесінде сақталған факультет студентінің телефон номерін өзгерту қажет болсын. Бұл жағдайда UPDATE операторы бір өрістің және бір жазудың ғана мәнін өзгертуі керек. Сондықтан WHERE сөйлемінде бізге қажетті жазуды таңдайтын шартты көрсету керек. Қарапайым шешімі. «Код_физического_лица» алғашқы кілтің өрісін қажетті жазуды таңдауға пайдалану болып табылады. Осы өрісте сақталатын мәнде қызметкерді бізмәнді анықтайды. Онда телефон номерін өзгертуді орындайтын UPDATE операторының түрі келесідей болады:

```
UPDATE Физические_лица
SET Телефон=(095)2347890
WHERE Код_физического_лица=16
```

Бұл оператор мәліметтер қорында 146 номерімен тіркелген қызметкерге сәйкес келетін жазу үшін телефон номерінің мәнін өзгертеді. Егер біз жоғарыда келтірілген операторда шектеулі шартты бермесек, онда телефон номерінің мәні кестенің барлық жазулары үшін өзгереді.

Кестенің бірнеше өрістеріндегі мәндерді өзгерту

UPDATE операторының көмегімен бір уақытта кестенің бірнеше өрістерінің мәндерін өзгертуге болады. Ол үшін SET кілттік сөзінен кейін бір емес, бірнеше өрістерді көрсету керек:

```
UPDATE кестенің_атауы
SET бағаның_атауы_1=мән_1
    бағаның_атауы_2=мән_2
    ...
    бағаның_атауы_N=мән__N
```

[WHERE шарт]

Кестеден мәліметтерді өшіру

Кестеден мәліметтерді өшіру DELETE операторының көмегімен орындалады. Бұл оператор жеке өрістердегі мәліметтерді емес, бүкіл жазуды толығымен өшіреді. DELETE операторының синтаксисі келесідей:

```
DELETE FROM кестенің_атауы
[WHERE шарт]
```

Өшірілетін жазулар міндетті емес WHERE сөйлемінің көмегімен берілген шартқа сәйкес анықталады. WHERE сөйлемі DELETE операторында болмаса, мәліметтер бүкіл кестеден өшіріледі.

2.4 Мәліметтер қорының қауіпсіздігін басқару

Мәліметтер қорын басқарудың ең маңызды есептерінің бірі мәліметтердің қауіпсіздігін қамтамасыз ету, яғни мәліметтерді рұқсатсыз пайдаланудан қорғау.

Қолданушылардың артықшылықтары

Артықшылықтары деп қолданушылардың өкілеттік деңгейлерін айтады. Мәліметтер қорында сақталатын ақпараттарға кіру рұқсатын шексіздендіру артықшылықтарының көмегімен реттеледі.

Артықшылықтардың екі түрін бөледі: жүйелік артықшылықтар; объектілік артықшылықтар.

Жүйелік артықшылықтар. Жүйелік артықшылықтар мәліметтер қорының пайдаланушысына, оны администрациялаумен байланысты әрекеттерді орындау мүмкіндігін береді: мәліметтер қорының, сондай-ақ жүйелік артықшылықтар мәліметтер қорымен олардың жеке объектілерінің күйін өзгерту хұқы береді.

Мүмкін болатын жүйелік артықшылықтар пайдаланылатын МҚБЖ тәуелді болады. Бірақ кез-келген жағдайда олар мыналарға хұқы бар анықтамаларды қосады: кестені құру, ұсыныстарды құру, сақталатын процедураларды құру, кестелерді өшіру, ұсыныстарды жою, сақталатын процедураларды жою.

Бұл тізім одан әрі қарай кеңейі мүмкін және де әрбір артықшылықтың әртүрлі МҚБЖ-де өз ерекшеліктері бар.

Объектілік артықшылықтар. Объектілік артықшылықтар дегеніміз мәліметтер қорының объектілеріне таралатын пайдаланушының өкілеттік деңгейлері. Ол дегеніміз мәліметтер қорының объектілерімен қандай да бір әрекеттерді орындау үшін пайдаланушының сәйкес хұқы болуы керек.

ANSI стандартымен келесі объектілік артықшылықтар қарастырылған:

- SELECT – көрсетілген кестеден мәліметтер таңдамасын жүргізуге рұқсат береді;
- INSERT(имя_поля) – көрсетілген кестенің қандай да бір өрісіне мәліметтерді қосуды орындауға рұқсат береді;

- INSERT-көрсетілген кестенің барлық өрістеріне мәліметтерді қосуға рұқсат береді;
- UPDATE(имя_поля)-көрсетілген кестенің берілген өрісінде мәліметтері түрлендіруге рұқсат береді;
- UPDATE-көрсетілген кестенің барлық өрістерінде түрлендіру жүргізуге рұқсат береді;
- REFERENCE- көрсетілген кестенің барлық өрістеріне сілтелуге рұқсат береді.

Мәліметтер қорының кіру рұқсатын басқару. Пайдаланушылардың мәліметтер қорына кіру рұқсатын басқару үшін SQL тілінде екі оператор бар: GRANT; REVOKE. Ереже бойынша бұл операторлар мәліметтер қорының администраторымен немесе оның қауіпсіздік бойынша көмекшісімен пайдаланылады.

GRANT операторы. GRANT операторы пайдаланушыға жүйелік және объектілік артықшылықтарды беру үшін пайдаланылады. Бұл операторының синтаксисі келесідей:

GRANT артықшылық _1 [.артықшылық _2]

ON объектінің аты

TO қолданушының атауы [WITH GRANT OPTION]

USER атты пайдаланушыға ҚЫЗМЕТКЕРЛЕР кестесінен мәліметтерді таңдау хұқын беру келесі оператордың көмегімен орындалады:

GRANT SELECT

ON ҚЫЗМЕТКЕРЛЕР

TO USER

Бір GRANT операторының көмегімен бірнеше артықшылықтарды беруге болады. Мысалы, келесі оператор USER пайдаланушыға ҚЫЗМЕТКЕРЛЕР кестесінің мәліметтерін карау және жаңасын қосу хұқын береді:

GRANT SELECT.INSERT

ON ҚЫЗМЕТКЕРЛЕР

TO USER.

Шақырғанда GRANT операторы міндетті емес WITH GRANT OPTION сөйлемін де пайдалануы мүмкін. Бұл сөйлем артықшылық берілетін пайдаланушы осы объектіге артықшылық хұқын алады дегенді білдіреді. Мысалы, егер жоғарыда карастырылған операторды шақырса, онда USER атты пайдаланушы осы артықшылықтарды басқа пайдаланушыларға беру хұқын алады:

GRANT SELECT.INSERT
ON ҚЫЗМЕТКЕРЛЕР
TO USER
WITH GRANT OPTION

REVOKE операторы. REVOKE операторы пайдаланушыға берілген артықшылықтарды тойтару үшін пайдаланылады. Бұл оператор екі параметрдің біреуімен шақырылады –RESTRICT немесе CASCADE RESTRICT варианты пайдаланғанда REVOKE операторы табысты орындалады, егер оның орындалуы калып-калған артықшылық пайда болуына әкелмесе.

CASCADE режимін пайдаланғанда басқа пайдаланғанда басқа пайдаланушыларға калып-калған барлық артықшылықтар жойылады.

REVOKE операторының синтаксисі:

REVOKE артықшылық _1[.артықшылық _2]

On объектінің аты

FROM қолданушының атауы [RESTRICT/ CASCADE]

Бақылау сұрақтар

- 1 SQL тілі қандай мәліметтерді өңдеу құралы болып табылады?
- 2 SQL тілінің теориялық негізі не болып табылады?
- 3 SQL тілінің функцияларын атаңыз
- 4 SQL тілінің қай операторы МҚ-ға қол жеткізуді басқарады?
- 5 SELECT, INSERT, DELETE, UPDATE операторлары не болып табылады?
- 6 REVOKE операторы нені орындайды?
- 7 SQL тілінің алғашқы стандартын кім және қашан жасаған?

3 Мәліметтер қорының моделі

3.1 Мәліметтер қорын пайдалану

Мәліметтер қорын пайдалану файлдық құрылым мәліметтерінде болған шектеулер мен кемшіліктерді жоюға мүмкіндік берді. Мәліметтер қорының көмегімен шешілетін әралуан есептер мәліметтер қорын құрылымдау мен мәліметтердің арасындағы байланыстарды ұйымдастырудың түрлі әдістерін қалыптастыруға әкеледі. Соның нәтижесінде мәліметтер қорының бірнеше моделі пайда болды. Мәліметтер қорының моделі деп мәліметтер қорының логикалық кейіптеуін атаймыз. Алғашқыда мәліметтер қорының иерархиялық, желілік, реляциялық модельдері пайда болды, олар кейіннен классикалық, негізгі модельдерге айналды. Соңғы уақытта осылардың негізінде постреляциялық, көпөлшемдік, объектілі-бағытталған жаңа модельдер пайда болды және барынша дами отыра, қолданысқа енгізіле бастады.

Сонымен қатар мәліметтердің өзге модельдеріне негізделген белгілі модельдерді кеңейтетін алуан түрлі жүйелер жасалынуда. Олардың қатарында объектілі-реляциялық, дедуктивтік-объектілі-бағытталған, семантикалық, тұғырнамалық және бағытталған модельдерді атап кетуге болады. Бұл модельдердің кейбіреулері мәліметтер қорын, білім қоры мен программалау тілдерін интеграциялауға арналған. Кейбір МҚБЖ-лар бір мезгілде бірнеше мәліметтер модельдерімен жұмыс істейді.

Иерархиялық модель. Алғашқыда мәліметтер қорының иерархиялық модельдері пайда болды. Иерархиялық модельде мәліметтер арасындағы байланыстар реттелген граф (немесе бұтақ) түрінде берілген.

Иерархиялық МҚ-ның құрылымын (сұлбасын) сипаттау үшін кейбір программалау тілінде «бұтақ» мәліметтер типі қолданылады.

«Бұтақ» типі ПЛ/1 және Си программалау тілдерінің «құрылым» атты мәліметтер типімен және Паскаль тілінің «жазба» типімен ұқсас. Оларда типтердің бір-біріне салынуына жол беріледі, олардың әрқайсысы белгілі бір деңгейде орналасқан.

«Бұтақ» типі құрамдас болып келеді. Оған ішкі типтер («кіші бұтақтар») кіреді, олардың әрқайсысы өз кезегінде «бұтақ» типі болып табылады. «Бұтақ» типінің әрқайсысы бір «түбірлік» типтен және бағыныңқы типтердің реттелген жиынтығынан (бос болуы да мүмкін) тұрады.

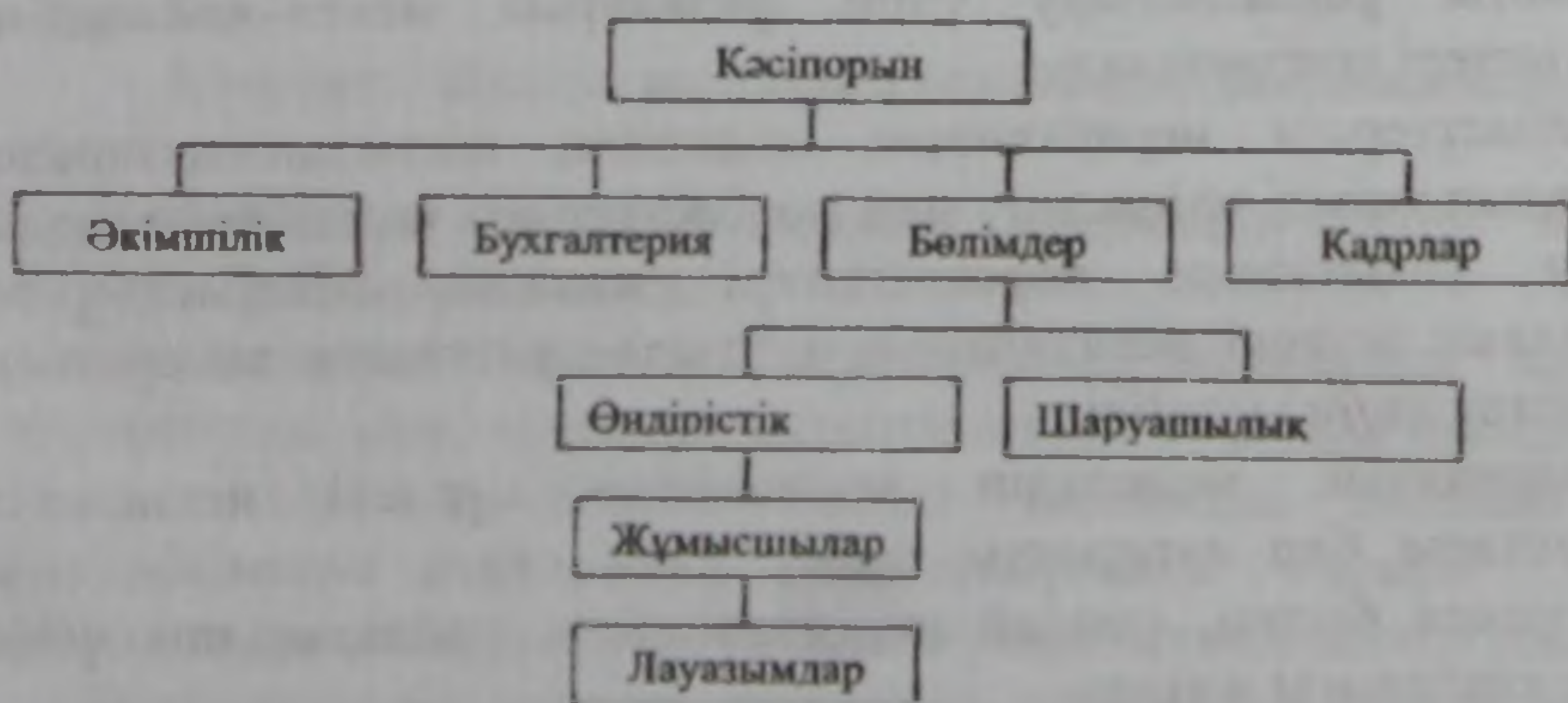
«Бұтақ» типіне енгізілген қарапайым типтердің әрқайсысы «жазба» атты жай немесе құрама тип болып табылады. Қарапайым «жазба» бір типтен тұрады, мысалы, сандық типтегі, ал құрама «жазба» типтердің белгілі бір жиынтығын, мысалы, бүтін, символдар жолы және көрсеткіштерді (сілтеме) біріктіреді.

Түбірлік деп бағыныңқы типтері бар және өзі ішкі тип болмайтын тип аталады. Бағыныңқы тип (ішкі тип) өзі үшін ата тегі (ата-ана) рөлін атқаратын типке қатысты ұрпақ болып табылады. Бір типтің ұрпақтары бір-біріне қатысты егіздер болып табылады.

Тұтастай алғанда «бұтақ» типі «жазба» типтерінің иерархиялық тұрғыдан ұйымдастырылған жиынтығын сипаттайды.

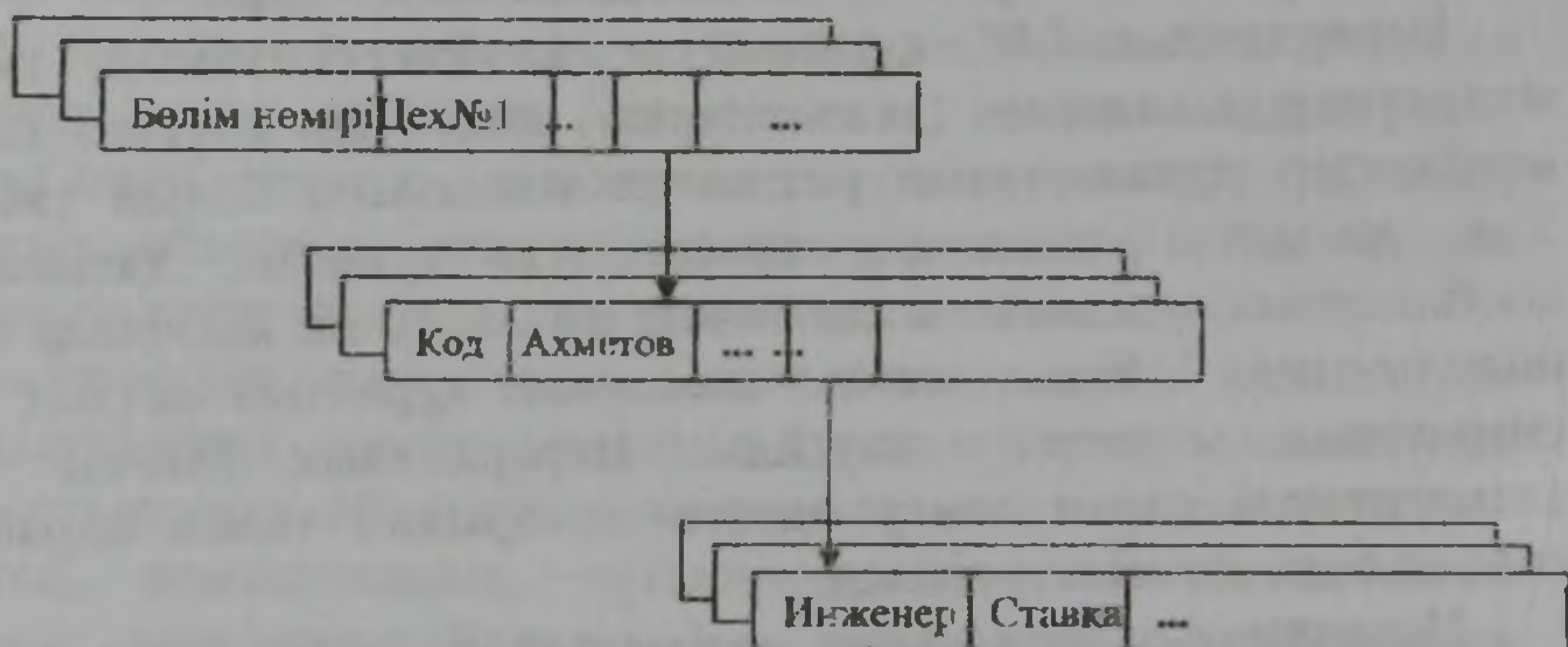
Иерархиялық МҚ құрамында «жазба» (жазбалар) типіндегі мәліметтер даналарын (экземплярларын) қамтитын «бұтақ» типіндегі мәліметтер даналарының реттелген жиынтығы болып табылады. Көп жағдайда типтердің арасындағы туыстық қатынастарды жазбалардың арасындағы қатынастарға көшіреді. Жазбалар өрістері шындығында МҚ-ның негізгі мазмұнын құрайтын сандық немесе символдық мәндерді сақтайды. Иерархиялық МҚ-ғы барлық элементтерді қарап шығу әдетте жоғарыдан төмен қарай және солдан оңға қарай жүргізіледі.

Иерархиялық модельдің мәліметтер қорлары бұтақ түріндегі құрылымы бар нысандарды сипаттауға ыңғайлы. Мысалы, белгілі бір кәсіпорынның құрылымын келесі түрде келіруге болады (3.1-сурет).



3.1-сурет – Кәсіпорынның иерархиялық құрылымы

Өзінен басқа жазбаға сілтеме жасалатын жазба «ата тегі» немесе бағындыратын жазба деп аталады. Сілтеме жасалатын жазба «ұрпақ» немесе бағыныңқы жазба деп аталады. Иерархиялық модельде әрбір ұрпақтың тек бір ғана ата тегі болады, басқаша айтқанда бір ұрпаққа екі және екіден артық ата тегі сілтеме жасай алмайды. Сілтемені нұсқау үшін ата тегі-жазбаның өрісі болуы тиіс, ол жерге тасымалдағыштағы ұрпақ-жазбаның физикалық мекен-жайы жазылады. Ұрпақ-жазба сақталатын тасымалдағыштағы осы физикалық мекен-жай көрсеткіш деп аталады. Жазбаларды байланыстырудың мысалы (3.2-сурет):



3.2-сурет – Жазбаларды байланыстыру

Сонымен, иерархиялық модельдегі мәліметтер арасындағы байланысты ұйымдастыру үшін физикалық мекен-жайлардың көрсеткіштері қолданылады.

Мәліметтердің иерархиялық моделінің артықшылықтарына ЭЕМ жадын тиімді пайдалану мен мәліметтермен негізгі амалдарды орындау уақытының көрсеткіштері жатады. Мәліметтердің иерархиялық моделі иерархиялық тұрғыдан реттелген ақпаратпен жұмыс істеу үшін ыңғайлы.

Иерархиялық модельдің кемшілігіне күрделі логикалық байланыстары бар ақпаратты өңдеу үшін оның көлемінің тым орасан үлкен болуы, сондай-ақ қарапайым пайдаланушы үшін түсінуге қиындығы жатады.

Желілік модель. Желілік модельдерде мәліметтердің арасындағы өзара байланыстар еркін граф түрінде болады, ондағы әрбір ұрпақтың екі және одан да көп ата тегі болуы мүмкін.

Желілік модельдің мәліметтер қоры жазбалар жинағы мен байланыстар жинағынан құралады. Байланыстардың жинағы физикалық көрсеткіштері бар өрістен тұрады.

Осылайша, желілік МҚ сұлбасын сипаттау үшін типтердің екі тобы: «жазба» және «байланыс» қолданылады. «Байланыс» типі «жазба» типінің ата тегі және ұрпақ типтері үшін анықталады. «Байланыс» типінің айнымалылары байланыстардың даналары болып табылады.

Желілік МҚ жазбалар жинағы мен сәйкес байланыстар жинағынан тұрады. Байланыстарды қалыптастыруға ерекше шектеулер қойылмайды. Егер иерархиялық құрылымдарда ұрпақ-жазбаның бір ғана ата тегі – жазбасы болса, ал мәліметтердің желілік моделінде ұрпақ-жазба ата тегі-жазбалардың (өгей ата-аналар) кез келген санына ие бола алады.

Келесі суретте (3.3-сурет) инженер лауазымына сәйкес келетін жазбаға екі жазба сілтеме жасайды.



3.3-сурет – Желілік модельдегі жазбаларды байланыстыру

Иерархиялық модельді желілік модельдің дербес жағдайы ретінде қарастыруға болады.

Мәліметтердің желілік моделінің артықшылығы жадының шығындалуы мен жеделдік көрсеткіштері бойынша тиімді түрде жүзеге асыру мүмкіндігі болып табылады. Желілік-модель иерархиялық модельмен салыстырғанда еркін түрдегі байланыстардың құрылу тұрғысынан үлкен мүмкіндіктер ұсынады.

Мәліметтердің желілік моделінің кемшілігі – оның негізінде құрылған МҚ сұлбасының жоғары дәрежедегі күрделілігі мен қатқылдығы, сондай-ақ қарапайым пайдаланушы үшін МҚ-дағы ақпаратты өңдеуді түсіну мен орындаудың қиындығы. Бұған қоса,

мәліметтердің желілік моделінде жазбалардың арасында еркін түрдегі байланыстардың орнау мүмкіндігінің салдарынан байланыстардың біртұтастығына қатысты бақылау әлсірейді.

Мәліметтер қорының иерархиялық және желілік модельдерінде физикалық көрсеткіштерді пайдалану, мәліметтерді өндеудің жылдамдығын едәуір арттыра түсті. Сонымен бірге, бұл модельдерде мәліметтердің өзара байланыстарының біртұтастығын сақтауға қатысты елеулі кемшіліктер анықталды. Байланыстардың біртұтастығының сақталуын бақылау ата-тегсіз ұрпақ болмайды деген ұстанымды сақтауға негізделген. Мәліметтер қорына өзгертулер енгізген жағдайда байланыстарды, физикалық көрсеткіштерді қайтадан анықтау қажеттігі туындайды, бұл үлкен қиындықтармен, пайдаланушының уақыты мен күш-жігерінің шығындарымен байланысты болады.

3.2 Реляциялық модель. Реляциялық алгебра

Иерархиялық және желілік модельдерде жазбаларды байланыстыру үшін физикалық көрсеткіштерді пайдалану мәліметтерді манипуляциялау мүмкіндіктерін едәуір шектейді және осы модельдердің мәліметтер қорларын өндеу күрделі және көп еңбекті қажет ететін процесс болып табылады.

1970 жылы IBM фирмасының қызметкері Эдгар Кодд реляциялық модель деп аталған мәліметтер қорының жаңа моделін ұсынған бірқатар еңбектерін жариялаған болатын. Сонымен қатар, реляциялық мәліметтер қорларын өндеу үшін Кодд мәліметтерді өндеудің тілдерін – реляциялық алгебра мен реляциялық есептеуді әзірлеп шығарды, олар бір командамен бүкіл мәліметтер қорын өндеуден өткізуге мүмкіндік береді, ал иерархиялық және желілік модельдерде бір жолы тек бір ғана жазба өндеуден өткізіледі. Физикалық көрсеткіштердің орнына Кодд мәліметтердің тасымалдағышта физикалық орналасуына тәуелсіз оларды өздерінің ішкі логикалық қарым-қатынастарына сәйкес байланыстыру идеясын ұсынды.

Реляциялық модельдің мәні – ол мәліметтерді реляциялар немесе қатынастар деп аталатын екі өлшемді кестелер түрінде беруді ұйымдастырады және ұсынады. Бір кесте бір нысанды, мысалы, сауда фирмасының клиенттерін сипаттайды. Кестенің

бағандары мәліметтер қорының өрістеріне сәйкес келеді және атрибуттар деп аталады. Сол арқылы атрибут нысанның жеке қасиетін сипаттайды. Кестенің жолдары жазбаларға сәйкес келеді және кортеж деп аталады. Сонымен, кортеж нысанның бір элементіне арналған атрибуттар мәндерінің жиынтығын, ал қатынас – кортеждердің жиынын сипаттайды. Мәліметтер қорының құрылымын әзірлеу және сипаттау кезінде әрбір кестедегі атрибуттарға атаулар беріледі және олардың типтері анықталады, мысалы, атауы Name, типі CHAR немесе атауы Amount, ал типі REAL. Сонымен қатар INTEGER типі, DATA типі және басқа да типтер болуы мүмкін.

Кестеге де атау беріледі. Мәліметтер қорының біреуі бірнеше кестеден тұруы мүмкін және қандай да бір құрамдас нысанды сипаттайды. Мәліметтер қорына да атау беріледі. Мысалы, қандай да бір сауда фирмасын сипаттайтын Orion оқу мәліметтер қоры келесі үш кестеден: Salespeople, Customers, Orders кестелерінен тұрады (1 қосымша).

Егер кестенің элементі әлі анықталмаған болса немесе ешқандай мәнге ие болмаса, онда оған Null - мәні меншіктелді.

Мәліметтердің арасындағы логикалық байланыс бастапқы және сыртқы кілттердің көмегімен жүзеге асырылады. Бастапқы кілт деп кестенің әрбір жолын бір мәнді анықтайтын кестедегі атрибуттардың ең аз жиынтығы аталады. Мысалы, Salespeople кестесінде бастапқы кілт Snum бағаны, ал Customers кестесінде Snum бағаны болып табылады. Осылайша, бастапқы кілтті құрайтын атрибуттар мәндерінің комбинациясы әр жазба үшін бірегей, қайталанбайтын болып табылады. Сыртқы кілт деп мәндерінің комбинациясы кейбір жазба үшін бастапқы кілттің жиынтығы аталады. Сыртқы кілттер бір мәліметтер қорының кестелеріндегі мәліметтердің арасындағы қажетті байланыстарды қамтамасыз етеді.

Мысалы, Orion мәліметтер қорында Orders кестесінің Snum және Snum бағандары сыртқы кілттер болып табылады. Мәліметтер арасындағы байланыстарды келесі сурет түрінде көрсетуге болады (3.4-сурет)

Orders

Onum	Amount	Date	Cnum	Snum
0813006	1098.16	10.03.2005	2016	1007

Customers

Cnum	Cname	City
2016	Pereira...	1004...

Salecpeople

Snum	Sname	City	Manager
1007...	Rifkin	Barsel ona	1004

3.4-сурет – Реляциялық модельдегі жазбаларды байланыстыру

Сонымен реляциялық кестенің анықтамасынан келесілер шығады:

- бастапқы кілттің және оған сәйкес келетін сыртқы кілттің атрибуттарының атаулары бірдей болуы міндетті емес, бірақ та бір типті болуы тиіс;
- мәліметтер қорының әрбір кестесінің бірнеше сыртқы кілттері болуы мүмкін, ал бастапқы кілт тек біреу ғана бола алады;
- жалпы жағдайда кестеде екі және одан да көп бірдей кортеждер болуы мүмкін емес;
- кестеде кортеждер реттелмеген, еркін түрде болуы тиіс;
- атрибуттар да реттелмеген, еркін түрде болуы тиісті.

Мәліметтердің реляциялық моделінің артықшылығы оны ЭЕМ-де физикалық жүзеге асырудың қарапайым, түсінікті және ыңғайлы болуында. Пайдаланушы үшін дәл осы қарапайымдылығы мен түсініктілігі оларды кең түрде қолдануға негізгі себеп болды. Осы типтегі мәліметтерді өңдеу тиімділігінің мәселелері техникалық тұрғыдан шешілетін болып шықты.

Реляциялық модельдің негізгі кемшіліктері:

- жеке жазбаларды идентификациялаудың стандарттық құралдарының болмауы;
- иерархиялық және желілік байланыстарды сипаттаудың күрделілігі жатады.

3.3 Мәліметтер қорын программалық жабдықтау

Қазіргі уақыттағы МҚБЖ-лар мәліметтер қорымен жұмыс істеу бойынша ауқымды есептерді қосымшаны жасамай-ақ шешуге мүмкіндік береді. Дегенмен де, қосымшаны жасау дұрыс болатын жағдайлар да бар. Мысалы, егер мәліметтермен манипуляциялауды автоматтандыру қажет етілсе, МҚБЖ-ның терминалдык интерфейсі жеткілікті түрде дамымаған болса, немесе МҚБЖ-дағы ақпаратты өңдеуден өткізу жөніндегі стандарттық қызметтер пайдаланушының қажеттерін қанағаттандырмайтын болса.

Мәліметтер қорын қолданбалы программалық қамтамасыз ету дегеніміз – бұл нақты мәселелерді шешу үшін пайдаланушылардың өздері құрастыратын программалар мен жүйелер. Қазіргі уақыттағы программалаудың алгоритмдік тілдерінде – Delphi, C++, Visual Basic және т.б. – SQL тілін пайдалана отырып мәліметтер қорына қол жеткізетін программалау құралдары бар. Сонымен қатар, көптеген МҚБЖ-ларда жүйе ішіне орнатылған программалау тілі бар. Dbase форматындағы мәліметтер қорын программалауға арналған арнайы Clipper алгоритмдік тілі бар.

Қосымшаларды жасау үшін МҚБЖ –ның программалық интерфейсі болуы тиіс, оның негізін сәйкес программалау тілінің функциялары және/немесе процедуралар құрайды. Қолданыстағы МҚБЖ –лар қосымшаларды жасаудың келесі технологияларына (және олардың комбинацияларын) қолдау жасайды:

- программаларды қолмен кодтау (Clipper, FoxPro, Paradox);
- генераторлардың көмегімен қосымшалардың мәтіндерін құрастыру (FoxPro-де FoxApp, Paradox-та Personal Programmer);
- визуалды программалау әдістерімен дайын қосымшаны автоматты түрде генерациялау (MS Visual Studio 2015 , Delphi, Access, Paradox for Windows).

Қолмен кодтау кезінде программалаушылар қосымшалар программаларын мәтінін қолмен тереді, сосын олардың жұмысын жүргізіп баптайды.

Генераторларды пайдалану қосымшаларды жасауды жеңілдетеді, себебі бұл жағдайда программалық кодты қолмен термей-ақ алуға болады. Қосымшалардың генераторлары қосымшалардың негізгі элементтерін (мәзір, экрандық форма,

сұраныстар және т.б.) жасауды жеңілдетеді, алайда көбінесе қолмен кодтауды жоққа шығара алмайды.

Қосымшаларды визуалды программалау құралдары қосымшалардың генераторларын қолдану идеясының дамуы боып табылады. Бұл ретте қосымша ыңғайлы біріктірілген ортаның көмегімен дайын түрдегі «құрылыс блоктарынан» құрылады. Қажет болған жағдайда жасаушы қосымшаға өзінің кодын оңай енгізеді. Біріктірілген орта, әдетте, қосымшаларды құрастыру, баптау мен өзгертудің күшті құралдарын ұсынады. Визуалды программалау құралдарын пайдалану қысқа мерзім ішінде алғашқы екі әдіспен алынған қосымшалардан да сенімді, тартымды және тиімді қосымшаларды құрастыруға мүмкіндік береді.

Жасалған қосымша әдетте амалдар жүйесінің бір немесе бірнеше файлдарынан құралады.

Егер де қосымшаның негізгі файлы орындалатын файлы (мысалы, exe-файл) болса, онда бұл қосымша МҚБЖ ортасынан дербес түрде орындалатын тәуелсіз қосымша болып табылады. Іс-жүзінде тәуелсіз қосымшаны алу, мәтінді қолмен теру сияқты, сондай-ақ қосымшаның генераторының немесе визуалды программалау ортасының көмегі сияқты тәсілдерімен алынған программаның бастапқы мәндерін компиляциялау жолымен жүзеге асырылады.

Тәуелсіз қосымшаларды мысалы, FoxPro МҚБЖ-лары және Delphi визуальді программалау жүйесін алуға мүмкіндік береді. Айта кететін нәрсе, Delphi құралдарының көмегімен әдетте, тәуелсіз қосымшаларды жасамайды, себебі бұл өте көп еңбекті қажет ететін үдеріс, оның орнына МҚБЖ ядросының рөлін атқаратын BDE (Borland DataBase Engine) мәліметтер қорының процессорын пайдаланады. Дербес ЭЕМ-дарға арналған қосымшаларды жасаудың ең алғашқы құралының бірі «таза түрдегі компилятор» болып саналатын Clipper жүйесі болып табылады.

Көп жағдайларда қосымша МҚБЖ ортасы болмағанда орындалмайды. Қосымшаны орындау МҚБЖ қосымша файлдарының құрамдас бөлігін (жеке жағдайда бұл бастапқы программаның мәтіні) талдайды және орындалатын қажетті машиналық командаларды автоматты түрде құруына негізделген. Былайша айтқанда, қосымша интерпретациялау әдісі арқылы орындалады.

Интерпретациялау режимі қазіргі заманғы көптеген МҚБЖ-ларда, мысалы, Access, Visual FoxPro және Paradox сияқты сондай-ақ, FoxBase және FoxPro тәрізді бұрынғы МҚБЖ-ларда жүзеге асырылған.

Сонымен қатар, компиляция мен интерпретацияның аралық нұсқасын — былайша айтқанда (псевдокомпиляция) жалған компиляцияны пайдаланатын жүйелер болады. Мұндай жүйелерде бастапқы программа компиляция арқылы аралық кодқа (псевдокодқа) өзгереді және дискіде жазылады. Бұл түрде оны кейбір жүйелерде тіпті редакциялауға рұқсат етіледі, алайда жалған компиляцияның басты мақсаты — программаны оны интерпретациялау үдерісін жылдамдатуға мүмкіндік беретін түрге өзгерту. Мұндай тәсіл DOS басқаруымен жұмыс істейтін МҚБЖ-ларда, мысалы, Foxbase+ және Paradox 4.0/4.5 for DOS-та кеңінен қолданылған.

Windows басқаруымен жұмыс істейтін МҚБЖ-ларда (псевдокод) жалған код көбінесе қосымшаны модификациялауға тыйым салу үшін қолданылады. Бұл жұмыс істеп жатқан программаны кездейсоқ немесе әдейі бұзудан сақтау үшін пайдалы болады. Мысалы, мұндай тәсіл Paradox for Windows МҚБЖ-да қолданылған, онда әзірленген экрандық формалар мен есеп берулерді редакциялауға сәйкес нысандарға өзгертуге жол беріледі.

Кейбір МҚБЖ-лар пайдаланушыға қосымшаны жасау нұсқасын тандап алу мүмкіндігін ұсынады: МҚБЖ-мен интерпретацияланатын программалық код ретінде немесе тәуелсіз программа ретінде.

Тәуелсіз қосымшаларда қолданудың артықшылығы — машиналық программаны орындау уақыты, әдетте интерпретация кезіндегі уақытпен салыстырғанда қысқалау болады. Мұндай қосымшаларды әлсіз машиналарда және қосымшаны пайдаланушылардың тарапынан өзгертуге қарсы жабу қажет болатын, жүйелерді, «қолдануға даяр» етіп орнатқан жағдайда пайдаланған дұрыс.

Интерпретацияланатын жүйелердің тағы бір үлкен артықшылығы — жақсы МҚБЖ-ларды әдетте мәліметтердің бүтіндігін бақылау мен оларға рұқсатсыз қол жеткізуден қорғауға арналған күшті құралдар болады, ал компиляциялаушы түрдегі жүйелер туралы олай деп айта алмаймыз. Сонғыларында аталған

қызметтерді қолмен программалауға, немесе әкімшелердің құзырына қалдыруға болады.

Қосымша жасау құралдарын таңдау кезінде келесі негізгі үш факторды: компьютердің ресурстарын, қосымшаның ерекшеліктерін (программаның қызметтерін модификациялаудың қажеттігі, жасауға жұмсалатын уақыт, қол жеткізуді бақылаудың қажеттілігі және ақпараттың тұтастығын қамтамасыз ету) және жасаудың мақсатын (шеттетілетін программалық өнім немесе өзінің күнделікті іс-әрекетін автоматтандыру жүйесі) ескерген дұрыс.

Қазіргі заманғы компьютері бар және онша күрделі емес қосымшаны құруды жоспарлап отырған пайдаланушы үшін интерпретациялау типіндегі МҚБЖ көбірек сай болады. Еске сала кейік, мұндай жүйелер жеткілікті дәрежеде күшті болып келеді, олардың жоғары деңгейлі құралы бар, олар жасау мен баптауға ыңғайлы, жасау жұмысын жылдам түрде орындауға мүмкіндік береді және қосымшаға еріп жүру мен оны модификациялаудың ыңғайлы болуын қамтамасыз етеді.

Сипаттамалары нашар компьютерді пайдаланған кезде тәуелсіз қосымшаларды жасау құралдары бар жүйені таңдаған дұрыс.

Бұл ретте ескертетін нәрсе, қосымшадағы кішігірім өзгеріс программалау, программаны компиляциялау және баптау кезеңдерін циклдік түрде қайталануына әкеледі. Тәуелсіз қосымшаны орындау мен қосымшаны интерпретация режиміндегі орындаудан айырмашылығы тәуелсіз қосымшаның пайдасына миллисекунд айналасында тербелуінде болады. Сонымен қатар, қосымшаны оны пайдалануға дайындау уақытындағы айырмашылық әдетте интерпретациялау жүйелерінің пайдасына минуттар-сағаттар дәрежесіндегі шама құрайды.

SQL және мәліметтер қорының қауіпсіздігі. МҚБЖ-нің қызметтерінің біріне мәліметтер қорының қауіпсіздігін қамтамасыз ету жатады. Реляциялық МҚБЖ-лардың қауіпсіздік жүйесінің негізі SQL тілі болып табылады. Мәліметтерді қорғаудың негізгі үш принципі қолданылады:

- мәліметтерді оқуға, енгізуге және жаңартуға арналған кез келген SQL-команда қандай да бір пайдаланушының атынан орындалады;

- қорғау объектілері негізінен кестелер мен бейнелеу болады;

- әрбір пайдаланушыға қандай да бір әрекеттерді жасауға арналған белгілі бір артықшылықтар (құқықтар) беріледі.

Реляциялық мәліметтер қорының әрбір пайдаланушысына идентификатор – пайдаланушыны бірімәнді анықтайтын қысқаша атау меншіктеледі. Коммерциялық, өнеркәсіптік мәліметтер қорларында пайдаланушыға атауды мәліметтер қорының әкімшісі береді. Дербес ЭЕМ-дарда мәліметтер қорында мәліметтер қорын құрған пайдаланушының тек бір ғана идентификаторы болуы мүмкін. Көптеген МКБЖ-лар үшін идентификатор ретінде операциялық жүйеде тіркелген пайдаланушылардың атаулары қолданылады. Әдетте, пайдаланушы жұмыс алдында идентификаторды және сонымен байланысты парольді енгізуі тиіс. Әртүрлі пайдаланушылардың бірдей идентификаторлары болуы мүмкін.

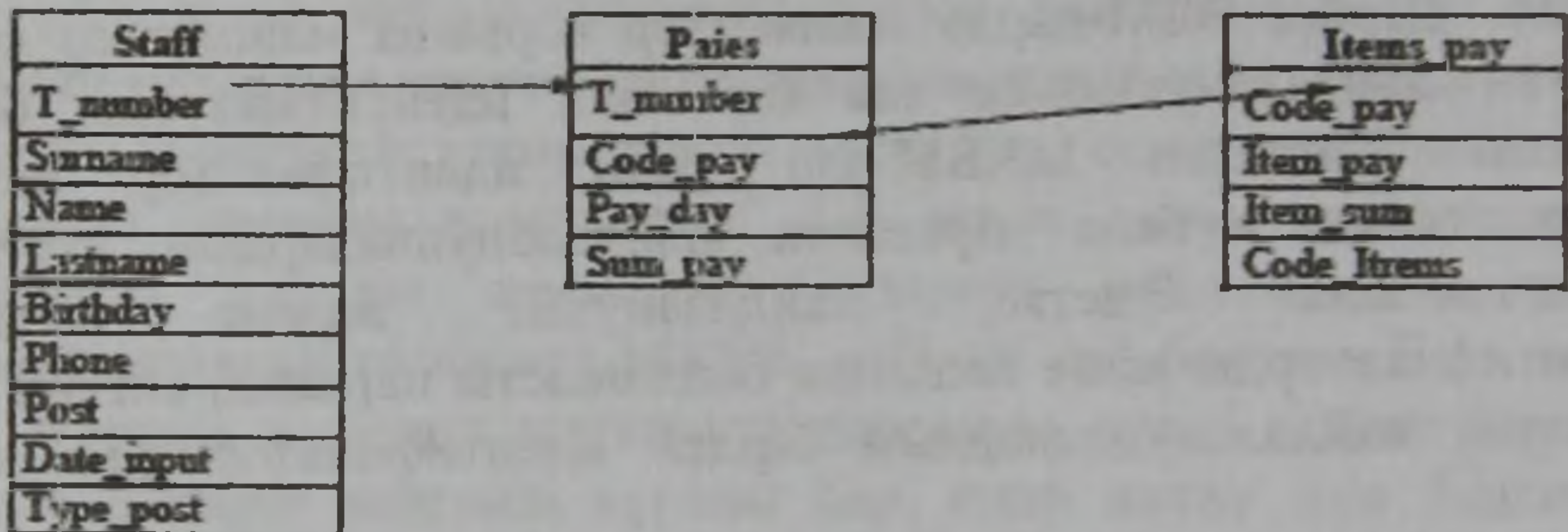
CREATE TABLE және CREATE VIEW операторлары көмегімен кестені немесе бейнелеуді құрған пайдаланушы автоматты түрде осы объектілердің иесіне айналады және осыларға қатысты жоғарыда аталған барлық артықшылықтарға ие болады.

Бақылау сұрақтары

- 1 Иерархиялық және желілік мәліметтер қорының тұтастығы нені білдіреді?
- 2 Мәліметтер қорының желілік моделі дегеніміз нені білдіреді?
- 3 Мәліметтер қорының желілік моделі дегеніміз нені білдіреді?
- 4 Реляциялық мәліметтер қоры дегеніміз не?
- 5 Қандай мәліметтер қорларында мәліметтерді байланыстыру үшін физикалық көрсеткіштер қолданылады?
- 6 Нысанның бір элементі үшін өрістер мәндерінің жиынтығы не болып табылады?
- 7 Мәліметтер қорында өріс деп нені айтамыз?
- 8 Кортөж дегеніміз не?
- 9 Мәліметтер қорының құрылымы мәліметтердің берілуінің қай деңгейінде сипатталады?
- 10 Мәліметтердің берілуінің қандай деңгейі тасымалдағыштағы мәліметтердің орналасу тәсіліне сәйкес?

Тәжірибелік бөлім

Мәліметтер қоры арқылы қызметкерлердің есебі МҚ сақтаудың схемасы. Мәліметтер қорының фрагменті «Төлем ақы» төменде көруге болады.



Кестелер аралағында байланыс екі баға көмегімен өткізіледі «бірге көп» сәйкес болатындағы 1-5 кестелерде көруге болады:

1. Staff.T_number- Paies.T_number.
2. Paies.Code_pay - Items_pay.Code_pay.

1-Кесте – Қызметкерлердің тізімі (кесте «Staff»)

Бағаның өрісі	Бағаның типі
T number	Character
Name	Character
Lastname	Character
Birthday	Date
Phone	Numeric
Post	Character
Type post	Character
Date input	Date

2-Кесте – Берілген төлем ақысының есебінің «Paies» кестесі

Бағаның өрісі	Бағаның типі
T number	Integer
Code day	Integer
Pay day	Date
Sum day	Numeric

3-Кесте – Әрбір төлемақысынын «Items_pay»

Бағаның өрісі	Бағаның типі
Code day	Integer
Item pay	Character
Item sum	Numeric
Code items	Integer

4-Кесте – «Staff» кестенің толтырудың мысалы

T_number	Surname	Name	Lastname	Birthday	Phone	Post	Type_post
1	Ахметов	Арман	Нурла-нулы	12.07.1972	766565	Бухгалтер	Қызметкер
2	Абенова	Роза	Алмазовна	24.04.1971	435232	Кадрлардың бөлімнің бастығы	ИТҚ
3	Советов	Аскар	Маултаевич	30.10.1988	223243	Кадрлардың бөлімнің маманы	Қызметкер
67	Ризаев	Аман	Мирасович	14.11.1968	342342	Бас инженері	ИТҚ
4	Матаева	Айсұлу	Амировна	11.05.1983	342322	Құрылысшы	Жұмыскер
11	Сакенов	Ренат	Айлулович	06.07.1987	456544	Бухгалтер	Жұмыскер
15	Камилова	Рафия	Жумашевна	23.12.1989	756443	Құрылысшы	Жұмыскер

5-Кесте -- «Paies» кестенін толтырудың мысалы

T number	Code pay	Pay day	Sum pay
1	1	01.01.2003	2544.00
1	2	01.02.2003	4521.00
1	3	01.03.2003	12542.00
2	4	01.01.2003	1452.00
2	5	01.02.2003	2145.00

Тапсырмалардың нұсқаулар:

1. Staff кестегі қызметкерлердің барлық мәліметтерді шығару және олардың нәтижелерді табельдық нөмірі бойынша сұрыптау:

```
SELECT * FROM Staff ORDER BY T_number
```

SELECT, FROM – кілттік сөз

2. Қызметкерлердің аты, жөнің, тегілердің тізімі, олардың кеңселерді шығару, нәтижелерді кеңсе атаулардың артуы және тегілерді кемуі бойынша сұрыптау

```
SELECT Surname, Name, Lastname, Post FROM Staff ORDER BY Post ASC, Surname DESC
```

3. Paies кестенің қызметкерлердің табельдық нөмірлері және төлемақы алатын даталар тандау және нәтижелерді датаны кемуі бойынша сұрыптау.

```
SELECT T_number, Pay_day FROM Paies ORDER BY Pay_day DESC
```

4. Staff кестенің қызметкерлердің мәліметтерді бағандарды шығару солай: Name, Lastname, Surname, Post, Date_input, Phone, Birthday, T_number, Type_post (см. рис. 3):

```
SELECT Name, Lastname, Surname, Post, Date_input, Phone, Birthday, T_number, Type_post FROM Staff
```

5. Paies кестенің бағандарды тандау солай: Sum_pay, Pay_day, T_number, Code_pay:

```
SELECT Sum_pay, Pay_day, T_number, Code_pay FROM Staff
```

6. Тізімі, аты және әкесінің аты қызметкерлері (кен тегі, аты, Тегі), сондай-ақ олардың жалақы құны (Sum_pay далалық) және алғаннан (Sum_pay далалық) күні шығару:

```
SELECT Surname, Name, Lastname, Sum_pay, Pay_day FROM Staff, Paies WHERE Staff.T_number = Paies.T_number
```

7. Кадрлық сандар, жалақы алған және баптары бойынша өз теңестіру күні, нәтижесі қызметкерлер санының (Сурет 4.) бойынша сұрыпталған шығару:

```
SELECT T_number, Pay_day, Item_pay, Item_sum FROM Paies, Items_pay WHERE Paies.Code_pay = Items_pay.Code_pay ORDER BY T_number
```

8. Қызметкерлерінің аттары және кадрлық сандар тізімі, сондай-ақ олардың жалақы күны мен мақалалар әрбір жалақы бар макетін алған күні шығару:

```
SELECT Surname, Staff.T_number, Sum_pay, Pay_day, Item_pay, Item_sum FROM Staff, Paies, Items_pay WHERE (Staff.T_number = Paies.T_number) AND (Paies.Code_pay = Items_pay.Code_pay)
```

9. «Басшысы» сөзбен басталғанда қызметкерлерінің тізімі шығару:

```
SELECT Surname, Name, Lastname, Post FROM Staff WHERE Post = 'Басшысы'
```

Мынау сұрағыстар өткізу:

- 1) SELECT Surname, Name, Lastname FROM Staff, Paies, Items_pay WHERE (Staff.T_number = Paies.T_number) AND (Paies.Code_pay = Items_pay.Code_pay) AND (Item_pay = 'премия')
- 2) SELECT Surname, Name, Lastname FROM Staff WHERE Surname = 'Ива';
- 3) SELECT Surname, Name, Lastname, Post FROM Staff WHERE NOT(Type_post = 'служащий');
- 4) SELECT Name, Lastname, Surname, Date_input FROM Staff WHERE NOT(Post = 'бухгалтер');
- 5) SELECT Name, Lastname, Surname, Sum_pay FROM Staff, Paies WHERE (Staff.T_number = Paies.T_number) AND Pay_day BETWEEN "1-JAN-2003" AND "1-MAR-2003";
- 6) SELECT Name, Lastname, Surname, Post FROM Staff WHERE Date_input BETWEEN "12-MAR-2000" AND "15-JUN-2000"
- 7) SELECT Name, Lastname, Surname, Phone FROM Staff WHERE Phone BETWEEN 111111 AND 222222;
- 8) SELECT Name, Lastname, Surname FROM Staff WHERE Surname BETWEEN 'P' AND 'Y';

- 9) SELECT Name, Lastname, Surname, Post FROM Staff WHERE Post IN('начальник отдела кадров', 'специалист отдела кадров', 'операционист отдела кадров');
- 10) SELECT Name, Lastname, Surname FROM Staff, Paies, Items_pay WHERE (Staff.T_number = Paies.T_number) AND (Paies.Code_pay = Items_pay.Code_pay) AND (Item_pay IN('премия', 'оплата учебы', 'поощрение'));
- 11) SELECT Name, Lastname, Surname, T_Number FROM Staff WHERE T_Number IN(4, 67, 45, 77);
- 12) SELECT DISTINCT Item_pay FROM Items_pay WHERE Item_pay LIKE "н%";
- 13) SELECT Name, Lastname, Surname FROM Staff WHERE Lastname LIKE "%ва%";
- 14) SELECT DISTINCT Post FROM Staff WHERE Post LIKE "%ль";
- 15) SELECT Name, Lastname, Surname FROM Staff, Paies WHERE (Staff.T_number = Paies.T_number) AND Pay_day = '15-MAR-2003' AND ((Sum_pay >= 2000) AND (Sum_pay < 3000));
- 16) SELECT DISTINCT Name, Lastname, Surname, Staff.T_number FROM Staff, Paies WHERE (Staff.T_number = Paies.T_number) AND ((T_Number BETWEEN 12 AND 30) OR Sum_pay > 5000);
- 17) SELECT Name, Lastname, Surname, Birthday, T_number FROM Staff WHERE (Birthday BETWEEN '01-JAN-1950' AND '01-JAN-1960') OR (T_number >= 10 AND T_number <= 150);
- 18) SELECT Name, Lastname, Surname FROM Staff INNER JOIN Paies INNER JOIN Items_pay ON Paies.Code_pay = Items_pay.Code_pay ON Staff.T_number = Paies.T_number WHERE Item_pay IN('премия', 'оплата учебы', 'поощрение');
- 19) SELECT DISTINCT Name, Lastname, Surname FROM Staff INNER JOIN Paies ON Staff.T_number = Paies.T_number WHERE (Sum_pay >= 2000) AND (Sum_pay < 3000);
- 20) SELECT Paies.Code_pay FROM Paies INNER JOIN Items_pay ON Paies.Code_pay = Items_pay.Code_pay WHERE Item_pay = 'за бездетность';
- 21) SELECT DISTINCT Name, Lastname, Surname FROM Staff INNER JOIN Paies INNER JOIN Items_pay ON Paies.Code_pay

= Items_pay.Code_pay ON Staff.T_number = Paies.T_number
WHERE Item_pay = 'за бездетность' Вычисления;

22) SELECT Name, Lastname, Surname, Post, (Date() -
Date_input)/365.25 FROM Staff ORDER BY Date_input;

23) SET DATE TO GERMAN && необходима для установки &&
даты в формате дд.мм.гг

24) SELECT Name, Lastname, Surname, Post, Birthday,
CTOD(str(day(Birthday))+ '.' + str(month(Birthday))+ '.' + str(YEAR(
Date())))- DATE() FROM Staff Where CTOD (str(day(Birthday))
+ ' . ' + str (month(Birthday)) + ' . ' + str (YEAR(Date())))-
DATE() >0;

25) SELECT Staff.T_number, Name, Surname, Pay_day, Sum_pay,
(Sum_pay- Item_sum) FROM Staff INNER JOIN Paies INNER
JOIN Items_pay ON Paies.Code_pay = Items_pay.Code_pay ON
Staff.T_number = Paies.T_number WHERE Item_pay = 'за
бездетность';

26) SELECT AVG(Sum_pay) FROM Paies

AVG () - бағанында қамтылған барлық мәндерінің орта мәнін
есептеп береді.

COUNT () - функциясы бағанда қамтылған мәндер санын
санайды.

COUNT (*) - функциясы сұрау нәтижелерінің кестесінде жолдар
санын санайды.

MAX () - функциясы бағанда қамтылған барлық мәндердің
арасында ең жоғары табады.

MIN () - бағанында қамтылған барлық мәндердің арасында ең
төменгі табады.

SUM () - функциясы бағанда қамтылған барлық сандарының
косындысын есептеп береді.

27) SELECT Name, Lastname, Surname, Staff.T_number,
SUM(Sum_pay) FROM Staff, Paies WHERE (Staff.T_number =
Paies.T_number) GROUP BY Staff.T_number

GROUP BY сіз жиындар сұрауын жасауға мүмкіндік береді.

Әдеттегі сұрау дерекқордағы әрбір жол үшін бір қатарда
нәтижесін қамтиды. Барлығы сұрау, керісінше, нақты
сипаттау үшін деректер базасын бірінші жолын қамтиды,
содан кейін сұрау нәтижесінің әр топтың бір қорытынды
жолды қамтиды.

Тармақта GROUP бұл жағдайда әрбір қызметкердің жалпы жалақы есептеу, әр топтың ішіндегі қорытындыларын есептеу мүмкіндік береді. Біз топтың пайдалануы жоқ, болса, онда нәтиже ажыратқыштар қызметкерлерінің барлық қызметкерлердің еңбекақысының мөлшерін алған еді.

- 28) SELECT Name, Lastname, Surname, Staff.T_number, AVG(Sum_pay) FROM Staff, Paies WHERE (Staff.T_number = Paies.T_number) AND (Pay_day BETWEEN '01-JAN-2002' AND '31-DEC-2002') GROUP BY Staff.T_number
- 29) SELECT Post, Count(T_number) FROM Staff GROUP BY Post
- 30) SELECT Min(Date_input), Max(Date_input) FROM Staff
- 31) SELECT Name, Lastname, Surname, Staff.T_number, SUM(Sum_pay) AS Itog FROM Staff, Paies WHERE (Staff.T_number = Paies.T_number) GROUP BY Staff.T_number\
- 32) SELECT Staff.T_number, Name, Surname, Pay_day, Sum_pay, (Sum_pay- Item_sum) AS Sum_With_Nalog FROM Staff INNER JOIN Paies INNER JOIN Items_pay ON Paies.Code_pay = Items_pay.Code_pay ON Staff.T_number = Paies.T_number WHERE Item_pay = 'подходный налог';
- 33) SELECT (Surname + ' ' + Name + ' ' + Lastname) AS FIO FROM Staff;
- 34) SELECT (Surname + ' ' + Name + ' ' + Lastname + ' в должности ' + Post) AS FIO_Post FROM Staff;
- 35) SELECT Name, Lastname, Surname FROM Staff WHERE Date_Input BETWEEN Perem_B AND Perem_E;
- 36) SELECT Name, Lastname, Surname FROM Staff WHERE ((Day(Birthday)+Month(Birthday)*30.5)/365.25-Year(Birthday)+Year(Date())) < Perem;
- 37) SELECT Name, Lastname, Surname FROM Staff WHERE Surname = Perem

Тест

1. Ұқсастықты табыңыз. Мәлімет қоры – бұл...

- А) Пайдаланушылар мәліметтермен жұмыс істеуге арналған бағдарлама ;
- В) Заттық салада қарастырылатын объектінің жағдайын және байланысын бейнелейтін атақталған мәлімет жыйынтығы;
- С) Ақпаратты басқару жүйесі объектіні;
- Д) Ақпараттан тұратын негізгі құрылымдық басқару жүйесі элементі ;
- Е) Көптеген пайдаланушыларға ақпаратты бірлесіп қолдануға , жасауға және жүргізуге арналған тілдік және бағдарламалық құрал-жабдықтардың жиынтығы .

2. Ұқсастықты табыңыз. Мәлімет қорын басқару жүйесі (МҚБЖ)-бұл...

- А) Көптеген пайдаланушыларға ақпаратты бірлесіп қолдануға , жасауға және жүргізуге арналған тілдік және бағдарламалық құрал-жабдықтардың жиынтығы ;
- В) Заттық салада қарастырылатын объектінің жағдайын және байланысын бейнелейтін атақталған мәлімет жыйынтығы;
- С) Ақпараттан тұратын негізгі құрылымдық басқару жүйесі элементі ;
- Д) Статикалық және динамикалық процесстердің мәлімет өзгеруін бейнелеуге арналған техника-бағдарламалық құралдар кешені ;
- Е) Көптеген пайдаланушыларға мәлімет қорын бірлесіп қолдануға тыйым салатын тілдік және бағдарламалық құрал-жабдықтардың жиынтығы .

3. МҚБЖ –ні тілдік қатынас қолданылуы бойынша сұрыптайды:

- А) Ашық және жабық;
- В) Ақпараттық және операциялық;
- С) Мәселе-нысандалған және әмбебаптық;
- Д) Дара қолданылатын және көп қолданылатын;
- Е) Реляциялық және объектілік.

4. Қолдайтын үлгі мәліметі бойынша МҚБЖ –ні сұрыптайды:

- A) Ашық және жабық;
- B) Ақпараттық және операциялық;
- C) Мәселе-нысандалған және әмбебаптық;
- D) Дара қолданылатын және көп қолданылатын;
- E) Реляциялық және объектілік.

5. Қолдану саласы бойынша МҚБЖ –ні сұрыптайды:

- A) Ашық және жабық;
- B) Ақпараттық және операциялық;
- C) Мәселе-нысандалған және әмбебаптық;
- D) Дара қолданылатын және көп қолданылатын;
- E) Реляциялық және объектілік.

6. Әдеттегі мәлімет қоры болып табылады

- A) Пассивті
- B) Активті;
- C) Итерационды;
- D) Итерационно-активті;
- E) Итерационно-пассивті;

7. Егер бекітілген транзакцияқате болып табылса, онда осы транзакцияның әрекетін алып тастайтын транзакцияны орындауды талап етеді:

- A) Қалпына келтіру;
- B) Релаксационды;
- C) Тиым салушы;
- D) Өтемелі;
- E) Реструктуризациялы.

8. COMMIT және ROLLBACK оператор функциясын және транзакцияның моделін анықталған SQL стандарты

- A) ANSI/ISO;
- B) ANS/ISO;
- C) ASI/ISO;
- D) ASSI/ISO;
- E) ANSI/ISO.

9. Тұтастық шектелудің екі түрін айырады:

- A) Сол мезетте тексерілетін және қалтырылатын ;
- B) Тексерілетін тексерілмейтін;
- C) Сол мезетте тексерілетін және тексерілмейтін;
- D) Тексерілмейтін және қалтырылатын;
- E) Дұрыс жайап жоқ.

10. Транзакцияны бекіту және откат мүмкіндігі МҚБЖ-дегі мына функция арқылы қамтамасыз етіледі

- A) Түзету транзакцияның талдауы;
- B) Транзакция журналын жүргізу;
- C) Мәлімет буферизациясы ;
- D) Айырбастау буферіндегі өордын көшіру мәліметі ;
- E) Жауабы А және С.

11. Мәлімет қорының сапасын талдауда компоненттерді нысаналы түрде қарастыру:

- A) Мәліметті басқару бағдарламаның жүйесі
- B) Кейбір ережелерде реттелген мәліметтер жиынтығы
- C) Объектінің толық жинақталған тізімі
- D) А, В
- E) А, В, С

12. Бағдарлама құралдар сапасын қарастыратын стандартты таңда:

- A) Үлгі 9000
- B) Үлгі ISO 9120
- C) Үлгі т ISO 9126
- D) Үлгі ISO 9012
- E) Үлгі ISO 91264

13. Ақпараттың сапа құрылымының мінездемесіне кірмейтін мінездемені көрсет

- A) Қорғаныштық
- B) Тәжірибелік
- C) Сопровождаемость
- D) Мобильдік
- E) Баға құны.

14. Өпдеуге және сақтауға қол жетерлік нақтылы толық санмен салыстырғандағы жазбалардын қатыстық саны бұл..

- A) Мәлімет көлемі
- B) Оперативтілік
- C) Ретроспективаның тереңдігі
- D) Динамикасы
- E) Ұқсастығы.

15. Нақтылы объектілер мәліметтерінің өзгеру жағдайын динамикалық сәйкестігінің дәрежесі

- A) Мәлімет көлемі
- B) Оперативтілік
- C) Ретроспективаның тереңдігі
- D) Динамикасы
- E) Ұқсастығы

16. Ең алғашқы құжаттың шыққан күнінен мәлімет қорына жаралған және /немесе қазіргі кезеңге дейінгі уақыт интервалы

- A) Мәлімет көлемі
- B) Оперативтілік
- C) Ретроспективаның тереңдігі
- D) Динамикасы
- E) Ұқсастығы

17. «Table» компоненті арналған:

- A) BDE жүйесі арқылы мәлімет кестесіне тікелей қол жетерлік үшін
- B) BDE жүйесі арқылы қордың кестесінде жазбаларды түзету/қорғап шығу үшін
- C) Кесте тізбесін және файл атын білмей мәлімет нәтижесін ашу үшін
- D) Мәлімет кестесінде орын ауыстыру үшін
- E) Визуалдық компоненттер және «Table» компоненттер арасындағы байланыс үшін

18. «DBGrid» компоненті арналған:

- A) BDE жүйесі арқылы мәлімет кестесіне тікелей қол жетерлік үшін
- B) BDE жүйесі арқылы қордың кестесінде жазбаларды түзету/қорғап шығу үшін
- C) Кесте тізбесін және файл атын білмей мәлімет нәтижесін ашу үшін
- D) Мәлімет кестесінде орын ауыстыру үшін
- E) Визуалдық компоненттер және «Table» компоненттер арасындағы байланыс үшін

19. «DBNavigator» компоненті арналған:

- A) BDE жүйесі арқылы мәлімет кестесіне тікелей қол жетерлік үшін
- B) BDE жүйесі арқылы қордың кестесінде жазбаларды түзету/қорғап шығу үшін
- C) Кесте тізбесін және файл атын білмей мәлімет нәтижесін ашу үшін
- D) Мәлімет кестесінде орын ауыстыру үшін
- E) Визуалдық компоненттер және «Table» компоненттер арасындағы байланыс үшін

20. «DataSource» компоненті арналған:

- A) BDE жүйесі арқылы мәлімет кестесіне тікелей қол жетерлік үшін
- B) BDE жүйесі арқылы қордың кестесінде жазбаларды түзету/қорғап шығу үшін
- C) Кесте тізбесін және файл атын білмей мәлімет нәтижесін ашу үшін
- D) Мәлімет кестесінде орын ауыстыру үшін
- E) Визуалдық компоненттер және «Table» компоненттер арасындағы байланыс үшін

21. «Query» компоненті арналған:

- A) BDE жүйесі арқылы мәлімет кестесіне тікелей қол жетерлік үшін
- B) BDE жүйесі арқылы қордың кестесінде жазбаларды түзету/қорғап шығу үшін
- C) Кесте тізбесін және файл атын білмей мәлімет нәтижесін ашу үшін
- D) Мәлімет кестесінде орын ауыстыру үшін
- E) Визуалдық компоненттер және «Table» компоненттер арасындағы байланыс үшін

22. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады.мәлімет қорды ашу әдісін көрсетіңіз:

- A) Open
- B) Close
- C) Refresh
- D) Edit
- E) Post

23.«TTable» компонентімен жұмыс істегенде кейбір әдістер болады.мәлімет қорды жабу әдісін көрсетіңіз:

- A) Open
- B) Close
- C) Refresh
- D) Edit
- E) Post

24. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады. мәлімет қорды жаңарту әдісін көрсетіңіз:

- A) Open
- B) Close
- C) Refresh
- D) Edit
- E) Post

25. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады Мәлімет қорын ағымдағы жазбаны жөндеу тәртібіне ауыстыру әдісін көрсетіңіз

- A) Open
 - B) Close
 - C) Refresh
 - D) Edit
 - E) Post
- \$\$\$ 26

26. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады .Ағымдағы жазбаны жөндеуіне арналған есте сақтау өзгерісінің әдісін көрсетіңіз:

- A) Open
- B) Close
- C) Refresh
- D) Edit
- E) Post

27. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады.Ағымдағы жазбаны жөндеуіне арналған өшіру өзгерісінің әдісін көрсетіңіз

- A) Cancel
- B) Insert
- C) Append
- D) Delete
- E) First

28. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады. МҚ жаңа бос жазбаны жою әдісін көрсетіңіз:

- A) Cancel
- B) Insert
- C) Append
- D) Delete
- E) First

29. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады. МҚ-ның соңына жаңа бос жазбаны қосу әдісін көрсетіңіз...

- A) Cancel
- B) Insert
- C) Append
- D) Delete
- E) First

30. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады... МҚ ағымдағы жазбаны жою әдісін көрсетіңіз:

- A) Cancel
- B) Insert
- C) Append
- D) Delete
- E) First

31. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады..МҚ бірінші жазбасына көшу әдісін көрсетіңіз:

- A) Cancel
- B) Insert
- C) Append
- D) Delete
- E) First

32. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады.МҚ-ның соңғы жазбасына өту әдісін көрсетіңіз:

- A) Cancel
- B) Insert
- C) Last
- D) Append
- E) Delete

33. «TTable» комноентімен жұмыс істегенде кейбір әдістер болады. МҚ-ның келесі жазбасына өту әдісін көрсетіңіз:

- A) Cancel
- B) Insert
- C) Next
- D) Append
- E) Delete

34. «TTable» компонентімен жұмыс істегенде кейбір әдістер болады. МҚ-ның алдыңғы жазбасына өту әдісін көрсетіңіз:

- A) Cancel
- B) Insert
- C) Append
- D) Prior
- E) Delete

35. “DBText” компоненті қызмет етеді:

- A) Нақтылы мәлімет қоры өрісінен мәні жағынан МҚ нақты жазбаны таңдау
- B) Тізбе кестесімен файл таын білмей-ақ мәлімет кестесін ашу
- C) Мәлімет кестесінде орын ауыстыру (навигации)
- D) МҚ жазбасы нақты бір ағымдық өрістен мәнді бейнелеуі
- E) МҚ ағымдық жазбасына арналған бір нақтылы тето-өрісінің мәндік бейнеленуі.

36. “DBLookupComboBox” компоненті қызмет етеді:

- A) Нақтылы мәлімет қоры өрісінен мәні жағынан МҚ нақты жазбаны таңдау
- B) Тізбе кестесімен файл таын білмей-ақ мәлімет кестесін ашу
- C) Мәлімет кестесінде орын ауыстыру (навигации)
- D) МҚ жазбасы нақты бір ағымдық өрістен мәнді бейнелеуі
- E) МҚ ағымдық жазбасына арналған бір нақтылы тето-өрісінің мәндік бейнеленуі.

37. “DBMemo” компоненті қызмет етеді:

- A) Нақтылы мәлімет қоры өрісінен мәні жағынан МҚ нақты жазбаны таңдау
- B) Тізбе кестесімен файл таын білмей-ақ мәлімет кестесін ашу
- C) Мәлімет кестесінде орын ауыстыру (навигации)
- D) МҚ жазбасы нақты бір ағымдық өрістен мәнді бейнелеуі
- E) МҚ ағымдық жазбасына арналған бір нақтылы тето-өрісінің мәндік бейнеленуі.

38. Мәліметтерді ұжымды көп мақсаттылы қолдану және орталық жиналуын қамтамасыз ету үшін арналған тілдік, техникалық, бағдарламалық, ұйымдастыру-методикалық, құралдық, мәлімет-қордың- ұйымдық мәліметтік –арнайы орісжүйе бұл...

- A) Мәлімет банкі;
- B) Мәлімет қоры;
- C) МҚБЖ;
- D) МҚ архитектурасы ;
- E) Дұрыс жауап жоқ..

39. МҚ қайнар түсінігі болып табылады:

- A) Ақпараттық жүйе;
- B) Мәлімет сәйкестігі;
- C) Қосынды ақпарат;
- D) Сауал тілдері;
- E) Ақпараттың нақтылығы.

40. МҚБЖ-ні іске асыруының ең қабілетті әдісі америкалық үлгі коммитеті ұсынған МҚ-ды ұйымдастыратын үшдеңгейлік жүйесі болып табылады :

- A) ANS;
- B) ASI;
- C) ANCI;
- D) ANSI;
- E) ASSI.

41. МҚБЖ-нің үшдеңгейлік үлгісіне жататын деңгейді көрсетіңіз:

- A) Сыртқы үлгілер, концептуалды, логикалық;
- B) Сыртқы үлгілер , физикалық, логикалық;
- C) Сыртқы үлгілер, концептуалды, физикалық;
- D) Физикалық, аналитикалық, концептуалды;
- E) Аналитикалық, логикалық, концептуалды.

42. Үш деңгейлі МҚБЖ-нің мәліметтерді өзіндік жүргізетін үлгісінің ең жоғары деңгейін көрсетіңіз:

- A) Концептуалды;
- B) Логикалық;
- C) Физикалық;
- D) Аналитикалық;
- E) Сырқы үлгі.

43. МҚ мәліметтерімен жұмыс істейтін барлық қосымшалар қолданылатын мәліметтерді біріктіретін және МҚ жалпы түрде көрініп өзіне орталық басқару түрін көрсететін деңгейді көрсет:

- A) Аналитикалық;
- B) Концептуалды;
- C) Физикалық;
- D) Логикалық;
- E) Сырқы үлгі .

44. Сырқы ақпарат тасушыларда орналасқан құрылым парақтары немесе файлдар түрінде өзімен мәлімет көрсететін деңгейді көрсет.

- A) Аналитикалық;
- B) Концептуалды;
- C) Физикалық;
- D) Логикалық;
- E) Сырқы үлгі .

45. МҚБЖ-мен біртұтас болып қарастырылатын МҚ-мен операциялар тізімі:

- A) Транзакция;
- B) Журнализациялау;
- C) Буферизациялау;
- D) Минимизациялау;
- E) Оптимизациялау.

46. Көп таралған реляциялық МҚБЖ-ның үлгі тілі болып табылады:

- A) DML;
- B) SDL;
- C) SLQ;

- D) SQL;
- E) IDS.

47. Семантикалық оптимизацияның мақсаты – бұл...

- A) Қате тұжырымдалған немесе карама-қайшы талаптары бар кейбір сауалдарды алып тастау;
- B) Қате тұжырымдалған немесе карама-қайшы талаптары бар кейбір сауалдарды алып тастау және түзету;
- C) Қате тұжырымдалған немесе карама-қайшы ұсынылған кейбір сауалдарды орындау мүмкіндігі;
- D) Қате тұжырымдалған талаптардың түрөзгеріс процедурасын жасау;
- E) Мәліметтерді қайта қалпына келтіретін процедурасын жасау

48. Сауал қате болып есептеледі. Егер ...

- A) Олардың нәтижесі тек қана мәліметтер жиынтығы түрінде алынады;
- B) Олардың нәтижесі мәліметтер жиынтығы түрінде алынбайды ;
- C) Олардың нәтижесі құрылымдалған түрде бейнеленбейді;
- D) Олардың нәтижесі тек қана құрылымдалған түрде бейнеленеді ;
- E) Олардың нәтижесі мәліметтер жиынтығы түрінде және құрылымдық түрде алынбайды.

49. Процедуралық ұсыныс немесе жоспарлық сауалды орындау болып оның осындағы ұсынысы болады:

- A) Мәлімет қорының логикалық деңгейіне рұқсат ету операциясының орындалу тәртібі бөлшектенген;
- B) Мәлімет қорының аналитикалық деңгейіне рұқсат ету операциясының орындалу тәртібі бөлшектенген;
- C) Мәлімет қорының барлық деңгейіне рұқсат ету операциясының орындалу тәртібі бөлшектенген;
- D) Мәлімет қорының физикалық деңгейіне рұқсат ету операциясының орындалу тәртібі бөлшектенген;;
- E) Дұрыс жауап жоқ..

50. Сауалды орындау жоспарын таңдау өлшеуіші боып табылады:

- A) Баға функциясын максимизациялау ;
- B) Баға функциясын минимизациялау;
- C) Баға функциясын нольге айналдыру;
- D) Баға функциясы тұрақты болу керек ;
- E) Дұрыс жауап жоқ.

51. МҚБЖ-ның даму көрсеткіші:

- A) Транзакция механизімін ұстап тұру;
- B) Транзакция механизімінің болмауы ;
- C) Салынған процедуралардың қолданылуы;
- D) МҚ үлестірілген әдісті қолдану ;
- E) Дұрыс жауап жоқ.

52. Транзакцияны бекіту – бұл ...

- A) Транзакцияны орындау барысында жасалған мәлімет қорындағы өзгерістерді шұғыл жадыға жазуды қамтамасыздандыратын әрекет .
- B) Істен шығу жағдайында мәліметтерді қалыптастыруды қамтамасыз ететін әрекет ;
- C) Транзакцияны орындау барысында МҚ-ғы өзгерістерді таспаға жазуды қамтамасыз ететін әрекет ;
- D) Бұл мәліметтерге қате үндеуде қамтамасыз етілген және бүтіндігі расталған барлық мәліметтерді оқшаулауға бағытталған әрекет;
- E) Дұрыс жауап жоқ.

53. Транзакцияның откаты – бұл...

- A) Аяқталмаған транзакция ағынындағы SQL операторымен жасалған мәліметтердегі барлық өзгерістерді өшіруді қамтамасыз ететін әрекет .
- B) Аяқталған транзакция ағынындағы SQL операторымен жасалған мәліметтердегі барлық өзгерістерді өшіруді қамтамасыз ететін әрекет
- C) Осы мәліметтерге қате үндеуден қамтамасыз етілген және бүтіндігі расталған барлық мәліметтерді оқшаулауға бағытталған әрекет;

- D) Ерекше асуал пайда болған жағдайда мәліметтерді қалыптастыруды қамтамасыз ететін әрекет ;
- E) Дұрыс жауап жоқ.

54. Біршама уақыт интервалында объектінің өзгермелі салыстырмалы санын жалпы санға жарылуын басылым кезеңінде анықтайтын МҚ болжамы дегеніміз –бұл...

- A) Мәлімет көлемі
- B) Оперативтілік
- C) Ретроспектива тереңдігі
- D) Динамикасы
- E) Ұқсастық

55. Объектілердің жиналған жазбалардың толықтығы – бұл....

- A) Берілген тематика бойынша объектілердің жалпы санға немесе МҚ-ғы жалпы санға қатысты мәлімет қорында сақталған құжаттардың немесе объектілердің салыстырмалы саны
- B) Кемістік және қателік болмайтын объектілердің салыстырмалы жазылу санының МҚ объектілер құжаттарының жалпы санына қатынасы
- C) МҚ объектілердің ескірген мәліметтердің салыстырмалы санының, жинақталған және өңделетін мәліметтердің жалпы санына қатынасы
- D) Мәліметтер өзгерісінің динамикасын объектілердің шынайы жағдайының сәйкестік дәрежесі
- E) Сыртқы ортадағы шынайы объектілердің толық санымен, өңдеуге және сақтауға рұқсат етілген санымен салыстырғандағы жазбалардың қатыстық саны

56. Ұқсастық – бұл....

- A) Берілген тематика бойынша объектілердің жалпы санға немесе МҚ-ғы жалпы санға қатысты мәлімет қорында сақталған құжаттардың немесе объектілердің салыстырмалы саны
- B) Кемістік және қателік болмайтын объектілердің салыстырмалы жазылу санының МҚ объектілер құжаттарының жалпы санына қатынасы

С) МҚ объектілердің ескірген мәліметтердің салыстырмалы санының, жинақталған және өңделетін мәліметтердің жалпы санына қатынасы

Д) Мәліметтер өзгерісінің динамикасын объектілердің шынайы жағдайының сәйкестік дәрежесі

Е) Сыртқы ортадағы шынайы объектілердің толық санымен, өңдеуге және сақтауға рұқсат етілген санымен салыстырғандағы жазбалардың қатыстық саны

57. Актуалдық – бұл....

А) Берілген тематика бойынша объектілердің жалпы санға немесе МҚ-ғы жалпы санға қатысты мәлімет қорында сақталған құжаттардың немесе объектілердің салыстырмалы саны

В) Кемістік және қателік болмайтын объектілердің салыстырмалы жазылу санының МҚ объектілер құжаттарының жалпы санына қатынасы

С) МҚ объектілердің ескірген мәліметтердің салыстырмалы санының, жинақталған және өңделетін мәліметтердің жалпы санына қатынасы

Д) Мәліметтер өзгерісінің динамикасын объектілердің шынайы жағдайының сәйкестік дәрежесі

Е) Сыртқы ортадағы шынайы объектілердің толық санымен, өңдеуге және сақтауға рұқсат етілген санымен салыстырғандағы жазбалардың қатыстық саны

58. ANSI ұсынған МҚБЖ-нің үшдеңгейлік үлгісі.:

Үшдеңгейлік үлгіге жатпайтын деңгейді таңдаңыз:

А) Сыртқы модель деңгейі;

В) Концептуалды деңгейі;

С) қосымша деңгейі;

Д) Физикалық деңгейі;

Е) Дұрыс жауап жоқ..

59. МҚБЖ-ге жатпайтын талапты таңдаңыз:

А) Операциондық, жасау-жабдық және кең диапазондағы әртүрлілік заттық саладағы МҚ бүтіндігін қамтамасыз ету;

В) Тармақталған МҚ басқару, тұтас емес МҚ-ның интеграциясы;

- C) Мәліметтер мультибазасын басқару (жалпы жағдайда олардың тұтастығы емес);
- D) Ақпарат ағыны арқылы қорды басқару;
- E) Техникалық білім қорын жетілдіру, білім және мәлімет қоры концепциясының интеграциясы, дедукциялық МҚ-ның орындалуы.

60. МҚБЖ сұрыпталады:

- A) Қолданылатын тілдік катынас бойынша;
- B) Орындалатын функциясы бойынша;
- C) Қолдану ортасы бойынша ;
- D) Б, В нұсқасы;
- E) А, Б, В нұсқасы.

61. МҚ-на жатпайтын моделді тап:

- A) Иерархиялық;
- B) Реляциондық;
- C) Транзиттік
- D) Желілік;
- E) Дұрыс жауап жоқ.

62. Өрістегі мәндер қалай жіктелуі керек:

- A) 1, 2, 3, ... ,100;
- B) 1, 3, 2, 4;
- C) 1, 10, 100, 2, 20, 200;
- D) 1, 3, 7, 15, ...;
- E) 100, 99, 98, 97,....

63. Отчет құрастыру кезінде Delphi 6 қандай компоненттер жиынтығы қолданылады:

- A) Win 32;
- B) Additional;
- C) Standart;
- D) BDE;
- E) Qreport.

64. Сыртқы жадыдағы мәліметтерді басқару функциясы:

- A) Істен шыққан жағдайда мәліметтерді қалпына келтіру операциясы;
- B) Мәліметтерді тікелей сақтауға және қызметтік максатқа арналған сыртқы жадыдағы керекті құралды қамтамасыз ету;
- C) Сыртқы жадыда, мәліметтерді тек қана сақтауға арналған керекті құрылымды қамтамасыз ету;
- D) Сыртқы жадыда, қызметтік максатқа арналған керекті құрылымды қамтамасыз ету;
- E) Қызметтік мәліметтерді қалпына келтіру операциясы.

65. Логикалық құрылымдарда ешқандай өзгерісті тудырмай-ақ, мәліметті сақтау ортада организациялау әдістерінің модификациялауға кейбір бостандық берудің МҚБЖ-нің қабілеттілігі:

- A) «Логикалық» бостандық;
- B) «Алгоритімдік» бостандық;
- C) «Физикалық» бостандық;
- D) «Реакциондық» бостандық;
- E) «Полиморфикалық» бостандық.

66. Жаңа тұжырымдалған сұлбаға сәйкесінше келесі берілген мәлімет қорындағы концептуалды модификациялық сұлба:

- A) Физикалық реструктуризация;
- B) Алгоритімдік реструктуризация;
- C) Полиморфикалық реструктуризация;
- D) Мнемондық реструктуризация;
- E) Логикалық реструктуризация.

67. Тілдік сауалда қалыптасқан, бірінші сатыдағы сауал ұшырайды:

- A) Логикалық оптимизациялауға;
- B) Физикалық оптимизациялауға;
- C) Синтаксисті анализге;
- D) Синтаксисті және аналитикалық анализге;
- E) Аналитикалық анализге және логикалық оптимизацияға.

68. Ішкі көрсетілімдегі тілдік сауалда қалыптасқан, екінші сатыдағы сауал келесіге ұшырайды:

- A) Логикалық оптимизациялауға;
- B) Физикалық оптимизациялауға;
- C) Синтаксисті анализге;
- D) Синтаксисті және аналитикалық анализге;
- E) Аналитикалық анализге және логикалық оптимизацияға.

69. Екінші фазаны орындағаннан кейін сауалды өңдеу оның ішкі көрсетілімінде келесідей болып қалады :

- A) Тәуелсіз;
- B) Процедуралық емес;
- C) Процедуралық жоспар түрінде;
- D) Жабық;
- E) Жөнделмейтін.

70. Сақталатын процедуралар – бұл...

- A) Бағдарламалық кодтың бір фрагмент ретінде орындалған және атақталған объект ретінде SDL-дің ұсынысы;
- B) Бағдарламалық кодтың бір фрагмент ретінде орындалған және атақталған объект ретінде DML -дің ұсынысы мәлімет қорында сақталған жиынтық терім;
- C) Бағдарламалық кодтың бір фрагмент ретінде орындалған және атақталған объект ретінде SQL ұсынысы мәлімет қорында сақталған жиынтық терім;
- D) Бағдарламалық кодтың бір фрагмент ретінде орындалған және атақталған объект ретінде DML және SDL ұсынысы мәлімет қорында сақталған жиынтық терім;
- E) Дұрыс жауап жоқ..

71. Триггер (МК қатысты) – бұл...

- A) Сақталған функцияның әдейі түрі;
- B) Сақталған оюьектінің әдейі түрі
- C) Сақталған айнымалының әдейі түрі;
- D) Сақталған процедураның әдейі түрі;
- E) Сақталған типтік тұрақтының әдейі түрі.

72. Тікелей басқа триггерлерді шақыратын триггерлер аталады :

- A) Полиморфикалық;
- B) Итерациондық;
- C) Ортогоналды;
- D) Рекурсивті;
- E) Ішіне салынған.

73. Триггер әрқашан кейбір әрекетті орындайды және ...

- A) Әрқашан пайдаланушыларға мәліметтерді қайтарады;
- B) Кейде пайдаланушыларға мәліметтерді қайтарады;
- C) Істен шыққын және ерекше жағдайда пайдаланушыларға мәліметтерді қайтарады ;
- D) Өткізілген операцияның отчетін пайдаланушыларға береді ;
- E) Мәліметтерді ешқашан пайдаланушыларға қайтармайды .

74. МҚ-ын белсенді ортаға айналуын қамтамасыз ететін қайнар компонентер:

- A) Сақталатын процедуралар;
- B) Триггерлер;
- C) Сақталатын функциялар;
- D) DML процедурасы;
- E) SDL процедурасы.

75. Бірінші қасиеттің таяу кезінде, мысалы DEC RSX және VMS фирмасының операциялық жүйесін файылдық жүйе түрінде пайдаланушылар ұсынады...

- A) Жазбалар тізбегі ;
- B) Блок бөлігі;
- C) Виртуалдық жадының парағы;
- D) Байттар тізбегі;
- E) Файлдар позициясы.

76. UNIX операциялық жүйесімен бірге, тармақталған екінші таяу келесіден тұрады, әр файл өзімен ұсынады :

- A) Жазбалар тізбегі;
- B) Блок бөлігі;
- C) Виртуалдық жадының парағы;
- D) Байттар тізбегі;
- E) Файлдар позициясы.

77. МҚБЖ-ні таңдаған кезде өткізілетін мерекеге жатпайтын мереке:

- A) Өткізілетін оқу компетенциялық саланы анықтағанда;
- B) Өнімдерді 3-4 талапкерге тізімді қысқарту ;
- C) Өнім бағасы;
- D) Отчетті дайындау алдында және дәйектілікті таңдау өткізгенде ;
- E) Компетенциялық саланы тексергенде .

78. Мәлімет қабыршағының үлгісі неге негізделген:

- A) Квадраттық бағдарламалық әдісін қолдануда;
- B) Сызықтық бағдарламалық әдісін қолдануда;
- C) Физикалық бағдарламалық әдісін қолдануда;
- D) Ортогональдық бағдарламалық әдісін қолдануда;
- E) Барлық жауаптар дұрыс.

79. Әдеттегі МҚБЖ-ның құрастырылуы. МҚБЖ ядросы келесіге жауап береді:

- A) Сыртқы жадыда мәліметтерді басқару, оперативтік жадының буферларын басқару, журнализацияны және транзакцияны басқару.
- B) Кейбір орындалатын бағдарламада мәлімет қорының тілдік операторлардың компиляциясы.
- C) Мәлімет қорында тілді қолданғанда асыра қиын орындау процедурасы, мысалы, МҚ-ын жүктеу, статистиканы жөндеу, бүтіндікті глобалды тексеру.
- D) Барлық жауаптар дұрыс.
- E) Дұрыс жауап жоқ.

80. Әдеттегі МҚБЖ-ның құрылымы. Тілдік МҚ-ның компилятордың негізгі функциясы.

- A) Сыртқы жадыда мәліметтерді басқару, оперативтік жадының буферларын басқару, журнализацияны және транзакцияны басқару.
- B) Кейбір орындалатын бағдарламада мәлімет қорының тілдік операторлардың компиляциясы.

С) Мәлімет қорында тілді қолданғанда асыра қиын орындау процедурасы, мысалы, МҚ-ын жүктеу, статистиканы жөндеу, бүтіндікті глобальді тексеру.

Д) Барлық жауаптар дұрыс.

Е) Дұрыс жауап жоқ.

81. Әдетте утилиттерді жеке утилиттерге айырады

А) Сыртқы жадыда мәліметтерді басқару, оперативтік жадының буферларын басқару, журнализацияны және транзакцияны басқару.

В) Кейбір орындалатын бағдарламада мәлімет қорының тілдік операторлардың компиляциясы.

С) Мәлімет қорында тілді қолданғанда асыра қиын орындау процедурасы, мысалы, МҚ-ын жүктеу, статистиканы жөндеу, бүтіндікті глобальді тексеру.

Д) Барлық жауаптар дұрыс.

Е) Дұрыс жауап жоқ.

82. Әдеттегі МҚБЖ-ның құрылымы. Көрсетілім – бұл...

А) Бір немесе бірнеше нақтылы МҚ кестесінен мәліметтерді көрсететін виртуалдық кесте.

В) Бағдарламалық кодтың бір фрагмент ретінде орындалған және атақталған объект ретінде SQL ұсынысы мәлімет қорында сақталған жиынтық терім

С) SQL операторлар сәйкестік көмегімен берілген кестеде мәліметтердің өзгеруі немесе жойылуы, қосылуы кезіндегі әдейі сақталатын процедура типі.

Д) Алдыңғы жауаптар дұрыс.

Е) Дұрыс жауап жоқ.

83. Әдеттегі МҚБЖ-ның құрылымы. Триггер – бұл...

А) Бір немесе бірнеше нақтылы МҚ кестесінен мәліметтерді көрсететін виртуалдық кесте.

В) Бағдарламалық кодтың бір фрагмент ретінде орындалған және атақталған объект ретінде SQL ұсынысы мәлімет қорында сақталған жиынтық терім

С) SQL операторлар сәйкестік көмегімен берілген кестеде мәліметтердің өзгеруі немесе жойылуы, қосылуы кезіндегі әдейі сақталатын процедура типі.

Д) Алдыңғы жауаптар дұрыс.

Е) Дұрыс жауап жоқ.

84. Әдеттегі МҚБЖ-нің транзакциясын өңдеудегі кіші жүйесі. Транзакция менеджері орындайды:

А) Қолданбалы бағдарламамен орындалатын транзакция жұмысын координациялау.

В) Транзакцияның бір-біріне параллелдік әсер ету ерекшеліктеріне шартына сәйкес, параллельдік максималды мүмкіндігіне жету деңгейі, себебі ол МҚ келісімділігінің бұзылуына негіз бола алады.

С) МҚ-ның бастапқы транзакцияның орындалу моментіне автоматты түрде қалпына келуіне гарантияның берілуі.

Д) Компьютердің негізгі жадысында және екінші реттік дисктік жадыға мәліметтерді ауыстыру.

Е) Дұрыс жауап жоқ.

85. Жоспарлаушының жұмысы:

А) Қолданбалы бағдарламамен орындалатын транзакция жұмысын координациялау.

В) Транзакцияның бір-біріне параллелдік әсер ету ерекшеліктеріне шартына сәйкес, параллельдік максималды мүмкіндігіне жету деңгейі, себебі ол МҚ келісімділігінің бұзылуына негіз бола алады.

С) МҚ-ның бастапқы транзакцияның орындалу моментіне автоматты түрде қалпына келуіне гарантияның берілуі.

Д) Компьютердің негізгі жадысында және екінші реттік дисктік жадыға мәліметтерді ауыстыру.

Е) Дұрыс жауап жоқ.

86. Қайта қалпына келтіру менеджерінің міндеті:

- A) Қолданбалы бағдарламамен орындалатын транзакция жұмысын координациялау.
- B) Транзакцияның бір-біріне параллелдік әсер ету ерекшелігі шартына сәйкес, параллельдік максималды мүмкіндігіне жету деңгейі, себебі ол МҚ келісімділігінің бұзылуына негіз бола алады.
- C) МҚ-ның бастапқы транзакцияның орындалу моментіне автоматты түрде қалпына келуіне гарантияның берілуі.
- D) Компьютердің негізгі жадысында және екінші реттік дисктік жадыға мәліметтерді ауыстыру.
- E) Дұрыс жауап жоқ.

87. Буфер менеджер жауап береді:

- A) Қолданбалы бағдарламамен орындалатын транзакция жұмысын координациялау.
- B) Транзакцияның бір-біріне параллелдік әсер ету ерекшелігі шартына сәйкес, параллельдік максималды мүмкіндігіне жету деңгейі, себебі ол МҚ келісімділігінің бұзылуына негіз бола алады.
- C) МҚ-ның бастапқы транзакцияның орындалу моментіне автоматты түрде қалпына келуіне гарантияның берілуі.
- D) Компьютердің негізгі жадысында және екінші реттік дисктік жадыға мәліметтерді ауыстыру.
- E) Дұрыс жауап жоқ.

88. Транзакцияны параллельдік өңдеген кезде жойылған өзгерістер. Бұл жағдай келесіден туады:

- A) Екі транзакция өзгерістерді бір жазбада орындаған жағдайда.
- B) Бірінші транзакция қалыптастырған мәліметтерге, екінші транзакцияны орындау барысында рұқсаттың ашылу есебінен.
- C) Екінші транзакция оқыған мәліметтер нәтижесін бірінші транзакция операторы өзгерте алу есебінен.
- D) МҚ келісім қалыпта болғанымен басқа қосымшасы қате жұмыс істегендіктен.
- E) Дұрыс жауап жоқ.

89. Транзакцияны параллельді өндегенде өтпелі мәліметтердің мәселесі. Бұл жағдай келесіден туады:

A) Екі транзакция өзгерістерді бір жазбада орындаған жағдайда.

B) Бірінші транзакция қалыптастырған мәліметтерге, екінші транзакцияны орындау барысында рұқсаттың ашылу есебінен.

C) Екінші транзакция оқыған мәліметтер нәтижесін бірінші транзакция операторы өзгерте алу есебінен .

D) МҚ келісім қалыпта болғанымен басқа қосымшасы қате жұмыс істегендіктен .

E) Дұрыс жауап жоқ.

90. Транзакцияны параллельді өндегенде келісілмеген мәліметтердің мәселесі. Бұл жағдай келесіден туады:

A) Екі транзакция өзгерістерді бір жазбада орындаған жағдайда.

B) Бірінші транзакция қалыптастырған мәліметтерге, екінші транзакцияны орындау барысында рұқсаттың ашылу есебінен.

C) Екінші транзакция оқыған мәліметтер нәтижесін бірінші транзакция операторы өзгерте алу есебінен .

D) МҚ келісім қалыпта болғанымен басқа қосымшасы қате жұмыс істегендіктен .

E) Дұрыс жауап жоқ.

91. Транзакцияны параллельді өндегенде жатық жолдардың мәселесі. Бұл жағдай келесіден туады:

A) Екі транзакция өзгерістерді бір жазбада орындаған жағдайда.

B) Бірінші транзакция қалыптастырған мәліметтерге, екінші транзакцияны орындау барысында рұқсаттың ашылу есебінен.

C) Екінші транзакция оқыған мәліметтер нәтижесін бірінші транзакция операторы өзгерте алу есебінен .

D) МҚ келісім қалыпта болғанымен басқа қосымшасы қате жұмыс істегендіктен .

E) Дұрыс жауап жоқ.

92. Транзакция мониторы- бұл...

- A)Тармақталған ортада ақпаратты-есептік қормен тиімді басқаруды қамтамасыз ететін бағдарлпма жүйесі.
- B) Екі немесе одан да көп транзакция блокировканы босатуды күтеді.
- C) Кейбір сериялы жоспармен транзакцияны орындау механизмі. Сондай механизмді орындау транзакциямен басқаруға жауап беретін МҚБЖ компонентінің негізгі функциясы болады.
- D) Алғашқы транзакция әрекетін жоятын қайта транзакция.
- E) Аяқталмаған транзакция ағынындағы SQL операторымен жасалған мәліметтердегі барлық өзгерістерді өшіруді қамтамасыз ететін әрекет.

93. Тығырық жағдай (dead-locks) – бұл...

- A)Тармақталған ортада ақпаратты-есептік қормен тиімді басқаруды қамтамасыз ететін бағдарлпма жүйесі.
- B) Екі немесе одан да көп транзакция блокировканы босатуды күтеді.
- C) Кейбір сериялы жоспармен транзакцияны орындау механизмі. Сондай механизмді орындау транзакциямен басқаруға жауап беретін МҚБЖ компонентінің негізгі функциясы болады.
- D) Алғашқы транзакция әрекетін жоятын қайта транзакция.
- E) Аяқталмаған транзакция ағынындағы SQL операторымен жасалған мәліметтердегі барлық өзгерістерді өшіруді қамтамасыз ететін әрекет.

94. Транзакцияны сериялдау – бұл...

- A)Тармақталған ортада ақпаратты-есептік қормен тиімді басқаруды қамтамасыз ететін бағдарлпма жүйесі.
- B) Екі немесе одан да көп транзакция блокировканы босатуды күтеді.
- C) Кейбір сериялы жоспармен транзакцияны орындау механизмі. Сондай механизмді орындау транзакциямен басқаруға жауап беретін МҚБЖ компонентінің негізгі функциясы болады.
- D) Алғашқы транзакция әрекетін жоятын қайта транзакция.

Е) Аяқталмаған транзакция ағынындағы SQL операторымен жасалған мәліметтердегі барлық өзгерістерді өшіруді қамтамасыз ететін әрекет.

95. Компенсиялайтын транзакция – бұл...

А)Тармақталған ортада ақпаратты-есептік қормен тиімді басқаруды қамтамасыз ететін бағдарлпма жүйесі.

В) Екі немесе одан да көп транзакция блокировканы босатуды күтеді .

С) Кейбір сериялы жоспармен транзакцияны орындау механизмі. Сондай механизмді орындау транзакциямен басқаруға жауап беретін МҚБЖ компонентінің негізгі функциясы болады.

Д) Алғашқы транзакция әрекетін жоятын қайта транзакция.

Е) Аяқталмаған транзакция ағынындағы SQL операторымен жасалған мәліметтердегі барлық өзгерістерді өшіруді қамтамасыз ететін әрекет.

96. Транзакция откаты (roll back)- бұл...

А)Тармақталған ортада ақпаратты-есептік қормен тиімді басқаруды қамтамасыз ететін бағдарлпма жүйесі.

В) Екі немесе одан да көп транзакция блокировканы босатуды күтеді.

С) Кейбір сериялы жоспармен транзакцияны орындау механизмі. Сондай механизмді орындау транзакциямен басқаруға жауап беретін МҚБЖ компонентінің негізгі функциясы болады.

Д) Алғашқы транзакция әрекетін жоятын қайта транзакция.

Е) Аяқталмаған транзакция ағынындағы SQL операторымен жасалған мәліметтердегі барлық өзгерістерді өшіруді қамтамасыз ететін әрекет.

97. Транзакцияны бекіту (commit) – бұл...

А)Тармақталған ортада ақпаратты-есептік қормен тиімді басқаруды қамтамасыз ететін бағдарлпма жүйесі.

В) Екі немесе одан да көп транзакция блокировканы босатуды күтеді.

С) Кейбір сериялы жоспармен транзакцияны орындау механизмі. Сондай механизмді орындау транзакциямен басқаруға жауап беретін МҚБЖ компонентінің негізгі функциясы болады.

Д) Алғашқы транзакция әрекетін жоятын қайта транзакция.

Е) Транзакцияны орындау барысында жасалған мәлімет қорындағы өзгерістерді таспаға жазуды қамтамасыздандыратын әрекет.

98. Контролді нүкте – бұл...

А)Тармақталған ортада ақпаратты-есептік қормен тиімді басқаруды қамтамасыз ететін бағдарлпма жүйесі.

В) Екі немесе одан да көп транзакция блокировканы босатуды күтеді.

С) Транзакцияның тіркеу журналы және МҚ арасындағы синхронизация моменті.

Д) Алғашқы транзакция әрекетін жоятын қайта транзакция.

Е) Транзакцияны орындау барысында жасалған мәлімет қорындағы өзгерістерді таспаға жазуды қамтамасыз ететін әрекет.

99. Ойды аяқтаңыз. Авторизация – бұл...

А) МҚ объектілерін иесіне заңды түрде рұқсат берілу құқығы.

В) Пайдаланушы өзін рас өзі екенін анықтайтын механизм.

С) Қойылған талаптарға сай қорғаныштық деңгей сәйкес келе ме және қарастырылған басқару құралы әрекеттелген екеніне тексеріс .

Д) Жүйеден бөлек сакталатын,мәлімет қорының көшірмесін алу және тасушыда журнал өзгерістерін алуының периодты түрде орындалатын процедура.

Е) Мәлімет қорының әр түрлі кестелерінен заңды жолмен алынған мәліметтерді комбинерлеу арқылы жаңа ақпараттарды алу әдісі.

100. Ойды аяқтаңыз. Аутентификация – бұл...

- A) МҚ объектілерін иесіне заңды түрде рұқсат берілу құқығы.
- B) Пайдаланушы өзін рас өзі екенін анықтайтын механизм.
- C) Қойылған талаптарға сай қорғаныштық деңгей сәйкес келе ме және қарастырылған басқару құралы әрекеттелген екеніне тексеріс .
- D) Жүйеден бөлек сақталатын, мәлімет қорының көшірмесін алу және тасушыда журнал өзгерістерін алуының периодты түрде орындалатын процедура.
- E) Мәлімет қорының әр түрлі кестелерінен заңды жолмен алынған мәліметтерді комбинерлеу арқылы жаңа ақпараттарды алу әдісі.

Тест бойынша кілттері

6-кесте – Тесттің кілттері

1-күрделік деңгейі		2 -күрделік деңгейі		3 -күрделік деңгейі	
Сұрақ нөмері	Дұрыс нұсқасы	Сұрақ нөмері	Дұрыс нұсқасы	Сұрақ нөмері	Дұрыс нұсқасы
1	B	31	E	71	D
2	A	32	C	72	E
3	A	33	C	73	E
4	E	34	D	74	B
5	C	35	D	75	A
6	A	36	B	76	D
7	D	37	E	77	E
8	E	38	A	78	B
9	A	39	B	79	A
10	B	40	D	80	B
11	D	41	C	81	C
12	C	42	E	82	A
13	E	43	B	83	C
14	A	44	E	84	A
15	B	45	A	85	B
16	C	46	D	86	C
17	B	47	A	87	D
18	A	48	B	88	A
19	D	49	D	89	B
20	E	50	B	90	C

21	D	51	A	91	D
22	A	52	C	92	A
23	B	53	A	93	B
24	C	54	D	94	C
25	D	55	A	95	D
26	E	56	B	96	E
27	A	57	C	97	E
28	B	58	C	98	C
29	C	59	D	99	A
30	D	60	E	100	B
	E	61	C		
	C	62	C		
		63	E		
		64	B		
		65	C		
		66	E		
		67	D		
		68	A		
		69	B		
		70	C		

Әдебиеттер

- 1 Андасова Б. З. Деректер базасының теориясы : оқулық – Астана: Еуразия ұлттық университеті, 2010. – 224 б.
- 2 Андасова Б. З. Реляциялық деректер базасын жобалау : оқу құралы. - Астана, 2010. – 189 б.
- 3 Сыдықов Б. Д., Тюлепбердинова Г. А., Абишева А. Ж. Программалау тілдері : электрондық оқулық. – Алматы : Абай ат. ҚазҰПУ, 2012. - 254 б.
- 4 Ануарбекова Г. Д., Тойганбаева Н. А. Access: электрондық оқулық. – Алматы, 2012. – 237 б.
- 5 Бөрібаев Б., Мадьярова Г. Web-технологиялар. – Алматы : Қаз ККА баспасы, 2007. – 188 б.

Мазмұны

	Кіріспе	3
1	Ақпараттық жүйелер және мәліметтер қоры	4
1.1	Деректер базасының архитектурасы	4
1.2	Ақпараттық жүйелердің архитектурасы	6
1.3	Мәліметтер қоры және Интернет	22
2	Деректер базасының стандартты SQL тілі	26
2.1	SQL тілі	26
2.2	Мәліметтер қорының объектілерін басқару	32
2.3	Индекстарды құру және жою.	
	Ұсыныстармен жұмыс	34
2.4	Мәліметтер қорының қауіпсіздігін басқару	45
3	Мәліметтер қорының моделі	48
3.1	Мәліметтер қорын пайдалану	48
3.2	Реляциялық модель. Реляциялық алгебра	52
3.3	Мәліметтер қорын программалық жабдықтау	55
	Тәжірибелік бөлім	60
	Тест	67
	Тест бойынша кілттері	94
	Әдебиеттер	96

Саринова А. Ж.

**АВТОМАТТАНДЫРЫЛҒАН ЖҮЙЕЛЕР
БАҚЫЛАУДЫҢ БАЗАЛЫҚ БЕРІЛГЕНДЕРІ
ЖОБАЛАУЫ**

Оқу-әдістемелік құралы

Пішімі 60x100 1/16

Тығыздығы 80 гр./м². Қағаздың ақтығы 95%.

Қағазы офсеттік. РИЗО басылымы.

Көлемі 100 бет. Шартты баспа табағы 6,25



«Эверо» баспасында басылымға
дайындалды және басып шығарылды
ҚР, Алматы, Байтұрсынұлы к., 22.
тел.: 8 (727) 233 83 89, 233 83 43,
233 80 45, 233 80 42
e-mail: evero08@mail.ru