

ЛЕКЦИЯ 5. ДЕРЕКТЕР ҚОРЛАРЫ ЖҮЙЕЛЕРІНІҢ НЕГІЗДЕРІ

Деректер қоры (ДҚ) туралы түсінік

Деректер қоры (ДҚ) – қолданбалы бағдарламаларға қарамастан, белгілі бір ережелер бойынша ұйымдастырылған мәліметтер жиынтығы, яғни деректерді сипаттау, сақтау және манипуляциялау принциптері.

Деректер қорының көпшілігі мәліметтерді сақтау және әңдеу үшін қолданылатын компьютерлік құралдар – аппараттық және бағдарламалық қамтамасыз етуді қамтиды. Мұндай мәліметтер қоры әдетте деректермен қарапайым операциялар орындауға мүмкіндік беретін белгілі бір файлдық жүйеге негізделеді.

Деректерді басқару жүйесі дегеніміз – көлемді мәліметтер жиынын тұтынушыларға ыңғайлы түрде бейнелеп, белгілі бір форматта сақтап қана қоймай, оны ары қарай өңдеуге арналған программалар кешені.

Деректер қорын басқару жүйесі (ДҚБЖ) (ағылш. *database management system (DBMS)*) – деректер қорын құру, күту және қолдану қызметін көрсететін бағдарламалық жасақтама.

ДҚБЖ қолдану

а) Банк қызметі: клиент, ақпарат, шот және несиелер мен транзакциялар үшін.

ә) Әуе компаниясы: брондау және кесте туралы ақпарат үшін. Әуе компаниясы деректер қорын географиялық түрде бөлінген алғашқылардың бірі болып, дүние жүзінде орналасқан терминалдар телефон желілері және басқа да деректер желісі арқылы орталық дерекқор жүйесіне қол жеткізді.

б) Университеттер: білім алушылар туралы ақпарат, курстарды тіркеу және бағалар үшін.

в) Кредиттік карта операциясы: несие карталары бойынша сатып алулар және ай сайынғы үзінді көшірмелер жасау үшін.

г) Сату: тұтынушы, өнім және сатып алу туралы ақпарат үшін.

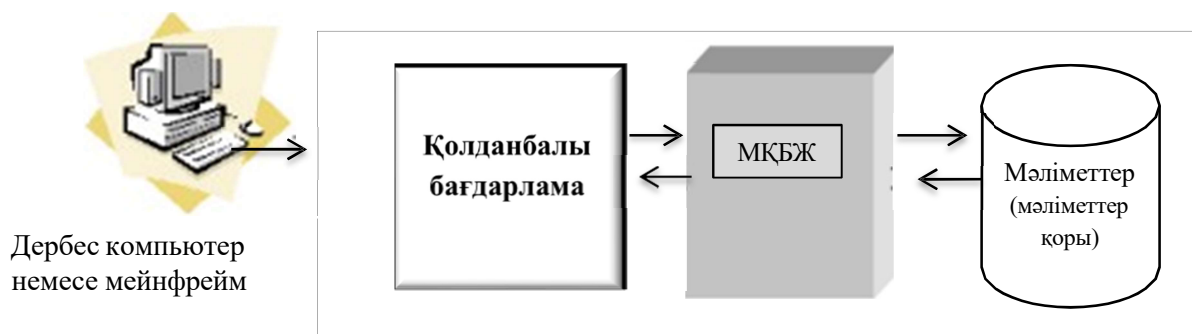
д) Адам ресурстары: қызметкерлер, жалақылар, жалақы салықтары мен жеңілдіктері туралы ақпарат алу, сондай-ақ жалақыны қалыптастыру үшін.

Деректер қорының архитектурасы

Деректер қоры түсінігі бірнеше пайдаланушылардың көптеген мәселелерді шешу мүмкіндігі. Осыған байланысты қазіргі ДҚБЖ ең маңызды сипаттамасы көп қолданушы технологиясының болуы болып табылады.

Орталықтандырылған архитектурасы

Бұл технологияны пайдалану кезінде деректер қоры, ДҚБЖ және қолданбалы бағдарлама (қосымша) бір компьютерде орналасқан (сурет 5.1). Ұйымдастырудың бұл әдісі желіні қолдануды қажет етпейді және барлық деректер қоры бір компьютермен жұмыс жасай береді.



Сурет 5.1. Орталықтандырылған архитектура

Файл-сервер архитектурасы

Тапсырмалардың күрделілігінің артуы, дербес компьютерлер мен жергілікті есептеу желілерінің пайда болуы жаңа файл-сервер архитектурасының пайда болуына негіз болды. Бұл желіге қолжетімді дерекқор архитектурасы желідегі компьютерлердің бірін дерекқор файлдары сақталатын арнайы сервер ретінде белгілеуді қамтиды. Пайдаланушылардың сұраныстарына сәйкес файлдық серверден деректерді әндеудің негізгі бөлігі жүзеге асырылып, пайдаланушының жұмыс станцияларына жіберіледі (сурет 5.2).

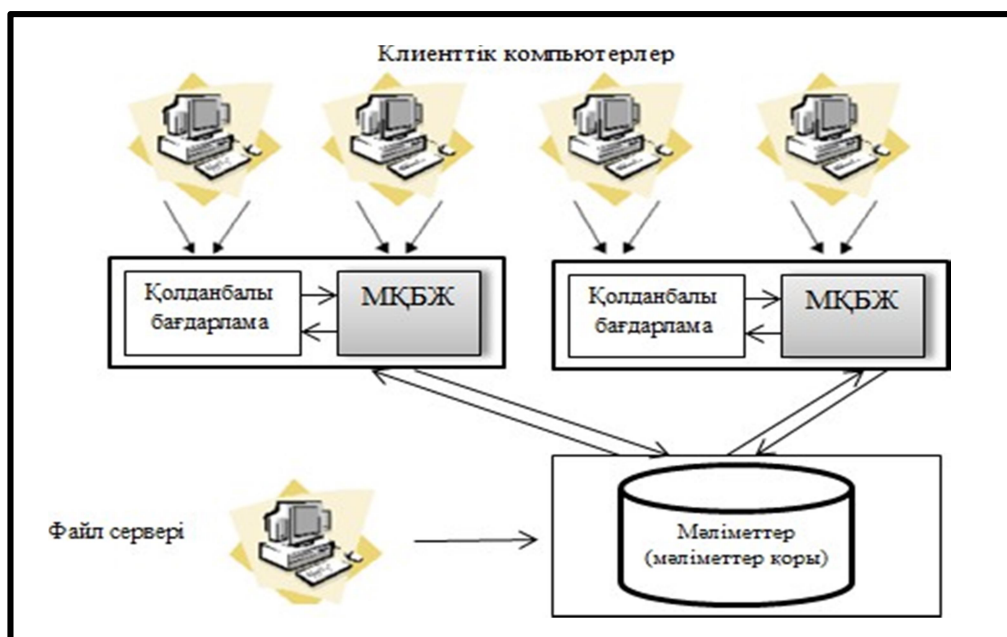
Клиент-сервер архитектурасы

«Клиент-сервер» технологиясын қолдану желіге біріктірілген компьютерлердің белгілі бір санының болуын қажет етеді, олардың бірі (желілік сервер болып табылады) арнайы функцияларды орындайды.

Сонымен, «клиент-сервер» архитектурасы функцияларды пайдаланушы (клиент) және сервер қосымшалары деп екіге бөледі.

Клиент қосымшасы серверге сұраныс жасайды, онда SQL (Structured Query Language) ДҚ реляциялық мәліметтер қоры жұмыс істейді. Қашықтағы сервер сұранысты қабылдайды және оны SQL ДҚ серверіне бағыттайды.

SQL сервері – қашықтағы дерекқорды басқаратын арнайы бағдарлама. SQL сервері сұранысты түсіндіруді, оның дерекқорда орындалуын, сұраныстың нәтижесін орындап және оны клиент-қосымшаға беруді қамтамасыз етеді.



Сурет 5.2. Файл-сервер архитектурасы

Бұл жағдайда клиенттік компьютердің ресурстары сұраныстың физикалық орындалуына қатыспайды; клиенттік компьютер тек сұранысты серверлік мәліметтер қорына жібереді және нәтижені алады, содан кейін оны пайдаланушыға ұсынады (сурет 5.3).

Клиенттік қосымшаға сұраныстың нәтижесі жіберілгендіктен, тек клиентке қажет деректер желі арқылы «саяхаттайды». Нәтижесінде желіге жүктеме азаяды. Сұраныстың орындалуы деректер сақталған жерде (серверде) болатындықтан, үлкен деректер пакеттерін жіберудің қажеті жоқ. Сонымен қатар, SQL сервері, егер мүмкін болса, алынған сұранысты ең аз үстеме шығындармен минималды уақытта орындалатын етіп оңтайландырады.

Деректер қоры архитектурасының үш деңгейі

1. Ішкі деңгей (физикалық деңгей):

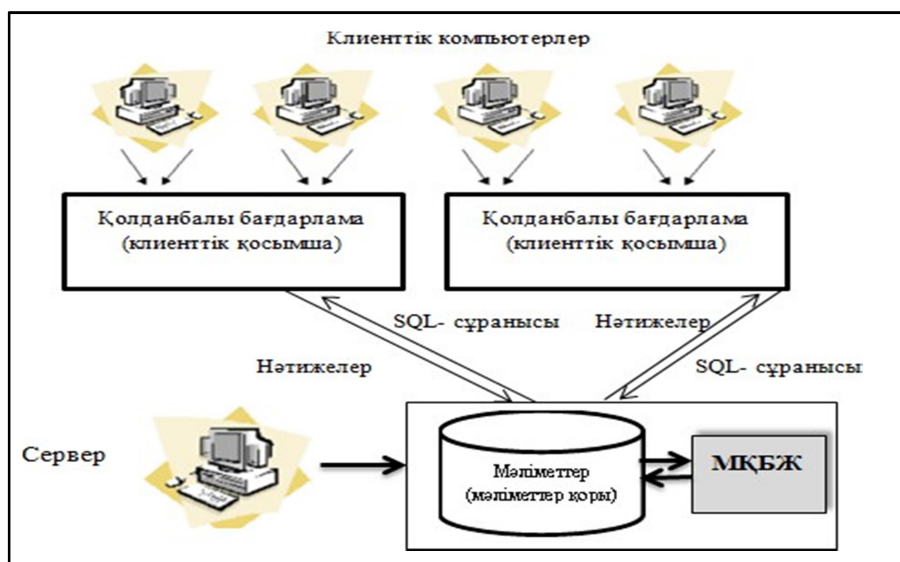
- Ол деректерді тасымалдаушыда қалай сақтайтынын сипаттады.
- Бұл мәліметтер қорының бағдарламашылары.
- Дерекқор абстракциясының ең төменгі деңгейі.

2. Тұжырымдамалық деңгей (концептуалдық деңгей)

- Ол дерекқорда қандай деректер сақталатынын сипаттайды.
- Бұл мәліметтер қорының менеджерлері.
- Бұл дерекқордың абстракциясының аралық деңгейі.

3. Сыртқы деңгей (көрініс деңгейі)

- Ол жеке пайдаланушының деректерді қарау әдісін сипаттайды.
- Бұл пайдаланушылар деңгейі.



Сурет 5.3. Клиент-сервер архитектурасы

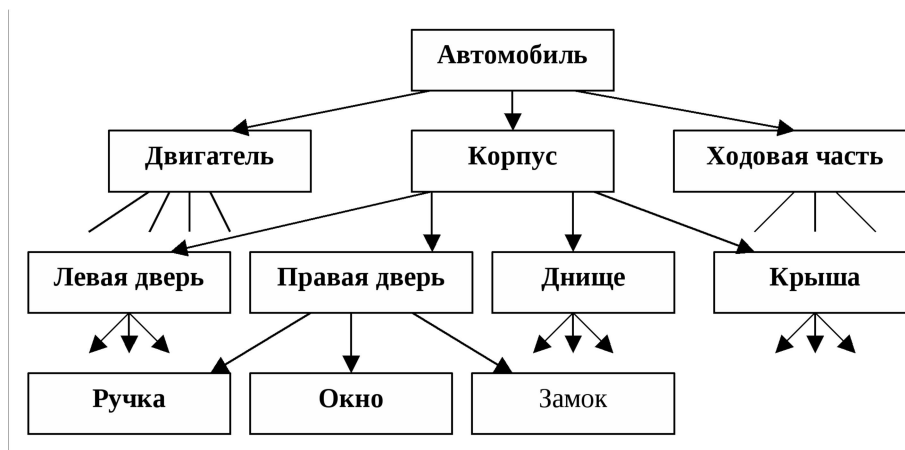
Деректер қорында сақталған деректер белгілі бір логикалық құрылымға ие. ДҚБЖ қолдайтын деректерді ұсынудың кейбір моделімен (деректер моделі) сипатталады. Деректер моделі деректерді ұйымдастыру тәсілін, тұтастықты шектеуді және объектіге рұқсат етілген кәптеген операцияларды анықтайды.

Деректер моделі

Деректер базасының көмегімен шешілетін әралуан есептер деректер базасын құрылымдау мен деректердің арасындағы байланыстарды ұйымдастырудың түрлі әдістерін қалыптастыруға әкеледі. Соның нәтижесінде деректер базасының бірнеше моделі пайда болды. Деректер моделінің 3 түрі бар: иерархиялық, желілік, реляциялық.

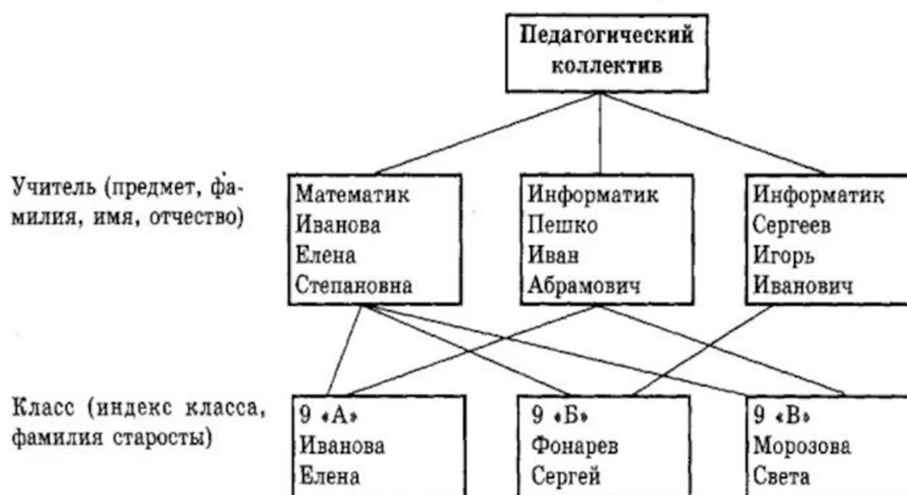
Иерархиялық ДҚ-да жазба элементтері реттеліп жазылады да, оның бір элементі негізгі, қалғандары бағыныңқы элементтер, ал екінші элементке бірінші элементтен басқасының барлығы бағыныңқы болып есептеледі. Мұнда элементтер нақты тізбек бойынша сатылы түрде болады. Онда берілгендерді іздеп табу саты бойынша төмен бағытта жүргізіледі. Мысалы: Windows бумасындағы файлдар жүйесі, Интернет ресурсындағы каталогтар жиыны.

Иерархиялық деректер үлгісінде ақпаратты ұсынудың негізгі құрылымы ағаш болып табылады. Ағаш – циклдері жоқ байланысқан график ретінде анықталады (сурет 5.4).



Сурет 5.4. Иерархиялық модель

Желілік ДҚ-да объектілердің арасындағы өзара байланыс түрлері қарастырылады. Оларда ешқандай ереже болмайды. Егер бұл модельді графикалық түрде бейнелесек – жазықтықта бағыттауыш сызықтармен бейнеленген байланыстар шығады. Мысалы: сілтемемен байланысқан WWW-құжаты (сурет 5.5).



Сурет 5.5. Желілік модель

Реляциялық ДҚ қарым-қатынастарға негізделген модель болып табылады. Реляциялық модельдің негізгі деректер құрылымы – белгілі бір нысан (мысалы, студент) туралы ақпарат бағандар мен жолдарда ұсынылған кесте болып табылады. Бағандар нысанның әртүрлі атрибуттарын (сипаттамалары) сипаттайды (мысалы, студенттің аты, мекенжайы, тіркеу нөмірі). Жолдар (сонымен бірге жазбалар деп аталады) нысанның (мысалы, нақты студент) даналарын көрсетеді (сурет 5.6).

Студенттің аты-жөні	Туылған жылы	Мекен жайы

Сурет 5.5. Реляциялық модель

Реляциялық ДҚБЖ мысалдары:

MicroSoft Access, Paradox, dBASE, FoxPro, Clarion, DB2, Oracle

Сонымен қатар, соңғы жылдары мынадай: постреляциялық, көпөлшемді, объектіге бағытталған деректер моделі пайда болды және іс жүзіне белсендірек енгізіле бастады.

Реляциялық модельде байланыстардың түрлері



Қалыпқа келтіру

Қалыпқа келтіру – бұл деректер қорындағы деректерді ұйымдастыру процесі, соның ішінде кестелерді жасау және деректерді қорғауды қамтамасыз етеді, олардың арасында байланыс орнату және артық және сәйкес келмейтін тәуелділіктерді жою арқылы деректер қорын икемді етеді. Деректердің артық болуы дискілік кеңістікті босқа жұмсайды және деректер қорына қызмет көрсетуді қиындатады. Мысалы, бірнеше жерде сақталған деректерді өзгерту қажет болса, сол орындардың барлығында бірдей өзгертулер оған енгізілуі керек.

Қалыпқа келтіру процесі дерекқордың бастапқы қатынастарын қарапайым қатынастарға ыдыратудан тұрады. Сонымен қатар, осы процестің әр сатысында қарым-қатынас схемалары қалыпты формаларға келтіріледі.

Қалыпқа келтірудің әр сатысы үшін мәліметтер қорының қатынастары қанағаттандыруы керек шектеулер жиынтығы бар. Қалыпқа келтіру процесі атрибуттардың функционалдық тәуелділігі ұғымына негізделген: В атрибуты (В→А) атрибутына байланысты, егер кез келген уақытта В атрибутының әрбір мәні А атрибутының ең көп дегенде бір мәніне сәйкес келсе.

Әрбір негізгі емес атрибут бүкіл құрама кілтке тәуелді және оның бәліктеріне тәуелді емес тәуелділік толық функционалдық тәуелділік деп аталады. Егер А атрибуты В атрибутына байланысты болса, ал В атрибуты С (С В А) атрибутына байланысты болса, бірақ кері байланыс жоқ болса, онда А-ға тәуелділік әтпелі деп аталады.

Дерекқордағы деректер кез-келген түрде болуы мүмкін: сандар, пайыздар, мәтін. Қалыпқа келтіру – деректерді ұйымдастыру тәсілі. Қалыпқа келтірілген дерекқорда қайталанатын деректер жоқ, онымен жұмыс істеу оңайырақ және оның құрылымын әртүрлі тапсырмалар үшін өзгертуге болады. Қалыпқа келтіру процесінде деректер аз орын алатындай етіп түрлендіріледі, ал элементтер бойынша іздеу тез және тиімді болады. Ол үшін қосымша кестелер жасалынады және қайталанатын элементтері жоқ баған кілттерімен бір-бірімен байланыстырылады.

Көптеген ресми ережелер мен сипаттамалар сияқты, нақты сценарийлер әрқашан тамаша сәйкестікті қамтамасыз ете бермейді. Әдетте, қалыпқа келтіруді орындау үшін қосымша кестелер жасау керек, ал кейбір клиенттер оны қажет емес деп санайды. Қалыпқа келтірудің негізгі қағидасы

1. Деректер қорының көлемін азайтады және кеңістікті үнемдейді.
2. Іздеуді жеңілдетеді және қормен жұмыс істеуді ыңғайлы етеді.
3. Қателер мен ауытқулардың ықтималдығын азайтады.

Деректер тұтастығына шектеу

Тұтастық шектеулерін арнайы құралдар ретінде анықтауға болады, олардың негізгі мақсаты жарамсыз деректер қорына кіруге рұқсат жоқ (мысалы, деректерді енгізу кезінде пайдаланушылардың қателіктеріне жол бермейді).

Тұтастықтың барлық шектеулерін үш үлкен санатқа бөлуге болады:

1. Бірінші санат – домендік тұтастықты қамтамасыз ету құралдары. Олар дерекқордың тиісті әрісінде рұқсат етілген мәндердің болуына жауап береді. Мысалы, тегі әдетте әріптерден, ал пошта индексі сандардан тұруы керек. Дерекқорларда мұндай тұтастық әдетте мән шарттарымен, бос мәндерге тыйым салумен, триггерлермен және сақталған процедуралармен, сондай-ақ кілттермен қамтамасыз етіледі.

2. Екінші санат – маңызды тұтастық. Мұндағы басты міндет-бір субъект туралы мәліметтер дерекқорға екі рет түспеуі үшін. Бірегейлікті шектеу және негізгі кілт арқылы қамтамасыз етіледі.

3. Үшінші санат – анықтамалық тұтастық, бастапқы және сыртқы кілттер жүйесімен қамтамасыз етіледі. Мысалы, осы құралдар көмегімен бізде мәліметтер қорында жоқ сатып алушыларға тапсырыстар берілмейтініне көз жеткізуге болады.

SQL негіздері

SQL тілі – компьютердің деректер базасында сақталған ақпараттарды таңдауға және өңдеуге арналған құрал. SQL – құрылымдалған сұраныстар тілінің аббревиатурасы (Structured Query Language).

SQL – құрылымдық сұраныс тілі дегенді білдіреді. Бұл кестелерде сақталатын өзара байланысты ақпараттардың жиыны болып табылатын

реляциялық мәліметтер қорын құруға және жұмыс істеуге мүмкіндік беретін тіл.

Ақпараттық кеңістік біртұтас болып келеді. Бұл компьютерлік ортаның көптеген түрлерінде қолдануға болатын стандартты тілді құру қажеттілігіне әкелді. Стандартты тіл командалардың бір жиынтығын білетін пайдаланушыға оларды дербес компьютерде, желілік жұмыс станциясында немесе әмбебап компьютерде жұмыс істейтініне қарамастан, ақпаратты құру, табу және беру үшін пайдалануға мүмкіндік береді.

Оның компьютерлік технологиядан тәуелсіздігі, сондай-ақ реляциялық дерекқор технологиясындағы сала көшбасшыларының қолдауы SQL-ді негізгі стандартты тілге айналдырды.

SQL тілі реляциялық мәліметтер қорындағы деректерді басқаруға, мәліметтер қорының құрылымын анықтауға және көп қолданушы ортасындағы деректерге қол жеткізу құқықтарын басқаруға арналған.

Сондықтан SQL тіліне құрамдас бөліктер ретінде мыналар кіреді:

1. Деректерді манипуляциялау тілі (Data Manipulation Language, DML).
2. Деректерді анықтау тілі (Data Definition Language, DDL).
3. Деректерді басқару тілі (Data Control Language, DCL).

Пайдаланушыларға, бағдарламаларға және компьютерлік жүйелерге арналған реляциялық деректер қорында қамтылған ақпаратқа қол жеткізу SQL (Structured Query Language) арқылы қамтамасыз етіледі.

Объектіге бағытталған модельде мәліметтерді ұсыну кезінде мәліметтер қорының жеке жазбаларын анықтауға болады. Мәліметтер қорының жазбалары мен оларды өңдеу функциялары арасында объектіге бағытталған бағдарламалау тілдеріндегіге ұқсас механизмдер арқылы байланыс орнатылады.

Артықшықтары:

- Арнайы ДҚБЖ-дан тәуелсіздік – барлық жалпы ДҚБЖ-лар SQL тілін пайдаланады.

- SQL көмегімен жасалған қолданбаларды жергілікті деректер қорлары үшін де, клиент-сервер жүйелері үшін де пайдалануға болады.

- SQL операторлары интерактивті және бағдарламалық қатынас үшін де пайдаланылады, сондықтан деректер қорын қамтитын бағдарламалардың бөліктері алдымен интерактивті түрде тексеріліп, содан кейін бағдарламаға енгізілуі мүмкін.

SQL мүмкіндіктері

- Құрылымының толық сипаттамасымен мәліметтер қорын және кестені құру;
- Кестелерден деректерді енгізу, өзгерту және жою сияқты негізгі деректерді өңдеу әрекеттерін орындау;
- Деректерді түрлендіретін қарапайым және күрделі сұрауларды іске

қосу.

Деректер қорын жобалау және құру

Жобалау процесі ұзақ және көп уақытты қажет етеді және әдетте бірнеше айға созылады.

Жасалған деректер қорының тиімділігінің төмендігінің негізгі себептері мыналар болуы мүмкін:

1. Талаптардың терең талдауының жеткіліксіздігі (жобалаудың бастапқы кезеңдері), оның ішінде деректер қатынастары семантикасы.

2. Құрылымдау процесінің ұзақ болуы, бұл процесті жалықтырады және қолмен өңдеу кезінде орындау қиынға соғады.

Мұндай жағдайларда автоматтандыру мәселелері бірінші орынға шығады.

Мәліметтерді жобалаудың негізгі бірнеше кезеңдері бар:

1. *Тапсырмаларды нақтылау.* Бірінші кезеңде, негізінен, осы қолданба арқылы шешуге болатын барлық негізгі міндеттердің тізімі құрастырылады, соның ішінде бүгінгі күні қажет емес, бірақ болашақта пайда болуы мүмкін.

2. *Тапсырмалар тізбегі.* Қолданбаның логикалық және ыңғайлы жұмыс істеуі үшін негізгі тапсырмаларды тақырыптық топтарға топтастыру, содан кейін әр топтың тапсырмаларын олардың орындалу ретімен орналасатындай етіп орналастырған дұрыс. Кейбір тапсырмалар әртүрлі топтармен байланысты болады немесе қандай да бір тапсырманың орындалуы басқа топқа жататын екіншісінің орындалуынан бұрын болуы керек.

3. *Мәліметтерді талдау.* Тапсырмалар тізімін жасағаннан кейін ең маңызды қадам - әрбір тапсырманы шешуге қажетті барлық деректердің егжей-тегжейлі тізімін жасау. Кейбір деректер бастапқы деректер ретінде қажет болады және өзгермейді. Тапсырма орындалу барысында басқа деректер тексеріліп, өзгертіледі. Кейбір деректер элементтері жойылуы немесе қосылуы мүмкін. Ақырында, кейбір деректер есептеулер арқылы алынады: олардың шығысы тапсырманың бір бөлігі болады, бірақ олар деректер қорына енгізілмейді.

4. *Деректер құрылымын анықтау.* Барлық қажетті деректер элементтерін алдын-ала талдағаннан кейін оларды объектілер бойынша ұйымдастырып, объектілерді кестелермен және мәліметтер базасының сұраныстарымен байланыстыру керек. Access типті реляциялық мәліметтер базасы деректерді сақтаудың ең тиімді және икемді жолын жасау үшін қалыпқа келтіру деп аталатын процесті пайдаланады.

5. *Қолданба макетін және пайдаланушы интерфейсін әзірлеу.* Қолданба кестелерінің құрылымын көрсеткеннен кейін, Microsoft Access бағдарламасында пішіндер арқылы оның макетін құру және оларды қарапайым макростар немесе оқиғаларды әндеу процедуралары арқылы байланыстыру оңай. Алдын ала жұмыс схемасын тұтынушыға оңай көрсетуге болады және қолданба тапсырмаларын егжей-тегжейлі орындағанға дейін оның мақұлдауын алуға болады.

6. Қосымшаны құру.

Өте қарапайым жағдайда жасалған макет іс жүзінде аяқталған қосымша болып табылады. Дегенмен, жобаға жоспарланған барлық тапсырмаларды толығымен автоматтандыруға мүмкіндік беретін процедураларды жазу өте жиі кездеседі. Сондықтан сізге бір тапсырмадан екіншісіне өтуді қамтамасыз ететін арнайы байланыстырушы формалар жасау қажет болады.

7. *Тестілеу және жетілдіру.* Қосымшаның жекелеген компоненттері бойынша жұмыстар аяқталғаннан кейін, мүмкін болатын режимдердің әрқайсысында қолданбаның жұмысын тексеру керек. Макростардың жұмысын тексеру керек, ол үшін белгілі бір макро команда орындалатын қадамдық жәндеу режимін пайдалану керек.

Қолданбаларға арналған Visual Basic қолданбасын пайдаланған кезде дизайнердің қолында қолданбаның жұмысын тексеруге, қателерді анықтауға және түзетуге мүмкіндік беретін әртүрлі жәндеу құралдары бар.

Қосымшаның дербес бәлімдері әзірленгендіктен, олардың жұмыс істеуін тексеру және белгілі бір өзгерістер енгізу қажеттілігі туралы пікір алу үшін оларды тапсырыс берушіге берген дұрыс.

Тапсырыс беруші қосымшаның жұмысымен танысқаннан кейін, жобаны алдын ала зерделеу қаншалықты мұқият болғанына қарамастан, оның әрқашанда жақсарту бойынша қосымша ұсыныстары болады.

Пайдаланушылар тапсырманы орнату процесінде өте маңызды және қажет деп айтқан кейбір нәрселердің қолданбаны іс жүзінде пайдалануда маңызды рөл атқармайтынын жиі байқайды. Қолданбаны әзірлеудің бастапқы кезеңдерінде қажетті өзгерістерді анықтау кейінгі қайта өңдеу уақытын айтарлықтай қысқартуы мүмкін.

Жобалаудың жеті кезеңін тереңірек ойластырған сайын, жүйені оңай жүзеге асыруға болады. Жобалау – үлкен көлемді құжаттар құру емес, оның негізгі мақсаты - қажетті қосымшаны іске асырудың айқын жолын анықтау.

1-кезең. Концепциядан бастап іске асуға дейінгі жалпы жобалау.

2-кезең. Базадағы есепті жобалау: өрістерді орналастыру.

3-кезең. Деректердің құрылымын жобалау кестенің құрылымында қандай өрістер болуы қажет.

4-кезең. Кестелерді жобалау және біріктіру.

5-кезең. Өрістерді жобалау деректерді енгізу және олардың мүмкін мәндерін тексеру ережелері.

6-кезең. Үлгілерді жобалау енгізу.

7-кезең. Автоматтандыру тәсілдерін жобалау.

Бақылау сұрақтары:

1. Деректер қоры дегеніміз не?

2. ДҚБЖ дегеніміз не?
3. ДҚБЖ жіктелуі?
4. Деректер моделін тізімдеңіз?
5. SQL тілі. SQL сұрауларының мысалдары?
6. Реляциялық мәліметтер базасын жобалау мәселелері?
7. ДҚ құру принциптері. Қалыпқа келтіру дегеніміз не?
8. Қалыпты формалар?
9. ДҚ жобалау кезеңдері?
10. ORM бағдарламалау технологиясын сипаттаңыз?