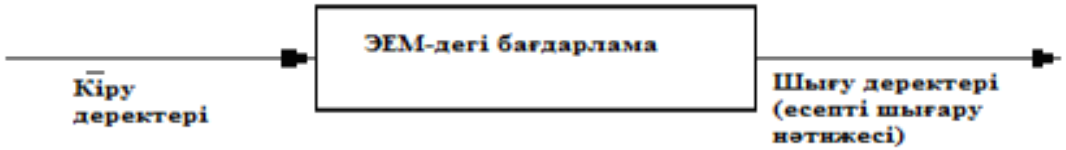
МӘЛІМЕТТЕР БАЗАСЫНЫҢ АРХИТЕКТУРАСЫ

2-дәріс

- Кез келген есептетуіш үрдіс келген деректердің шығу деректеріне
- бейнеленуін (белгілі бір алгоритм бойынша) білдіреді.



- 1.1 сурет – Алгоритм
- Өңделетін деректер мен есептеу алгоритмдерін ұсыну күрделілігінің
- қатынасы есептердің екі класын анықтайды:
- 1) Есептеуіш есептер – есептеу дерекетерін ұсынудың жеткілікті
- қарапайым және күрделі көп операциялық үрдісі.
- 2) Деректерді өңдеудің есептері (есептеуіш емес есептер) – деректерді
- өңдеудің қарапайым алгоритмі және өңделетін деректердің күрделі ұсынылуы.

- Өңделетін деректер мен есептеу алгоритмдерін ұсыну күрделілігінің қатынасы есептердің екі класын анықтайды:
- 1) Есептеуіш есептер – есептеу дерекетерін ұсынудың жеткілікті қарапайым және күрделі көп операциялық үрдісі.
- 2) Деректерді өңдеудің есептері (есептеуіш емес есептер) – деректерді өңдеудің қарапайым алгоритмі және өңделетін деректердің күрделі ұсынылуы.
- Бағдарламалауды үйренудің бастапқы сатысында есептерді шешу алгоритмін жасақтауға негізгі көңіл бөлінеді. Алайда, нақты есепті шешу мүмкіндігі (немесе мүмкінсіздігі) таңдалған алгоритмге ғана байланысты емес, сонымен қатар өңделетін деректерді ұсыну үшін қандай түсініктер
- қолданылатынына да байланысты.
- Келесі формула бойынша есептелетін қарапайым мысалды қарастырайық:
- Y=X2+5X,
- мұндағы X және Y – анықталған сандар, олар мұнда деректердің
- қарапайым бірлігі (деректер элементтері) болып табылады.

- Осы есепті шешу алгоритмін бағдарламалау кезінде (формуланы бағдарламалау) деректердің қарапайым түрі – қарапайым айнымалылар (X және Y бағдарламада қарапайым айнымалылар ретінде ұсынылады) қолданылады.
- Бағдарламалау жүйесінде қарапайым айнымалы, бағдарламалау кезінде таңдалуға тиісті, оның мәндерінің белгілі бір типімен сипатталады. Тіпті бұл қарапайым жағдайда да айнымалының типін дұрыс таңдау қажет, себебі қолданбалы есептің нақты шешімінің мүмкіндігі немесе мүмкінсіздігі осы таңдауға байланысты (мысалы, нақты деректерді ұсыну үшін берілген
- разрядтар жетпейді).
- Басқа мысалды қарастырайық:
- S=a1+a2+...+aN.

- Бұл есептің шешімін, жалпы жағдайда тек қана қарапайым айнымалыларды қолдана отырып, алу мүмкін емес. Мұнда жай сан емес, ал сандар тізбегі өңделеді. Бұл жағдайда бағдарламалау кезінде, жиын (массив) секілді деректер түрі қолданылады. Жиын (массив) – элементтер жиынтығы, элементтердің әрқайсысы, индекс деп аталатын, реттелген бүтін сандар жиынымен байланысқан. Барлық элементтердің мәндерінің типі бірдей болуы қажет, ол жиын типі де болады. Бұл жағдайда a1, a2,..., aN сандары бағдарламада A(1), A(2),..., A(N) жиынымен беріледі.
- Қарастырылған мысалдар есеп түрінің өзгеруі басқа деректер түрлерін қолдану қажеттілігін міндеттейді.
- Бағдарламалаудың бұрынғы тілдері (ФОРТРАН, АЛГОЛ-60) ғылыми техникалық есептеуіш есептерді шешу үшін арналған. Бұл тілдерде тек жоғарыда көрсетілген деректер түрлері (қарапайым айнымалылар және жиындар) қолданылған.
- 60-шы жылдардың соңынан бастап, әртүрлі түрдегі құжаттарды өңдеумен байланысты, есептелмейтін деп аталатын есептерді шешуі үшін компьютерлерді қарқынды қолдана бастайды. Деректердің жаңа түрлерінің пайда болуын деректерді өңдеудің жеңілдетілген есептер мысалында қарастырайық.

Тапсырма 1. Жалақыны есептеу.

Есепті екі жеңілдетілген болжаммен қарастырайық:

1) Қызметкерге жалақы оның айлығы (оклад) негізінде есептеледі.
2) Бірде-бір салықтар мен ұстап қалулар ескерілмейді.

Осы есепті шешуге арналған қызметкерлер бойынша мәліметтер келесі «ЕСЕПТЕП ЖАЗУ» карточкасында берілген:

Тегі, аты-жөні, әкесінің аты	Айлық	Бір айдағы жұмыс жасаған күндер саны	Есептелген сомасы
FIO	O	K_o	S

Әр қызметкер үшін белгілі-бір айға есептелген сома келесі формула бойынша есептеледі:

$$S=K_oO/K_r,$$

мұндағы K_r – берілген айдағы жұмыс күндерінің саны.

Әр қызметкер үшін сәйкес деректер нақты мәнге ие, мысалы:

Иванов Иван Иванович	180000	24	180000

- Бұл мәндердің мәні бір бірімен байланыс кезінде ғана бар. Жеке таңдалған 180000 саны өзінің мағыналық мәнін жоғалтатындықтан, мұнда оны
- қарапайым айнымалы деректер түрінде қолдануға болмайды. Сонымен бірге,нақты қызметкерді сипаттайтын, сәйкес мәндер жиыны әртүрлі типте
- (символдық және сандық) болады, яғни оларды ұсыну үшін жиын сияқты
- деректер түрінде қолдануға болмайды. Осылайша, «қарапайым айнымалы»
- және «жиын» түсініктері, сәйкес карточканы ұсыну үшін, жеткіліксіз.
- Есептелмейтін есептердің пәндік аймақтарында осындай деректерді
- ұсынуды бейнелеу үшін жаңа түсініктер реті енгізілді [1.1].

- Деректер элементі (өріс) – атаулы деректердің ең кіші бірлігі.
- Берілген мысал үшін деректер элементі FIO, O, Ko, S болып табылады.
- Қызметкер карточкасын сипаттау үшін «Логикалық жазба» түсінігі
- қолданылады.
- Логикалық жазба – деректер элементтерінің (өрістер) атаулы жиынтығы.
- Логикалық жазба данасы (экземпляр) – жазба элементтерінің ағымдағы мәні.
- Қызметкерлер карточкаларының барлық жиынын ұсыну үшін
- «Логикалық файл» түсінігі қолданылады.
- Логикалық файл – берілген типтегі барлық жазба даналарының атаулы
- жиынтығы.
- «ТӨЛЕМ» (НАЧИСЛЕНИЕ) логикалық файлының мысалы:

Иванов Иван Иванович	180000	24	180000
Абаев Абай Абаевич	220000	20	183000
----------	----------	----------	----------
Акпаева Лиза Маратовна	250000	24	250000

Осылайша, енгізілген түсініктердің көмегімен сәйкес деректерді бейнелеуге болады. Заманауи бағдарламалау тілдерінде, есептеуіш есептерге және де деректерді өңдейтін есептерге арналған, осы түсініктерді бейнелеу үшін жаңа деректер түрі енгізілді.

Паскаль алгоритмдік тілінде, жазба (RECORD) – әртүрлі типте бола алатын, бірнеше компоненттері бар күрделі айнымалылар секілді, деректер түрлері енгізіледі. Сонымен қатар, жазба (өріс) компоненттеріне рұқсат индекс бойынша емес, ал атау бойынша жүзеге асады. Тапсырма 1 Паскаль бағдарламалау тілінде бағдарламалау кезінде «ТӨЛЕМ» (НАЧИСЛЕНИЕ) логикалық жазба (логикалық файл) «RECORD» деректер түрімен ұсынылады, қызметкерлердің логикалық жазба даналарының жиыны, Паскаль тілінің құралдарымен және операциялық жүйемен құрастырылатын, «физикалық» файлмен ұсынылады.

- Salary = RECORD
- FIO: string;
- O: real;
- Ko: real;
- S: real;
- END;
- Мұндай есептеуіш емес есептердің маңызды ерекшелігін атап өтейік.
- Мұндай есептерге үлкен көлемді деректер (қызметкерлердің үлкен саны,
- өндірілетін өнімнің үлкен саны т.с.с.) сипаты тән. Көрсетілген деректер,
- ережеге сәйкес, есептерді шешу үшін бірнеше рет қолданылады (еңбек ақы
- үнемі әр ай сайын есептеледі), сондықтан деректер ЭЕМ жадында жеткілікті
- ұзақ сақталуы қажет. Ұзақ уақыт сақтау үшін үнемі сыртқы жады қолданылады.

- Осыған байланысты тапсырма 1 екі сатыдан тұрады.
- 1. бастапқы деректерді енгізу және оларды сыртқы жадыға кіргізу.
- type
- Salary = RECORD
- FIO: string;
- O: real;
- Ko: real;
- S: real;
- END;
- FSalary = File of Salary;
- var
- F: FSalary;
- ...
- { Бастапқы деректерді енгізу }
- repeat
- write('Қызметкерлер санын енгізіңіз (көп емес',
- MaxN,'): ');
- readln(N);
- until (N>0) AND (N<=MaxN);
- For I := 1 to N do

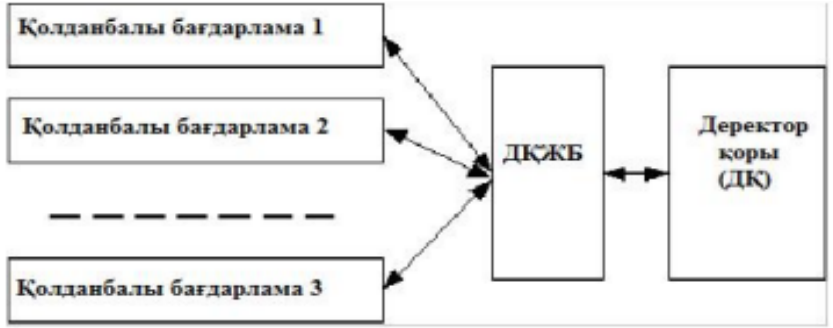
```
Begin
 Write(I,'нөмірлі қызметкер тегін енгізіңіз: ');
 ReadLn(Sotr[i].FIO);
 Write(I,'нөмірлі қызметкердің еңбек ақысын енгізіңіз: ');
 ReadLn(Sotr[i].O);
Write(I,'нөмірлі қызметкердің жұмыс жасаған күндер санын енгізіңіз:');
 ReadLn(Sotr[i].Ko);
End;
{ Деректерді сыртқы жадыға кіргізу }
Assign(F, 'MyFile.fsf');
Rewrite(F);
For I := 1 to N do
 Write(F, Sotr[i]);
Close(F);
...
```

- 2. Сыртқы жадыдан бастапқы деректерді оқу, есептелген соманың есебі
- және оларды баспаға шығару.
- ...
- { Сыртқы жадыдан мәліметтерді оқу }
- Assign(F, 'MyFile.fsf');
- Reset(F);
- For I := 1 to N do
- Read(F, Sotr[i]);
- Close(F);
- { есептелген сома есебі және баспаға шығару }
- For I := 1 to N do
- Begin
- Sotr[i].S := Sotr[i].O * Sotr[i].Ko / Kr;
- WriteLn(Sotr[i].FIO, ': ', Sotr[i].S);
- End;
- ...

- Ұсынылған бағдарламалар қойылған есепті жасалған болжамдар
- бойынша шешеді. Бұған қажетті деректер, тек осы есепті шешу үшін арналған, MyFile.fsf файлында сақталады. Атап өтейік, бұл жағдайда деректерді сипаттау қолданбалы бағдарламаға қосылған. Жазба файлының форматын өзгерту кезінде қолданбалы бағдарламаны да өзгерту қажет. Осылайша, қойылған есепті шешетін, бағдарламалық жүйе өзінің өзіне меншікті деректерін анықтайды және оларды басқарады. Мұндай бағдарламалық жүйелер файлдық жүйелер деп аталады

- Есептерді шешу кезінде бір немесе бірнеше жеке файлдарды қолданатын, қолданбалы бағдарламада деректердің сақталуына және нақтылығына, осы есеппен жұмыс жасайтын бағдарламалаушы жауап береді. Деректер қорын қолдану әртүрлі қолданушылардың есебін шешетін, бірнеше қолданбалы бағдарламалардың онымен жұмысын талап етеді.
- Демек, қолданбалы есептің біреуін шешетін, бағдарламалаушы деректердің сақталуына және интегралданған деректердің нақтылығына енді жауап бере алмайды. Сонымен қатар, деректер қорын қолданған шешілетін есептер шеңберінің кеңеюі жазбалардың жаңа типінің және олардың арасындағы қатынастың пайда болуына алып келуі мүмкін. Деректер қоры құрылымының мұндай өзгерісі, деректер қорымен жұмыс жасайтын, бұрын жасалған және сәтті жұмыс жасап тұрған қолданбалы бағдарламалар жүйесінің өзгерісіне алып келмеуі қажет. Басқа жағынан қарағанда, кез келген қолданбалы бағдарламалардың мүмкін өзгерістері, өз кезегінде деректер құрылымының өзгерісіне алып келмеуі қажет. Жоғарыда айтылғандардың барлығы деректерді қолданбалы бағдарламалардан бөлу қажет екенін көрсетеді.

- Олардың тәуелсіздігін қамтамасыз ететін, қолданбалы бағдарламалар мен деректер қорының арасындағы интерфейстің ролін бағдарламалық кешен - деректер қорын басқару жүйесі (ДҚБЖ) ойнайды (2.1 сурет).
- ДҚБЖ – көптеген қолданушылармен (қолданбалы бағдарламалармен) дерекетер қорын құру, енгізу және қолдану үшін арналған, дерекетрдің интегралданған жиынын қолдауының бағдарламалық кешені.



2.1 сурет – Қолданбалы бағдарламалар мен деректер қорының тәуелсіздігін қамтамасыз ету

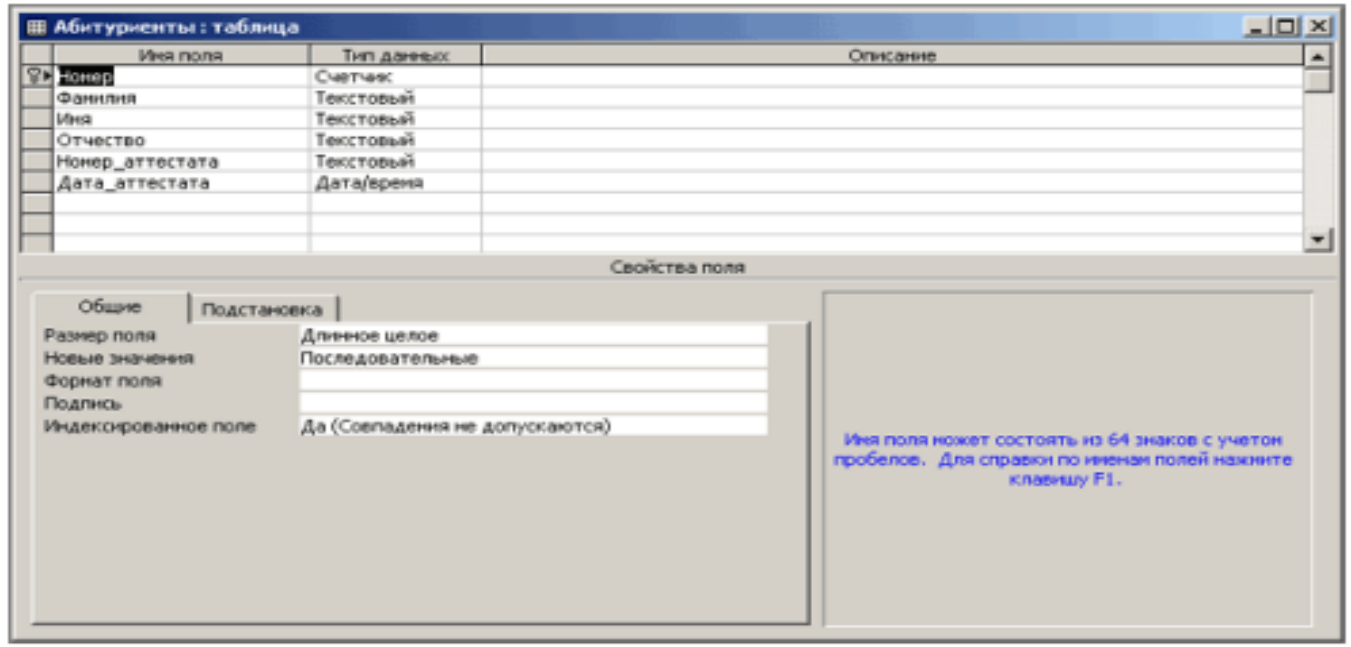
Тағы да бір ұғымды анықтайық.

Деректер банкі – дерекетрдің интегралданған жиынын қолдауының, тілдік, алгоритмдік, бағдарламалық, техникалық және ұйымдастырушылық құралдары, сонымен қатар деректер қоры ретінде ұсынылған, осы деректердің өзі.

Деректер қорын басқару жүйесінің негізгі функцияларын атап өтейік:

1) Құрылатын деректер қорының құрылымының анықтамасы, оның инициализациясы және бастапқы жүктемені жүргізу.

Ережеге сәйкес, деректер қорының құрылымын құру сұхбат режимінде жүреді. ДҚБЖ қолданеушыдан қажетті деректерді жүйелі түрде сұрайды. Көпшілік заманауи ДҚБЖ деректер қоры кестелердің жиынтығы түрінде беріледі. Қарастырылатын функция жадыда кестенің құрылымын сипаттауға және құруға, кестеде деректердің бастапқы жүктемесін жүргізуге мүмкіндік береді. MS Access ДҚБЖ үшін мұндай әрекеттердің мысалы 2.2 сурете келтірілген.

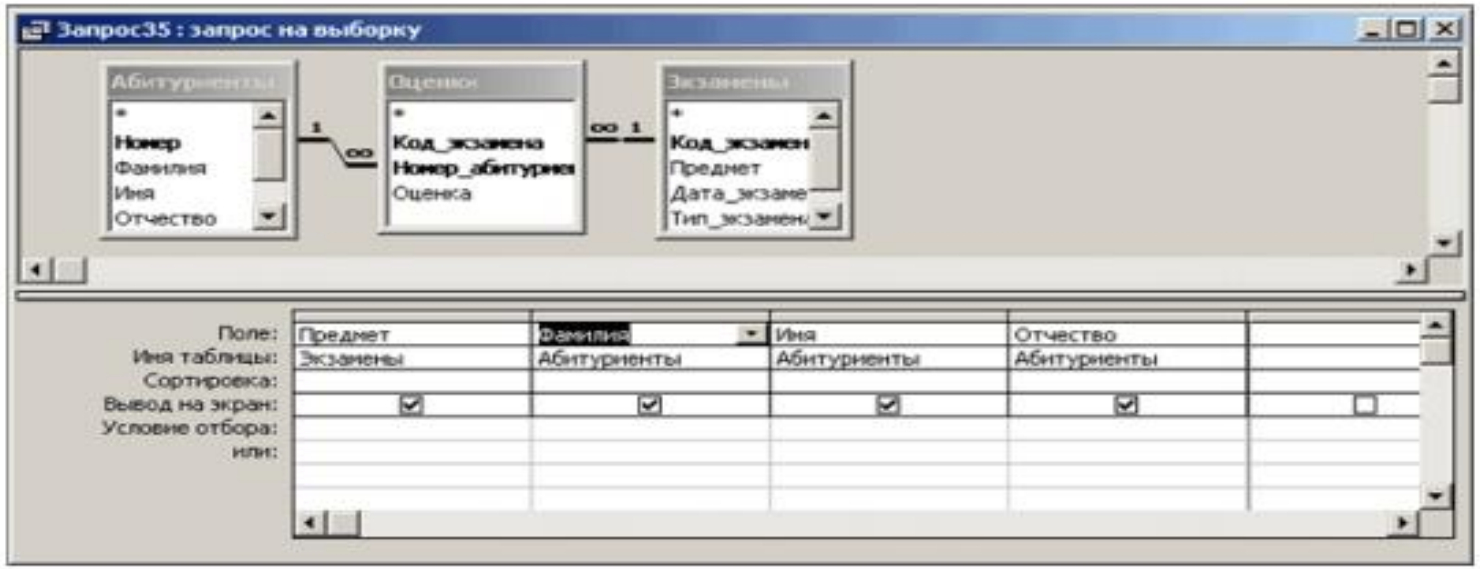


2.2 сурет – ДҚБЖ MS Access деректер қоры құрылымының құрылуы

2) Қолданушыларға дерекетрмен күрделі әрекеттер жасау мүмкіндігі берілу (қажетті деректерді іріктеу, есептеуді орындау, енгізу\шығару интерфейсін жасақтау, визуализациялау).

ДҚБЖ мұндай мүмкіндіктер, ДҚБЖ құрамына кіретін, арнайы бағдарламалау тілін қолдану негізінде, немесе графикалық интерфейс көмегімен беріледі.

MS Access-те берілген функцияны жүзеге асыру сұраныстарды құру және графикалық интерфейстің көмегімен енгізу формасын құру арқылы жүзеге асырылады (2.3 сурет).



2.3 сурет – ДҚБЖ Access-те іріктеу үшін сұраныстарды құру

Клиент-серверлік ДҚБЖ арналған, сұраныстарды орындауға мүмкіндік беретін, құралдар және қолданушының интерфейсін құруға мүмкіндік беретін, бағдарламалық құралдар бар.

3) Қолданбалы бағдарламалар және деректердің тәуелсіздігін қамтамасыз ету (логикалық және физикалық тәуелсіздік).

- ДҚБЖ маңызды қасиеті деректер қорына екі тәуелсіз көзқарасты – деректерді логикалық ұсынуды іске асыратын, және қолданбалы бағдарламалардағы оның бейнесі, «қолданушы көзқарасын»; және «жүйе көзқарасын» - ЭЕМ жадысында деректерді физикалық ұсыну қолдау мүмкіндігі болып табылады. Деректердің логикалық тәуелсіздігін қамтамасыз ету деректерді сақтаудың логикалық құрылымын өзгертуді қажет етпей деректер қорын логикалық ұсынуын өзгерту (белгілі шектерде) мүмкіндігін береді.
- Осылайша, қолданбалы бағдарламада деректердің логикалық ұсынуын өзгерту деректерді сақтау құрылымының өзгеруіне алып келмейді. Деректердің физикалық тәуелсіздігін қамтамасыз ету деректердің «логикалық» ұсынуын өзгерту қажеттілігін шақырмай-ақ ЭЕМ жадында деректер қорын ұйымдастыру тәсілдерін өзгерту (белгілі шектерде) мүмкіндігін береді. Осылайша, деректер қорының ұйымдастыру тәсілдерін өзгерту қолданбалы бағдарламалардың өзгеруіне алып келмейді.

- 4) Деректер қорының логикалық толықтығын қорғау.
- Осы функцияны жүзеге асырудың негізгі мақсаты деректер қорында деректердің нақтылығын көтеру болып табылады. Деректердің нақтылығы оларды ДҚ енгізу кезінде немесе теріс деректерді ДҚ алатын және енгізетін, деректерді өңдеу үрдістерінің заңсыз әрекеттері кезінде бұзылуы мүмкін.
- Жүйеде деректердің нақтылығын көтеру үшін, белгілі бір жағдайда дұрыс немесе деректерді «аулап алатын», толықтық шектеулері жарияланады.
- Осылай, барлық заманауи ДҚБЖ, құрылымды құру кезінде сипатталған, енгізілетін деректердің олардың типіне сәйкестігі тексеріледі. Жүйе сандық типтегі өріске символды, мүмкін емес күнді және т.с.с. енгізуге жол ермейді.
- Дамыған жүйелерде толықтық шектеулерін бағдарламалаушы, есептің мазмұндық мәніне қарай, сипаттайды және оларды тексеру деректерді әр өңдеу кезінде жүзеге асырылады. Деректер қорының логикалық толықтығының түрлі көріністері нақтырақ келесі бөлімдерде қарастырылатын болады.

- 5) Физикалық толықтықты қорғау.
- ЭЕМ жұмысы кезінде жұмыста іркілулер (мысалы, электр қуатын айырудың әсерінен), деректерді машиналық тасушылардың бұзылуы болуы мүмкін. Сонымен бірге деректердің арасындағы байланыста бұзылуы мүмкін.
- Бұл болашақтағы жұмыстың мүмкінсіздігіне алып келеді. Дамыған ДҚБЖ
- деректер қорын қалпына келтіру құралдары бар. Қолданылатын маңызды ұғым «транзакция» ұғымы болып табылады. Транзакция – бұл, деректер қорымен өндірілетін, әрекеттер бірлігі. Транзакция құрамына деректер қорын өзгертетін бірнеше операторлар кіруі мүмкін, бірақ осы операторлардың барлығы орындалады, немесе бірде біреуі орындалмайды. ДҚБЖ, шынында деректер қорын енгізуден басқа, сонымен қатар транзакция журналын да жүргізеді.

- Деректер қорында транзакцияны қолдану қажеттілігін келесі қарапайым мысалда көрсетейік. Деректер қоры кейбір банкте қолданылады және клиенттердің бреуі басқа клиенттің шотына ақша аударғысы келеді деп болжайық. Деректер қорында әр клиенттің ақшасының саны туралы ақпараттар сақталған. Бізге деректер қорында екі өзгерту жүргізу керек – клиенттердің біреуінің шотындағы ақша сомасын азайтуымыз қажет, ал басқа шоттағы ақша сомасын көбейтуіміз қажет. Әрине, нақты өмірде банкте ақша аударымы, көптеген кестелерді, ал тіпті, көптеген деректер қорын қозғайтын, өте күрделі үрдіс болып табылады. Алайда, мәнісі бірдей – барлық әрекетті орындауымыз қажет (бір клиенттің шотын өсіруіміз, ал екінші клиенттің шотын азайтуымыз қажет), немесе осы әрекеттердің бірде біреуін орындамаймыз. Бір шоттағы ақша сомасын азайтып, ал екінші шоттағы ақша сомасын өсірмеуге болмайды.
- Сонымен бірге, бірінші әрекеттің біреуін орындағаннан кейін (бірінші клиенттің шотындағы ақша сомасын азайту) іркілу болды деп болжайық.
- Мысалы, клиенттік компьютер мен деректер қорының арасындағы байланыс үзілуі мүмкін немесе операциялық жүйенің қайта жүктелуіне алып келетін, клиенттік компьютерде жүйелік іркілу (сбой) болуы мүмкін. Бұл жағдайда деректер қорымен не болады? Бірінші клиенттің шотындағы ақшаны азайту бұйрығы орындалды, ал екінші бұйрық – басқа шоттағы ақшаны өсіру – жоқ, деректер қорының қарама-қайшы, ескірген жағдайына не алып келуі мүмкін.

- Транзакция механизмін қолдану осындай жағдайлардың шешімін табуға мүмкіндік береді. Бірінші әрекетті орындаудың алдында транзакцияның басы бұйрығы шығады. Транзакцияға бір шоттан ақшаны шешу және басқа шоттағы ақшаны өсіру операциясы қосылады. Транзакцияның аяқталу операторы әдетте COMMIT деп аталады. Бірінші әрекет орындалғаннан кейін транзакция аяқталмағандықтан, өзгерістер деректер қорына кіргізілмейді. Өзгерістер тек транзакция аяқталғаннан кейін ғана кіргізіледі (бекітіледі). Осы оператор шықпайынша қорда деректердің сақталуы жүрмейді.
- Біздің мысалымызда, транзакцияны бекіту операторы шықпағандықтан, деректер қоры бастапқы күйіне «кері шегінеді» - басқаша сөзбен айтқанда, клиенттердің шотындағы сома транзакция басынынң алдындағы күйіндей қалады. Деректер қорының әкімшілігі транзакция жағдайын қадағалай алады және қажет болған жағдайда транзакцияны қолмен «кері егіндіреді».
- Сонымен қатар, айқын жағдайда ДҚБЖ транзакцияның «кері шегінуі» туралы шешімді өзбетімен қабылдайды.

- Транзакцияның қысқа болуы міндетті емес. Бірнеше сағатқа немесе бірнеше күнге созылатында транзакциялар болады. Бір транзакцияның шегінде әрекеттер санының өсуі орын алатын жүйелік ресурстардың өсуін талап етеді.
- Сондықтан мүмкіндігінше транзакцияны қысқа істеген дұрыс. Транзакция журналында барлық транзакциялар – бекітілгенде, «кері шегінумен» аяқталғанда транзакциялар енгізіледі. Сақтық қорда сақталған көшірмелерді құрумен бірге жүргізілген транзакция журналы деректер қорының жоғарғы сенімділігіне жетуге мүмкіндік береді.
- ДҚБЖ сервері қондырылған, компьютерде аппараттық іркілудің нәтижесінде деректер қоры зақымданды деп болжайық. Бұл жағдайда соңғы жасалған деректер қорының сақтық қорға сақталған көшірменсін және транзакция журналын қолдану қажет. Сонымен бірге деректер қорына, сақтық қорған сақталған көшірмесі құрылғаннан кейін бекітілген, тарнзакцияларды ғана қолдану қажет.
- Көптеген заманауи ДҚБЖ әкімшілікке, сақтық қорға сақталған көшірмесін және транзакция журналын пайдалана отырып, деректер қорын жаңадан жасауға мүмкіндік береді. Мұндай жүйелерде белгілі бір уақыт мезетінде ДҚ сақтық қордағы жинақтауыштарда көшірмесі алынады. ДҚ барлық жүгінулер өзгертулер журналына бағдарламалық түрде жазылып отырады. Егер деректер қоры бұзылған болса, онда қалпына келтіру процедурасы жүктеледі, ол үрдіс кезінде сақтық қордағы көшірмеге өзгертулер журналынан барлық жүргізілген өзгертулер енгізіледі.

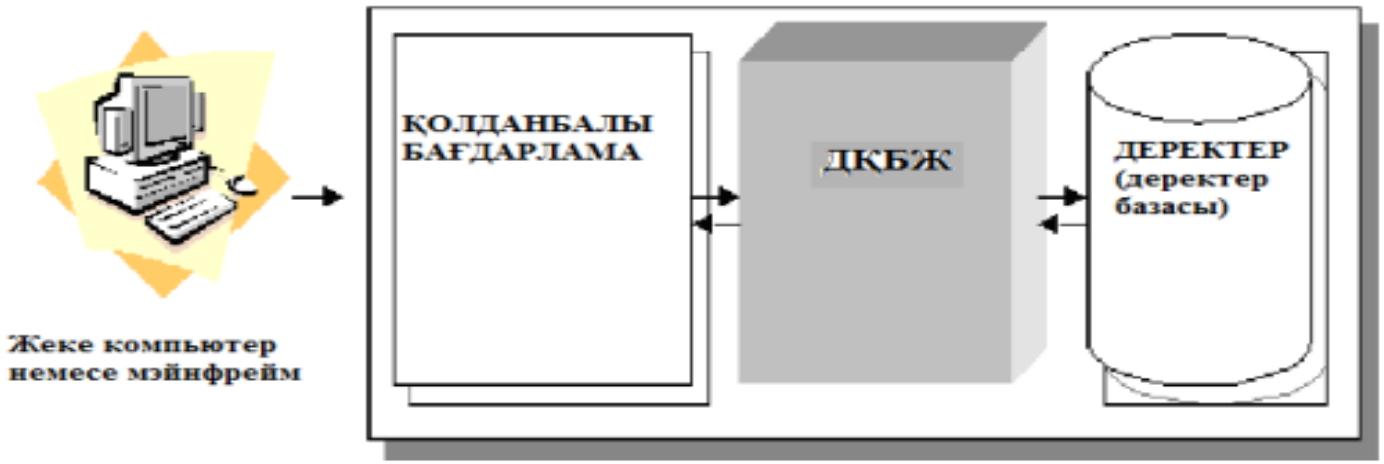
- 6) Деректер қорына рұқсатқа қолданушылардың өкілдеттігін басқару.
- Әртүрлі қолданушылар деректермен жұмыс бойынша әртүрлі өкілеттікке ие (кейбір деректер қолжетімсіз болуы мүмкін; белгілі-бір қолданушыларға деректерді жаңартуға рұқсат берілмейді және т.с.с.). ДҚБЖ, құпия сөз принциптеріне, немесе өкілеттіктердің сипаттамасына негізделген, рұқсат өкілеттігін шектеу механизмі қарастырылады.
- 7) Бірнеше қолданушылар жұмысын синхронизациялау.
- Бірнеше қолданушылар бір уақыт мезгілінде бірдей деректерді жаңарту операциясын орындауы мүмкін жағдай жеткілікті жиі орын алады. Мұндай қайшылықтар деректердің логикалық толықтығының бұзылуына алып келеді, сондықтан жүйе осы деректермен қолданушы жұмысын толық аяқтағанша басқа қолданушылардың осы деректі жаңартуына жол бермейтін, шараларды қарастыруы қажет. Мұнда қолданылатын негізгі ұғым «оқшаулау» болып табылады. Оқшаулау түрлі қолданушыларға деректер қорымен бір уақыт мезгілінде жұмыс жасауға тыйым салу үшін қажет. Себебі ол көптеген қателіктерге алып келеді.

- Бұл тыйымды жүзеге асыру үшін ДҚБЖ, транзакция қолданатын, нысандарға оқшаулау орнатады. Оқшаулаудың түрлі типтері бар – кестелік, беттік, жолдық және де басқа, олар бір-бірінен оқшауланған жазбалар санымен ажыратылады. Басқаларынан жиі беттік оқшаулау – бір жолға транзакциясының жүгінуі кезінде тек осы жол оқшауланады, ал қалған жолдар өзгертулерге қол жетімді болып қалады.
- Осылайша, деректер қорына өзгерістерді енгізу үрдісі келесі әрекеттер тізбегінен тұрады: транзакция басы операторы шығады, деректерді өзгерту операторы шығады, ДҚБЖ операторды талдайды және оның орындалуына қажетті, оқшаулауды орындауға тырысады, оқшаулау сәтті болған жағдайда оператор орындалады, содан кейін үрдіс келесі транзакция операторы үшін қайталанады. Транзакция ішіндегі барлық операторлар сәтті орындалғаннан кейін транзакцияны бекіту операторы орындалады. ДҚБЖ транзакциямен жасалған өзгерістерді бекітеді және оқшаулауды алады. Транзакцияның қандайда бір операторлары сәтсіз аяқталған жағдайда «кері шегінеді», деректер алдыңғы мәндеріне ие болады, оқшаулау алынып тасталады.

- 8) Сақтау ортасының ресурстарын басқару.
- ДҚ ЭЕМ-нің сыртқы жадында орналасады. Жұмыс жасау кезінде ДҚ жаңа деректер енгізіледі (жадыдан орын алынады) және деректер жойылады (жады босайды). ДҚБЖ жаңа деректер үшін жады ресурстарын бөледі, босаған жадыларды қайта бөледі, сыртқы жадыға сұраныс кезектерін енгізуді ұйымдастырады және т.с.с.
- 9) Жүйелік қызметкерлердің қызметтерін қолдау.
- Деректер қорын пайдалану кезінде ДҚБЖ параметрлерін өзгерту, рұқсаттың жаңа әдісін таңдау, сақталған деректердің құрылымын өзгерту (белгілі-бір шектерде), сонымен қатар ортақ жүйелік әрекеттер қатарын орындау қажеттігі пайда болуы мүмкін. ДҚБЖ ДҚ қызметін қолдау үшін ДҚ қызмет көрсететін жүйелік қызметкерлерге, ДҚ әкімшілігі деп аталатын, осы және де басқа әрекеттерді орындау мүмкіндігін береді.

- **Көпқолданушылық ДҚБЖ жүзеге асыру кезінде қолданылатын әртүрлі архитектуралық шешімдер**
- Жоғарыда аталып өткендей, деректер қоры ұғымы басынан бірнеше қолданушылары бар көптеген есептерді шешу мүмкіндігі бар деп болжанды.
- Осыған байланысты, заманауи ДҚБЖ маңызды сипаттамасы жұмыстың көпқолданушылық технологиясының бар болуы мүмкін. Осындай технологияларды түрлі уақыт мезетінде әртүрлі жүзеге асыру есептеуіш техникасының негізгі қасиеттерімен де, сонымен қатар бағдарламалық қамтаманың дамуымен де байланысты. Осы технологияларға хронологиялық ретпен қысқаша сипаттама берейік.
- Осы технологияны қолдану кезінде деректер қоры, ДҚБЖ және қолданбалы бағдарлама (қосымша) бір компьютерде (мэйнфреймде немесе дербес компьютерде) орналасады (3.1 сурет). Ұйымдастырудың мұндай тәсілі үшін желінің қолдауы қажет немесе және барлығы автономиялық жұмысқа сай болады.

- Жұмыс келесі түрде құрылған:
- 1)Файлдар жиыны түріндегі деректер қоры компьютердің қатты дискіде орналасады.
- 2) Сол компьютерде ДҚБЖ және ДҚ жұмысқа арналған қосымшалар орнатылған.
- 3) Қолданушы қосымшаны жуктейді. Қосымшамен ұсынылатын қолданушы интерфейсін пайдалана отырып, ол ақпараттарды іріктеу\жаңарту үшін ДҚ жүгінуді бастамшылық етеді.
- 4) ДҚ барлық жүгінулер ДҚБЖ арқылы жүреді, олар ДҚ физикалық құрылымы туралы барлық мәліметтерді өз ішінде қапшықтандырады.
- 5) ДҚБЖ, қолданушылар сұраныстарын орындайды қамтамасыз ете отырып (деректерге жасалатын қажетті операцияларды жүзеге асыра отырып), деректерге жүгінуді бастамшылық етеді.
- 6) Нәтиже ДҚБЖ қосымшаға қайтарады.
- 7) Қолданушы интерфейсін қолдана отырып, қосымша сұраныстардың
- орындалу нәтижесін бейнелейді.

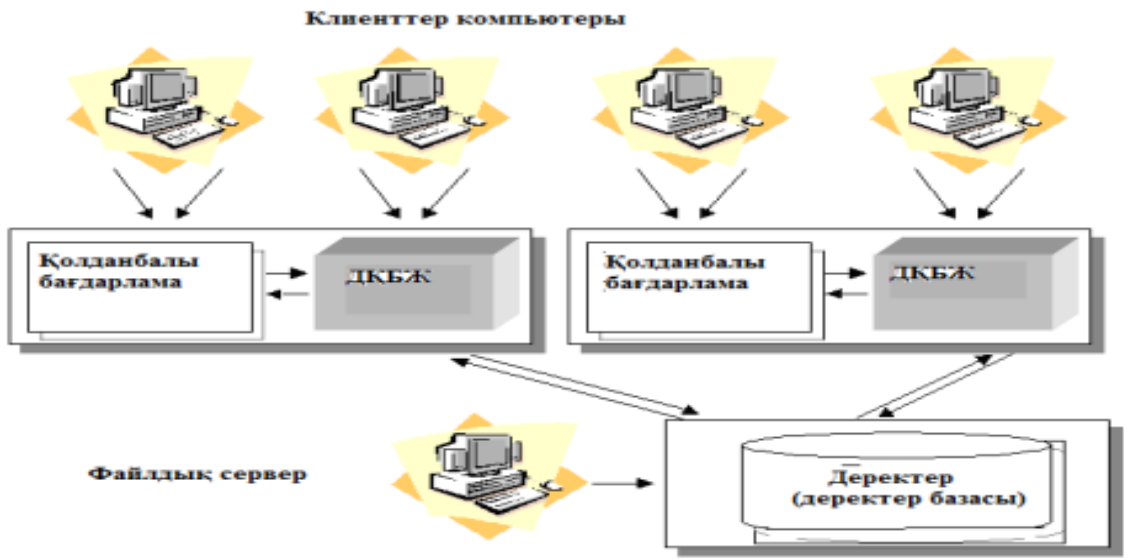


3.1 сурет – Орталықтандырылған архитектура

Мұндай архитектура DB2, Oracle, Ingres ДҚБЖ бірінші нұсқаларында қолданылды [3.1].

Жұмыстың көпқолданушылық технологиясы мультибағдарламалық режиммен (бір уақыт мезетінде процессор және сыртқы құрылғылар жұмыс жасай алады, мысалы, бір қолданушының қолданбалы бағдарламасында сыртқы жадыдан деректерді оқу жүріп жатса, басқа қолданушының бағдарламасы процессормен өңделіп жатады), немесе уақытты бөлу режимімен (қолданушыларға кезекпен олардың бағдарламаларын орындау үшін уақыт кванты бөлінеді) қамтамасыз етіледі. Мұндай технологиялар үлкен ЭЕМ-дің (IBM-370, ЕС-1045, ЕС-1060) «үстемдігі» уақытында таралды. Бұл үлгінің негізгі кемшілігі қолданушылар саны өскен кезде өнімділіктің күрт төмендеуі

- Есептердің қиындығының жоғарлауы, дербес компьютерлердің және локальды есептеуіш желілердің пайда болуы жаңа файл-сервер архитектурасының пайда болуының алғышарты болып табылды. Желілік рұқсаты бар деректер қорының бұл архитектурасы, деректер қорының файлдары сақталатын, бөлінген сервер ретінде компьютерлердің біреуінің тағайындалады деп болжайды [3.2].
- Қолданушылар сұранысына сәйкес файлсерверден файлдар, деректерді өңдеудің негізгі бөлігі жүзеге асатын, қолданушылардың жұмыс бекеттеріне беріледі. Орталық сервер, деректердің өзін өңдеуге қатыспай, негізінен тек файлдарды сақтаушы қызметін атқарады.



3.2 сурет – «Файл-сервер» архитектурасы

Жұмыс келесі түрде тұрғызылған:

1) Деректер қоры файлдар жиыны түрінде арнайы бөлінген компьютердің (файлдық сервер) қатты дискісінде орналасады.

2) Клиенттік компьютерлерден тұратын, локальдық желі болады, олардың әр қайсысында ДҚБЖ және ДҚ жұмыс жасайтын қосымшалар орнатылған.

3) Клиенттік компьютерлердің әр қайсысына қолданушылардың қосымшаларды жүктеу мүмкіндігі бар. Қосымшамен ұсынылған қолданушылық интерфейстерді пайдалана отырып, ол ДҚ жүгінуді ақпаратты сұрыптауға\жаңартуға бастамшылық етеді.

- Файл-сервер. Данная архитектура систем БД предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Каждый пользователь может запускать приложение, расположенное на сервере, при этом на компьютере пользователя запускается копия приложения. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном производится обработка. Когда пользователь сети работает с БД, на его компьютере появляется локальная копия общей БД. Эта копия периодически обновляется данными, содержащимися в БД, расположенной на сервере. Архитектура файл-сервер обычно используется в таких сетях, где имеется немного компьютеров. Для ее реализации предназначены персональные СУБД, например Paradox и DBase. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает.

- 4)ДҚ барлық жүгінулер ДҚБЖ арқылы жүреді, ол өзінің ішінде, файлдық серверде орналасқан, ДҚ физикалық құрылымы туралы барлық мәліметтерді бумаға жинайды.
- 5)ДҚБЖ, файлдық серверде орналасқан, деректерге жүгінуге бастамшылық етеді, соның нәтижесінде ДҚ файлдарының бір бөлігінің клиенттік компьютерде көшірмесі алынады және өңделеді, ол қолданушының сұранысын қамтамасыз етеді (деректерге жасалатын қажетті операциялар жүзеге асады).
- 6)Қажет болған кезде (деректердің өзгерген жағдайында) деректер ДҚ
- жаңарту мақсатымен файлдық серверге кері қайтады.
- 7)Нәтижесін ДҚБЖ қосымшаға қайтарады.
- 8)Қолданушы интерфейсін қолдана отырып, қосымша сұраныстардың
- орындалу нәтижесін бейнелейді.

- «Файл-сервер» архитектурасының шеңберінде күнделікті қажетті ДҚБЖ деп аталатын, мысалы, dBase және Microsoft Access секілді, атақты бірінші нұсқаларын орындалды. [3.2] әдебиеттерде бұл архитектураның келесі негізгі кемшіліктері көрсетіледі:
- 1)Бір деректерге бірдей уақыт мезетінде көптеген қолданушылардың жүгінуінен жұмыс өнімділігі кенет төмендейді, себебі деректермен жұмыс
- жасап отырған қолданушы жұмысын аяқтағанға дейін күту қажет. Кері жағдайда, бір қолданушылармен жасалған, басқа қолданушылармен
- өзгертілген, өзгертулер өшірілуі мүмкін.
- 2)ДҚ рұқсат кезіндегі барлық ауыртпалық клиент қосымшасына түседі,
- себебі кестедегі ақпаратты іріктеуге сұранысты беру кезінде ДҚ барлық
- кестесінің көшірмесі клиенттік машинаға алынады және іріктеу клиентте
- жүзеге асады.
- Осылайша, клиенттік компьютер және желі ресурстары тиімсіз
- жұмсалады. Нәтижесінде желілік трафик өседі және қолданушы компьютерінің аппараттық қуаттылығына талап жоғарылайды.

- 3)Ережеге сәйкес, жеке жазбалармен жұмысқа бағытталған,навигациялық амал қолданылады.
- 4)Файл-сервердегі ДҚ, тікелей аспаптық құралдардан тұратын (мысалы, Paradox және dBase файлдарына арналған Borland фирмасының Database Desktop утелиттерінен), қосымшаларға тоқтамай, жеке кестелерге өзгерістер енгізу әлдеқайда жеңілдеу; мұндай мүмкіншілік, іс жүзінде ДҚБЖ беректер қоры физикалық ұғымға қарағанда логикалық ұғымға жақын, жағдайымен жеңілденеді, себебі ДҚ деп, дисктегі жеке каталогта бірге жүретін, жеке кестелер жиынын түсінеміз. Осының барлығы, қымқыру және зиян келтіру тұрғысынанда және қате өзгерістер енгізу тұрғысынанда қауіпсіздік деңгейінің өте төмендігі туралы айтуға мүмкіндік береді.
- 5)Транзакцияның жеткіліксіз дамыған аппараты бір жазбаға бір уақыт мезетінде өзгерістер енгізу кезінде ақпараттың мағыналық және сілтемелік тұтастығының бұзылуындағы ықтимал қате көзінің қызметін атқарады.

- «Клиент – сервер» технологиясын қолдану бір желіге біріктірілген, олардың біреуі ерекше басқарушы қызметін атқаратын (желі сервері болып табылатын), біршама компьютерлер санын талап етеді.
- Демек, «клиент – сервер» архитектурасы қосымша қызметтерін қолданушы (клиент деп аталатын) және сервер қызметі деп бөледі. Клиент қосымшасы, реляциялық ДҚ әлемінде өндірістік стандарт болып табылатын, құрылымдық SQL (Structured Query Language) сұраныс тілінде, ДҚ орналасқан, серверге сұраныстарды құрастырады. Алшақтатылған сервер сұранысты қабылдайды және ДҚ оны жаңа мекен-жайға жібереді.

- Клиент-сервер. В этой концепции подразумевается, что помимо хранения централизованной БД сервер базы данных дожжен обеспечивать выполнение основного объема обработки данных. Технология клиент-сервер разделяет приложение на две части: клиентскую и серверную. Клиентская обеспечивает интерактивный интерфейс, сервер обеспечивает управление данными, разделение информации, администрирование и безопасность. Для получения данных приложение-клиент формирует и отсылает запрос удаленному серверу, на котором размещена БД. Запрос формируется на языке SQL, который является стандартом доступа к серверу при использовании реляционных баз данных. После получения запроса удаленный сервер направляет его SQL-серверу (серверу баз данных). SQL-сервер – это программа, которая управляет удаленной БД и обеспечивает выполнение запроса и выдачу клиенту его результатов – требуемых данных. Вся обработка запроса выполняется на удаленном сервере. Для реализации архитектуры клиент-сервер обычно применяются многопользовательские СУБД, например Qracle, MS SQL Server, InterBase и др. Подобные СУБД называют промышленными, так как они позволяют организовать информационную систему, состоящую из большого числа пользователей.

- SQL-сервер – алшақтатылған деректер қорын басқаратын, арнайы бағдарлама. SQL-сервер сұранысты түсіндіруін, деректер қорында оның орындалуын, сұранысты орындауда нәтиженің құрастырылуын және оның клиент-қосымшасына берілуін қамтамасыз етеді. Сонымен бірге клиенттік компьютер ресурстары сұраныстардың физикалық орындалуына қатыспайды; клиенттік компьютер тек сұранысты серверлік ДҚ жібереді және нәтижесін алады, содан кейін оны қажетті түрде түсіндіріп, қолданушыға ұсынады. Клиенттік осымшаға
- сұраныстың орындалуының нәтижесі жіберілетіндіктен, желі бойынша тек клиенттке қажетті деректер ғана жүреді. Нәтижесінде желіге түсетін ауырпалық жеңілдейді. Сұраныстың орындалуы, деректер сақталатын жерде үретіндіктен, үлкен деректер пакетін тасымалдау қажеттілігі жоқ. Сонымен қатар, SQL-сервер, егер ол мүмкін болса, азғантай уақыт мезетінде азғантай шығынмен алынғын сұранысты оңтайландырады [3.2,3.3]. Жүйе архитектурасы 3.3 суретте келтірілген.

- Осының барлығы жүйенің жылдамдығын көтереді және сұраныс нәтижесін күту уақытын төмендетеді. Сервермен сұранысты орындау кезінде деректер қауіпсіздігінің деңгейі де жоғарылайды, себебі деректердің толықтығының ережесі сервердегі деректер қорында анықталады және осы ДҚ қолданатын, барлық қосымшалар үшін бір болып табылады. Осылайша, бірдей деректерді бір уақыт мезетінде әртүрлі қолданушылармен өзгерту мүмкіндігін болдырмау жоққа шығарылады және апаттық жағдаймен аяқталған, ДҚ өзгертулер енгізу кезінде бастапқы мәндерге кейін шегіну мүмкіндігі ұсынылады [3.2, 3.3].

3.3 сурет – «Клиент – сервер» архитектурасы

Сонымен, нәтижесінде жұмыс келесі түрде тұрғызылды:

1) Файлдар жиыны түріндегі деректер қоры арнайы бөлінген компьютердің (желі серверінің) қатты дискісінде болады.

2) ДҚБЖ де желі серверінде орналасады.

3) Клиенттік компьютерлерден тұратын, локальды желі болады, олардың әрқайсысында ДҚ жұмыс жасауға арналған қосымшалар орнатылған.

4) Клиенттік компьютерлердің әрқайсысында қолданушыларға қосымшаларды жүктеу мүмкіндігі бар. Қосымшамен ұсынылған қолданушы интерфейсін қолдана отырып, ақпараттарды іріктеу\жаңарту үшін, серверде орналасқан, ДҚБЖ жүгінуді түсіндіреді, яғни клиенттен серверге желі бойынша тек сұраныс мәтіндері ғана беріледі.

- 5) ДҚБЖ өз ішінде, серверде орналасқан, ДҚ физикалық құрылымы туралы барлық мағұлматтарды оқшаулайды.
- 6) Серверде жатқан деректерге жүгінулерді бастамшылық етеді, нәтижесінде серверде деректердің барлық өңдеуі жүзеге асады және тек
- сұраныстың орындалу нәтижесінің клиенттік компьютерге көшірмесі алынады. Осылайша, ДҚБЖ қосымшаға нәтижені қайтарады.
- 7) Қолданушы интерфейсін қолдана отырып, қосымша сұраныстарды орындау нәтижесін бейнелейді.
- 8) Сервер мен клиенттер арасындағы функцияларды шектеу қалай болатынын қарастырайық.
- 9) Клиент-қосымшасының функциялары:
- - серверге сұраныстарды жіберу;
- - серверден алынған, сұраныс нәтижелерін түсіндіру;
- - нәтижелерді біршама формада (қолданушы интерфейсі) қолданушыларға ұсыну.
- 10) Серверлік бөліктің функциялары:
- - клиент-қосымшасынан сұраныстарды қабылдау.
- - сұраныстарды түсіндіру.
- - ДҚ сұраныстарды тиімдеу және орындау.
- - нәтижелерді клиент-қосымшасына жіберу.
- - қауіпсіздік жүйесін қамтамасыз ету және рұқсатты шектеу.
- - ДҚ толықтығын басқару.
- - жұмыстың көп қолданушылық режимінің тұрақтығын жүзеге асыру.

«Клиент–сервер» архитектурасында «өндірістік» деп аталатын ДҚБЖ жұмыс жасайды. Олар өндірістік деп, тек осы кластың ДҚБЖ ғана орташа көлемдегі және ірі көлемдегі мекемелердің, ұйымдардың, банктердің ақпараттық жүйе жұмысын қамтамасыз ете алатын болғандықтан аталады. Өндірістік ДҚБЖ қатарына MS SQL Server, Oracle, Gupta, Informix, Sybase, DB2, InterBase және т.б. жатады [[3.2]].

Ережеге сәйкес, SQL-сервер жеке қызметкермен немесе қызметкерлер тобымен (SQL-сервер әкімшіліктері) қызмет көрсетіледі. Олар деректер қорының физикалық сипаттамаларын басқарады, тиімділікті, баптауларды және ДҚ түрлі компоненттерінің қайта анықталуын жүргізеді, жаңа ДҚ құрады, барларын өзгертеді және т.б., сонымен қатар түрлі қолданушыларға айрықша құқықтар береді (нақты ДҚ, SQL-серверге белгілі-бір деңгейдегі рұқсатқа рұқсат береді) [[3.2]].

Берілген архитекетураның "файл-сервер" архитектурасымен салыстырғандағы негізгі жетістіктерін қарастырайық:

1) Желілік трафик барынша кішірейеді.

2) Клиенттік қосымшалардың күрделілігі азаяды (ауыртпалықтың үлкен бөлігі серверлік бөлікке түседі), демек, клиенттік компьютерлердің аппараттық қуаттығына талап төмендейді.

3) Арнайы бағдарламалық құралдың – SQL-сервердің – бар болуы жобалық және бағдарламалаушылық есептің маңызды бөлігінің шешіліп тұруына алып келеді.

- 4) Демек ДҚ толықтығы мен қауіпсіздігі де жоғарылайды.
- Кемшіліктер қатарына аппаратық және бағдарламалық асақтамаларға жоғары финанстық шығындарды, сонымен қатар, әртүрлі орындарда орналасқан клиенттік компьютерлер снының көбі барлық клиенттік- компьютерлерде клиенттік қосымшаларды уақытылы жаңартулардың белгілібір қиындықтарын тудыруды жатқызуға болады.
- Дегенмен, «клиент – сервер» архитектурасы өзін тәжірибеде жақсы ұсынды, қазіргі уақытта, осы архитектураға сәйкес тұрғызылған, ДҚ көптеген саны бар және жұмыс істейді.