**Дәріс 1**

 **Тақырыбы: Деректер қоры ұғымы және деректер қоры модельдері. Серверлік деректер қорын басқару жүйелері және заманауи тенденциялар.**

**Жоспар:**

**1.1 Деректер қоры.**

**1.2 Клиент-сервер архитектурасының үлгілері және деңгейлері**

 **1.3 Серверлік деректер қорын басқару жүйелері және заманауи тенденциялар**

 **Мақсаты:** Деректер қоры ұғымы және деректер қоры модельдері**н айқындау. Серверлік деректер қорын басқару жүйелерінің қызметін түсіндіру, заманауи тенденцияларға шолу жасау.**

**Негізгі түсініктер:** Клиент-сервер жүйелік архитектурасы, сервер, клиент, деректер қоры сервері.

Негізгі әдебиет: [1-3]

Қосымша әдебиет: [2-4]

**Деректер қоры (ДҚ)** — құрылымы берілген тәсіл бойынша ұйымдастырылған арнайы форматтағы файл (деректер қорын , кейде **берілгендер қоры** деп те атайды). ДҚ-да сақталатын берілгендер кесте түрінде дайындалады. Яғни *дискілік файл — деректер қоры,* оған енгізілетін кесте - ДҚ құрамында сақталушы **объект** (объект — қасиеттері, өрістері бар, өзімен түрлі әрекеттер орындауға болатын нәрсе, мысалы, кесте). Деректердің байланысты элементтер тобын **жазба** (не **жазу)** деп атайды.

Деректерді және олардың арасындағы байланыстарды ұйымдастырудың түрлі *типтері* бар: *иерархиялық, реляциялық* және *тораптық.*

**Иерархиялық** ДҚ-ында жазба элементтері реттеліп жазылады да, оның бір элементі негізгі, қалғандары бағыныңқы элементтер деп есептеледі. Мұнда элементтер нақты тізбек бойынша сатылы түрде реттеліп қойылады. Онда берілгендерді іздеп табу саты бойынша төмен бағытта жүргізіледі. Мысалы, *Norton Commander* программасындағы каталогтар, білім жүйесіндегі иерархиялық кластар тізімі, т.б.

 Иерархиялық модельде берілгендер арасында байланысты реттелген граф немесе (ағаш) арқылы көрсетуге болады.



**Берілгендердің желілік моделі.**

Жүйелік құрылымда деңгей, элемент (түйін), байланыс ұғымдар иерархиялық модельдегідей болады. Кез-келген элемент әр элементпен байланыса алады. Байланыстар типы



**Реляциялық ДҚ** — кесте түрінде дайындалған берілгендер. Ол ең көп тараған берілгендер қоры (relation — қатыс, байланыс), кестелер арасындағы байланыстар - жиі пайдаланылатын ең маңызды ұғым. Деректер қорында **жазба** үшін кесте жолы түсініледі, кестеде қайталанатын бірдей жолдар болмайды. Кесте бағанын құрайтын жазба элементтері **өріс** деп аталады. Өрістің негізгі ерекшелігі — бір өріс элементтері бірдей типті етіп құрылады, мысалы, саңдық не символдық.

Бұл — реляциялық деректер қорының Excel-ден негізгі айырмашылығы (Excel-де бір өріс элементтері түрлі типті бола береді және ол тек бір пайдаланушыға ғана арналған, ал реляциялық ДҚ-мен көптеген адамдар жұмыс істей береді). Мұндай ДҚ-да кестелерді байланыстырып, олардың кейбір өрістері арқылы түрлі жаңа кестелер құруға да болады. Мұндай кестелерді **сұраныстар** деп атайды (сұраныс жөнінде кітаптың 4-тақырыбын қараңыз).

Реляциялық деректер қорын құру және онымен жұмыс істеуді басқару үшін көптеген арнайы программалар дайындалған: **dBASE, FoxBase, FoxPro, Access,** т.б. Оларды **Деректер қорын басқару жүйелері (ДҚБЖ)** деп атайды.

**Клиент-сервер моделі және деректер қоры**

*Клиент-сервер* жүйелік архитектурасының негізгі принциптері мыналар: АЖ- лер желінің әр түрлі шетіндегі- клиенттік және серверлік бөлім болып екі бөлімге бөлінеді.

Серверлік жақ өзімен қоса қуатты аппараттандырылған құралы, қажетті стандартты бағдарламалық қамтамасыз етуі, деректер қорын басқару жүйесі және ДҚ бар арнайы мамандандырылған аппараттандырылған комплексте жұмыс істейді.

Қолданбалы программа немесе пайдаланушы жүйенің клиенттік бөлімімен әрекеттеседі. Қосымшаның клиенттік жағы пайдаланушының жұмыс орнында жұмыс істейді.

Жүйенің клиенттік бөлімі қажеттілік жағдайында серверлік бөлімге желі (локальді немесе глобальді)  бойынша байланыс жасайды. Сонымен бірге клиент пен серверге қарағанда қатынас айқын орындалады. Бұл қатынасты орындайтын желілік компоненттің өзімен бірге жиынтығында  қажетті желілік жабдықтар, желілік түйіндер арасындағы байланыстарды қамтамасыз ететін бағдарламалық технологиялық жиынтық және де сұраныстар мен олардың орындалу нәтижелерінің айырбасы үшін  интерфейстік бағдарламалық қабаттар (протокол немесе протоколдар) бар.  Клиенттік және серверлік бөлімдердің арасында SQL деректер қорының тілі негізгі интерфейс болып табылады.

 Клиент-сервер архитектурасының ең қарапайым формасы - бұл екі бөлек процесстердің: клиент және сервер арасындағы есептеуiш жүктеменiң бөлiнуi. Сонымен бірге бұл екі процесстің арасында қосымшаның интерактивті стандартты функциясының бес тобы бар:

1) диалогтiң функциялары (Presentation Logic, PL) немесе көру компоненті. Қолданбалы есептiң көру компонентi пайдаланушы арқылы  ақпаратты еңгізуді және сонымен қатар ақпаратты экранға немесе баспаға шығаруды жүзеге асырады. Клиент-сервер архитектурасының көру компоненті пайдаланушының (себебі ол бағдарламалардың жұмыс нәтижесін бақылау керек )жұмыс орнында орындалады. Призентациялық логиканы ұйымдастыру үшін көбінесе GUI (User Interface Graphical)  пайдаланушысының графикалық интерфейсінің моделі немесе Web-интерфейс қолданылады;

2) қолданбалы функциялар немесе деректердi өңдеудiң (Business Logic, BL) алгоритмдарымен бірге қосымшаның кодының бір бөлігі. Қолданбалы логиканың компоненті шындығында деректерді өңдеуге байланысты  есепті шешеді және пайдаланушының іс әрекетіне немесе ішкі әрекеттерге қосымшаның реакциясының алгоритмдерін, деректерді өңдеу ережесін көрсетеді. Бұл компонент клиенттік және серверлік бөлімнің арасына қолданылатын модельіне қарай әртүрлі орналасуы мүмкін. Әдетте бұл код бағдарламалаудың жоғарғы деңгейіндегі тілдерде бағдарланады: C, C++, Visual Basic, Object Pascal және т.б.;

3) қосымша ішіндегі деректерді өңдеудің функциялары (деректерді тікелей ДҚБЖ басқарады) манипуляция жасауға қатысты қосымша кодының бөлімі. Тілі  SQL.

|  |  |
| --- | --- |
|   |  |

1 - сурет - «Клиент-сервер» қосымшасының типтік құрылымы

4) ақпараттық қорларды басқару функциясы (Database Manager System, СУБД). Деректер қорын сақтау компоненті деректерді сақтауға, ДҚ- дағы ақпаратты оқумен және ондағы қолданбалы есепке байланысты жазбалармен байланысты физикалық операцияларды жүзеге асырады. Клиент-сервер архитектурасындағы бұл компонент әрқашан серверде орындалады;

5) алдыңғы функция топтарының арасындағы байланыс рольін атқаратын қызметші функциялар.

**1.2 Клиент-сервер архитектурасының үлгілері және деңгейлері**

Құрама бөлімнің санына қарай клиент- серверлік жүйелер екі деңгейлі және үш деңгейлі болып бөлінеді.

Екі деңгейлі жүйелер тек қана клиенттен және серверден тұрады. «Клиент-сервер» архитектурасының  екі деңгейлі модельіне мыналар жатады:

1) файлдік сервердің модельі (FS-модель);

2) алып тастаған рұқсаттың модельі (RDA-модель);

3) белсенді сервердің модельі (DBS-модель).

Үш деңгейлі модель «клиент – сервер қосымшасының» схемасына негізделген «клиент-сервер» архитектурасының бір түрінде көрсетілген. Мұндай архитектура жүйенің функцияларын және бағдарламалы- аппаратты комплекстің компоненттерінiң арасындағы жүктеменi  икемді үлестіреді, сонымен қатар пайдаланушының жұмыс орнының қорына талаптарды азайтады.

*ДҚ- ның серверінің моделі (кейде белсенді сервер деп атайды)*

Сервер моделі (Data Base Server, DBS)- қолданбалы компонент (биснес- логика) жартылай немесе толық деректер қорының серверында орналасатындығын білдіретін тұжырымдама. Бизнес-логика сақталатын программалық бiрлiктер (ПрБ), триггерлер, ДҚ- да сақталған, серверде басқарылатын мәлiметтердiң қолданбалы түрiнде ұйымдасқан. Деректер қорында сонымен қатар деректер жиналады және метамәлiметтердiң базасы (БМД) қалыптасады. Клиенттік қосымша серверге ПрБ бастау командамен бағытталады, ал сервер қажет жағдайда оны орындайды және ДҚ- да өзгерісін бекітеді. Сервер сұраныстың нәтижесін сыртқы құрылғыға шығару немесе клиентте орналасқан бизнес- логиканың бөлігін орындау үшін клиентке қайтарады. Деректерді өңдеудің үлестірулерінің осындай әдістерінде желілік трафик күрт төмендейді.

*Триггер – ДҚ күйімен байланысты арнайы оқиғаларды зерттеп отыру механизмі. ДҚ- дағы белгілі бір жағдайлардың пайда болуында  істейтін триггер тумблер болып табылады.   ДҚБЖ ядросы (барлық оқиғаларға барлау өткiзедi) жасалған және сипатталған триггерлерді ДҚ- ға шақырады және сәйкес келетін оқиғалар кездессе сервер сәйкес триггерді қосады=> триггер- бұл ДҚ- ға жасалатын және сақталған процедураларды шақыратын программа.*

Сервердің берілген моделі белсенді болып келеді себебі тек клиент қана емес сервер де триггердің механизмін пайдаланады.



2 - сурет -ДҚ белсенді серверінің моделі

DBS-моделінің артықшылықтары:

- қолданбалы функциялардың орталықтандырылған басқаруына мүмкiндiк;

- процедураның өшірілген шақыруы мүмкiндiктері есебiнен желілік трафиктiң төмендеуі;

- процедураларды бірнеше қосымшалар арасында бөлісу мүмкіндігі және компьютер қорларын SQL- сұраныстың  орындалуына жасалған жоспарды қолдану есебiнен үнемдеу. (өйткені, сақталған процедуралар және триггерлер ДҚ сөздігінде сақталады және оларды бірнеше клиенттер пайдалана алады => әр түрлі клиенттік қосымшадағы деректерді өңдеу алгоритмдерінің қайталануы төмендейді).

Кемшіліктері: сервердің өте көп уақытта жүктеліу.

Сервердің функциялары:

1. сипатталған триггерлерге байланысқан оқиғаларды барлауды жүзеге асырады;

2. триггерлерге қатысты оқиғалардың пайда болуында, олардың автоматты жұмыс iстеуiн қамтамасыз етедi;

3. әр триггердің ішкі программасының орындалуын қамтамасыз етеді;

4. пайдаланушының сұранысы бойынша сақталған процедураларды жібереді;

5. триггерлерде сақталған процедураларды жібереді;

6. клиентке қажет деректерді қайтарады;

7. ДҚБЖ- нің барлық функцияларын қамтамасыз етеді: деректерге қол жеткізу, ДҚ- дағы деректердің бүтіндігіне бақылау және қолдау, бiртұтас ДҚ мен барлық қолданушыларды жұмыспен қамтамасыз ету. Шын мәнінде, сервер функциялардың көптеген жиынтығын орындайды және көптеген клиенттерге қызмет жасайды. Серверді жеңілдету үшін көп деңгейлі моделдер жасалған, сонын бірі үш деңгейлі модель.

*Қосымша сервері*

Қосымша сервері (Application server, AS)-  «клиент - сервер қосымшасы» схемасына негізделген  «клиент-сервер»  көп деңгейлі архитектурасының әр түрі.

Бұл модельде клиент және сервер арасында қосымша аралық деңгей жүргізіледі. Бұл аралық деңгейде қосымшаның бір немесе бірнеше сервері бар. Осыдан- «қосымша серверінің модельі» AS (Application Server) атауы.

Бұл моделде АЖ барлық компоненттері үш пайдаланушы арасында бөлінеді.

Клиент пайдаланушы мен интерфейсті ұйымдастырады, қосымшаның призентациялық логикасын қамтамасыз етеді, клиент машинасына жергілікті және глобальді желіге қол жеткізудің коммуникациялық функцияларын орындайтын клиент қосымшасының шектелген кодын жұмысқа қосады, авторизация интерфейсін, шифрлеудің алгоритмін орындайды. Клиент және қосымша сервері API арқылы араласады ал сервер қосымшасы мен ДҚ- ның сервері  SQL тілінің сұранысымен араласады.

|  |  |
| --- | --- |
|   |  |

3 - сурет - Қосымша серверінің моделі

AS-моделінің артықшылықтары:

- сервердің жеңілдету есебінен өнімділігін жоғарылату;

- АЖ пайдаланудың арзандауы;

- өнiмдiлiктердің шығынысыз бизнес- логиканың күрделену мүмкiндiгі;

- БҚ етуді басқаруды қажет етпейді;

- жоғары қауіпсіздік;

- жоғары сенімділік;

- клиент пен қосымша серверінің арасындағы жылдамдыққа талаптың төмен болуы;

- олардың бағасының төмендігі сияқты, клиенттің техникалық және өндірістік сипаттамаларына талаптың төмен болуы.

Кемшіліктері:

- серверлік бөлімінің күрделілігі жоғарылауда және басқаруға және қызмет етуге қаражаттың жоғарылауы;

- қосымшаны құрудың жоғары күрделілігі;

- басқаруда және таратуда күрделірек;

- қосымша серверіне және деректер қорының серверіне өнімділіктің жоғарылығы, серверлік құралдың қымбат болуы;

- деректер қорының серверіне және қосымшалар серверлерінің арасындағы желінің жылдамдығына жоғарғы талап.

Қазіргі кездегі Веб жүйенің атақты архитектурасы  үшдеңгейлік түрде көрсетуге болады.



4 - сурет - Веб-қосымшасының архитектурасы

 **1.3 Серверлік деректер қорын басқару жүйелері және заманауи тенденциялар**

«Клиент-серверлік» жүйелер клиент пен сервер арасындағы міндеттер бөлінісінің бірнеше типіне иек тірейді:

- «Интеллектуалды» клиенттер (қалың клиент - жұқа сервер);

- «Интеллектуалды» сервер (жұқа клиент  - қалың сервер);

- араласқан жүйелер;

- көп деңгейлі жүйелер.

Іске асыру схемасын талап анализі негізінде таңдайды:

- желілік трафик бойынша;

- клиент және сервер ресурсы бойынша;

- деректер қорын өндіру бойынша.

«Интеллектуалды» клиенттер -  «клиент-серверлік» қосымшасы ұйымдастырудың ең көп таралған әдісі. «Интеллектуалдық» клиентке  ұсыныстың сервисінің орындайтынына, бизнес- логиканы және деректерді басқаруды сенуге болады. Бұл жағдайда сервер функциясы шындығында деректер (деректерді сақтау) қорының қолдауымен шектелген. Барлық ақпараттар клиентте жергiлiктi қаралудан өткендіктен сервер қоры босайды

«Интеллектуалды» клиенттің артықшылықтары:

- өңдеуді және жүйені бақылап отыруды жеңілдететін архитектураның қарапайымдылығы;

- клиенттік қосымшаны өңдеудің белгілі және едәуір  қуатты құралдары.

«Интеллектуалды» клиенттің кемшіліктері:

- клиентте бизнес ережелерді орындау кейде, клиентке ереже негізінде шешім қабылдау үшін барлық мәліметтерді беру керек болғандықтан желілік трафикті жоғарылатады;

- бизнес- логиканы түрлендіру үшін барлық клиенттерді күшейту керек.

«Интеллектуалды» серверлер. Барлық бизнес- ережелерді,  сақталатын процедуралар және ДҚ- ның триггерлерiнiң түрлерiнде жүзеге асырылатын SQL Server- ге көшіре отырып  ДҚ-ны сақтауға және функциялауға «Интеллектуалды» сервер» құрылады. Сонымен қатар, сервердің интеллекті деректерге (SQL-сұраныстарды орындайды) моннипулияция жасау қабілеттіліктерінде және клиентке қорытынды деректер жиынтығын қайтаруда айқындалады.

Араласқан жүйелер. Интеллектуалды серверлер мен интеллектуалды клиенттердің артылықшылықтарына ие. Серверде әдетте ДҚБЖ-ны, деректермен манипуляция жасауды,көптеген клиенттерге ортақ болатын биснес- логиканың бөлімін қопарады.  Клиентте дәл осы клиентке тән мәліметтердің бөліну логикасын және бизнес- логика бөлімін қопарады.

Көп деңгейлі жүйе (кейде оны үш деңгейлі деп атайды) пайдаланушының интерфейсін, бизнес- ережесін және деректер қорын  бөлуге мүмкіндік береді. Көп деңгейлі жүйеде ДҚ сервері мен клиентінің арасына орналастыратын бағдарламалық компонентті бизнес- ереже қосымшаның серверіне салады. Клиент тек қана пайдаланушысы бар интерфейске, ДҚ сервері- ДҚ- ны сүйемелдеумен және жұмысына жауап береді. Аралық қабат пайдаланушы үшін сервер болып табылады, ал деректер қорын басқару  жүйесі үшін- клиент. Клиенттер қажет жағдайда қосымшаның серверіне қарайды, ал ол өз кезегінде клиенттердiң сұраныстарын iске асыру үшiн қажеттi деректер үшін ДҚ серверіне қарайды.