**1.Деректер базасының архитектурасы**

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 архитектурада 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 броузерлер арқылы жүзеге асады.

**2. CORBA (Common Object Request Broker Architecture) технологиясы және оның қолданылуы**

CORBA (Common Object Request Broker Architecture) таратылған объекттерді құрудың

ең тиімді технологиясы болып табылады. Бұл технология инфрақұрылымнан, желілік

протоколдардан, операциялық жүйелер мен программалау тілдерінен тәуелсіз таратылған

жүйелерді құруды ұсынады. CORBA технологиясына қатысты үш түсінік қолданылады:

- әрекеттесуге қатысатын барлық объекттердің интерфейстерін сипаттайтын OMGIDL

(Interface Definition Language) интерфейстерді сипаттау тілі;

- жүйенің аралық звеносы болып табылатын және клиенттік қосымшаның серверлік

объекттермен әрекеттесу логикасынан тұратын ORB (Object Request Broker) объектті

сұраныстар брокері;

- клиенттің сұраныстары мен сервердің жауабын транспорттық қолдауды және ORB

арасындағы әрекеттесуді қамтамасыз ететін IIOP (Internet Inter-ORB Protocol) протоколы.

Бұл технологияның артықшылықтарына келесілер жатады:

· платформалық тәуелсіздік;

· тілдік тәуелсіздік;

· динамикалық шақырулар;

· масштабталу;

· CORBA-сервистер.

Java-технологиясы броузер желі арқылы осы апплетке қажетті Java-класстарды жүктеп

оны орындауға жіберуіне негізделген. Бұл кезде қолданушының машинасында жүктелген

апплет орындалатын Java Virtual Machine (JVM) жіберіледі. CORBA-клиент болып

табылатын Java-апплет, жүйенің басқа (серверлік) қосымшаларымен барлық қажетті

байланыстарды орнатады және ол арқылы қолданушыға барлық ақпарат түседі. Берілген

таратылған жүйе үшін апплет қолданушылық интерфейстің ролін ойнайды. Java-ға қойылған

көпағындық сияқты құралдар апплеттердің басқа да қосымшалармен синхронды,

асинхронды әрекеттесуін жеңіл іске асыруға мүмкіндік береді. Осыған байланысты Java және

CORBA технологиялары www технологиясына негізделген жүйелерді таратылған

ақпараттық жүйелермен біріктіру мәселесін шешудің әмбебап, қуатты құралы ретінде

қолданыла алады.

**3. ADO.NET жүйесіне толық түсінік беріңіз.(шифры)**

ADO (ағылшыннан аударғанда ActiveX Data Objects - «ActiveX мәліметтерінің объектісі») – Microsoft компаниясы жасап шығарған, ActiveX технологиясына негізделген интерфейс бағдарламасы. ADO бағдарламасы әртүрлі файлдарды объектіге бағытталған түрде көрсетеді.

ADO моделі бірнеше объектілерден тұрады:

* Connection (жойылған мәліметтер қорын қосуға арналған);
* Recordset (мәліметтер қорынан алынған мәліметтерді көрсетеді);
* Command (командаларды орындауға және SQL параметрлерін қолдануда қолданылады);
* Stream (ағымдағы ақпараттарды жазуға және оқуға қолданылады);
* Errors (қате екенін көрсетеді);
* Fields (мәліметтер қорының бағана мен кестені көрсетеді);
* Parameters (SQL – құрылымының параметрлерін көрсетеді);
* Properties (объектінің құрылымын көрсетеді).

Microsoft ActiveX Data Objects технологиясы – деректер қорының (ДҚ) қосымшаларынан дерек көздерiне әмбебап қатынауды қамтамасыз етедi.

ADO технологиясы және OLE DB-ның интерфейстерi әр түрлi дерек көздерiне қатынауға рұқсат беретін бiртұтас қосымшалары үшiн қамтамасыз етедi. Мысалы, ADO қолданушы қосымшасы бiрдей күрделi операциялар мен электрондық кестелерге жергiлiктi ДҚБЖ-да SQL-дың бiрлескен серверiнде және сақталған мәлiметтерге қолдана алады. SQL кез-келген дерек көзiне ADO арқылы бағытталған сұрау салуды орындайды.

ADO терминологиясына сәйкес кез-келген мәліметтер көзі провайдерлердің көмегі арқылы мәліметтерді сақтайды. ADO провайдерлері мәліметтерді бір-бірімен байланыстырады. Мәліметтерді сақтау типінің әр қайсында ADO провайдерлері болуы керек. Провайдер – мәліметтердің қай жерде сақталуын, қалай сақталғандығын біледі.

ADO компоненті жоғары дәрежелі командалар тілінде қолданылады, яғни, [VBScript](http://ru.wikipedia.org/wiki/VBScript)-тың [ASP](http://ru.wikipedia.org/wiki/Active_Server_Pages), [JScript](http://ru.wikipedia.org/wiki/JScript)-тың[WSH](http://ru.wikipedia.org/wiki/WSH), [Visual Basic](http://ru.wikipedia.org/wiki/Visual_Basic), [Delphi](http://ru.wikipedia.org/wiki/Delphi_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)).

Delphi-дің компонеттеріне Delphi (VCL), және интерфейс, және ADO обьектілері жатады. Ары қарай біз компонет ретінде Delphi-ді айтамыз, ал обьект немесе интерфейсін ретінде обьектіні немесе ADO интерфейсінайтамыз. Байланыс жасау үшін Delphi-дің ADO компоненттері пайдаланылды:

TADOConnection компоненті BDE Tdatabase компонентіне қатысты және мәліметтер базасын қолдануда және транзакциямен жұмыс істеудеқолданылады.

TADOTable – ADO арқылы жұмыс істейтін кесте.

ADOQuery – мәліметтер базасына кіру. Бұл мәліметтер базасын шығару және шақырмау кезінде қолданылады.

**ADO.NET** — технология, предоставляющая доступ к [данным](http://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5) для приложений, основанных на [Microsoft .NET](http://ru.wikipedia.org/wiki/Microsoft_.NET). Является не развитием более ранней технологии [ADO](http://ru.wikipedia.org/wiki/ADO), а самостоятельной технологией, частью [фреймворка .NET](http://ru.wikipedia.org/wiki/.NET_Framework).

Классы ADO.NET находятся в сборке System.Data.dll.

Многие другие части фреймворка .NET, как и сторонние библиотеки, используют те или иные части ADO.NET. Например:

* Части .Net Framework
  + [Entity Framework](http://ru.wikipedia.org/wiki/ADO.NET_Entity_Framework)
  + [Linq](http://ru.wikipedia.org/wiki/Linq) to SQL
* Сторонние библиотеки

**4. Деректерді ұсынудың модельдері**

70-80 жылдардың соңында деректер базасын басқару жүйесінің дамуы негізінде деректерді ұсынудың әртүрлі модельдері пайда болды. Әрбір модельдің өзінің ерекшеліктері мен жетіспеушіліктері болды.

Деректер моделі деп берілген талаптарға сәйкес мәліметтерді интерпретациялауды жүзеге асыратын құралдарды айтамыз. Сонымен қатар, мәліметтер моделі мәліметтердің мағлұматтық мазмұнын көру мүмкіндігін беретін абстрактілі ұғым болып табылады.

Мәліметтер моделі екі класқа бөлінеді:

- Мықты типтендірілген моделі, мұнда барлық мәліметтер қандай да бір категорияға қатысты болады.

- Әлсіз типтендірілген моделі, категорияға қатысты ешқандай болжамдармен байланысты емес.

Деректер моделінің үш негізгі типін қарастырайық: иерархиялық, желілік, реляциялық.

Иерархиялық деректерді моделдеу .Иерархиялық құрылым бір-бірімен белгілі бір ережелер бойынша байланысқан элменеттер жиынтығын береді. Иерархиялық байланыстағы объектілер бағытталған граф (төңкерілген ағаш) құрады.

Желілік деректерді моделдеу. Желілік құрылымда алдыңғы аталған негізгі ұғымдардағы (деңгей, түйін, байланыс) әр элемент кез-келген басқа элементпен байланыста бола алады. Иерархиялық модельдің жақсартылған түрі болып табылатын желілік модель көп қолданушылық режимде құрылатын модель. Деректерді желілік ұсыну элементарлы мәліметтер және олардың арасындағы қатынастар бағытталған желі түрінде көрсетіледі (төбелері – мәліметтер, доғалар - қатынастар). Мысалы, деректер базасында тапсырыстарды сақтауға арналған бір тапсырыс әртүрлі үш байланысқа қатынаса алады.

Реляциялық деректерді моделдеу. Иерархиялық және желілік модельдердің жетіспеушіліктері деректердің реляциялық моделінің пайда болуына әкелді. Реляциялық модель әрекеті деректер базасының құрылымын оңайлатты. Мұндағы барлық мәліметтер жолдар мен бағандардан құралған қарапайым кесте түрінде көрсетілген.

Реляциялық деген ұғым (ағыл.relation - қатынас) деректер базасы жүйелерінің маманы белгілі американдық ғалым Е.Коддонның есімімен байланысты.

Бұл модельдер деректер құрылымының қарапайымдылығымен, пайдаланушыға ыңғайлы кесте түрінде сипаттайды және бұл модельде қатынастардың және деректерді өңдеуде реляциялық есептеулердің формальды алгебралық аппаратын қолдану мүмкіндігі бар.

**5. CORBA архитектурасына жалпы шолу. (7)**

CORBA (Common Object Request Broker Architecture) таратылған объекттерді құрудың

ең тиімді технологиясы болып табылады. Бұл технология инфрақұрылымнан, желілік

протоколдардан, операциялық жүйелер мен программалау тілдерінен тәуелсіз таратылған

жүйелерді құруды ұсынады. CORBA технологиясына қатысты үш түсінік қолданылады:

- әрекеттесуге қатысатын барлық объекттердің интерфейстерін сипаттайтын OMGIDL

(Interface Definition Language) интерфейстерді сипаттау тілі;

- жүйенің аралық звеносы болып табылатын және клиенттік қосымшаның серверлік

объекттермен әрекеттесу логикасынан тұратын ORB (Object Request Broker) объектті

сұраныстар брокері;

- клиенттің сұраныстары мен сервердің жауабын транспорттық қолдауды және ORB

арасындағы әрекеттесуді қамтамасыз ететін IIOP (Internet Inter-ORB Protocol) протоколы.

Бұл технологияның артықшылықтарына келесілер жатады:

* · платформалық тәуелсіздік;
* · тілдік тәуелсіздік;
* · динамикалық шақырулар;
* · масштабталу;
* · CORBA-сервистер.

Java-технологиясы броузер желі арқылы осы апплетке қажетті Java-класстарды жүктеп

оны орындауға жіберуіне негізделген. Бұл кезде қолданушының машинасында жүктелген

апплет орындалатын Java Virtual Machine (JVM) жіберіледі. CORBA-клиент болып

табылатын Java-апплет, жүйенің басқа (серверлік) қосымшаларымен барлық қажетті

байланыстарды орнатады және ол арқылы қолданушыға барлық ақпарат түседі. Берілген

таратылған жүйе үшін апплет қолданушылық интерфейстің ролін ойнайды. Java-ға қойылған көпағындық сияқты құралдар апплеттердің басқа да қосымшалармен синхронды,асинхронды әрекеттесуін жеңіл іске асыруға мүмкіндік береді. Осыған байланысты Java жәнеCORBA технологиялары www технологиясына негізделген жүйелерді таратылған ақпараттық жүйелермен біріктіру мәселесін шешудің әмбебап, қуатты құралы ретінде қолданыла алады.

**6. Заттық облыстағы концептуальды модель.**

Деректер базасын жобалау процесі пайдаланушылардың концептуалдық талаптарды құрудан басталады. Бұл талаптар жеке пайдаланушылардың талаптарын біріктру арқылы құралып, **деректер базасының коцептуалды моделі** деп аталады. Басқаша айтқанда, концептуалды модель деректер базасымен жұмыс жасайтын барлық приложениялар жұмыс жасайтын деректерді біріктіріп, осы приложениялардың талаптардың барлығын ескереді. Концептуалды модель оьъектілерді және олардың байланыстарын оларды сақтаудың физикалық әдістерінсіз көрсетеді. Сонымен концептуалды модель пәндік(заттық) облыс болып табылады.

Концептуалды модель кейін таңдалған ДББЖ сәйкес келетін деректер моделіне айналады. ДББЖ қамтамасыз ете алатын концептуалды моделдің версиясы логикалық модель деп аталады. Пайдаланушыларға бұл логикалық моделдің сыртқы моделдер деп аталатын құраушылары бөлінеді. Логикалық модель физикалық жадыға ауысады.Оны стандартизация бойынша американдық комитет ANSI (American National Standards Institute) ұсынды.

**8. Серверлік компоненттер және функциялары.**

MySQL серверін қайнардың өзінен жинап орнатуға болады. Бірақ бұл біршама уақыт қажет ететін және жалықтыратын процесс. Бұл операция көбінесе егер сізге ерекше конфигурацияда сервер қажет болса ғана істеледі. Linux операциялық жүйесінің бағдарлама пакеттерін басқаруға арналған жүйесі бұндай жұмыстарды минималды қимылмен орындатқызады. Дей тұрғанмен, әдетте орнатудан кейін қауіпсіздік және оңтайландыруға байланысты конфигурациялық жұмыстар жасау керек болады.

SS – МК ядросы (мәліметтер қоры). Әрбір SS данасы үшін өзіндік SS қызметі болады. MS SS Analysis Services (AS) – ол аналитикалық функцияларды іске қосады. SS даналары санына тәуелсіз жалғыз түрде орнатылады.

SS Agent – бұл SS администраторлық алыдн-ала жоспарланған іс-әрекетін орындайды. Әрбір SS данасының өзіндік қызметі бар.

MS Search (іздеу) – толық мәтіндік іздеу ядросы. Барлық даналарға ортақ.

MS DTS (Distributed Transaction Coordinator) – бір уақытта бірнеше SS даналарында орындалатын таралынған транзакторларды басқару үшін қолданылады

**9. Модель элементтерін құрастыру жолдары.**

Мәліметтердің реляциялық, моделінің элементтері мен оларға сәйкес ұғымдар көрсетілген. Мәліметтер қорын құрайтын кестенің әрбір жазбасын даралау үшін, яғни кестеде бірдей жазбалардың болмауын қадағалау үшін кестенің бір немесе бірнеше атрибуттарынан кестенің кілтін немесе негізгі кілтін анықтауға болады. 1970 ж. Коддом ойлап шығарған реляциялық мәліметтер моделі иерархиялық желілік модельдердің кемшіліктерін жойды. Бұл модельде нақты көрсеткіштер жоқ болды, барлық мәліметтер бағандар мен жолдарға бөлінген жай кесте ретінде көрсетілді.

Кесте кілтінің мәні бірде-бір рет кайталанбайтын ерекше болуы шарт, яғни кестеде кілт бағанасындағы (бағаналарындағы) мәндері өзара бірдей екі немесе одан да көп жазбалар болуы мүмкін емес.

Мысалы, Student (ҒІО, Special, Datard) кестесінде кілт ретіңде ҒІО немесе Special немесе Datard жеке жиектерін қолдануға болмайтыны түсінікті:

- FIO жиегінің кілт болмайтын себебі — кестеде аты-жөндері бірдей студенттер болуы мүмкін;

- Special жиегінің кілт болмау себебі — бір мамандықга бірнеше студентгердің дәріс алатыны;

Datard жиегінің кілт болмайтын себебі — туған күндері бірдей студенттер болуы мүмкін.

Бұл кестенің кілті ретінде FIO, Special жөне Datard жиектерінің жиынын да алуға болмайтыны анық: жоғары оку орнында бір мамандық бойынша бір күнде туылған аты-жөндері бірдей бірнеше білімгерлердің кездеспейтініне кім кепіл? Мұндай жағдайда кестеге әрбір жазбадағы мәні ерекше болатын арнайы жиек енгізген дұрыс; мысалы натурал сандарды кабылдайтын білімгерлердің коды — Kodstud атрибуты: Student(Kod\_stud, FIO, Spe¬cial, Datard) (кілт-жиектің асты сызылған). Әдетте мұндай жиектің мәні қолданбалы программада программалық жолмен немесе мәліметгер қорында автоматтандыру арқылы беріледі.

Кесте кілтінің атқаратын қызметі:

І.Жазбалардың қайталануын болдырмау

2.Кортеждерді кілт жиектің (жиектердің) мөндерінің өсу немесе кему ретімен реттеу

3. Кестедегі жазбаларды окуды тездету

4.Кестелерді өзара байланыстыру.

Реляциялық моделде кестелердің байланысы сыртқы кілттер (Foreign Key — FK) арқылы ұйымдастырылады. Сыртқы кілт — мәңдері басқа бір кестенің жазбалары арқылы көрсетілген, яғни басқа бір кестенің негізгі кілтінің мәндерінен тұратын атрибут.

**10. "Бірге-көп" байланысы.**

МҚ-да қатыстардың байланысының негізгі үш түрі кездеседі:

- "бірге-бір " (1:1) — one-to-one relationship;

-"бірге-көп" (1:М) — one-to-many relationship немесе "көпке-бір"(М:1);

-"көпке-көп" (М:М немесе M:N) —many-to-many relationship.

Егер әрбір клиент нөл, бір немесе бірнеше тапсырыс жасай алатын болса, онда бұл кестелер "бірге-көп" немесе *master- detail* катыста, Customer — *master* кесте, Order — *detail кесте.*

"Бірге-көп" байланысына (1:М) қоймадан тұтынушыларға жіберілген заттарды есепке алу тарифі бойынша Tovar және Rashod кестелерінің байланысын көрсетуге болады (3,4—суретгер), мұңдағы Tovar— заттың аты, Edjzm — тауардың өлшем бірлігі, Сепа — тауардың бір бірлігінің сату бағасы, Dataras— тауардың жібірілген күні, Kolras — жіберілген заттың саны. Бұл мысалдағы 14.05.04 -не дейінгі мәліметтер бойынша күріш әлі сатылмаған, соңдықган Rashod кестесіңце күрішке сәйкес жазба жок, Rashod кестесінің кілті Tovar және Rashod жиектерінен тұрады.

МҚ құрылымының кең тараған IDEF1X нотациясында "бірге-көп" қатысы "көп" жағында дөңгелекпен аяқталған сызық арқылы көрсетіледі

**11. BDE ( Borland Database Engine) – технологиясына жалпы шолу**

Borland Database Engine (BDE) механизмы

Негізгі BDE механизмі деректер базасын құрайтын компоненттерді қамтамасыз етеді. Үстеме мен деректер базасының арасындағы интерфейс түрінде қызмет атқарады.

Жүйелік DDL түрінде орындалған.Компоненттік обьектілер арасындағы өзара қарым-қатынас - нақты деректер базасына немесе ақпарат алмастыруға ешқандай бағыты жоқ және одан тәуелді емес.

Дәл осы BDE драйверлерге жатады. Деректер базасы үшін спецификалық, қосымша үшін нақтылық деректер болады.

BDE - ODBC (Open Database Connectivity) аналогтық драйвер контреллерінде қызмет атқарады. Инструменталды құралдарға құрылушы қолданбадан тыс мәліметтер қорына қызмет етуді қамтамасыз ететін арнайы программалар мен дестелер жатады. Компоненттер мәліметтер қорымен операцияларды жүзеге асыратын қолданбаларды құруға арналады.

Мәліметтер қорымен операциялар жасау үшін Delphi келесі инструменталды құралдар жинағын ұсынады:

- Borland Database Engine (BDE) – Delphi-қолданбасынан мәліметтер қорына қатынауды ұйымдастыруға арналған динамикалық кітапханалар мен драйверлер жинағынан тұратын мәліметтер қорының процессоры. BDE мәліметтер қорына қатынаудың орталық звеносы болып табылады.

- BDE Administrator – BDE-нің түрлі параметрлерін баптауға арналған утили-талар.

- Database Desktop – SQL-сұраныстар мен QBE сұраныстарды, кестелерді құруға және редакциялауға арналған программа.

- SQL Explorer – мәліметтер қоры мен мәліметтер сөздігін қарап шығуға және редакциялауға мүмкіндік беретін мәлі-меттер қорының жол серігі.

- SQL Builder – SQL –сұраныстарын визуалды түрде құру программасы.

- SQL Monitor – қашықты мәліметтер қорына SQL –сұраныстарының орындалу ретін бақылайтын программа.

- Data Pump – мәліметтер қоры арасында мәліметтерді тасымалдауға ар-налған программа.

**12. ADO Connection компоненттері.**

ActiveX - ActiveX- тің активті элементтері. Servers - кең таралған COM серверлеріне арналған VCL қабықшалары; ADO- бұл реляциялық және реляциялық емес мәліметтер қоры, Электрондық пошта, жүйелі, мәтінді, графикалық файлдардан тұратын мәліметтерден кез келген типіне сәйкес келетін қолданушы интерфейсі. мәліметтермен байланысу OLEDB технологиясының қатысуымен жүзеге асырылады.

       ADO –  ны қолдану мәліметтермен  жұмыс істеуге қамтамассыз етеді. ADO – мен жұмыс істеу үшін  компьютерде ADO 2.1 және одан жоғары нұсқалы жүйесі орнатылуы қажет.

       ADO – Бұл кез-келген мәліметтер типінің пайдаланушы интерфейсі, реляционды және реляционды емес мәліметтер қорын енгізуші, электронды почталы, жүйелі, мәтіндік және графикалық файлды. Мәліметтермен байланысты жүзеге асыру OLE DB технологиясы болып табылады.

       ADO – пайдаланушы альтернативті Borland Database Engine (BDE)-ге келіп, мәліметтердің эффективті жұмысымен қамтамассыз етілген. Мұндай мүмкіндікті пайдалану үшін біздің компьютерде ADO.2 немесе оданда жоғарғы версиялар орнатылуы керек. Мысалы: Microsoft SQL Server, сіз жұмыстап отырғанда мәліметтер қоры үшін OLE DB драйверіне ODBC бетінде орналасады. Олар (Connection обьектілері, Command және Record set  сияқты) ADO-ның мұндай обьектілері инкапсулдайды. Бұл сәйкестірілген жаңа Delphi ADOConnection, ADOCommand және ADODataset компененттерімен қамтамассыз етіледі.

        Мәліметтер қорының байанысы ADO технологиясының  нақты бауымен іске асырылады: мәліметтер жиынтығы          мәліметтер негізі (Data Source  компененті)           басқару және мәліметтерді тартып алу  компененттері (DBGrid, DBedit, және басқа).

       Тек ғана алғашқы бөлімде мұндай баулар компененттерде пайдаланылып, ADO беттерінде орналасады.

**13. Кесте кілтінің атқаратын қызметіне толық түсінік беріңіз**

Кесте кілтінің атқаратын қызметі:

І.Жазбалардың қайталануын болдырмау

2.Кортеждерді кілт жиектің (жиектердің) мөндерінің өсу немесе кему ретімен реттеу

3. Кестедегі жазбаларды окуды тездету

4.Кестелерді өзара байланыстыру.

Реляциялық моделде кестелердің байланысы сыртқы кілттер (Foreign Key — FK) арқылы ұйымдастырылады. Сыртқы кілт — мәңдері басқа бір кестенің жазбалары арқылы көрсетілген, яғни басқа бір кестенің негізгі кілтінің мәндерінен тұратын атрибут. Сыртқы кілті анықталған қатыс (кесте) оған сәйкес атрибуты (жиегі) негізгі кілті болып табылатын екінші бір қатысқа (кестеге) сілтеме жасайды деп айтады, мысалы, Customer және Order (Клиент жене Тапсырыс) кестелерін қарастырайық (1-сурет), мүңцағы Customer — басты

- (негізгі, аналық) кесте, Order – қосымша (бағыныңқы, балалық) кесте. Мазмұндауды

- жеңілдету үшін MS Access пакетінің құрамыңдағы Борей немесе NorthWind мәліметтер қорына сүйенейік:

Мәліметгер қорындағы деректер бірмәнді және өзара біртұтас үйлесімді болуы үшін реляциялық моделде шектеуші шартгар анықгалады. Шектеуші шарттар дегеніміз мәліметтердің мүмкін мәндерін қадағалаушы тәртіптер; мұндай үйлесімділік (бүтіңдік) шарттары мәліметтер қорын өңдеу және өзгерту кезінде кетуі мүмкін қателерді болдырмауға мүмкіндік береді.МҚ-ның шектеулері категориялык, бүтіндік және сілтеме

**14. Реляциялық деректерді моделдеу**

Иерархиялық және желілік модельдердің жетіспеушіліктері деректердің реляциялық моделінің пайда болуына әкелді. Реляциялық модель әрекеті деректер базасының құрылымын оңайлатты. Мұндағы барлық мәліметтер жолдар мен бағандардан құралған қарапайым кесте түрінде көрсетілген.

Реляциялық деген ұғым (ағыл.relation - қатынас) деректер базасы жүйелерінің маманы белгілі американдық ғалым Е.Коддонның есімімен байланысты.

Бұл модельдер деректер құрылымының қарапайымдылығымен, пайдаланушыға ыңғайлы кесте түрінде сипаттайды және бұл модельде қатынастардың және деректерді өңдеуде реляциялық есептеулердің формальды алгебралық аппаратын қолдану мүмкіндігі бар.

Реляциялық модель деректерді екі өлшемді кесте түрінде ұйымдастыруға бағытталған. Әр реляциялық кесте екі өлшемді массив түрінде беріледі де, мына қасиеттерге ие болады:

- кестенің әр элементі деректердің бір элементі;

- кестедегі бағандардың барлығы біркелкі, яғни бағандағы барлық элементтер бір типке (сандық, таңбалық т.б.) және бәрінің ұзындығы бірдей;

- әр бағанның өзіне тән аты бар;

- кестеде бірдей жолдар жоқ;

- кестеде жолдар мен бағандардың орналасу тәртібі әркелкі.

Қатынастар кесте түрінде беріледі. Кестедегі жолдар мен кортеж немесе жазба деп, ал бағандар қатынастар атрибуты немесе домен, өрістер деп аталады. Бір мәні жазбаны анықтайтын өрісті – қарапайым кілт (кілттік өріс) деп атайды. Егер жазбалар бірнеше өрістермен анықталса, онда мұндай кестенің құрамды кілті бар деп есептеледі. Екі реляциялық кестені ұйымдастыру үшін бірінші кестенің кілтін екінші кестенің кілтінің құрамына енгізу керек (кілттер бір-бірін қайталуы мүмкін). Басқа жағдайда бірінші кестенің құрылымына сыртқы кілтті енгізу қажет. Сыртқы кілт екінші кестенің кілті болады.

Реляциялық деректер моделінің **артықшылығы** ЭВМ-ғы жай, түсінікті және ыңғайлы физикалық жүзеге асырушылық болып табылады. Сол себепті жай және түсініктілік қолданушылар арсында кең қолданыс тапты.

Реляциялық деректер моделінің негізгі **кемшіліктері** келесідей: стандартты құралдардың жоқтығы, жолдардың атауы және иерархиялық және желілік байланыстарды суреттеу күрделілігі.

Реляциялық СУБД ПЭВМ-ға мысал ретінде: dBaseIII Plus, dBaseIV (Ashton-Tate), FoxBase (Fox Software), Paradox (Borland), Visual FoxPro, Access (Microsoft), Oracle (Oracle) және т.б.

Реляциялық СУБД-ның соңғы версиялары обьектілік – бағдарламалық жүйелердің кейбір қасиеттері еңгізілген мынадай СУБД обьектілік реляция деп атайды. Мұндай жүйенің мысалы ретінде Oracle 8x өнімі деп санауға болады. Жүйенің елесі версиялары Oracle 7x дейін таза реляциялық деп саналады.

**15. ADO технологиясының негізі.**

ADO (ағылшыннан аударғанда ActiveX Data Objects - «ActiveX мәліметтерінің объектісі») – Microsoft компаниясы жасап шығарған, ActiveX технологиясына негізделген интерфейс бағдарламасы. ADO бағдарламасы әртүрлі файлдарды объектіге бағытталған түрде көрсетеді.

ADO моделі бірнеше объектілерден тұрады:

Connection (жойылған мәліметтер қорын қосуға арналған);

Recordset (мәліметтер қорынан алынған мәліметтерді көрсетеді);

Command (командаларды орындауға және SQL параметрлерін қолдануда қолданылады);

Stream (ағымдағы ақпараттарды жазуға және оқуға қолданылады);

Errors (қате екенін көрсетеді);

Fields (мәліметтер қорының бағана мен кестені көрсетеді);

Parameters (SQL – құрылымының параметрлерін көрсетеді);

Properties (объектінің құрылымын көрсетеді).

ADO технологиясы және OLE DB-ның интерфейстерi әр түрлi дерек көздерiне қатынауға рұқсат беретін бiртұтас қосымшалары үшiн қамтамасыз етедi. Мысалы, ADO қолданушы қосымшасы бiрдей күрделi операциялар мен электрондық кестелерге жергiлiктi ДҚБЖ-да SQL-дың бiрлескен серверiнде және сақталған мәлiметтерге қолдана алады. SQL кез-келген дерек көзiне ADO арқылы бағытталған сұрау салуды орындайды.

**16. Кестелердің арасындағы байланыс түрлері.**

Көп жағдайда бір кесте (қатыс) жеке бір файлда сақталады. Жалпы МҚ алға қойылған мақсатқа сәйкес анықталған бір немесе өзара байланыстағы бірнеше кестелердің жиынынан тұрады, мысалы, MS Access МҚБЖ-де МҚ файлында кестелермен қатар сұрыптамалар, қорытындылар, үлгілер, макростар және модульдер сияқты басқа да МҚ-ның барлық объектілері бірге сақгалады.

Жоғарыда айтылғандай МҚ-ның екі кестесінің арасында бір-біріне тәуелділік қатысы болуы мүмкін; өзара байланыстағы екі кестенің біреуі master— басыңқы (негізгі, аналық), екіншісі de¬tail — бағыныңқы (қосымша, балалык) кесте болып табылады. Реляциялық МҚ-да қатыстардың байланысының негізгі үш түрі кездеседі (2-кесте):

- "бірге-бір " (1:1) — one-to-one relationship;

-"бірге-көп" (1:М) — one-to-many relationship немесе "көпке-бір"(М:1);

-"көпке-көп" (М:М немесе M:N) —many-to-many relationship.

Егер Customer кестесіндегі әрбір клиенттің ешқандай немесе тек қана бір тапсырыс жасау мүмкіндігі болса, онда Customer және Order кестелері "бірге-бір" қатысымен байланыста болады. Ал егер әрбір клиент нөл, бір немесе бірнеше тапсырыс жасай алатын болса, онда бұл кестелер "бірге-көп" немесе master- detail катыста, Customer — master кесте, Order — detail кесте (2-сурет).

"Бірге-бір" байланысы (1:1) екі кестенің де байланыстырушы жиектері олардың кілті болған жағдайда орын алады, сондықган бұл кестелердің өзара дәрежелері бірдей. Бұл байланыс МҚ-да сирек қолданылады, себебі мұндай кестелердегі мәліметтерді ЭЕМ жадын тиімді пайдалану мақсатында бір кестеге біріктіруге болады.

"Көпке-бір" байланысы (М:1) негізгі және қосымша кестені анықгау барысына байланысты МҚ-нда кеңінен қодданылатын "бірге-көп " байланысының (1:М) бір түрі болып табылады.

"Бірге-көп" байланысына (1:М) қоймадан тұтынушыларға жіберілген заттарды есепке алу тарифі бойынша Tovar және Rashod кестелерінің байланысын көрсетуге болады (3,4—суретгер), мұңцағы Tovar— заттың аты, Edjzm — тауардың өлшем бірлігі, Сепа — тауардың бір бірлігінің сату бағасы, Dataras— тауардың жібірілген күні, Kolras — жіберілген заттың саны.

"Көпке-көп" байланысы (М:М) негізгі кестенің бірнеше жазбасьша қосымша кестенің бірнеше жазбасы сәйкес болған жағдайда орын алады. 5-суретге "көпке-көп" байланыстағы оку топтарындағы пәндер мен сабақжүргізетін окзытушылар арасындағы катыстар көрсетілген: әрбір топка бірнеше окщушы сабақбереді;әрбір оқытушы біріншіден, бірнеше пәндерді, екіншіден, әртүрлі топтарда сабақ беруі мүмкін.

Мәліметтер қорын жобалау барысында бірнеше кесте өзара байланыста болып, бір кесте калған кестелермен әртүрлі қатыста болуы мүмкін. Байланыста тұрған кесте басқа кестелермен тағы да байланысқан жағдайда кестелердің иерархиясы түзіледі.

**17. ODBC және OLE DB технологиялары.**

Microsoft SQL Server, сіз жұмыстап отырғанда мәліметтер қоры үшін OLE DB драйверіне ODBC бетінде орналасады. Әлдеқашан анықталған OLE DB немесе ODBC сыртқы деректер көзіне қосылу үшін, деректерді қосу шеберін қолдануыңызға болады. Деректерді қосу шеберін ашу үшін, **Деректер**қойындысындағы **Сыртқы деректерді алу** тобында **Басқа көздерден**түймешігін басып, **Деректерді қосу шеберінен**параметрін таңдаңыз.Деректерді қосу шеберінде **Басқа/қосымша** деректер көзі параметрі таңдалса, **Деректермен байланыс сипаттары** тілқатысу терезесінде қол жетімді OLE DB жеткізушілерінің тізімін көруге болады. Оған қоса, ODBC драйверлеріне арналған Microsoft OLE DB жеткізушісі ODBC деректер көздеріне де қол жеткізуге мүмкіндік береді. Осы тілқатысу терезесіндегі әрбір қойындыны пайдалану туралы қосымша ақпарат алу үшін **Деректермен байланыс сипаттары** тілқатысу терезесіндегі **Анықтама** түймешігін басыңыз.

Жалпы, **Деректермен байланыс сипаттары** тілқатысу терезесінде қосылым мәліметтерін анықтау үшін төмендегі әрекеттерді орындаңыз:

* **Жеткізуші** қойындысын нұқып, OLE DB жеткізушісін таңдаңыз да, **Келесі** түймешігін басыңыз. Бұл кезде**Қосылым** қойындысы көрсетіледі, оның ішінде OLE DB жеткізушісі үшін нақты қосылым мәліметтерін енгізуге болады.

Windows-тың тағы бір ерекшелігі — Windows ортасында жұмыс істейтін түрлі қосымшалардың өзара әрекеттестігі. Мысалы, алмастыру буфері арқьлы не өз терезесін пайдаланып бір қосымшаньң берілгеңдерін екіншісіне ауыстыруға болады (буфер жөніңде 6.12-тақырыпты, ал, терезені пайдалану мысалын VII тарауға енгізілген Word 97 мәтіңдік редакторы пайдаланатын Символдар программасьн қараңыз);

OLE (Object Linking and Embedding - Объектіаерді байланыстыру және сіңдіру) - DDE технологиясының дамытылған түрі. Delphi7-де BDE мен ODBC- ден басқа кеңірек қарастырылып отырған ADO механизмі қолданылады.

OLE DB құрылымы.

OLE DB құрылымында, деректерге қатынайтын бағдарлама деректер тұтынушысы деп аталады (мысалы, Excel), ал деректерге қатынауға мүмкіндік беретін бағдарлама дерекқор жеткізушісі деп аталады (мысалы, Microsoft OLE DB Provider for SQL Server).

Қосылым мәліметтерін анықтау.

Әмбебап деректерді беру арнасы файлы (.udl) деректер көзіне сол деректер көзінің OLE DB жеткізушісі арқылы қатынау үшін деректер тұтынушысы пайдаланатын қосылым мәліметтерін қамтиды. Қосылым мәліметтерін төмендегі әрекеттердің бірін орындау арқылы жасауға болады:

Деректерді қосу шеберінде OLE DB жеткізушісі үшін деректерді беру арнасын анықтау үшін Деректерді беру арнасының сипаттары тілқатысу терезесін пайдаланыңыз. Қосымша ақпаратты Деректерді қосу шеберінің көмегімен деректерді импорттау бөлімінен қараңыз.

Файл кеңейтімі .udl бар бос мәтіндік файл жасап, Деректерді беру арнасының сипаттары тілқатысу терезесін көрсететін файлды өңдеңіз.

**18. "Көпке-көп" байланысы**

"Көпке-көп" байланысы (М:М) негізгі кестенің бірнеше жазбасьша қосымша кестенің бірнеше жазбасы сәйкес болған жағдайда орын алады. "көпке-көп" байланыстағы оку топтарындағы пәндер мен сабақжүргізетін окзытушылар арасындағы катыстар көрсетілген: әрбір топка бірнеше окщушы сабақбереді;әрбір оқытушы біріншіден, бірнеше пәндерді, екіншіден, әртүрлі топтарда сабақ беруі мүмкін.Grpred кестесінің бірінші және үшінші жазбаларына Pred\_prep кестесінің бірінші жазбасы, ал төртінші жазбасына— екінші және төртінші жазбалары сәйкес келеді. Бұл кестелерді Pred жиектері бойынша біріктіру арқылы әр топтағы пәндерді жүргізетін оқытушылар туралы мәліметгерден тұратьш Gr\_pred\_prep кестесін алуға болады .

Мәліметтер қорын жобалау барысында бірнеше кесте өзара байланыста болып, бір кесте калған кестелермен әртүрлі қатыста болуы мүмкін. Байланыста тұрған кесте басқа кестелермен тағы да байланысқан жағдайда кестелердің иерархиясы түзіледі.

**19. Мәліметтер қорына тиесілі технологияларға шолу.**

CORBA (Common Object Request Broker Architecture) таратылған объекттерді құрудың ең тиімді технологиясы болып табылады. Бұл технология инфрақұрылымнан, желілік протоколдардан, операциялық жүйелер мен программалау тілдерінен тәуелсіз таратылған жүйелерді құруды ұсынады. CORBA технологиясына қатысты үш түсінік қолданылады:

әрекеттесуге қатысатын барлық объекттердің интерфейстерін сипаттайтын OMGIDL (Interface Definition Language) интерфейстерді сипаттау тілі;

жүйенің аралық звеносы болып табылатын және клиенттік қосымшаның серверлікобъекттермен әрекеттесу логикасынан тұратын ORB (Object Request Broker) объектті сұраныстар брокері;

клиенттің сұраныстары мен сервердің жауабын транспорттық қолдауды және ORB арасындағы әрекеттесуді қамтамасыз ететін IIOP (Internet Inter-ORB Protocol) протоколы.

Объектілерді ендіру және байланыстыру технологиясы (Технология связывания и внедрения объектов; object linking and embedding (OLE) — Microsoft фирмасы жасаған қолданбалы программалар арасында объектілер алмасу технологиясы. LE 2.0 (Object Linking and Embedding – объектілерді байланыстыру және ендіру) – қосымша бағдарламалардың интеграциясының ережелерін сипаттайтын стандарт. Access бағдарламасының бұрынғы нұсқалары суреттер мен құжаттарды сақтау үшін OLE деп аталатын технологияны қолданған.

ADO (ағылшыннан аударғанда ActiveX Data Objects - «ActiveX мәліметтерінің объектісі») – Microsoft компаниясы жасап шығарған, ActiveX технологиясына негізделген интерфейс бағдарламасы. ADO бағдарламасы әртүрлі файлдарды объектіге бағытталған түрде көрсетеді

ADO технологиясы және OLE DB-ның интерфейстері (1 сурет) әр түрлі дерек көздеріне қатынауға рұқсат беретін біртұтас қосымшалары үшiн қамтамасыз етедi. Мысалы, ADO қолданушы қосымшасы бірдей күрделі операциялар мен электрондық кестелерге жергiлiктi ДҚБЖ-да SQL-дың бірлескен серверінде және сақталған мәліметтерге қолдана алады. SQL кез-келген дерек көзіне ADO арқылы бағытталған сұрау салуды орындайды.

**20. Мәліметтер қорын жобалау кезеңдері.**

Қазіргі кезде адам өмірінде әртүрлі ақпараттар өңдеу көп. Сондықтан оларды басқару және ұйымдастыру үшін кең көлемдегі мәліметтер базасымен қорын құруға тура келеді. Қазіргі кездегі кез келген есепт ер ақпараттар мен мәліметтерді түрлендірумен байланысты.

Мәліметтер базасы мағынасына байланысты электронды картотека түрінде мәліметтердің электрондық қоймасы түрінде компьютерлерде бір немесе бірнеше файл түрінде сақталады. Сондықтан мәліметтер базасының жұмыс атқару барысында бірнеше амалдарды орындауға тура келеді:

• Құрылған немесе бар мәліметтер базасының файлынан жаңа ақпараттар қосу;

• Мәліметтер базасына жаңа бос файлды қосу;

• Мәліметтер базасындағы файлындағы ақпаратты өзгерту (модификациялау);

• Мәліметтер базасынан ақпаратты іздестіру;

• Мәліметтер базасындағы бар файлындағы ақпаратты жою;

• Мәліметтер базасынан файлдарды өшіру;

Мысал ретінде:

Қала университет факультет студент

Алматы ату фииит Адина

Кестедегі әр жолды жазба түрінде қарастыру керек. Сондықтан мұндай ақпарат сәйкес өрістерге толтырылады. Сонымен қатар барлық жазбалар бірдей өрістерден тұрады.

Мәліметтер базасының негізгі қызметі ең алдымен ондағы ақпараттарды жылдам іздестіру болып табылады. Ақпаратты ұйымдастыру үшін және оларды кестеге орналастыру үшін көптеген программалар бар, ондай программалар мәліметтер базасын басқару жүйесі деп аталады.

МББЖ–нің негізгі ерекшелігі бұл тек қана мәліметтерді енгізу және сақтау құралы ғана емес сонымен қатар олардың құрылымын сипатайтын құрал.

МББЖ–нің қызметіне келесілер жатады:

• МБ–да мәліметтерді тікелей басқару:

Бұл қызмет МБ–на тікелей енетін барлық мәліметтердің сақталуын қамтамасыз етеді және МБ–ң жұмысын және қызметтік ақпаратын қамтамасыз етеді.

•МБ–ң тілдерін қолдау:

Бұл МБ–мен жұмыс істеу үшін арнайы тілдер МБ тілдер деп аталады.

**21.Логикалық модельдер. .**

Концептуалды модель кейін таңдалған ДББЖ сәйкес келетін деректер моделіне айналады. ДББЖ қамтамасыз ете алатын концептуалды моделдің версиясы логикалық модель деп аталады. Пайдаланушыларға бұл логикалық моделдің сыртқы моделдер деп аталатын құраушылары бөлінеді. Логикалық модель физикалық жадыға ауысады.

Жоғарыда көрсетілген моделдерге сәйкес деректер базасын ұйымдастырудың үш деңгейлі жүйесі бар. Оны стандартизация бойынша американдық комитет ANSI (American National Standards Institute) ұсынды.

Бұл архитектура деректермен жұмыс істеу барысында логикалық (1 мен 2 деңгейлер арасында) және физикалық ( 2 мен 3 деңгейлер арасында) тәуелсіздікті қамтамасыз ете алады. Логикалық тәуелсіздік берілген деректер базасымен жұмыс істейтін приложенияны басқа приложенияларды өзгертпей өзгертуге мүмкіндік береді. Логика (гр. λογική - «талдауға құрылған», λόγος — «сөз», «сөйлем», «ойлау», «ақыл») — ойлау, оның формалары мен заңдылықтары туралы ғылым. Логика дәлелдеу мен теріске шығарудың белгілі бір әдіс-тәсілдері қаралатын ғылым теориялар жиынтығын құрайды. Логикалық модель пән саласының түсінігін, олардың өзарабайланысын, сонымен қатар пән саласымен қойылатын мәліметтерге шектеулер қояды. Логикалық моделдің негізгі құралы ER-диаграмма (Entity-Relationship, мән – байланыс диаграммасы).

**22. Мәліметтер қорына OLE нысандарын сақтау.**

Windows-тың тағы бір ерекшелігі — Windows ортасында жұмыс істейтін түрлі қосымшалардың өзара әрекеттестігі. Мысалы, алмастыру буфері арқьлы не өз терезесін пайдаланып бір қосымшаньң берілгеңдерін екіншісіне ауыстыруға болады.OLE (Object Linking and Embedding - Объектіаерді байланыстыру және сіңдіру) - DDE технологиясының дамытылған түрі. Алғашқы рет енгізілген DDE технологиясыңда көшірілетін объект бір типті және ол файлдар түріндегі объектілерді байланыстыратып (DDE {Dynamic Data Exchange — беріпгендерді динамикалык, турде алмастыру) технологиясы — бір программаньщ жұмыс істеу барысында (динамикалық түрде) кейбір қажетті мәліметгерді басқа программадан алып, объект құру әдісі). Windows-ты пайдаланып, суретті графикалық редактордан мәтіндік редакторға кірістіруге болады. Мұңда графикалық редактор сервер, ал мәтіндік редактор клиент қызметін атқарады. Бірақ, кез келген қосымша программа объект дайындай алмайды, кейбір қосымшаның тек сервер болуы, объектіні дайьшдау мүмкіңдігі жоқ болуы да мүмкін. Мысалы, Word 97 редакторы пайдаланатын Снмволдар программасы мен Формулалар редакторы тек серверлер.OLE технологиясының ерекшеліктерінің бірі — Жылжыту және Сол жерде қалдыру (Drag — & — Drop) принципі. Алғашқы Windows 1.X, З.Х нұсқаларыңда программалар топтарының, түрлі каталоггар мен жеке программалардың арнайы белгілемелері (Icon) болатын. Windows 95-те каталоггар орнынаа бумалар (папкалар), файлдық белгілемелер орнына кішкене төртбұрыш сияқты белгішелер пайдаланылған. Барлық программалар мен берілгеңдер файлдары, типті логакалық дискілер де осындай бумадар ішіңде сақталады. Windows терезесінде көрінген буманы ашу үшін оны тышқан түймесімен 2 рет шерту (Click) жеткілікті.

**23. Сұраныстар тілі – SQL. SQL тілінің стандарты.**

Structured Query Language реляциялық ДҚБЖдардың мәлiметтерiнiң басқаруы үшiн қолданылатын процедуралық емес тiл болады.

SQL автономды қолданылмайды,әдетте ол орнатылған тілдер арасында СУБД-ны бағдарламалау(мысалы,Visual Basic for Application СУБД Access ) SQL орнатылған.

Орнатылған SQL-ді негізгі 2 әдіс бойынша ажыратады:статистикалық және динамикалық.

Статистикалық тілді қолданғанда бағдарламаның мәтіні SQL тілін шақыру функциясы орындалып жатқан модельге компиляциядан кейін қатаң қосылады.

IBM жобасын зерттеу нәтижесінде SQL тiлі 70-шi жылдары реляциялық мәлiметтердiң манипуляцияның тiлiнiң жасауы болатын. Ол бастапқы (Structured English Query Language ) SEQUEL деп аталды, содан соң - SEQUEL/2, содан соң - жай ғана SQL. SQL ресми түрде стандарты (American National Standards Institute - стандартизацияның Ұлттық институты, АҚШ) ANSIда 1986 жылы жариялаған. Осы стандарт 1989 және 1992 жылдары кеңiтілген сондықтан SQLның соңғы стандарты SQL92дiң атауын таныс. Қазіргі кезде объективтiк – орентиралық кеңейтілуі бар, SQL3 тіліне жұмыстар жүргізіліп жатыр. кейбiр -хабар кеңейту болатын стандартының үстiнде жұмысты дәл қазiр апарады.

ANSIдің стандарты үш деңгейі бар: бастапқы, аралық және толық. Серверлiк ДҚБЖдардың көп өндiрушiлерi, IBM, Informix, Microsoft, Oracle және Sybase, өздерін реализациялық тіжірибелерін қолданып, берілген ДҚБЖ үшін ANSI стандартын негіздеп және кейбiр кеңейтілулер негiзделген меншiктi iске асырулары ерекше қолданады.

**24. Инфологиялық және даталогиялық тәсілдің қасиеті. (айырмашылықтары)**

Инфологиялық (семантикалық) деректер моделдері жобалаудың алғашқы кезеңдерінде приложенияны жасау процесінеде қолданылатын ДББЖ байланыссыз деректердің құрылымын суреттеу үшін қолданылады.

Даталогиялық моделдер нақты ДББЖ қодлауды талап етеді.

* Документалды моделдер негізінен құжаттардың, текстердің әдеттегі тілмен еркін құрылуына негізделген. Құжаттарды белгілеу тілдеріне негізделген моделдер стандартты жалпы белгілеу тілмен SMGL (Standart Generalised Markup Language) байланысты.
* Фактографиялық реляциондық моделдерде пәндік облыстың абстрактілік моделдерін және объектілер арасында семантикалық бпайланыстарды көрсететін қатынас схемалары құрылады. Талдаудың негізі қатынас атрибуттары арасындағы функционалдық байланыс болып табылады.
* Даталогиялық модельдер ішінде ең қарапайым модельдерінің бірі – иерархиялық модель болып табылады және сол модельдің ішінде ең бірінші пайда болған иерархиялық модельде деректер бұтақ тәріздес құрылым арқылы көрсетіледі.

Пәндік облыстағы зат арасындағы байланыстар, инфологиялық  модельде сипаталады , даталогиялық модельде инфологиялық элементердің бірге орналасуына байланысты суретелуі немесе олардың арасындағы байланыстың жариялауында. Байланыс іштей жазу сырттай жазу деңгейде де берілуі мүмкін.

Даталогиялық жобалау мәліметтер базасының  логикалық құрылымын жобалау болып МББЖ – мен ұсынылатын мәліметтерді физикалық мүмкіндіктері әсер етеді. Сондықтан да, мәліметтерді физикалық ұйымдастыру ерекшеліктерін білү логикалық құрылымды жобалау барысында пайдалы болып табылады.

**25. OLE DB (Object Linking және Embedding Database) технологияларына түсінік.**

Microsoft SQL Server, сіз жұмыстап отырғанда мәліметтер қоры үшін OLE DB драйверіне ODBC бетінде орналасады. Әлдеқашан анықталған OLE DB сыртқы деректер көзіне қосылу үшін, деректерді қосу шеберін қолдануыңызға болады. Деректерді қосу шеберін ашу үшін, **Деректер**қойындысындағы **Сыртқы деректерді алу** тобында **Басқа көздерден**түймешігін басып, **Деректерді қосу шеберінен**параметрін таңдаңыз.Деректерді қосу шеберінде **Басқа/қосымша** деректер көзі параметрі таңдалса, **Деректермен байланыс сипаттары** тілқатысу терезесінде қол жетімді OLE DB жеткізушілерінің тізімін көруге болады. Оған қоса, ODBC драйверлеріне арналған Microsoft OLE DB жеткізушісі ODBC деректер көздеріне де қол жеткізуге мүмкіндік береді. Осы тілқатысу терезесіндегі әрбір қойындыны пайдалану туралы қосымша ақпарат алу үшін **Деректермен байланыс сипаттары** тілқатысу терезесіндегі **Анықтама** түймешігін басыңыз.

Жалпы, **Деректермен байланыс сипаттары** тілқатысу терезесінде қосылым мәліметтерін анықтау үшін төмендегі әрекеттерді орындаңыз:

* **Жеткізуші** қойындысын нұқып, OLE DB жеткізушісін таңдаңыз да, **Келесі** түймешігін басыңыз. Бұл кезде**Қосылым** қойындысы көрсетіледі, оның ішінде OLE DB жеткізушісі үшін нақты қосылым мәліметтерін енгізуге болады.

OLE DB құрылымы.

OLE DB құрылымында, деректерге қатынайтын бағдарлама деректер тұтынушысы деп аталады (мысалы, Excel), ал деректерге қатынауға мүмкіндік беретін бағдарлама дерекқор жеткізушісі деп аталады (мысалы, Microsoft OLE DB Provider for SQL Server).

Қосылым мәліметтерін анықтау.

**26. SELECT таңдау операторына толық түсінік.**

SQL тілі командалардан тұрады. SQL командасының әрекет ету объектісі бір кесте және де кестелер тобынан тұруы мүмкін.

Жолдарды таңдау сәйкестігіне байланысты бөледі, жолдар нәтижесінде берілген ретіне қарай сұрыптайды және топтастырады. SELECT командасының орындалуы берілген деректер базасындағы деректерді өзгертпейді.

Qbasic тілінде таңдау командасы  үшін арналған ыңғайлы, құрылымдық SELECT – CASE операторы бар. (Select – таңдау, Case – жағдай).

             Жазылу үлгісі:

      SELECT CASE <айнымалы>   
CASE<1-мән>: <1-блок>

CASE<2-мән>: <2-блок>

   END SELECT

Бұл оператор ойын құрүға рұқсат етеді.Егер бағаналардың аты ойында көрсетілмесе,онда сұрауда көрсетілген бағаналардың аты қолданылады, SELECT операторы арқылы жүзеге асырылады.

**27. SQL стандарты кезеңдері. (жалпы шолу)**

SQL тілі әр алуан болып келеді. Шын мәнісінде бұл программалау тілінің әр алуан реализациялауға жатады. (Мысалы Си тілі мен Borland компаниясының диалектерін салыстырып көріңіз) Стандартқа сәйкес келетін, кез келген компиляторда жүретін Халықаралық Си тілі бар. Ашық стандартты жүйемен кездесуі қолданбалы программаларды шығаруңа мүмкіндік береді.

1989ж алғашқы халықаралық SQL ( SQL-89) тілінің стандарты шыққаннан кейін, 1982ж екінші халықаралық SQL-92 стандарты қабылданды. Осыдан кейін МББЖ бейімделген SQL тілі туралы айтуға болатын болды. Кез келген SQL тілді қолдану үшін стандарт тілдерін білу қажет.

SQL тілінің стандарттау жұмысы оның ең алғашқы коммерциялық реализациясы басталғанмен тұспа–тұс келеді. Стандарт негізінде SQLSYSTEMR қолдануға тиым салынған еді. Оны реализациялау қиынға түскен еді.

1989ж қабылданған халықаралық SQL стандарттарының көп бөлімі бірдей болып келеді. Бұл стандартта МБ схемасын басқару және динамикалық SQL деген тарау мүлдем жоқ.

Осының негізінде SQL-2 стандартын шығару жұмысы басталған болатын. Бұл да быраз жылға созылып, ақырында 1992 ж наурыз айында біткен еді (оны енді SQL-92 деп атайды). Бұл стандарт толық және реализацияға қажет аспектілерді қамтиды: МБ схемасын басқару, транзакция мен сессияларды басқару (сессия басқару дегеніміз транзакцияның уақыт қатынастарының сақтала тізбектелуі), МБ қосылуы, динамикалық SQL.

SQL-2 стандартын аяқтай келе, SQL-3 стандартын шығару жұмысы басталып еді. SQL-3 триггерлер механизмі және абстрактілі мәліметтерді қолдану мүмкіндігін қамтиды.

**28. Мәліметтер қорының қосымшаларын өңдеу жолдарын атап көрсетіңіз.**

Мәліметтер қоры (МҚ) – қандай да пәндік аймақтағы ақиқат өмірдің  нақты объектілері туралы мағлұматтардың өзара байланыстағы жиынтығы.

Мәліметтер қорымен жұмыс істейтін бағдарламаны бірнеше интегралдау орталарында құруға болады. Мысалы: Delphi , Visual Studio, My SQL және т.б.

Мәліметтер қорын өндеуді тездету, әртүрлі деңгейдегі қолданушылардың тек өзіне қажетті деректермен ғана жұмыс істеуі арқылы олардың жұмысын жеңілдету, жылдамдату мақсатында кейде бір кестедегі мәліметтерді әдейі екі немесе одан да көп бірнеше кестелерге жіктеуге болады, мысалы, бір мекеменің жүргізіп жатқан ғылыми-зерттеу жұмыстары туралы мәліметтер қорын құру керек болсын: ғылыми жұмыстың тақырыбы, мақсаты, шифры, басталған күні, аяқталған күні, зерттеу кезендерінің саны, басты орындаушы, қосымша мәліметтер. Мәліметтер қорының орналасуы көптеген жағдайда осы қорда бар мәліметтерді өңдейтін қосымшалардыңдайындалуына әсер етеді. Осылайша, келесі қосымшалар түрлерін бөліп көрсетуге болады:

* Локальды мәліметтер қорын пайдаланатын қосымшаларды бір деңгейлі (бір буынды) қосымшалар деп атайды. Себебі, қосымша мен мәліметтер қоры біртұтас файлдық жүйені құрайды;
* Жойылған мәліметтер қорын пайдаланатын қосымшаларды екі деңгейлі (екі буынды) және көп деңгейлі (көп буынды) қосымшалар деп бөледі. Екі деңгейлі қосымшаларда клиенттік және серверлік бөлшектер бар;
* Көп деңгейлі (әдетте үш деңгейлі) қосымшалар клиенттік және серверлік бөлшектер мен қоса қосымша бөлшектерге ие. Мысалға үш деңгейлі қосымшаларда клиенттік бөлшек, қосымшалар сервері мен мәліметтер қорының сервері бар.

Бір және екі деңгейлі Delphi қосымшалары келесі механизмдерді қолдана отырып локальды және жойылған мәліметтер қорына қол жеткізе алады:

**29. Мәліметтер қорын жетілдіру операторлары.**

Мәліметтер қорларын бағдарламалық қамтамасыз ету жалпыға арналған қамтамасыз ету, қолданбалы бағдарламалық қамтамасыз ету және мәліметтер қорларын жобалау мен жасауды автоматтандыру құрал-жабдықтары болып бөлінеді. SQL тілінің операторларын 2 шартты тілшеге бөлеміз: мәліметтерді анықтау тілі ( DDL – Data Definition Language ) және мәліметтерді басқару тілі ( DML – Data Manipulation Language). SQL тілі командалар (операторлар) жүйелерінен тұрады, олардың ішіндегі ең маңыздылары төмендегілер:

DML (Date Manipulation Language), мәліметтерді манипуляциялау тілі.

SELECT – мәліметтерді оқуға сұраныс жасау;

INSERT – мәліметтер қорына жаңа жазба қосу;

DELETE – мәліметтер қорынан жазбаларды жою;

UPDATE – мәліметтерді өзгерту.

DDL (Date Definition Language), мәліметтерді анықтау тілі.

CREATE TABLE – жаңа кестені жасау, оның құрылымын сипаттау;

DROP TABLE – кестені жою;

ALTER TABLE – кесте құрылымын өзгерту;

CREATE VIEW – бейнелеуді жасау;

DROP VIEW – бейнелеуді жою;

CREATE INDEX – индексті жасау;

DROP INDEX – индексті жою;

Қол жеткізуді басқару.

GRANT – пайдаланушыға артықшылықтар беру;

REVOKE – берілген артықшылықтарды алып тастау.

Транзакцияларды басқару.

COMMIT – транзакцияны аяқтайды;

ROLLBACK – транзакцияны алып тастайды.

SQL командаларын интерактивті режимде орындаумен қатар, жоғары деңгейдегі тілдерде бағдарламалауда пайдалануға болады.

**30. SQL-дің курсор арқылы динамикалық операторларымен жұмыс істеу тәсілдері.**

SQL тілін тиімді жұмыс істеу үшін курсор деген мағынаны енгіземіз**. Курсор дегеніміз** нұсқау сияқты,өңдеу кезінде жазулар жинақтарын орын ауыстыруын қолданылады.

Курсордың сипаттамасы мен қолданылуы келесідей орындалады. Бағдарламаны сипаттау бөлігінде ауыспалы түрлі курсор байланыс қызметін атқарады(CURSOR) SQL операторымен (әдетте SELECT операторымен).Бағдарламаның атқару бөлігінде курсордың ашылуы болады (OPEN <курсордың аты>), жазулар бойынша курсордың жылжуы(FETCH<курсордың аты>), қатысты өңдеумен қатар жүретін, (CLOSE<курсордың аты>). Орнатылған SQL-ді негізгі 2 әдіс бойынша ажыратады:статистикалық және динамикалық.

Статистикалық тілді қолданғанда бағдарламаның мәтіні SQL тілін шақыру функциясы орындалып жатқан модельге компиляциядан кейін қатаң қосылады.

Динамикалық тілді қолғанда динамикалық шақыруларды орнату ұйғарылады. SQL функциясында және осы шақырулардың интерпретациясы, мысалы,өшірілген мәліметтер базасын үндеу,осы бағдарламаны орындау кезінде болады.Бұл әдіс мынандай жағдайда қолданылады,егер қосымшалар алдын-ала белгісіз болғанда SQL шақыру түрі және қолданушы арасында диалог құрылады.

SELECT деген сөз кілтінен басталатын 3 түрлі синтаксистік конструкцияның 3 түрін ұсынады олар: Курсор спецификациясы (cursor specification), Таңдау операторы (select statement),(sub query)

**31.SQL тілін қосымша өңдеуде қолдану жолдары**

Программалау тіліне SQL операторын енгізуге SQLSystemR-де арнайы оператор бар. SQL-ң енгізудегі негізгі кедергі SQL тілі - реляциондық тіл болуы. SQL тілін тиімді жұмыс істеу үшін курсор деген мағынаны енгіземіз**. Курсор дегеніміз** нұсқау сияқты**,өңдеу** кезінде жазулар жинақтарын орын ауыстыруын қолданылады.

Курсордың сипаттамасы мен қолданылуы келесідей орындалады. Бағдарламаны сипаттау бөлігінде ауыспалы түрлі курсор байланыс қызметін атқарады(CURSOR) SQL операторымен (әдетте SELECT операторымен).Бағдарламаның атқару бөлігінде курсордың ашылуы болады (OPEN <курсордың аты>), жазулар бойынша курсордың жылжуы(FETCH<курсордың аты>), қатысты өңдеумен қатар жүретін, (CLOSE<курсордың аты>).

**32. SQL тілінің жалпы құрылымы мен қолданбалы программалармен интерфейсі.**

SQL тілі – компьютердің деректер қорында сақталатын ақпараттарды таңдауға және өңдеуге арналған құрал. SQL (Structured Query Language) – құрылымдалған сұраныстар тілінің аббревиатурасы. SQL аббревиатурасы әдетте «сиквел» деп оқылады, бірақ «эскюэль» альтернативті түрде айтылуы да қолданылады. SQL – қолданушының ДҚ өзара қатынасын ұйымдатыру үшін қолданылатын программалау тілі. Деректер қоры (ДҚ) – ұйымның ақпараттық қажеттіліктерін қанағаттандыру үшін арналған логикалы байланысқан деректердің жиынтығы мен оның сипатталуы. SQL тілінің құрылымдық сұрауы реляционды есептеулер мен ауыспалы кортеждер тілі бірнеше стандарттары бар,көбінесе қолданылатын олар SQL-89 және SQL-92.

Көптеген SQL тілінің кішігірім сипаттамасы стандартты интерфейсте орындалады.ODBC ( Open Database Connectivity – ашық мәліметтер базасы мен қосылуы) Microsoft фирмасы.    Обьектілер функциясын шақыру үшін арнайы интерфейс - диспинтерфейс(dipinterface) түсінігін енгізеді. Бұл технологияны, мысалы Microsoft Excel  қолдайды, ол басқа қосымшаларға өз қызметтерін ұсыны алады.

**33. Мәліметтер тұтастығының негізгі ережелері.**

Мәліметтер базасын құру себебі: мәліметтерді өдеу уақыты ықшамдау, мәліметтерді сақтауды артықшылықтары қысқарту, мәліметтерді қарама-қайшылықтары жою, тапсырысты көпқолданушылық режимде шешу, тапсырысты шешуші программалардан мәліметтерді тәуелсіздігі, мәліметтер базасында ақпаратты сақтау сенімділігі және т.с.с.

Мәліметтер базасымен жұмыс істеу үшін арнайы мәліметтерді басқару программалық құралдар қажет, олар мәліметтер базасында орналасады немесе оларды мәліметтер базасын басқару жүйесі деп атайды. Мәліметтер базасын басқару жүйесі- жалпыжүйелік кешенді программаға енгізу мен шығаруды, қатынауды, ақпараттық массивтерді түзетуді, АЖО-ны қолданушыларына қосымша тапсырыстағы мәліметтерге қол жеткізуді қамтамасыз етеді.

АЖО-ны құруда МББЖ InterBase мәліметтер базасын басқару жүйесі қолданамыз, ол келесі талаптарға сәйкес болуы тиісті:

o “клиент-сервер” архитектурасында жұмыс істеу керек;

o SQL-ді негізгі стандарттар тіліні сұраныстары қамтиды;

o Бірнеше қолданушылар бір уақытта жұмыс істеуі қамтамасыз кету;

o Ақпаратты қорғау құралдарыны жетілдірген түрлеріні бар болуы;

o Ішкі администрациялық құралдары бар және қолданушыларды қатынау дегейлері мен құқықтарын басқарады;

**34. Қондырылған SQL мен динамикалық SQL-дің айырмашылығы**

Динамикалық тілді қолғанда динамикалық шақыруларды орнату ұйғарылады. SQL функциясында және осы шақырулардың интерпретациясы, мысалы,өшірілген мәліметтер базасын үндеу,осы бағдарламаны орындау кезінде болады.Бұл әдіс мынандай жағдайда қолданылады,егер қосымшалар алдын-ала белгісіз болғанда SQL шақыру түрі және қолданушы арасында диалог құрылады. Орнатылған SQL-ді негізгі 2 әдіс бойынша ажыратады:статистикалық және динамикалық.

Сұраудың ашылуы

DoCmd.OpenQuery qdf.Name End Sub

OpenRecordset әдісі Recordset сияқты объектілерді ашу үшін және оларға амалдар қолдану үшін қолданылады.

Келесі процедурада SQL нұсқауы көмегімен динамикалық жазулар терімі сияқты Recordset объекті ашылады.Сөйлемде WHERE нұсқаулар SQL Year функциясы қосылған, бұл 1988 ж орнатылған тапсырыстар таңдауын анықтайды.

SubOrders98()

Dim dbs As Database, rst As Recordset, strSQL As String

Dim fid As Field

Setdbs = CurrentDb

strSQL = "SELECT DISTINCTROW Тапсырыс, Орнату мерзімі " &

"FROM Тапсырыстар WHERE ((Уеаг([Орнату мерзімі])=1998));"

Set rst = dbs.OpenRecordset(strSQL, dbOpenDynaset)

rst.MoveLast

Debug. Print rst.RecordCount

End Sub

**Мәліметтерді сорттау.** Компьютер жадына екілік формадағы сандарды ғана жазуға болады, оларды экранға немесе баспаға шығарғанда ондық сан  түрінде болады. **Қондырылған** бағдарламалар көмегімен сандарды өсу немесе кему ретімен орналастыруға болады. Компьютер жадындағы символдар сандық кодтар түрінде көрсетілгендіктен, олар да өсу немесе кему реті бойынша сортталады, яғни экранда олар алфвит бойынша немесе алфавитке кері орналасатын болады. Кез келген өрісті ерекшелеп, **Записи – Сортировка – По возрастанию** (немесе **По убыванию**) командаларын орындаңыз.

**35. Мәліметтер қорының тұтастығы және қауіпсіздігі.**

Жүйе қауіпсіздігін жүзеге асыруға ДҚБЖ жауапты. SQL тілі реляциялық ДҚБЖ қауіпсіздік жүйесінің іргетасы болып табылады: деректер қорындағы ақпаратты қорғау жүйесіне қойылатын талаптар SQL инструкциясының көмегімен құрылады. SQL-де деректерді қорғаумен негізгі үш концепция байланысқан:

* Қолданушылар деректер қорындағы негізгі жұмыс істейтіндер болып табылады. ДҚБЖ деректерді алғанда, қойғанда, жойғанда немесе өңдегенде әр уақытта қандай да бір қолданушы атынан жасайды. ДҚБЖ талап еткен әрекетті қай қолданушы сұрайтынына байланысты қабылдайды немесе орындамайды.
* Деректер қорының объектілері қорғанысы SQL көмегімен жүзеге асатын элементтер болып табылады. Әдетте, ұсыну және кесте қорғанысы қамтамасыз етіледі, бірақ басқа объектілер: формалар, қолданбалы программалар және толық деректер қоры қорғанысқа алынады. Көптеген қолданушыларға деректер қорының бір объектісін қолдануға рұқсат етіледі және басқаларын қолдануға тыйым салынады.
* Артықшылықтар – бұл қолданушының деректер қорындағы анықталған объектіге кейбір әрекет ету құқығы. Қолданушыға кейбір кестеден жолдарды алуға және оларды оған қосуға рұқсат етіледі, бірақ бұл кесте жолын жоюға немесе жаңартуға тыйым салынады. Басқа қолданушыда артықшылықтың басқа жиынтығы болуы мүмкін.

**Қауіпсіздік жүйесі сервер деңгейінде келесі түсініктерді басқарады:**

* аутентификация;
* тіркелу жазбасы;
* орнатылған сервер ролі.

**Деректер қоры деңгейінде келесі түсініктер қолданылады:**

* деректер қорын қолданушы;
* деректер қорының белгіленген ролі;
* деректер қорының қолданушылық ролі.
* SQL Server қолданушылар аутентификациясының екі тәртібін ұсынады:
* Windows NT/2000 құралдарымен аутентификация тәртібі;
* аралас аутентификация тәртібі (Windows NT Authentication and SQL Server Authentication).

o Ақпарат жүйені талабына сай және толық болуы керек;

**36. Таңдау операторы SELECT.**

SQL тілі командалардан тұрады. SQL командасының әрекет ету объектісі бір кесте және де кестелер тобынан тұруы мүмкін.

Жолдарды таңдау сәйкестігіне байланысты бөледі, жолдар нәтижесінде берілген ретіне қарай сұрыптайды және топтастырады. SELECT командасының орындалуы берілген деректер базасындағы деректерді өзгертпейді.

Qbasic тілінде таңдау командасы  үшін арналған ыңғайлы, құрылымдық SELECT – CASE операторы бар. (Select – таңдау, Case – жағдай).

             Жазылу үлгісі:

      SELECT CASE <айнымалы>   
CASE<1-мән>: <1-блок>

CASE<2-мән>: <2-блок>

   END SELECT

Бұл оператор ойын құрүға рұқсат етеді.Егер бағаналардың аты ойында көрсетілмесе,онда сұрауда көрсетілген бағаналардың аты қолданылады, SELECT операторы арқылы жүзеге асырылады.

**37. "Бірге-бір" байланысы**

Жоғарыда айтылғандай МҚ-ның екі кестесінің арасында бір-біріне тәуелділік қатысы болуы мүмкін; өзара байланыстағы екі кестенің біреуі *master—* басыңқы (негізгі, аналық), екіншісі *de­tail —* бағыныңқы (қосымша, балалык) кесте болып табылады. Реляциялық МҚ-да қатыстардың байланысының негізгі үш түрі кездеседі (2-кесте):

- *"бірге-бір " (1:1) — one-to-one relationship;*

*-"бірге-көп" (1:М) — one-to-many relationship немесе "көпке-бір"(М:1);*

*-"көпке-көп" (М:М немесе M:N) —many-to-many relationship.*

Егер Customer кестесіндегі әрбір клиенттің ешқандай немесе тек қана бір тапсырыс жасау мүмкіндігі болса, онда Customer және Order кестелері "бірге-бір" қатысымен байланыста болады. *"Бірге-бір" байланысы* (1:1) екі кестенің де байланыстырушы жиектері олардың кілті болған жағдайда орын алады, сондықган бұл кестелердің өзара дәрежелері бірдей. Бұл байланыс МҚ-да сирек қолданылады, себебі мұндай кестелердегі мәліметтерді ЭЕМ жадын тиімді пайдалану мақсатында бір кестеге біріктіруге болады.

**38. Желілік деректерді моделдеу.**

**Теоретико-графалық модельдерге иерархиялық және желілік модельдер** жатады. Бұл модельдер бұрыннан пайда болған, ал қазіргі таңда олар сирек қолданылады. Алайда бүгінгі күнге дейін осы негізде жұмыс істейтін модельдердің жүйесі бар (Мысалы, желілік- FoxBase + FoxPro немесе иерархиялық- IMS/VS). Желілік құрылымда алдыңғы аталған негізгі ұғымдардағы (деңгей, түйін, байланыс) әр элемент кез-келген басқа элементпен байланыста бола алады.

**Желілік деректер моделі.**

Бұл моделде деректер еркін граф түрінде көрсетіледі және элементтерінің

әртүрлі өзара байланысын көрсетуге мүмкіндік береді.

Осылайша, егер деректердің құрылымы күрделі болса иерархияға қарағанда, онда иерархиялық деректер базасының құрылымы оның кемшілігі болып табылады. Мысалы, кафедра деректер базасында бір пән барлық әртүрлі қатынастарда, оның тобымен байланысында қатыса алады. Пән

**Гибкость** – көптеген қатынасты предок/потомок желілік деректер базасында деректерді сақтау, жай иерархияға қарағанда күрделі құрылымдарды сақтауға мүмкіндік береді.

**Стандартизация** – желілік модельдің CODASYL стандарттың пайда болуы және Digital Equipment Corporation және Data General сияқты кіші-компьютер жабдықтаушылары желілік СУБД құрды.

**39.Иерархиялық деректерді моделдеу .**

**Теоретико-графалық модельдерге иерархиялық және желілік модельдер** жатады. Бұл модельдер бұрыннан пайда болған, ал қазіргі таңда олар сирек қолданылады. Алайда бүгінгі күнге дейін осы негізде жұмыс істейтін модельдердің жүйесі бар (Мысалы, желілік- FoxBase + FoxPro немесе иерархиялық- IMS/VS). Иерархиялық байланыстағы объектілер бағытталған граф (төңкерілген ағаш) құрады

**Иерархиялық деректер моделі.**

Даталогиялық модельдер ішінде ең қарапайым модельдерінің бірі – иерархиялық модель болып табылады және сол модельдің ішінде ең бірінші пайда болған иерархиялық модельде деректер бұтақ тәріздес құрылым арқылы көрсетіледі.

Әрбір бұтақ тәріздестің тамырлық түрі бар. Әрбір элементарлық түрдің соның ішінде бұтақ тәріздес түрдлер жай немесе күрделі жазба болып бөлінеді. Жай жазбалар бір түрден тұрады (Мысалы, сандық). Ал күрделі жазба бірнеше түрдің жиынтығын қосады.

**Тамырлық түр дегеніміз** - өзі бағынбайтын, өзіне бағынатын түрді айтамыз.

Иерархиялық түрдің СУБД санына FC/Focus, Team-Up, Data Edge жатқызуға болады. Бұл модельде әрбір деректер базасындағы жазба нақты бөлікті көрсетеді. Жазбаның арасында әрбір бөлімді бөліктермен байланыстыратын Предоктың / Потомокқа қатынасы бар. Деректер базасында қамтылған деректерді алу үшін мыналар қажет:

* Кафедрадан группаны табу
* Бірінші потомокқа төмен өту
* Предокқа жоғары өту
* Басұқа потомокқа жанына өту

Осылайша иерархиялық деректер базасында деректерді оқу үшін жазбаларды бір рет және бір жазбаға жоғары немесе төмен өтетін жазбалардың араласуын қажет етеді.