**MS Access ДББЖ байланыстыр** Access ДББЖ-не көптеген қосымша сервистiк мүмкiндiктер кiрiстiрiлген. MS Access ДББЖ-дегi Деректер базасы ақпаратты өңдеу, сақтау, басқару, тарату iстерiн жылдам әрi көрнекi орындайтын құралдар жиынтығын құрайды..          MS Access жүйесi мәлiметтермен жұмыс ұйымдастыруға мүмкiндiк беретiн кесте, форма, сұраныс, отчет объектiлерiнен тұрады. Сонымен қатар Access-тi қосымшалар дайындау ортасы деп қарауға да болады, ол макростар көмегiмен жүзеге асырылады. Макростар пайдаланушыға Access VBA тiлiн пайдаланбай-ақ, яғни Basic тiлiнде программалау бiлмесе де көптеген iс-әрекеттердi автоматтандыруға мүмкiндiк бередi MS Access құрамына жұмысты жеңiлдететiн жүздеген шеберлер кiредi. Access-тегi шеберлер дайын үлгiлерге сүйенiп  стандарт түрдегi кесте, форма, отчет, сұраныс, графиктер, пошталық жапсырмалар (наклейка) мен қосымшаларды  жобалауға және құруға мүмкiндiк бередi. Конструкторлар дайын кесте, формаларды түрлендiруге, күрделендiруге арналған және олар стандарт емес құжаттар дайындауға мүмкiндiк бередi. MS Access ДББЖ-мен жұмыс барысында анықтамалық жүйесiн қолдануға болады. нықтамалықты мәзiр арқылы да, тәуелдi мәзiр көмегiмен де алу мүмкiндiгi бар. Сонымен қатар қажет кезде жылдам шақыруға болатын Көмекшiнi де пайдалануға болады. MS Access ДББЖ-де Деректер базасының құрамына енетiн кесте, форма, отчет, сұраныс және басқа объектiлер бiр файлда сақталады. Деректер базасының негiзгi құрылымдық компонентi кестелер болып табылады. Кестелер өрiстер деп аталатын бағандар мен жазба деп аталатын қатарлардан (жолдар) тұрады. Кестенiң әрбiр жазбасы Деректер базасының жеке элементi туралы ақпаратты сақтайды. Мысалы, оқытушы туралы жазба оның фамилиясы, аты, әкесәнi аты, туған күнi, қызметi, мекен-жайы және т.б. мәлiметтерден тұруы мүмкiн. Кесте құрылымын анықтау барысында кесте өрiстерi, өрiс типтерi және өрiс өлшемдерi анықталады. Бiр кестеде аттас өрiстер болмауы керек, яғни өрiстер атаулары ерекше болуы тиiс. Өрiс атауы ондағы мәлiметтердi сипаттайтындай болғаны дұрыс. Әрбiр өрiске мiндеттi түрде сәйкестi типтер бекiтiледi.

**MS Access ДББЖ-де деректер базасының құрамына енетін кесте**, форма, отчет, сұраныс және басқа объектілер бір файлда сақталады. Деректер базасының негізгі құрылымдық компоненті кестелер болып табылады. Кестелер өрістер деп аталатын бағандар мен жазба деп аталатын қатарлардан (жолдар) тұрады. Кестенің әрбір жазбасы деректер базасының жеке элементі туралы ақпаратты сақтайды. Мысалы, оқытушы туралы жазба оның фамилиясы, аты, туған күні, қызметі, мекен-жайы және тағы басқа деректерден тұруы мүмкін. Кесте құрылымын анықтау барысында кесте өрістері, өріс типтері және өріс өлшемдері анықталады. Бір кестеде аттас өрістер болмауы керек, яғни өрістер атаулары ерекше болуы тиіс. Әрбір өріске міндетті түрде сәйкесті типтер бекітіледі. MS Access-те төмендегідей өріс типтері пайдаланылады:

|  |  |
| --- | --- |
| Деректер типтері | Сипатталуы |
| Текстілік (текстовый) | 255 символдан аспайтын текст немесе есептеуді қажет етпейтін сандар |
| Сандық (числовой) | Есептеу жүргізуге арналған әртүрлі форматтағы сандық деректер |
| Мерзімдік (дата/время) | 100 жыл мен 9999 жыл аралығындағы уақыт пен мерзімді сипаттайтын деректер |
| Ақшалық/қаражаттық (денежный) | Ақша, қаражатты сипаттайтын деректер, сандық өріске ұқсас, мұнда санның бөлшек бөлігіне автоматты түрде тек екі орын ғана бөлінеді. |
| МЕМО өріс | Ұзындығы 65535 символдан аспайтын түсініктеме (комментарий) сақтауға арналған. |
| Санағыш (счетчик) | Автоматты түрде жазбаның реттік номерін меншіктейтін, өзгермейтін арнайы сандық өріс |
| Логикалық | Ақиқат не жалған мәндерін ғана қабылдайтын өріс. |
| Ole объектілі өріс (поле объекта Ole) | Әртүрлі графиктік бейнелер, екілік кодтағы деректер, Access-ке кірістірілген текстілік құжат, диаграмма, электрондық кесте, аудио жазбалар сақтауға мүмкіндік беретін өріс |
| Гиперсілтеу (гиперссылка) | Әріптер мен цифрлардан тұратын және гиперсілтеу адресін сипаттайтын қатар. Адрес кем дегенде 3 бөліктен тұрады: өріске енгізілетін не басқару элементінде қолданылатын текст; сыртқы файлға сілтеме беретін жол, маршрут (UNC форматындағы жол); файл ішінде не парақтар (беттер) арасында көрсетілетін жол (URL адрес). Өріске гиперсілтеу адресін кірістіру үшін Вставка-Гиперссылка бұйрығы қолданылады. |
| Алмастыру шебері (мастер подстановок) | Тұрақты мәндерден не басқа кесте деректерінен тұратын өріс, тізім мәндерін таңдауға мүмкіндік беретін өріс құрады. Бұл типті өріс типі емес өрісті сақтау тәсілі деп қарауға болады. |

Access-те кесте құрудың 4 тәсілі қолданылады: - кесте режимі – бос кестені тікелей толтыру арқылы құру тәсілі, мұнда өрістерге атау беруде олардың типтерін Access автоматты түрде өзі анықтайды. - конструктор – конструктор режимінде кесте макетін, параметрлерін анықтауға мүмкіндік беретін тәсіл. - кесте шебері – дайын үлгі-шаблондарға сүйеніп кесте құрылымын, өрістерін анықтайтын тәсіл. - кестелерді импорттау – бұрын құрылған ДБ-сы не сыртқы файл кестелері, форма, отчеттарына сүйеніп, бір операцияда жаңадан ДБ құру тәсілі. Бұл тәсілді дайын ДБ-на жаңадан кесте, форма, отчет кірістіруге қолдануға болмайды. Кестелер арасында 4 түрлі байланыс орнауы мүмкін: біреуден біреуге (1→1), біреуден көпке (1→...), көптен біреуге (...→1), көптен көпке (...→...). Кестелер арасындағы байланыс көп жағдайда алғашқы кілт бекітілген өрістер негізінде орнатылады. Алғашқы кілт - жазбаларды ерекшелендіретін деректерді сақтайды, жалпы жағдайда алғашқы кілт деректері қайталанбайтын және өзгермейтін болғандықтан, санағыш типті өріске бекітіледі. Алғашқы кілтті басқа өрістерге де бекітуге болады. Алғашқы кілтті өрістерді индекстеуде де қолдануға болады, кілт бекітілген өріс автоматты түрде индекстеледі. Индекс – индекс құрамына кіретін өрістер мәні (өрнегі) мен индекстік өрнекке сәйкес келетін әрбір жазбаның кестедегі орны көрсетілген екі бағаннан тұратын ішкі кесте. Өрістерді индекстеу көпкестелік ДБ-нан деректерді жылдам іздеп табуға және реттеуге мүмкіндік береді. Ole және MEMO типті өрістер индекстелмейді.

**ER – диаграмма құру**  **Деректердің ER-моделі** «негіз-байланыс» моделі **(ER-модель)** (ағыл. *entity-relationship model*, ERM) — пәндік аймақтың концептуалды сұлбасын сипаттауға мүмкіндік беретін деректер моделі. ER-модель деректер қорының жоғарғы деңгейін (концептуалды) жобалағанда қолданылады. Оның көмегімен кілттік негізді оқшаулап және осы негіз арасында орналаса алатын байланыстарды белгілеуге болады.  Деректер қорын жобалау кезінде ER-моделі таңдалған деректер моделінің негізінде (реляционды, объектілі, желілік немесе т.б.) деректер қорының нақты сұлбаға түрлендіріледі. ER-моделі формальды конструкция болып табылады, ол өздігінен оның визуализациясында графиктік құрылғыларды жазбайды. Стандартты графиктік нотация ретінде ER-моделін визуализациялауға болатын «негіз-байланыс» диаграммасы ұсынылды **(ER-диаграмма)** (ағыл. *entity-relationship diagram*, ERD). ER-диаграмма түрінде пәндік аймақтың концептуалды моделін құру Енгізілетін түсініктер мен деректер қорының жобалау этаптарын нақты пәндік аймақтың мысалында қарастырамыз: ЖОО студенттері туралы деректерді көрсету. Қарастырылатын пәндік аймақтың қысқаша сипаттамасыг берейік. ЖОО-да бірнеше факультет бар, олардың әрқайсысында бірнеше пәндер мен бағыттар бойынша дайындық жүреді. Әр мамандық үшін факультетте сағат саны көрсетілген оқылатын піндер курсының тізімі көрсетілетін жеке жоспар болады. Студенттер сәйкес пәндерді оқиды, емтижан және сынақ тапсырады, баға алады. *ER-*диаграмманың құрылу үдерісі ER-модельдеу деп аталады. Пәндік аймақты сипаттауға болатын негізгі түсініктерді енгізейік. *Сущность* (Entity) немесе объект – ақпараттық жүйеде ақпарат жинақталады. Егер жүйеде факультет туралы ақпарат өңделсең онда факультет негіз болады, студенттер туралы болса, студент негіз және т.б. ER-модельдеу кезінде негіз атауы ереже бойынша бас әріппен жазылады. Әр негіз ақпараттық жүйеде сақталатын белгілі қасиеттер тізімінен тұрады (өткізілетін зерттелу шекарасында қолданушылар үшін тек қана қасиеттер қарастырылады). Мысалы, ФАКУЛЬТЕТ сущносттінің қасиеті ретінде факультет номерін, факультет атауын, СТУДЕНТ сущностінң қасиеті ретінде фамилияны, туылған мерзімін, туылған жерін, ал ЕМТИХАН сущностінің қасиеті ретінде пәнді, емтихан өткізілген уақытты, емтихан алушыны көрсетуге болады.

**SQL** (structured query language – «құрылымдық сұраныс тілі» ) ­­­­– формалдық бағдарламалық тіл, бірақ процедуралық бағдарламалауға жатпайды. Реляционнық деректер қорында  мәліметтерді құруға, өзгертуге және оларды басқаруға арналған. Арнайы деректер қорын басқару жүйесі (ДҚБЖ) арқылы басқарылады. Мәліметтермен манипуляция жасау операторлары (Data Manipulation Language, DML): [SELECT](http://www.ubukulov.kz/sql/select-operatory-turaly-maelimet.html) берілген шарттарды қанағаттандыратын мәліметтерді оқийды; [INSERT](http://www.ubukulov.kz/sql/insert-operatory-turaly-maelimet.html) жаңа мәліметтерді қосады; UPDATE бар мәліметтерді өзгертеді; DELETE мәліметтерді жояды. Кез келген ақпараттық жүйенің мақсаты – нақты өмірдің объектілері туралы деректерді өңдеу. Деректер базасы сөзін кеңінен алғанда – бұл қандайда бір пәндік облыста шынайы өмірдің нақты объектілері туралы деректер жиынтығы. Пәндік облыс деп шынайы өмірдің бір бөлігі деп түсінуге болады. Деректер базасын құрған кезде пайдаланушы ақпаратты әр түрлі белгілер бойынша сұрыптауға және белгілердің дербес құрылымын таңдауды тез жүзеге асыруға ұмтылады. Деректер базасын пайдаланушылар болып әр түрлі қолданбалы программалар, программалық кешендер, сонымен қатар пайдаланушылар деп аталатын деректерді пайдаланушылар рөлінде шығатын пәндік облыстың мамандары болуы мүмкін. Деректер базасының қазіргі технологиясында деректер базасын құру, оны қолдау және оған пайдаланушылардың кіруін қамтамасыз ету арнайы программалық инструментарий – д*еректер базасын басқару жүйесі* көмегімен жүзеге асыруды болжайды. **Деректер базасы** (ДБ) – бұл қандайда бір пәндік облысқа жататын құрылымдық деректердің аталынған жиынтығы.

**SQL DEVELOPER арқылы oracle ДББЖ қосылу**  Windows ОЖ:  Start-> Programs (или All Programs) -> Oracle Database 11g Express Edition -> Start Database. деректер базасын командалық жол аркылы қосу cmd: sqlplus-> user name, password -> conn hr

**Delphi-де мәліметтер қорымен байланысты ұйымдастыру** Delphi ортасында әртүрлі деректер базасын құруға болады. Деректер базасы құрылымының қиындығына қарамастан, мәліметтермен жұмыс істеуге мүмкіндік беретін негізгі механизм бар. Ол компоненттер триадасынан тұрады. Бұл компоненттер мыналар: Мәліметтерді енгізу компоненттері (Ttable, TQuery) DataSource компоненті Мәліметтерді енгізу компоненттері DataAccess бетінде орналасқан. Ал мәліметтерді көрсету компоненттері DataControls бетінде орналасқан. Қарапайым Деректер базасын құрудың күрделі Деректер базасын құрудан өзгешелігі шамалы. Формаға бірнеше компонентті қойып, олардың қажетті қасиеттерін өзгертсек, жұмыс істеп тұрған деректер базасын аламыз. Мәліметтерді енгізу компоненттері мен көрсету компоненттерінің арасындағы байланысты қамтамасыз ететін DataSourse компоненті. Бұл компоненттердің деректер базасындағы өзара байланысын қарастырсақ: Қосымша мен деректер базасының арасындағы байланысты қамтамасыз ететін BDE болып табылады. Процессорда сұраныстарды деректер базасына жіберетін драйвер орнатылған, сонымен қатар BDE –де деректер базасының псевдонимі тіркелген болуы керек. Ол деректер базасының файлдары қай жерде орналасқандығын және пайдаланатын драйвердің типін анықтайды.

***Oracle ДҚ физикалық архитектурасы 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.

**Oracle 11g express Edition Demo ДББЖ сұраныс құру.** Сұраныстар – бұл деректерді өңдеу үшін арналған арнайы құрылғы. Сұраныстардың көмегімен деректерді сұрыптауға, сүзгілеуге, өзгертуге, яғни өңдеуге болады. Сұраныстың негізгі қызметі – кестелерді байланыстыру және деректерді таңдап алу. Бұл жерде бірнеше кестедегі деректерді өзара байланыстыру мүмкіндігі туралы айтылады. Сонымен қатар деректерді таңдау барысында жаңадан қажетті мәліметтері бар уақытша кестелер құрылады. Бұл мақсат үшін MS Access-те сұраныстар қолданылады, ал уақытша қорытынды кестелерді Recordset объектісі деп атайды. Әдетте Recordset объектілерін виртуалды кестелер деп атайды, дегенмен олар сұраныстардың орындалу нәтижелерін деректер қорында емес, компьютердің жадында сақтайды. Дәлірек айтсақ, кестелер арасындағы байланыстар орнату мүмкіндігі реляциондық ДББЖ тізімдерді өңдеу үшін аранлаған тегіс файлдардың процессорлары деп аталатын қарапайым қосымшалардан өзгешелейді. Recordset объектісі – бұл бірнеше кестедегі деректерді тек біріктіріп қана қоймайды, сонымен қатар ондағы деректер іріктеудің нақты белгілеріне сәйкес келу керек. Сонымен бірге есептеу жолымен бір мәннен екінші мәнді алуға болады.Көптеген ДББЖ сұраныстырдың келесі түрлері қолданылады: - таңдама бойынша сұраныс кестедегі деректерді өзеріссіз қалдырып, форма немесе кесте түрінде болады; - өзгермелі сұраныс деректерді орналастыру немесе өзгерту (деректерді кірістіру, жою, жаңарту) үшін арналады; - қиылысқан сұраныстар статистикалық есептердің нәтижелерін көрсету үшін арналған; - параметрлері бар сұраныстар – Ал енді біз ең күрделі тапсырмалардың бірін қарастыруға – таңдамаға сұраныстар (немесе қарапайым сұраныс деп те атауға болады) құруға кірісеміз. Таңдамаға сұраныстарды қолдана отырып, деректер қорынан кестелік түрдегі деректердің бейнесін алуға болады. Сұраныстарға арналған деректер бір немесе бірнеше кестелерден, бұрыннан бар сұраныстардан немесе кестелер мен сұраныстардың комбинациясынан түсуі мүмкін. Деректерді алу үшін пайдаланылатын кестелер мен сұраныстар жазбалар көздері деп аталады.Таңдамаға қарапайым сұраныстар құру шеберді пайдалану мен құрастырушы режимінде жұмыс істеу кезінде де бірдей жағдайда орындалады. Бұл үшін сұранысқа енгізуді қажет ететін жазбалар көздері мен өрістерді таңдап алу қажет. Қажет болған жағдайда сұраныстың нәтижелерін нақтылау үшін шарт қоюға болады.Таңдамаға сұраныс құрғаннан кейін, нәтиже алу үшін оны орындау қажет. Сұранысты орындау үшін оны кестелер режимінде ашу жеткілікті. Алдағы жерде сұранысты қажеттілікке қарай қайтадан қолдануға болады, мысалы пішім үшін жазбалар, есептеулер нәтижелері немесе басқа сұраныстар көзі ретінде.Қарапайым сұраныс құру үшін келесі әрекеттерді орындаңыз:Ауысулар облысында, оның негізінде сұраныс құрылатын, кестені бөліп алыңызБұл кестеге, оның негізінде сұраныс құруға болатын, деректер енгізіңіз.Құру қосымшасына өтіңіз, Басқалар тобын таңдап алыңыз, одан кейінСұраныстар шебері пиктограммасында шертіңіз. Жаңа сұраныс шеберінің бірінші терезесінде сұраныстың түрін таңдап алыңыз, мысалы Қарапайым сұраныс. Одан кейін ОК батырмасында тышқанды шертіңіз.

**Oracle 11 g Exptss edition demo ДББЖ кілтті құру жіне жою........**

**Oracle 11 g Exptss edition demo ДББЖ типтері................**

**Oracle 11 g Exptss edition demo ДББЖ мәліметтерді енгізү және түзетү . 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. Компьютер жадындағы мәліметтерді басқару: Бұл қызмет алдымен үлкен МБ–мен және оның жүйесімен жұмыс атқарады, МБ–ның жұиысын жылдамдату үшін компьютердің оперативтік жадында мәліметтерді буферлеу қолданылады. Транзакцияны басқару: Бұл МББЖ–нің қызметі, яғни біртұтас бүтін ретінде Мб көптеген амалдарды орындайды. Бұндай амалдар компьютер жадында атқарылады. Транзакция ең алдымен көп қолданушылдық жүйеде МБ–ның логикалық үтіндігін сүйемелдеу үшін қажет. МБ–ғы өзгерістерді басқару және протоколдау: Бұл қызмет сақатлатын мәліметтің сенімділігіне байланысты , яғни МББЖ–нің мүмкіншілігі авариялық жағдайдағы МБ–ң күйін қалыптастыру.

**Oracle 11g Express Edition Demo ДББЖ қолдану .жаңа қолданушы құру .SQL-скрипті** **жүктеу.** ДББЖ мен жмыс істеу үшін қандай да бір аппараттық бағдарлама қажет. Ол өте кең аралықта - жалғыз дербес [компьютер](http://mvideo.ru/)не бір мейнфреймден көп санды компьютерлі желілерге дейін түрленуі мүмкін. Қолданылатын аппараттық қамтама нақты ұйымның және қолданылатын ДББЖ ның түріне байланысты болады. Кейбір ДББЖ лар операциялық жүйелердің не қондырғылардың тек нақты типтеріне арналса, басқалары кең ауқымды аппараттық қамтамалар және түрлі операциялық жүйелермен жұмыс істей алады. Әдетте ДББЖ мен жұмыс істеуге минимум оперативті және дискілі жады жұмсалады.

**Oracle АЛЫСТАТЫЛҒАН ДЕРЕКТЕР ҚҰРЫЛЫМЫ Түрдің 2 түрі бар: 1. Локальды 2. Алыстатылған.** Түр дегеніміз – нәтижелеріне кесте сияқты қатынас жасауға болатын сақталған сұраныстар. Түрге қосылу үшін Use командасы қолданылады. Түрде сақталған деректерді кестедегі сияқты қарауға, Browse терезелерде жөндеуге, Grid басқару элементінде қарап, жөндеуге болады. VisualFoxPro-да түрді View Designer жобалау терезесінде немесе Create sql View командасымен құруға болады. Жаңа түрді жоғарғы мәзірдің File – New командасын беріп, құрылатын файлдың типін View деп көрсетіп те құруға болады. Project Manager – жобалау диспетчері терезесінде локальды түр құру үшін келесі амалдарды орындау керек: 1. Project Manager терезесінде локальды түр құрылатын МҚ атауы көрсетіледі. 2. Бағынышты секцияда Databases, Local Views элементін таңдап, New пернесін басу керек. 3. Ашылған диалогтық Add Table or View терезесіндегі тізімнен кестені немесе түрді белгілеп, Add пернесін басады. Алыстатылған түрді құру. Локальды түрден өзгешелігі – алыстатылған деректер көзі болып табылатын серверді көрсету. Алыстатылған түр құру үшін келесі амалдарды орындайды: 1. Project Manager терезесінде Remote Views тізімінен элементті таңдап New пернесін басу 2. Select Connection or DataSource диалогтық терезе ашылады, одан алыстатылған деректер көзімен байланыс орнату үшін Available DataSources командасын таңдау керек. Байланыс ODBC драйвері бар кез келген деректер типімен орнатылады. Егер бұрын қандай да бір деректер көзімен байланыс орнатылған болса, онда Connection командасын таңдау керек. 3. Ашылған терезеде байланыс көзін, немесе деректер көзін тізімнен таңдап, OK – ді басады. 4. Егер деректер көзі серверде сақталған болса, оның қолданушы атауы мен паролі сұралады, оларды енгізгеннен кейін байланыс орнатуға болады. 5. Осыдан кейін серверде орналасқан кесте атауы таңдалынуы керек. Сосын түрді жобалау терезесі ашылады. Қалған баптаулар локальды түрді құру кезіндегімен бірдей орындалады.

**oracle ДББЖ администрациялау** Мұндай МҚ бір немесе бірнеше компьютерлерде орналасады. МҚ серверлері бір бірінен ақпаратты сақтау әдістерімен ерекшеленеді, бірақ барлығы бірігіп SQL- өқұрылымды сұраныстар тілін қолдану арқылы өзара байланыстарды орнатады. Осы қасиетіне қарап МҚ-мен жұмыс істейтін барлық серверлерді SQL-серверлер дейді. Бұл серверлерге қосылатын қосымшалар көпднңгейлі деп аталады, себебі ондай қосымшалар әртүрлі жүйелерде жұмыс істей береді. SQL-срверлердің көпшілігі SQL-командаларды және басқа да қосымша қызметтерді орындайды. Сондықтан әр сервердің өз диалекті болады.SQL тілі локальды МҚ-мен де жұмыс істейді, бірақ SQL-серверлермен салыстырғанда жұмысы баяу болады, себебі SQL-серверлер SQL-командаларды талдап, тиімді етіп орындайды, ал локальды базада ол жоқ. Сондықтан қосымшалар құрған кезде Paradox, dBase, FoxPro МҚБЖ-н тек басқа қолдануға тұрарлық программа жоқ болғанда ғана қолданады. SQL-серверлерге жататын программалар: InterBase, Oracle, SyBase, Informix, MS SQL Server, DB2. МҚ типін таңдағанда келесі факторларды ескеру қажет: — МҚ-на бірмезгілде қосылатын қолданушылар саны. SQL-серверлер бірмезгілде көпқолданушылықты ұйымдастырады және транзакция механизмімен жұмыс істейді. Ал локальды МҚ файлдық жүйенің қолданылуына негізделген блоктау механизмімен жұмыс істейді. Егер осыны ескерсек, локальды базаны аз қолданушылық режимде таңдайды. — Сақталатын деректер көлемі. SQL-серверлер көп деректерді сақтай алады, себебі локальды МҚ-да бір базадағы кестелер саны 10000-нан аса бастаса кедергілер туындайды. — Администраторлық қызметтерді қолдану қажеттілігі. Локальды МҚ-да SQL-серверге қарағанда ресурстар аз қолданылады және оларды басқару жеңіл. Ал SQL-серверде МҚ-н администрациялау үшін көп ресурс керек. — Сенімді қауіпсіздіктің қажеттілігі. МҚ көп жағдайда конфиденциальды ақпараттан тұрады, локальды МҚ-да қауіпсіздік тек қана бір кесте немесе атрибут деңгейінде орнатылса, SQL-серверде МҚ-н тұтасымен қорғау мүмкіндігі бар. Сонымен локальды МҚ келесі талаптарға сай қолданылады: — МҚ кестелерінде жазулар саны 10000-нан жоғары болмаса. — МҚ-н қолданатын қолданушылар саны аз болса. — Күрделі администраторлық, қорғау қызметтері қажет болмаса.

**pl sql деректер базасының тілі** SQL (structured query language – «құрылымдық сұраныс тілі» ) ­– формалдық бағдарламалық тіл, бірақ процедуралық бағдарламалауға жатпайды. Реляционнық деректер қорында мәліметтерді құруға, өзгертуге және оларды басқаруға арналған. Арнайы деректер қорын басқару жүйесі (ДҚБЖ) арқылы басқарылады. Бұл тілдің кеңейтілуі \*.sql. Алғашында SQL қолданушының деректер қорымен негізгі жұмыс істеу әдісі болатын және төмендегі келтірілген операцияларды орындауға мүмкіндік беретін: Дерек қорында жаңа кестені жасау үшін; Кестеге жаңа мәліметтерді қосуға; Мәліметтерді өзгертуге; т Мәліметтерді жоюға; Бір немесе бірнеше кестелерден мәліметтерді алуға; Кестелердің құрылымын өзгертуге. Уақыт өте келе SQL тілі жаңартылып жаңа конструкциялармен толықтырылды. Мысал ретінде индекстер, елестер, триггерлер және сақталатын процедуралар. SQL операторлары келесі түрлерге бөлінеді: Мәліметті анықтайтын операторлар (Data Definition Language, DDL): CREATE ДҚ объектін құрады (дерек қорының өзін, кестелерді, елестеулер, пайдаланушылар және т.б.); ALTER объектті өзгертеді; DROP объектті жояды. Мәліметтермен манипуляция жасау операторлары (Data Manipulation Language, DML): SELECT берілген шарттарды қанағаттандыратын мәліметтерді оқийды; INSERT жаңа мәліметтерді қосады; UPDATE бар мәліметтерді өзгертеді; DELETE мәліметтерді жояды. Мәліметтерге қол жетімділікті анықтайтын операторлар (Data Control Language, DCL): GRANT пайдаланушыға (топқа) объектпен белгіленген операцияларды орындауға рұқсат береді; REVOKE бұрын берілген мүмкіндіктерді қайтарып алады; DENY тыйым салады Транзакцияларды басқару операторлары (Transaction Control Language, TCL): COMMIT транзакцияны қабылдайды; ROLLBACK қазіргі транзакцияда істелінген барлық өзгерістерді қайтарады; SAVEPOINT транзакцияны кішігірім бөліктерге бөледі. Артықшылықтары және кемшіліктері. Артықшылықтары. Нақты ДҚБЖ дан тәуелсіздігі.Диалектінің және синтаксисінде айырмашылығының барлығына қарамастан, SQL сұранысының мәтіні DDL және DML құрамынан тұратын сұраныстарды бір ДҚБЖ дан басқасына оңай ауыстыруға болады. Стандартының барлығы. Декларативтігі, яғни бағдарламашы SQL арқылы өзіне керекті мәліметтерді шығарады немесе түрлендіреді. Кемшіліктері Реляционды деректер моделінің сәйкессіздігі. Реляционды деректер моделінің құрушылары Эдгар Кодд, Кристофер Дейт және олардың жақтаушылары SQL ді шынайы реляциондық тіл деп есептемейді. Олар реляциондық теория көзқарасы бойынша SQL дің келесі ақауларын көрсетеді:қатарлар-көшірме нұсқалар; анықталмаған мәндер (null); бағаналардың реттігін пайдалану; атсыз бағаналар және қайталанатын бағаналар аты; «=» операциясымен дұрыс істемеушілігі; көрсеткіштерді пайдалану; жоғары артықтық. Қиындығы. Алғашында SQL соңғы қолданушыға жұмыс істеу құралы ретінде ойластырылған болатын, бірақ соңына келе ол біршама өзгерістерге ие болып, бағдарламашының құралына айналды.Стандарттардан шегіну. Өзінің халықаралық ANSI SQL-92 стандарты болғанымен көптеген ДҚБЖ бағдарламашылары SQL тіліне өзгерістер енгізеді.Иерархиялық құрылымдықтармен жұмыс істеу қиындығы. Біршама ДҚБЖ бұрын SQL диалектілерін иерархиялық құрылымдарымен әртүрлі манипуляциялардың түрлерін ұсынбаған болатын. Кейбір ДҚБЖ өздерінің шешімдерін ұсынды. Мысалы, Oracle Database де CONNECT BY көрінісі пайдалынады. Қазіргі уақытта SQL DB2 диалектінен WITH рекурсивті конструкциясы ANSI да стандарттылған. Microsoft SQL Server де рекурсивті сұраныстар (Recursive Common Table Expressions) 2005 ші версиясында ғана пайда болды.Процедуралық кеңейтілімі. SQL процедуралық бағдаламалық тілдер қатарына енбейді. Негізінен әрбір ДҚБЖ де өзінің процедуралық тілі қолданылады, сонымен бірге Oracle Database де PL/SQL, Interbase және Firebird де PSQL, DB2 де SQL PL, Microsoft SQL Server және Adaptive Server Enterprise де Transact-SQL, PostgreSQL де PL/pgSQL.

**ms access дббж байланыстыру** Кез-келген ДББЖ да деректер базасының бiрнеше кестесiмен бiр мезгiлде жұмыс ұйымдастыруға болады. Мысалы, MS Access программасында дайындалған кестелердi басқа ДББЖ-лерiмен (Paradox, dBase) байланыстыруға болады және ол басқа ДББЖ-нде жасалған кестелердi пайдалана алады. Excel электрондық кестесi мәлiметтерiн де Access программасында қолдануға болады. Программалау тiлдерi (мысалы, С++) мүмкiндiктерiн пайдаланып, Access программасымен байланыса отырып басқа қосымшалармен, яғни мәлiметтер көзiмен жұмыс ұйымдастыруға арналған қолданбалы программа интерфейсiн дайындауға болады. MS Access-те бүкiләлемдiк Интернет желiсiмен байланысуда және Web қосымшаларын құруда жеңiлдiк беретiн құрылғылар жиынтығы бар Access ДББЖ-не көптеген қосымша сервистiк мүмкiндiктер кiрiстiрiлген. MS Access ДББЖ-дегi Деректер базасы ақпаратты өңдеу, сақтау, басқару, тарату iстерiн жылдам әрi көрнекi орындайтын құралдар жиынтығын құрайды.. MS Access жүйесi мәлiметтермен жұмыс ұйымдастыруға мүмкiндiк беретiн кесте, форма, сұраныс, отчет объектiлерiнен тұрады. Сонымен қатар Access-тi қосымшалар дайындау ортасы деп қарауға да болады, ол макростар көмегiмен жүзеге асырылады. Макростар пайдаланушыға Access VBA тiлiн пайдаланбай-ақ, яғни Basic тiлiнде программалау бiлмесе де көптеген iс-әрекеттердi автоматтандыруға мүмкiндiк бередi MS Access құрамына жұмысты жеңiлдететiн жүздеген шеберлер кiредi. Access-тегi шеберлер дайын үлгiлерге сүйенiп стандарт түрдегi кесте, форма, отчет, сұраныс, графиктер, пошталық жапсырмалар (наклейка) мен қосымшаларды жобалауға және құруға мүмкiндiк бередi. Конструкторлар дайын кесте, формаларды түрлендiруге, күрделендiруге арналған және олар стандарт емес құжаттар дайындауға мүмкiндiк бередi.MS Access ДББЖ-мен жұмыс барысында анықтамалық жүйесiн қолдануға болады. Анықтамалықты мәзiр арқылы да, тәуелдi мәзiр көмегiмен де алу мүмкiндiгi бар. Сонымен қатар қажет кезде жылдам шақыруға болатын Көмекшiнi де пайдалануға болады.

**sql dml dsl ddl** .60- жылдардың аяғында IBM фирмасының кызметкері Эдгар Кодд алғаш рет реляциялык деректер коймасының кұрылымдық принциптерін түжырымдады жэне SEQUEL тілін (Structured English Query Language- кұрылымданған сұраныстардың агылшьгнша тілі) жасап шығарды. Кейінірек бұл атау SQL (Structured Query Language- кұрылымданған сүраныстар тілі) болып кысқартылды. SQL тілі деректердің реляциялык үлгісін өңдеу нәтижесінде пайда болған тілдердің бірі болып табылады, ол казіргі уақытта өте кең қолданылады жэне реляциялык деректер корыныц стандартты тіліне айналды. SQL тілінің алғаш стандарты 1986 жылы (SQL-86 деп аталды) және 1987 жылы шыкты жэне 1989 жылы ол аздап өзгертілді. Одан кейін ДҚБЖ құрастырушыларыныц тілді өзгертуі 1992 жылы жаца кеңейілген стандартын қабылдауга әкелді. (ANSI SQL-92 немесе жай SQL2). Одан кейінгі стандарты- SQL: 1999 (SQL3). Қазіргі кезде аздап өзгеріске ұшыраган 2003 жылы кабылданган SQL:2003 стандарты колданылады.

SQL тілінлегі команлалардын негізгі категорнялары ДҚ объектілерін қүрып жзнс оларды манипуляішялау, кестеге бастапкы деректерді косу, акиараттарлм жанарту жэне ошіру, ДҚ-ла сүраныстарды орынлау. оган кол жеткізулі баскару және оны жалпы адмнннстраинялаумен катар лргурлі функцняларлы орынлау үшін арналган. SQL тілінлсгі команлаларлыц негізгі катсгориялары: DDL - деректерді аныкглу тілі; DML - леректерлі маннпуляннялау тілі: DQL - сүраныстар тілі; DCL - лерсктсрлі баскару тілі; DAC- лерсктерді алминнстрацнялау командалары: TCL - траизакиияны баскару команлалары. DDL <)еректер<)і анықтау тілі (Data Definition Language) деректер кормный күрмлымын жасауга жэне озгергугс мүмкінлік берелі.

DDL тілінін негізгі командшіары: CREATE SCHEMA - ДК схемасын жасау; DROP SHEMA - ДК схемасын жою; CREATE TABLE - кесте жасау; ALTER TABLE - кестені озгсргу; DROP TABLE - кестені жою; CREATE DOMAIN -ломеи жасау; ALTER DOMAIN - домеиді озгерту; DROP DOMAIN - доменді жою; CREATE INDEX - индекс жасау; DROP INDEX - индексті жою; CREATE COLLATION - тізбек жасау; DROP COLLATION - тізбекті жою; CREATE VIEW - корсетілім жасау; DROP VIEW -корсетілімді жою. DML ОеректерОі манипу.шцииау тиі (Data Manipulation Language) INSERT, UPDATE, DELETE үш негізгі комаішасы аркылы реляцнялык деректер корының объектілерініц ішіндегі акларатты маннпуляциялау үшін колданылады. INSERT - кестеге жол косу; UPDATE - ксстедегі жолды өзгерту; DELETE - кестедегі жолды жою.

DCL ОеректерОі басқару тиі (Data Control Language) ДК-нын ішінде орналаскан акпараттарга кол жеткізуді баскару үшін колданылады. Ереже боііынша, олар деректерге кол жеткізумен банланыскан объектілерді күру үшін колданылады жэне колданушылардын арасында женілдіктерді гаратуды бакыпау үшін кызмет етеді. DCL тілінін командалары: CREATE ASSERTION - шектеу жасау; DROP ASSERTION - шектеуді жою; COMMIT -енгізілген озгерісті бекітуг; ROLLBACK - енгізілген озгерісті болдырмау; GRANT - колданушыга немесе косым шага объектілерді озгертуте женіпдік беру;

## Ақпараттық қатынастар және деректердің өзара байланысыь **үннен-күнге ақпарат біздің күнделікті өмірде маңызды рөл атқарады. Біз XXI ғасырдың ақпаратты тәуелді қауымы болдық және сұраныстар мен командалар әлемінде өмір сүреміз. Ол ақпарат алудағы белгілі уақыт пен белгілі бір орындағы қажеттілікті көрсетеді. Күнделікті біз ақпарат іздеу, әлеуметтік желіде араласу, электронды поштаны жөнелту және алу, суреттер мен видеоларды, сонымен қатар басқа да тапсырмаларды орындау үшін Интернетке шығамыз. Бұл кезде туындайтын контент саны артады, (ағыл.*content* — құрамы) – ақпараттық ресурстың кез-келген ақпараттық-мәнді толықтыру (мысалы, газет, журнал, кітаптар, веб-беттер және т.б.) – оқытушы/тыңдарман/көрермен өзінің қарастыруы бойынша қолдана алатын барлық ақпарат. Мысал, мәтін және бейнелер: сурет, фото, сызбалар, құжаттарменөзара байланысу элементтері, аудио-визуалды мағлұматтар: Adobe, Excel, Word, Power Point, exe, rar форматындағы аудио-, видео- құжаттар, құрылғылар жеке кәсіпорынмен емес, жеке тұлғалармен құрылады. Жеке тұлғалармен құрылған ақпаратты басқа тұлғалармен алмастырғанда оның құндылығы артады. Әдетте, құрылу кезінде бұл ақпарат мобильді телефондар, камералар, ноутбуктер сияқты құрылғыларда орналасады. Алмастыру үшін деректерді сақтау орталықтарында оны желі арқылы жүктеу керек**

**ақпараттар және деректер базасының негізгі** түсініктеріМәліметтер базасы дегеніміз - ақпаратты сақтауға және жинақтауға арналған ұйымдасқан құрылым. Ең алғаш мәліметтер базасы ұғымы жаңадан қалыптасқан кезде онда шындығында мәліметтер сақталған. Бірақ қазіргі кездегі көптеген мәліметтер базасын басқару жүйелері өздерінің құрылымдарында тек мәліметтерді ғана емес, сонымен қатар олардың тұтынушылармен және басқа да ақпараттық - программалық кешендермен қарым - қатынасының әдістерін де қамтиды. Сондықтан біз қазіргі заманғы мәліметтер базасында тек мәліметтер ғана емес, ақпарат та сақталады деп айта аламыз.Мәліметтер базасын басқару жүйесі дегеніміз жаңа мәліметтер базасын құруға, оны мәліметтермен толтыруға, оның мазмұнын редакциялауға, оларды реттеуге, безендіруге және баспа құрылғыларына шығаруға немесе байланыс арналарына жөнелтуге арналған программалық құрылғылар кешені.Дүние жүзінде мәліметтер базасын басқару жүйелерінің түрлері өте көп. Олардың әрбіреулерінің түрлі объектілермен жұмыс істеу ерекшеліктеріне, қолданушыға ұсынатын қызмет түрінің өзгешелігіне қарамастан, оларда ортақ ұғымдар кешені қалыптасқан. Мәліметтер - ақпараттың құрамдас бөлігі. Тіркелу әдісіне сәйкес мәліметтер әр түрлі тасуыштарда сақталады және тасымалданады. Ең кең тараған мәліметтерді тасуыш қағаз болып табылады. Заттың оптикалық қасиеттерінің өзгерісі лазерлік сәулелердің көмегімен жазылатын тасуыштар CD - ROM - да қолданылады. Магниттік қасиеттердің өзгерісін қолданатын тасуыштар ретінде магниттік таспалар мен дискілерді алуға болады.Базаға ешқандай мәліметтер енгізілмеген жағдайда да ол мәліметтер базасы болып қала береді. Өйткені онда мәліметтер болмаған күннің өзінде де, қандай да бір ақпарат бар - бұл мәліметтер базасының құрылымы. Ол мәліметтерді енгізудің және сақтаудың әдістерін анықтап береді. Мәліметтер базасы көптеген объектілерден тұрады, бірақ солардың ең негізгісі кесте болып табылады. Ең қарапайым мәліметтер базасы бір ғана кестеден тұрады. Сәйкесінше, ең қарапайым мәліметтер базасының құрылымы сол кестенің құрылымымен анықталады.Біз білеміз, екі өлшемді кестенің құрылымы жолдар мен бағандардан тұрады. Олардың аналогы ретінде қарапайым мәліметтер базасында өрістер мен жазбаларды аламыз. Өрістердің құрамы мен қасиеттерін өзгерту арқылы, мәліметтер базасының құрылымын өзгертеміз. Егер мәліметтер реттелген, яғни қандай да бір берілген құрылымға сәйкестендірілген болса, онда олармен жұмыс істеуді автоматтандыру оңайға түседі. Мәліметтер құрылымының негізгі үш түрі бар: сызықтық, кестелік, және иерархиялық. Мәліметтердің сызықтық құрылымы дегеніміз әрбір элементтің адресі мен нөмірі бір мәнді анықталған реттелген құрылым. Сызықтық құрылымға мысал ретінде тізімдерді алуға болады

алғашқы баланстары, қоймадағы тауарлардың саны мен бағасы туралы

**Алғашқы деректердің интерпретациясы.** Алғашқы желілер тең құқылы қатынас жасауға кепілдік беретін. Қолданбалы бағдарлама белгілі-бір уақытқа байланыстың бөлінетін қорын иемденетін; қосымшаға басқа қосымша байланыстың осы ресурсын қолдануға кіріспестен бұрын өз жұмысын аяқтауға рұқсат берілетін. Бұны болғызбас үшін қазіргі компьютерлік желілерде дестелерді пайдалану қарастырылған. Желі бір компьютерге бір дестені жіберуге рұқсат береді де компьютерге тосқаул қояды, сосын басқа компьютерге өз пакетін жіберуге мүмкіндік береді де, осы компьютерге де тосқаул қойып, келесі компьютерге өз пакетін жіберуге мүмкіндік береді. Әр компьютер бір дестені жіберуге жетерлік уақытқа ғана бөлінетін қорды иемденеді, сосын келесі дестені жібермес бұрын басқа компьютер желіні қолданып болғанша тұрып күтуі тиіс. Дестелерді қолдану неліктен үздіксіз жабдықтауды қамтамасыз ететінін түсіну үшін, қандай бір қосымшаға желіні оның жұмысын толық аяқтағанға дейін толық қолдануға рұқсат береді.

әр біреуі екі қатысушылармен және бір келісімен байланысты. Диаграммадағы әрбір қатынас тізбек деген атағы бар – оның мағынасын жай

баланстары туралы емес.

**генераторлар мен триггер** Генератор(Generator) — 1) кіріс тілі проблемалы-бағытталған тіл болып табылатын аударғыштың бір түрі; 2) машиналық командаларды генерациялауды орындайтын аударғыштың құрамдас бөлігі. Жазбалар генераторы (Генератор записей; rekord generator) — тестілеуге қажетті жазбаларды құрастыруға арналған машиналык программа.Кездейсоқ сандар генераторы (Генератор случайных чисел; generator random numbers) — кездейсоқ сандар беретін қүрылым немесе программа. Әдетте, Кездейсоқ сандар генераторы - программалау тілдерінде стандартты функциялар мен процедуралар қүрамына кіретін программа. Мысалы, Паскаль тілінде RANDOM(N) функциясы 0 ден N-1 аралыгындағы бүтін сан қайтарады.Егер программа қайта орындалса, функция сол санды қайталайды. Кездейсоктық дәрежесін арттыру үшін тілде генерация базасын өзгертетін RANDOMIZE процедурасы бар, бұл процедура RANDOM функциясының алдында орындалуы қажет. Код генераторы (Генератор кода; code generator) — талдау және оңтайландыру нәтижелері бойынша машиналық программаны (объектілік модульді) құрастыратын аударғыштың машинаға тәуелді бөлігі.[1] Командалар генераторы (Генератор команд; code generator) — аударылатын программа операторларына сәйкес машиналық командалар тізбегін щығаратын аударгыш бөлігі. Қолданбалы программалар дестесі генераторы . (Генератор пакетов прикладных программ; package generator) Қолданбалы программалар дестесін нақтылы мәселелер классына бағыттау программасы. Қүжат дайындау генераторы (Генератор отчетов; report generator) — 1) берілген пішін бойынша мөліметтерді қалыптастыру жөңе шығаруға арналған объектілік программаларды генерациялауға колданылатын өндеу программасы; 2) берілген пішін бойынша мөліметтерді калыптастыру мен шығаруды орындайтын кейбір программалау тілдерінін (мыс., Кобол, Access, FoxPro, Delphi, жөне т.б.) қүралы. Мәліметтер генераторы (Генератор данных; data generator) — тізбекті қатынас құру әдісі негізінде бір тапсырма көлемінде мәліметтер жиынын кұруға арналған сервистік программа. Программалар генераторы (Генератор программ; program generator) кейбір операцияларды сипаттау негізінде осы операцияларды жүзеге асыратын программаны автоматты түрде жасайтын программа. Сүрыптау программаның осы программалардың алғашқы мысалы болып табылады. Ол файл пішімін және талап етілген сүрыптау түрін сипаттау негізінде сәйкес сүрыптау программасын құрады. Кейінгі уақытта қолданбалы программаларды шаблон, мәліметтер базасы сипаттамасының таблицасы, экрандық форма, меню сипаттамасы және т.б.. Объектілер негіздерінде құратын программалар кеңінен таралуда. Тактілік жиілік генераторы (Генератор тактовой частоты; generator clock speed) — белгілі бір уақыт аралықтары сайын импульстер тізбегін шығаратын құрылғы. Қатар екі импульстың арасындағы уақыт Ырғақ деп аталады. Кейбір процессор командалары бірнеше ырғақта орындалады. Импульстар барлық компьютер элементтері арқылы өтіп, оларды бір ырғақта (синхронды) жұмыс жасатады. Ырғақ импульстарының жиіліғі компьютердің жьлдамдығын анықтайды.[2] Триггер (Trigger, flip-flop) — екі орнықты қалып-күйінің біреуінде ғана болатын цифрлық техниканың элементі. Екілік жүйеде берілген ақпаратты сақтау үшін қолданылады. Триггердің RS, JK, Т, D, DV және т.б. түрлері болады;оқиға мен программалық процесс арасындағы байланыс. Мысалы, триггерлер (қосылған, жалғастырылған процедуралар) — Visual FoxPro оқиғаларының қызметтік көрінісі.Олар кестедегі (мәліметтер базасындағы) жазбаларды көру, өзгерту немесе жою кезінде не болып жатқанын бақылауға мүмкіндік береді; екі тұрақты қалып-күйдегі электронды логикалық схема.

**ДЕЛЬФИДЕ МҚмен байланысты ұйымдастыру**

**Деректер базасын басқару жүйесін басқару түрлері. *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.

**Деректер базасын жобалау этаптры. *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. **SELECT таңдау операторы** Оңай болу үшін осы оператордың сиетаксисін қайталайық: <select statement> ::= SELECT [ALL | DISTINCT] <select name> INTO <select target list><table expression> <select target list>::= <target specification> [{,<target specification>}...] Айтып өткендей, жалғыз оператордың нәтжесі кем дегенде 1 жолдан тұратын операция WHERE операторы. Егер табдицада WHERE бөлімі болса, онда келесі кезекте саналады. WHERE бөлімінің синтаксисі төмендегідей: <where clause> ::= WHERE <search condition> <search condition> ::= <boolean term> ( <search condition> OR <boolean term> <Boolean term> ::= <boolean factor> ( <boolean term> AND <boolean factor> <boolean factor> ::= [NOT] <boolean primary> <boolean primary> ::= <predicate> | (<search condition>)

**Деректер базасының басқару жүйесінің басқарылуы.** Қызметтің кез келген саласында жиі еректердің үлкен көлемдерімен жұмыс жасау қажет. Бұл кездегі негізгі операциялар – бұл ақпаратты жинау, оны өңдеу (қажет деректерді іздеу, сұрыптау және т.б.), деректерді қарап шығуға және баспаға шығаруға арналған формаларды құрастыру болып табылады. Осы барлық функцияларды қамтамасыз ететін ең таңымалды программалық құрал MS Access деректер базасын басқару жүйесі болып табылады.Бұл ақпарат, яғни деректер арнайы форматта (\*.mdb) сақталынатын файл (құжат). Деректер базасын басқару жүйесі – бұл программа. Ол арқылы ақпарат деректер базасына енгізілінеді, сұрыпталынады, сүзбеу жүргізілінеді, қажет ақпарат ізделінеді, қажет ақпарат басқа программаларға көшірілінеді. Деректер базасындағы бағандар өрістер деп аталады, ал жолдар – жазулар. Жазулар саны қатты дисктің сыйымдылығымен шектеледі. Өрістердің максималды саны – 255. Өрістер мен олардың қасиеттерінің жиынтығы деректер базаның құрылымын анықтайды. Жазуы жоқ деректер базасы да деректер базасы болып табылады, өйткені оның құрылымы бар. Деректер базасын құрастыру: 1 қадам. Оның құрылымын құрастыру. 2 қадам. Құрылымды ақпаратпен толтыру. Access-ті жүктеген кезде деректер қорының объектілерімен жұмыс жасауға арналған мәзір пайда болады.

деректер базасының құрылымын ұсынады. Ол көрсеткіштер, индекстер, жазу

**Деректер қорын жобалау кезеңдері.**Кез келген деректер қорын құрудың мақсаты қандай да бір пәндік саладағы ақпаратты сақтау және қолдану болып табылады. Бұл мақсатты тарату үшін келесі құралдар қажет:1. Деректердің реляциялық моделі -пәндік саладағы деректерді көрсетудің ыңғайлы әдісі.2. SQL тілі- осындай деректерді манипуляциялаудың әмбебап тәсілі.Бірақ расында, сол бірдей пәндік саладағы реляциялық қатынастарды көптеген әртүрлі тәсілдермен жобалауға болады. Мысалы, көп санды атрибуттармен бірнеше қатынастарды жобалауға болады, немесе керісінше, барлық атрибуттарды үлкен санмен ұсақ қатынастарға тарату. Қалай анықтауға болады, қандай белгімен атрибуттарды сол немесе басқа қатынастарға орналастыруды?Осы бөлімде реляциялық қатынасты жобалаудың "жақсы" немесе "дұрыс" тәсілдері қарастырылады. Алғашқыда біз, деректер моделінің "жақсысы" немесе "дұрысын" талқылаймыз. Содан кейін бірінші, екінші, үшінші қалыпты форма қатынас түсінігі енгізіледі (1ҚНФ, 2ҚФ, 3ҚФ) және көрсетілген "жақсысы" үшінші қалыпты форма болып табылады.Деректер қорын құру кезінде көбінесе бірнеше моделдеу деңгейлері ерекшеленеді, олардың көмегімен пәндік саладан нақты ДҚБЖ құралдарының деректер қорына өту жүргізіледі. Келесі деңгейлерді ерекше айтуға болады:· Сол пәндік сала ,· Пәндік сала моделі , Деректердің логикалық моделі· , Деректердің, физикалық моделі,· Негізінен деректер қоры және қосымша. Пәндік сала – деректер қорында біз көрсететін, бұл қоршаған ортаның бір бөлігі. Мысалы, пәндік сала ретінде кәсіпорынның, кадр бөлімінің, банктің, дүкеннің және т.б. бухгалтериясын таңдауға болады. Пәндік сала шексіз және қажетті түсініктер мен деректерден, сол сияқты аз ғана қажетті немесе тіпті қажет емес деректерден тұрады.

деректерден тұрады, бірақ сатып алынған тауарлардың бағасынан немесе

**Деректерді модельдеу түрлері.** 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 моделі деректердің құрылымдарының жиынтығының деректер бүтіндігін шектеу және деректермен операциялар жасау әрекеттерін орындайды. Жалпы айтқанда, ***деректер моделі*** - деректер құрылымы жиынтығы мен оларды өңдеу операциялары. Деректер базасын басқару жүйелері иерархиялық, желілік және реляциялық модельдеуге немесе осы модельдердің белгілі бір жиынындағы комбинациясына негізделген. Деректер моделінің үш негізгі типін қарастырайық: иерархиялық, желілік, реляциялық.

**ЕR диаграммалар.** E-R диаграммалар (Entity-Relationship, Мән-Қатынас) ДБ сызбасын

есептейді. Сыртқы сызба пайдаланушылардың логикалық ұсынысың

жасырып, ондық міннен тұрады. Мысал ретінде жеке саудада сатуларды

жеткізімді) немесе ЖАБЫҚ (қол жеткізімді) Деректер қорының құрылымы. Oracle деректер қорының физикалық,

жинақталуына рұқсат етіледі. Диаграммада әрбір мән мен қатынас арасында ‘1’ немесе ‘\*’ символдары болады. Олар екі мән арасында қатынас сипатын сипаттайды. Қатынастың сипаты бойынша үш тип болады: бірге бір (1:1), көпке бір (1:\*) және көпке көп (\*:\*).

жиынынан тұрады, олардың әр біреуі сызба деп аталады. Ішкі сызба

***Иерархиялық деректерді моделдеу .*** Иерархиялық құрылым бір-бірімен белгілі бір ережелер бойынша байланысқан элменеттер жиынтығын береді. Иерархиялық байланыстағы объектілер бағытталған граф (төңкерілген ағаш) құрады Иерархиялық құрылымның негізгі ұғымдарына мыналар жатады: деңгей, элемент (түйін), байланыс. Түйін дегеніміз белгілі бір объектіні сипаттайтын деректер атрибуттарының жиынтығы.

келтіретін қызмет ететін сыртқы сызбалар несие лимит пен клиенттердің

клиенттердің адресінен тұратын ақпараттан тұрады, бірақ олардың келесі

компанияның алғашқы банктік счеты туралы ақпараттарды қоспайды. Сырқы

концептуалды және сыртқы сызбасын өңдеу үшін тартылады. Сондықтан

**Компоненттер** ДҚ жұмыс жасау үшін қолданылатын компаненттер, DataAccess, DataControls, Midas, Qreport және DesisionCube беттерінде болады. DataAccess бетінде визуалды емес компонент орналасқан, олардың көмегімен мәліметке қатынасты ұйымдастыруға болады: DataSourse – мәліметтер көзі ; Table – ДҚ кестесіне негізделген мәліметтер жиыны ; Query – SQL сұрауына негізделген мәліметтер жиыны ; StoredProc – серверде сақталған процедураға негізі мәліметтер жиыны;

* DataBase - ДҚ мен байланысуы; - Session – БД да ағымдағы жұмыстар сеансы . DataControls бетінде келесі визуалды компоненттер орналасқан; DBGrid – тор ( кесте ) DBNavigator – навигациялы интерфейс DBText – жазу DBEdit – бір жолдық редактор DBImage – графикалық бейне DBListBox – қарапайым тізім DBComboBox – қиыстырылған тізім DBCheckBox – тәуелдісіз ауыстырып- қосқыш QReport бетінде келесі визуалды компоненттері орналасады - QuickRep - есептеу - QRSubDetail – байланысқан кестелерге арналған есепберу жолағы - QRBand – есепберу жолағы

**Деректер базасын компонентпен байланыстыру**. Ол үшін объектілер инспекторындағы Database қасиетін пайдаланамыз. Бұл жерде BDE – де тіркелген деректер базасының псевдонимін көрсету керек. Псевдонимді ашылатын тізімнен таңдап алуға болады.

**Деректер базасының кестесін компонентпен байланыстыру**. Ол үшін объектілер инспекторы терезесіндегі TableName қасиеті пайдаланылады. Бірінші кезеңді орындағаннан кейін берілген деректер базасының барлық кестелері тізімде көрінуі керек. Бұл тізімнің құрамы TableType қасиеті арқылы анықталады. Егер мәні ttDefault болса, онда таңдап алынған кестенің барлық файлдары кеңейтулерімен көрсетіледі. Кестені таңдап алған соң ол компонентпен байланысады.

**Компоненттің атын өзгерту**. Бұл міндетті емес, бірақ компоненттерге ойластырылған ат берген дұрыс. Көбінесе компоненттің аты онымен байланысқан кестенің аты мен Table сөзінің комбинациясынан тұрады.

**Концептуалды схема және д.м.** Деректер базасының сызбасы ДБ логикалық құрылымын сипаттайды.Сызбаның үш деңгейі бар: концептуалды, ішкі және сыртқы. Концептуалды сызба барлық ДБ ұйымына жалпы көзқарасын береді. Ол деректердің барлық элементтерін және олардың арасындағы қатынастарын

қабылдау ДБ өңдеу кезінде қиын сызбаларды көзқараспен алуы, бөлшектерді

қабылдау. REA моделі бұның негізінде берілген диаграмма құрылған 7 мәннен

**қатынастарды байланыстыру** Бөлімше немесе жұмыс тобы деректер сақтау үшін Microsoft SQL Server пайдаланса, Access бағдарламасында кейбір SQL Server деректермен жұмыс істеу керек болады.SQL Server нысандарынан (кестелерінен немесе көріністерінен) Access бағдарламасына екі жолмен алып келуге болады: импорттау немесе байланыстыру арқылы. Екі үрдістің айырмашылығы төмендегідей:Деректі импорттағанда Access бағдарламасы SQL Server дерекқорында шағылыспаған SQL Server деректері мен Access дерекқорындағы деректерге жасалған кез келген соңғы өзгерістердің көшірмелерін жасайды. Сондай-ақ, кез келген SQL Server кестесі мен көрінісінде жасалған және Access бағдарламасында шағылыспаған өзгерістердің де көшірмелерін жасайды.SQL Server дерегімен байланысқан кезде, тікелей дереккөзге жалғанасыз, осылайша Access бағдарламасында жасалған соңғы өзгерістер SQL Server ішінде шағылысады және керісінше болады. Байланыстыру қолайлы болатын жағдайлар Әдетте, SQL Server деректерімен келесі себептерге байланысты байланыстырасыз: SQL Server дерекқорындағы және Access дерекқорындағы ең соңғы жаңалықтарды қарауға және өңдеуге мүмкіндік алу үшін дереккөзге тікелей өту үшін қажет.SQL Server дерекқоры көптеген көлемді кестелерді қамтиды, ал олардың барлығын бір .accdb файлына импорттау мүмкін емес.Access дерекқорының мөлшері 2 гигабайт, оның ішінде жүйе нысандары алатын орын да бар.Деректердің көшірмесін жасамастан SQL Server дерекқоры деректері негізінде сұрауларды іске қосып, есептер жасау қажет болады, сонда SQL Server дерекқорымен байланыстырыңыз.Бөлімше немесе жұмыс тобы Access бағдарламасын есеп және сұрау жасау үшін және SQL Server бағдарламасын деректер сақтау үшін пайдаланады. Жеке топтар SQL Server кестелерін және көріністерін орталықтандырылған жад үшін жасайды, бірақ көбінесе осындай деректер бірігу және есеп жасау үшін жұмыс үстеліне жеткізіледі. Байланыстыру дұрыс таңдау деп табылады, себебі ол SQL Server дерекқорының да, Access дерекқорының да пайдаланушыларына деректерді қосуға және жаңартуға, және соңғы деректерді көруге және олармен істеуге рұқсат береді.Сіз SQL Server бағдарламасын пайдалануды жақында ғана бастаған Access пайдаланушысысыз. Бірнеше дерекқорларды топтың SQL Server бағдарламасына көшірдіңіз және осы дерекқорлардағы кестелердің көбі байланыстырылған кестелер. Қазірден бастап Access кестелерін жасаудың орнына SQL Server ішінде кестелер мен көріністер жасап, оларды Access дерекқорларымен байланыстыратын боласыз.Деректерді SQL Server ішінде сақтауды жалғастыруыңыз мүмкін, бірақ Access ішінде жасалған сұрауларды іске қосып, есептерді басып шығару үшін Access ішінде ең соңғы жасалған деректермен жұмыс істеуіңіз мүмкін.

мәліметтер жасырылған, ДБ концептуалды сызбасында өңдеуінде

**Концептуалды жобалау** Қажетті мәлімет қолданушылардың талабы Жоба пайдаланушының барлық негізгі талаптарын қанағаттандыру керек. Бұл жүйенің негізгі пайдаланушылары университет мұғалімдері мен студенттері болады. Негізгі талаптар алынған ақпараттың ұсынылатын тиімділік деңгейінде негізделеді. Жүйеге қойылатын негізгі талаптардың тізімі:  жүйе қолданыста қарапайым болу керек. интерфейс ыңғайлы болу керек;  жүйе қауіпсіздікті қамтамасыз ету керек;  жүйе дерекқордың тұтастығын қамтамасыз ету керек;  жүйе толығымен интеграциялану керек, яғни мәліметтердің артылуы және артық жаңартулар болмау керек

**Логикалық жобалау** Логикалық (даталогикалық) жобалау – нақты деректер үлгісі негізінде дерекқорлар схемасын құру, мысалы, деректердің реляциялық үлгісі. Деректердің реляциялық үлгілері үшін даталогиялық үлгі – әдетте біріншілік кілттерді көрсетумен, қатынастар, сонымен қатар ішкі кілттен тұратын қатынастар арасында «байланыстар» схемасының жиынтығы. Тұжырымдамалық үлгіні логикалық үлгіге түрлендіру, ережеге сай, формалық ережелер бойынша жүзеге асырылады. Бұл кезең едәуір дәрежеде автоматтандырылған болуы мүмкін. Логикалық жобалау кезеңінде деректердің нақты үлгісінің ерекшелігі ескеріледі, бірақ нақты ДБЖ ерекшелігі ескерілмеуі мүмкін. Кестелер құру Кестелердіі CREATE TABLE командасымен құруға болады. CREATE TABLE бағандар атын және кесте атын анықтайды және ол деректердің типін және бағандардың өлшемін анықтайды. Ең ыңғайлысы бұл генерацияланған скриптті Query Analyzer-де (МS SQL Server) қосу. Кесте құруға арналған скрипт А қосымшада көрсетілген

**Мәліметтер қорымен жжасауға арналған инструментті** құралдар Мәліметтер қорымен жұмыс жасауға арналған Delphi құралдарын екі түрге бөлуге болады: - иструменталды құралдар; - компоненттер. Инструменталды құралдарға құры-лушы қолданбадан тыс мәліметтер қорына қызмет етуді қамтамасыз ететін арнайы программалар мен дестелер жатады. Компоненттер мәліметтер қорымен опера-цияларды 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.

**мәліметтер типтері және моделдері** . Мәліметтердің немесе шамалардың типі деп, олардың қабылдай алатын мәндерінің және олармен орындауға болатын амалдардың жиының анықтауды айтады, яғни тип дегеніміз - шамалардың қабылдайтын мәндеріне берілетін сипаттама. Мәліметтердің әр типі тек өзіне ғана сәйкес келетін операциялар жиынын орындата алады. Мысалы, 1 мен 2 мәндері бүтін сандар типіне жатады, оларды қосуға, азайтуға, көбейтуге және бөлуге болады.АJI, «ІВМ» және «РС» мәндері сөз тіркесі типіне жатады, бұларды біріктіріп жазуға ғана болады («ІВМ РС»). Қосуға, азайтуға, көбейтуге, бөлуге болмайды. Кез келген тұрақты, айнымалы функция немесе өрнек өзіне тән бір типпен ғана сипатталады. Паскаль тілінде шамалардың типін көрсету міндетті болып табылады. Сондықтан программа алдына оның сипаттау бөлімінде пайдаланылатын барлық шама атаулары және оның типтері көрсетілуі қажет. Паскаль тілінде пайдаланылатын барлық типтер скалярлық (қарапайым) және құрылымдық (структуралық) болып үлкен екі топқа бөлінеді. Скалярлық (қарапайым) типке шамалардың стандартты типі және жасанды тип (тағайындалған) жатады. Стандартты типтерге: • Бүтін - INTEGER; • Нақты - REAL; • Логикалық - BOOLEAN; • Тіркестік - (STRING); • Мәтіндік (ТЕХТ) тәрізді типтер жатады. Құрылымдық типтері жиымдар - ARRAҮ, жазбалар - RECORD, жиындар - SET және файлдар - FILE түрлеріне бөлінеді. Бұлардан өзге типтер программаның типтері сипаттау бөлігінде анықталуы қажет. Тұрақты сандардың типін олардың сыртқы пішініне қарап - ақ ажыратуға болады, ал айнымалылар мен функциялар типтері программаның сипаттау бөлімінде көрсетіледі. Ол үшін type түйінді сөзі қолданылып, жазылу пішімін былай көрсетуге болады: Туре <тип атауы> =<тип мәндері>; Логикалық шамалар екілік сандар жүйесіне сәйкес «1» және «О» деген мәндерді ғана қабылдайды. Мұндағы «1» тексерілетін белгілі бір шарттың - логикалық тұжырымның орындалатынын, яғни ақиқат екенін, ал «О» сол шарттың орындалмайтынын - тұжырымның жалған екенін көрсетеді. Бұл екі мән программада ағылшын тіліндегі айтылуына сәйкес true (1) және false (О) болып жазылады. Символдық шамалар апострофқа алынған бір символды ғана мән ретінде қабылдай алады, мысалы, 'R','Л','%'тағы басқа. Бірақ әрбір таңбаны оның кодына байланысты реттелген бүтін сандармен де өрнектеу мүмкіндігі бар. Мәліметтердің қарапайым түрлерін стандартты типтер деп атау қалыптасқан. Стандартты типтердің real түрінен өзгелері реттелген деп аталады, өйткені оларды бүтін сан арқылы өрнектей отырып өсуі не кемуі бойынша реттеп орналастыруға болады. Стандартты типтер. Паскаль тілінде төмендегі қарапайым типтер пайдаланылады: • Бүтін типтер - SHORTINT, INTEGER, LONGINT, ВУТЕ, WORD; • Нақты тип - REAL, SINGLE, DOUВLE, EXTENDED, СОМР; • Логикалық - BOOLEAN; • Симводық - НAR; • Ауқымды (диапозонды) тип; • Саналатын тип;[1] Мәліметтер қорының моделі деп мәліметтер қорының логикалық кейіптеуін атаймыз. Алғашқыда мәліметтер қорының - иерархиялық, - желілік, - реляциялық модельдері пайда болды, олар кейіннен классикалық, негізгі модельдерге айналды. Соңғы уақытта осылардың негізінде постреляциялық, - көпөлшемдік, - объектілі-бағытталған жаңа модельдер пайда болды және барынша дами отыра, қолданысқа енгізіле бастады.

**Мәліметтер типтері және модельдері.** Мәліметтер қоры мен қосымшалардың өзара орналасуларына байланысты келесі мәліметтер қорларын бөліп көрсетуге болады:  Локальдымәліметтер қорлары; Жойылғанмәліметтер қорлары.  Локальды  мәліметтер қорларыменоперацияларжасау үшінлокальды қосымшалар, ал жойылғанмәліметтер қорларыменоперацияларжасау үшін клиент – серверлік қосымшалардайындалады. Мәліметтер қорының орналасуыкөптегенжағдайда осы қорда бар мәліметтерді өңдейтін қосымшалардыңдайындалуына әсеретеді. Осылайша, келесі қосымшалартүрлерінбөліпкөрсетугеболады: Локальдымәліметтер қорынпайдаланатын қосымшалардыбірдеңгейлі (бірбуынды) қосымшалардепатайды. Себебі, қосымша мен мәліметтер қорыбіртұтасфайлдық жүйені құрайды; Жойылғанмәліметтер қорынпайдаланатын қосымшалардыекідеңгейлі (екібуынды) жәнекөпдеңгейлі (көпбуынды) қосымшалардепбөледі. Екідеңгейлі қосымшалардаклиенттікжәнесерверлікбөлшектер бар; Көпдеңгейлі (әдетте үшдеңгейлі) қосымшаларклиенттікжәнесерверлікбөлшектер мен қоса қосымшабөлшектергеие. Мысалға үшдеңгейлі қосымшалардаклиенттікбөлшек, қосымшаларсервері мен мәліметтер қорының сервері бар. БіржәнеекідеңгейліDelphi қосымшаларыкелесімеханизмдерді қолданаотырыплокальдыжәнежойылғанмәліметтер қорына қолжеткізеалады:

мөлшері. Деректер қорын ашу мен жабу. Oracle деректер қоры АШЫҚ (қол

оқуға рұқсат етеді (кімге сатылатындығы? - клиенттерге, сатуға төленетін

операциялардың деректерге сәйкес орындалуы. Oracle Деректер Қоры – бұл бірлік ретінде қарастырылатын деректердің

**оракл архитектурасы** Oracle Database 11g ДҚБЖ бірден-бір болып табылады, есептеуіштерді орнатуға немесе бөлуге арналған (Grid). Oracle Database 11g шығара, Oracle компаниясы қолданушыларға басқару мүмкіншілігін жеңіл жолын көрсетті, соған байланысты клиентер өздерінің бизнесін іске асырып инновацияларды еңгізді. Oracle Database 11g өндірімділіктің ең асқақ деңгейлерін қамсыздандырады, масштабтандырады, қолжетімділік, қауіпсіздік және басқарманың желідегі жайлылығын стандартты сервер және сақтау жүйесімен танымалы. Oracle Database 11g ДҚБЖ тиімді ұңғылау үшін арналған және әр түрлі типтегі базалармен және құрылғылармен жұмыс істеуге ыңғайлы.Бұл өнім автоматизацияланғандағы басқару функциясын ұсынады және эффективті экплуатациялауды ұсынады. SQL - Structured Query Language-дің қысқартылуы (сұраныстардың құры- лымдық тілі). SQL реляционды мәліметтер базасымен жұмыс істеу үшін жасал- ған. Ол қолданушыға мәліметтер базасымен жұмыс істеуге (қарау, іздеу, қосуға және ақпаратты басқаруға) мүмкіндік береді. MySQL ANSI 92 SQL специфика- сына сәйкес келеді. MySQL – интернеттегі ең кең таралған және ең атақты ақпарат базасын басқару жүйесі. Ол үлкен көлемді ақпаратпен жұмыс істеуге арналмаған, бірақ үлкен, кіші интернет сайттар үшін өте ыңғайлы жүйе. MySQL үлкен және орташа бағдарламалармен жұмыс істеуге өте ыңғайлы. Сервердің негіздері көптеген платформаларда компиляция жасайды. Сервердің толығырақ мүмкіндіктері өнімділікті арттыратын көпағымдылықтың қолдауы бар Unix-серверлерде байқалады. MySQL жұмыстың жоғары жылдамдылығымен, сенімділігімен, ыңғайлы- лығымен ерекшеленіледі. Онымен жұмыс істеу үлкен қиындықтар тудырмайды.Тегінділігі де үлкен роль атқарады. MySQL ортақ GNU (GPL, GNU Public License) лицензиясы негізінде таратылады. Бұрын ақпаратты ұзақ уақыт сақтау үшін файлдармен жұмыс істелетін: бірнеше жолдар санына енгізіп, келесі жұмыстар үшін алынатын. Ақпаратты ұзақ уақытқа сақтау мәселесі Web – бағдарламаларын программалауда жиі кездеседі: 31 қолданушыларды санау, хабарламаларды форумда сақтау, сайттағы ақпаратты арақашықтықтан басқару т.б.

**ОРАКЛ ДББЖ деректер қоймасы.** Құрылымдар – бұл деректер қорының деректерін сақтайтын тізбектер жиыны. Операциялар. Операциялар – бұл нақты анықталған әрекеттер. Бүтінділік тәртіптері. Бүтінділік тәртіптері – бұл қандай

**ОРАКЛ ДББЖ ПРОГР-Қ КОМПОНЕНТТЕРІ** мәліметтер қорын құруға арналған компоненттерге тоқталатын бол-сақ, мәліметтер қорымен байланыс Delphi-дің басқа да басқару элементтеріндей компоненттер визуалды және визуалды емес болып екіге бөлінеді. Визуалды емес компоненттер кестедегі мәліметтер қорына қатынауды ұйым-дастыруға арналады. Визуалды компонеттер қолданбаның интерфейстік бөлігін құруға арналады. Олардың көмегімен қолданушы мәліметтер қорындағы мәліметтерді қарап шығу, редакциялау сияқты операцияларды орын-дай алады. Мәліметтер қорымен жұмыс жасауға арналған компоненттер DataAccess, Data-Controls, QReport т.б. беттерінде орналасады.Мәліметтер көзі ретінде мәліметтер қорын, мәтіндік файлдарды, электрондық кестелерді т.б. қарастыруға болады. Мәліметтер қорына арнайы программалар, яғни мәліметтер қорын басқару жүйесі қызмет жасайды. Мәліметтер қорын басқару жүйесі бір ғана қолданушыға арналған жергілікті қолданбалар және компьютерлерде-серверлерде қызмет жа-сайтын серверлік болып екіге бөлінеді. Мұндай классификациялаудың негізі мә-ліметтер қорын басқару жүйесіне түсетін жүктеме мен мәліметтер қорының көлеміне байланысты. Жүзеге асыру түрлерінің әр қилылығына қарамастан мәліметтер қоры қолданбасының архитектурасы өзгеріссіз қалады.Мәліметтерді алу және жіберу механизмі мәліметтер көзімен қосуды қамта-масыз етеді.Мәліметтердің ішкі бейнелеу механизмі мәліметтер қоры қолданбасының ядросы болып табылады. Ол алынған мәліметтерді қолданбада сақтауды және қолданбаның басқа бөліктерінің сұраныс-тарына байланысты оларды мәліметтермен қамтамасыз етеді.Мәліметтер қоры қолданбасы қалай-ша жұмыс жасайды деген сұрақ төңірегінде сөз өрбітетін болсақ, Delphi репозиториясында мәліметтер қоры қолдан-басының өзіндік шаблоны жоқ.Сондықтан Delphi ортасында жүзеге асырылатын кез келген қолданба тәрізді мәліметтер қоры қолданбасы қарапайым формадан басталады

орындалуына жатпайтын деректерді жасыру керек. Тапсырысты формаға

пайдаланады. АЖ үшін ДБ құрғанда ұйымның эксперттері (менеджерлер, бухгалтерлер)

пайдаланушылардың қажеттілік үшін арналған және олардан міндеттерінің

**релициялық ДББЖ.** 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.

**Реляционды ДББЖ. 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.

**Реляциялық алгебра.** Негізгі амалдар Берілген деректер қорымен моделінің реляциялық базисі: реляциялық алгебра, реляциялық операциялардың интерпретациясы, теориялық-көбейткіштік операциялардың семантикасының ерекшеліктері. Реляциялық модельдің үшінші бөлігі, манипуляциялық бөлік реляциялық деректер қорымен жиынына қатынау реляциялық алгебра немесе оған эквивалентті реляциялық есептеулер арқылы іске асырылатынын тұжырымдайды. Нақты реляциялық МҚБЖ-да шындығында қазір реляциялық алгебра да реляциялық есептеулер де таза күйінде қолданылмайды. Реляциялық деректер қорымен жиынына стандартты қатынау SQL(Structured Query Language). SQL тілі реляциялық алгебраның операторлар қоспасынан және синтаксис қолданатын, ағылшын тілінің фразасына жақын реляциялық алгебра мен реляциялық есептеуде кездеспейтін реляциялық есептеу өрнектерінен тұрады. Берілген деректер қорыменге қатынау тілі, егер ол сипаттау күші жағынан реляциялық алгебрадан кем болмаса реляциялық толы деп аталады, яғни реляциялық алгебраның кез келген операторы осы тілдің құралдарымен сипатталуы мүмкін. SQL тілі осындай тіл болып саналады. Берілген бөлімде реляциялық алгебраның негізі қарастырылады. Реляциялық алгебраның тұйықтылығы. Реляциялық алгебра қатынастарды аргумент ретінде және қайтаратын қатынастарды нәтиже ретінде көрсететін операторлар жиынынан тұрады. Осылайша , реляциялық оператор f қатынастармен аргумент ретінде функция түрінде беріледі.R=f (r1,r2,…,rn). Реляциялық алгебра тұйық болып табылады, яғни аргумент ретінде реляциялық операторға типі бойынша сәйкес келетін басқа реляциялық операторларды қоюға болады. R=f(f1,(R11,R12,…),f2(R21,R22) Осылайша, реляциялық өрнектерде күрделі құрылымда енгізілген өрнектерді қолдануға болады. Деректер қорымен жиыны шегінде әр қатынас бірегей атқа ие болуы тиіс.Реляциялық операцияны орындау нәтижесінде алынған қатынас аты теңдіктің сол жағында анықталады. Бірақ та егер бұл қатынас аргумент ретінде басқа реляциялық өрнектерге қойылатын болса, реляциялық өрнек нәтижесінде алынған қатынастың аты болуын талап етпеуге болады. Мұндай қатынастарды атаусыз қатынастар деп атаймыз. Атаусыз қатынастар деректер қорымен жиыны базасында болмайды, ол тек қана реляциялық оператор мәнін есептеу уақытында есептеледі. Дәстүрлі түрде коддтық ізімен екі топқа біріккен, сегіз реляциялық оператор анықталады. Теориялық көпмүшелік операторлар: · біріктіру, · қию, · азайту, · декарттық көбейту. Арнайы реляциялық операторлар: · таңдама, · проекция, · жалғау(қосу), · бөлу. Бұлардың бәрі тәуелсіз болып табылады, яғни бірқатар операторлар басқа реляциялық операторлар арқылы өрнектеледі. Типі (түрі) бойынша біріктірілетін қатынастар. Бірқатар реляциялық операторлар, (мысалы, біріктіру) қатынастар бірдей тақырыпқа ие болуын талап етеді. Екі қатынасты біріктіру операциясы сәйкес қатынастар денесімен алынған екі көпмүшелік тізімдерін жәй қосу болып табылады.Нәтиже қатынас бола ала ма? Біріншіден, егер бастапқы қатынас түрлі атрибут санына ие болса, онда мұндай әртүрлі біріктірулердің тізімі болып табылатын көпмүшелікті қатынас түрінде көрсетуге болмайды. Екіншіден, қатынас атрибуттардың бірдей санына ие болғанымен де, атрибуттар әртүрлі атауға ие болады. Көптеген тізімдерді қосу нәтижесінде алынған қатынастар тақырыбын қалай анықтауға болады?Үшіншіден, қатынас бірдей атрибут санына, атрибуттар әр түрлі домендерге анықталған бірдей атауға ие болсын. Сонда тізімдерді қайта біріктіру қатынас түзбейді. 1 Анықтама. Бірігу қатынастарының типтері бойынша айтамыз, егерде оларда ұқсас тақырыптары болса, тереңірек айтатын болсақ, ● Қатынастар бір немесе бірнеше атрибут аттарына ие болса, яғни бір қатынастағы кез келген атрибутқа сондай атпен басқа қатынастарда ат табылады. ● Бірдей аттағы атрибуттар тек сол бір доменде-ақ анықталады. Кейбір қатынастар типтері бойынша бірге алмайды, бірақ атрибут атын өзгерткеннен кейін солай қалады. Реляционды операцияларда сондай қатынастарды қолдану үшін атрибуттың атын өзгертетін көмекші оператор енгізіледі.

**Реляциялық алгебра.** *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.

сақталынады. Осы деңгейдің сызбалары арасында бейне деп аталатын

сәйкестіктер бар. Бұл бір ұсыныстың екіншіге пайда болуының ережесі. СУБД деректерді енгізу және шығару сызбасын бейнелеушілерді

сипаттайтын E-R диаграммасын қарастырайық (рис. 6.4). оны оңайлаттыру

сондай-ақ, логикалық құрылымы болады. Деректер қорының физикалық құрылымы. Oracle деректер қорының

суреттеу үшін арналған. Олар тікбұрыш түрінде мәнің және олардың арасындағы қатынасты сызық ретінде көрсетеді. Бұндай қарапайым

счеттар туралы ақпараттардан тұрады. Сыртқы деңгейде осы сызбадан әр

сызба қызметкерлерге товардың жеткізуін қамтамасыздандыратын

сызбаның осы этаптарының арасындағы айырмашылықтарды түсіну қажет. Мысалы, концептуалды деңгейде деректер базасын жекелеп сатуын сипаттау

төлем және т.б. басқа қатынастар үшін әрі өзіңіз жалғастырасыз). Арықарай мәннің әрбіреуі атрибут-бағана және экземпляр-жолдардан

**Транзакцияны басқару**Транзакция механизмін қолдау - ДҚБЖ даму деңгейінің көрсеткіші. Транзакцияның сыпайы басқарылуы бір мезгілде басты тұтас базаны қамтамасыз етеді. Сондай-ақ базистері көп пайдаланатын жүйелерінен айырғыш пайдаланушы құрайды. Транзакция арқылы бөлінбейтін көзқарастан ДҚ манипуляциялық басты операторлардың жүйелілігі, яғни осындай транзакция шығатын барлық операторлардың нәтижесі ДҚ көрсетіледі немесе осы операторлардың әсер етуі түгелімен жоқ болады. Транзакцияның ұраны "Бәрі немесе ештеңе".Транзакцияның COMMIT операторымен аяқталуының нәтижесі кепілді түрде сыртқы жадыда белгіленеді. Транзакцияның ROLLBACK операторларының аяқталуының нәтижесі сыртқы жадыда жоқ болуымен түсіндіріледі.Тұтастықтың шек қойылуы екі түрмен ажыратылады: тез арада тексеру және кейінге қалдыру. Тез арада тексерушілік тұтастықтың шек қойылуына мыналар жатады, мағынасыз тексеру немесе кейінге қалдыру доменге шек қойылу (қызметкерлердің жасы 50 жастан асып кетпейді,күрделі шек қойу; тексерісті кейінге қалдыру мүмкін емес; келесі мына кездерде: қызметкерлердің ақысын бір операция үшін бір миллионнан көп өсе алмайды. Тез арада тексеру тұтастығы шек қойылуы тек операторлардың МҚБЖ тіл деңгейіне лайықты болады. Ол бұзылуда транзакцияның шегінуі тасылмайды, тек қана лайықты операторлар жоққа шығады. Кейінге қалдыру кедергісі тұтастықтың бұл түрінің қандайда бір жеке бөлінгіші операцияға емес, басты базаға кедергі жасайды. Осындай үндемей жасалған кедергілер транзакцияның соңында тексермейді. Олардың бұзылуы COMMIT операторлардың автоматты түрде ROLLBACK операторына ауысуын тудырады. Тек қана кейбір жүйелер арнайы операторлар арқылы транзакцияны ішкі шек қойылуына күшпен тексеріс жүргізеді. Егер осындай оператордың орындалуынан кейін, тұтастық шартты орындалмағаны ұсталса пайдаланушы өзі операторын орындай алады немесе ішкі транзакция нақты базаның тұтас емес жүйесінің себептерін жоюға тырысады. Тағы да бір пікір сыртқы көріністің көзқарасын транзакцияның барлық кейінге қалдырған тұтастары осы нақты базадағы анықталып тексеріледі. Тек қана идулизация кезінде транзакцияның орындалуына, яғни шын мәнінде бұзылуға шек қойылған тұтастықта динамикалық бөлуге тырысады. Мысалы: егер транзакцияның нақты қоймасында орындалғанда қызметкер бөлімінде үстеме оператор немесе қызметкерлердің байланысы арқылы игерудің алыстауы орындалмайды. Жоғарыдағы шек қойған тұтастықты еске салуды тексеруді қажет етпейді (ал нақты бүтіндікті тексеру шын мәнінде үлкен жұмысты қажет етеді).Транзакцияға негізгі үш қасиет тән болып келеді деп айтады: • атомарлық (транзакцияға кіретін барлық амалдар орындалады немесе бірде-бір амал орындалмайды); • топтамалануы (сериалдануы) (бір мезгілде орындалатын транзакциялардың өзара әсері жоқ); • ұзақ мерзімділігі (тіпті жүйе күйреген жағдайдың өзінде бекітілген транзакция қорытындылары жоғалып кетпейді).

тұрады. Төлем мен сату оқиғалары келісім болып табылатындықтан, олардын

тұратын таблица түрінде жазылады. Бірақ E-R диаграммасында бұл

түрлі бағыныңқы сызбалар алынуы мүмкін, олардың әр біреуі жеке

ұзындығы және т.б. туралы ақпараттарды сипаттайды, деректер тәрізді

үшін клиенттер, сату, кассалық операцияларды, сатушы мен кассир, тауар,

үшін оқиғаның тек қана екі түрі моделденеді: сатуды формалау және төлемді

физикалық құрылымы сервер компьютерлерінің мүмкіндігіне сәйкес. Деректер қорының логикалық құрылымы. Oracle деректер қорының логикалық құрылымы жазылған бағдарламаға сәйкес.Осындай тәсілді пайдалану кезінде ақпараттық жүйе деректерді сақтауды ұйымдастыру топтық немесе корпоративті түрде іске асады. Осындай талқылаулар классикалық транзакция түсінігінің пайда болуын туындатты. Ақпараттық даму ДҚБЖ-сі транзакция түсінігін қолдайды. Транзакцияларға қатысты тағы да бір үлкен емес ескерту. Бірінші рет қарағанда, транзакциялар түсінігі кез-келген уақытта тек қана корпоративтік ақпараттық жүйелерінде табиғи себептер бойынша деректердің жалпы қорын реттейді. Транзакцияларды басқару. Транзакция – бұл ДҚ-ғы операциялардың тізбектілігі. Әрбір транзакция ДҚ-ның бүтінділік жағдайында басталады. Көп қолданушылық ДҚБЖ-де транзакцияларды басқарумен транзакцияларды сериялау мен транзакциялар іске асыруына сәйкес орындалады. Мəліметтер қоймасын құру 1. Файл – СОЗДАТЬ БАЗУ командасын орындап,пайда болған сұхбат терезесінің ОБЩИЕ вкладкасының ішінен НОВАЯ БАЗА ДАННЫХ таңдап ОК батырмасын басу керек. Терезенің төменгі бөлігіне Файлдың атын PRIMER. mdb енгізіп, СОЗДАТЬ батырмасына басу керек. Мəліметтер қоймасының терезесі ашылады. 2. 1-ші кестені құру үшін: мəліметтер қоймасы терезесінде ТАБЛИЦЫ таңдап алып, маустың сол жақ батырмасын басып, СОЗДАТЬ батырмасын басу керек. Экранда НОВАЯ ТАБЛИЦА сұхбат терезесі пайда болады. КОНСТРУКТОР-ды таңдап алып, ОК басу керек. Экранда екі бөліктен тұратын кесте құру терезесі ашылады.Терезенің жоғарғы бөлігінде өріс аты, мəліметтер типі, сипаттамалар енгізіледі. Екіншісінде - өріс қасиеттерінің сипаттамалары беріледі (параметрлер).<F6> пернесін басу арқылы терезе бөліктерінің бірінен-біріне ауысуға болады. ПОЛЕ бағанының бірінші жолына клавиатурадан КОД МОДЕЛИ сөзтіркесін енгізіп, ENTER батырмасын басу керек. Осы жолдың ТИП ДАННЫХ бағанында таңдау тізімін ашып «Текстовый» ерекшелеп, маустың сол жақ батырмасын басу керек. Кестенің төменгі жақ бөлігіндегі РАЗМЕР ПОЛЯ үшін 6 санын енгізіп,ОБЯЗАТЕЛЬНОЕ ПОЛЕ үшін «Да», ПУСТЫЕ СТРОКИ үшін «Нет» Таңдау керек. ИНДЕКСИРОВАННОЕ ПОЛЕ үшін «Да». «Да» жəне «Нет» мəндері таңдау тізімінен алынады. Осы тəсілмен келесі өрістер мен олардың параметрлері енгізіледі. Мәліметтер қорын дұрыс жобалау әрқашан құрылған бағдарламаның тыйымсыз болуына әкеледі.Сондықтан МҚ құру алдында оның жобасына көңіл бөлу керек. Мәліметтер қорын құру төмендегі бөлімдер бойынша жүргізіледі: 1.МҚ-ның мақсаттарын анықтау. 2.МҚ-да қандай мәліметтерден(кестелерден) тұратыны туралы шешім қабылдау. 3.Кестелерге кіретін өрістерді анықтау және ерекше мәндерге кіретін өрістерді таңдау. 4.Кестелер арсындағы байланыстардың мақсаты. 5.Кестелер құру,оларды бір-бірімен байланыстардың мақсаты. 6. Енгізілген мәліметтерді форма,сұрату және есеп құру.