



Цифрлық әдіскер

ақпараттық-білім беру ортасы

ДӘРІСТЕР ЖИНАҒЫ

Деректер құрылымы және бағдарламаларды әзірлеу процестері

Деректер қорының теориясы, реляциялық модель және SQL тілі — ақпараттық жүйелерді жобалаудың негізі.

Деректер қорының теориясы

Реляциялық модель

SQL тілі

Ақпараттық жүйелер архитектурасы

ДӘРІС ЖИНАҒЫ

1-ДӘРІС.

Тақырыбы: Кіріспе.Деректер қорының пайда болу және даму тарихы. Лекция жоспары:

- Файлдық құрылым Деректері
- Деректер қорының – объектілі-бағдарланған Деректер қорының, білім қорының дамуы және болашағы.
- Деректер қоры. Негізгі ұғымдар.
- Деректер қорының анықтамасы.
- Ақпарат және Деректер қоры Құзыреттіліктер- Деректер қорына деректерді ендіру және олармен жұмыс жасай алуы менгеру Лекция мазмұны Деректер қоры ақпараттық технологияда үлкен роль атқарады және үлкен көлемді ақпараттарды автоматты түрде өңдеудің маңызды құралы болып табылады. Деректер қоры Деректермен жұмыс істейтін қарапайым және ыңғайлы құрал және өндірістің әр түрлі салаларында қолданылады. Деректер қорының мүмкіндіктері Деректер қорының реляциялық жүйесі теориялық тұрғыдан жасалынып және жүзеге асырылғаннан кейін едәуір артты. Объектілі-бағытталған Деректер қоры мен білім қорының жүйесі кең көлемде таралымға ие болып отыр. Объектілі-бағытталған Деректер қоры реляциялық Деректер қорын басқару жүйелері шеше алмайтын қиын есептерді шеше алады. Білім қорларындағы Деректер қорының реляциялық жүйелерінде Деректерден аса жоғарғы деңгейдегі ақпараттарды алуға мүмкіндік беретін предикаттарды есептеу логикасы қолданылады. Есептеу желілерінің дамуы үлестірілген Деректер қорының ролін арттырып отыр. Клиент/сервер технологиясы Деректерге орталықтандырылған қол жеткізу және басқарумен үлестірілген Деректерді өңдеуді білдіреді. Деректер қорының жалпы теориясы, негізгі анықтамалар, Деректер қорының файлдық құрылымды Деректерден айырмашылығы мен артықшылықтары Деректер қорларын практика жүзінде қолдану кезінде қалыптасқан Деректер қорының әралуан модельдері, Деректер қорларын концептуальды жобалаудың принциптері мен бейнелеу деңгейлері беріледі. Теориялық негізін реляциялық алгебра, реляциялық есептеу құрайтын, қазіргі уақытта ең көп тараған реляциялық модель жеке қарастырылады. Деректер қорларын пайдаланатын ақпараттық жүйелер. Оларды жергілікті ақпараттық жүйелер және бірнеше компьютерге таралған жүйелер деп айырады. Қолдану салалары бойынша ақпараттық жүйелерді өндірісте, білім беру саласында, денсаулық сақтау саласында, ғылымда, әскери істе, әлеуметтік салада, саудада және де басқа салаларда пайдалынатын жүйелерге бөлуге болады. Ақпараттық жүйелерді мақсаттық функция бойынша басқарушы, ақпараттық-анықтамалық, шешім қабылдауды қолдау сияқты негізгі категорияларға шартты түрде бөлуге болады. Файл/сервер және клиент/сервер сәулеті (архитектурасы) ақпараттық жүйелердің ең көп тараған сәулеттері болып табылады. Деректерге қашықтан қол жеткізу моделі (Remote Data Access – RDA)бар ақпараттық жүйелерде ақпаратты ұсыну функциясы мен қолданбалы өңдеу логикасын жүзеге асыратын бағдарламалар біріктірілген және компьютер-клиентте орындалады. Деректерді басқару қызметіне жүгіну SQL тілініңоператорларының көмегімен немесе API арнайы кітапханасының функцияларын шақырып жеткізу ортасы арқылы орындалады. Деректер қорының серверінің(DataBase Server – DBS) алдыңғы модельден айырмашылығы компьютер-клиент функциясы ақпаратты бейнелеу функциясымен ғана шектеледі де, ал қолданбалы функциялар компьютер-сервердегі қосымшалармен қамтамасыз етіледі. Бұл модель RDA-модельге қарағанда барынша технологиялық болып табылады. Деректер қорларын бағдарламалық қамтамасыз ету жалпыға арналған қамтамасыз ету, қолданбалы бағдарламалық қамтамасыз ету және Деректер қорларын жобалау мен жасауды автоматтандыру құрал-жабдықтары болып бөлінеді. Деректер қорларымен жұмыс істейтін ақпараттық жүйелердегі жалпыға арналған бағдарламалық қамтамасыз ету Деректер қорларын басқару жүйелері дегеніміз – бұл Деректер қорларын жасауға, жүргізуге, қорғауға және енгізуге арналған тілдік және бағдарламалық құрал-жабдықтар кешені. Деректер қорларын қолданбалы бағдарламалық ету дегеніміз – бұл нақты қолданбалы есептерді шешу үшін бағдарламалық қамтамасыз етуді жасаушы, немесе пайдаланушылардың өздері жасап шығарылатын бағдарламалар мен жүйелер. Delphi, C#, Visual Basic және тағы басқалар қазіргі заманғы бағдарламалардың алгоритмдік тілдерінде SQL тілін пайдаланатын Деректер қорларына қол жеткізу мүмкіндігін бағдарламалау құрал-жабдықтары бар. Оған қоса, көптеген МҚБЖ-де жүйеге қоса орнатылған бағдарламалау тілі бар. CASE-құралдары деп аталған CASE-технологиясын жүзеге асыратын бағдарламалық-технологиялық құралдар Деректер қорларын жасау мен ақпараттық жүйелерді жобалауды автоматтандыруға арналған. CASE (Computer Aided Software Engineering) терминін компьютер көмегімен бағдарламалық қамтамасыз етуді жасау деп аударуға болады. CASE-құралдары дегеніміз – бұл талаптарды талдау мен тұжырымдау, Деректер қорлары мен қосымшаларын жобалау, кодты генерациялау, тестілеу, сапаны қамтамасыз

ету, конфигурацияны басқару және тағы басқалар сияқты ақпараттық жүйелерді жасау мен ілесе жүру үдерістерін қолдайтын бағдарламалық құралдар. Төртінші бөлімде SQL тілі қарастырылады. SQL тілі – қазіргі заманғы көптеген реляциялық МҚБЖ-ның стандартты тілі болып табылады. SQL дегеніміз – Structured Queries Language сөздерінің қысқартылған түрі, ол сұраныстардың құрылымдалған тілі деп аударылады. SQL тілі реляциялық түрдегі Деректер қорларымен жұмыс істеуге арналған. Ол реляциялық МҚБЖ пайдаланушыларға ұсынатын барлық негізгі мүмкіндіктерді жүзеге асырады. SQL тілі командалар (операторлар) жүйелерінен тұрады, олардың ішіндегі ең маңыздылары төмендегілер: DML (Data Manipulation Language), Деректерді манипуляциялау тілі. SELECT – Деректерді оқуға сұраныс жасау; INSERT – Деректер қорына жаңа жазба қосу; DELETE – Деректер қорынан жазбаларды жою; UPDATE – Деректерді өзгерту. DDL (Data Definition Language), Деректерді 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. Файлдық құрылым Деректері. Қазіргі ақпараттық технологияларда Деректер қоры үлкен роль атқарады. Деректер қорының пайда болуынан бұрын файлдық құрылым Деректері пайда болған. Деректерді жинақтау, сақтау және өңдеудің тәсілдері, әдістері есептеу техникасының өзгеруімен қатар өзгеріп отырды. Бірінші және екінші буындардағы компьютерлерде Деректер файлдар түрінде сақталатын және бұл файлдардағы ақпаратқа қол жеткізу біртіндеп іске асырылатын, оның себебі жинақтағыштардың магниттік таспалар болуымен байланысты болды. Үшінші буындағы ЭЕМ-де магниттік барабандар мен дискілердің пайда болуымен файлдық құрылым Деректеріне қол жеткізу еркін іске асырылып, қажетті ақпаратқа тікелей қол жеткізуге мүмкіндік туды. Бұл Деректерді өңдеу тиімділігін едәуір арттырды. Үшінші және төртінші буындағы ЭЕМ-дерде бір мезгілде Деректер файлдарымен қатар Деректер қоры пайда болды. Файлдық құрылым Деректерінің Деректер қорына қарағанда бірнеше кемшіліктері бар. Файлдық құрылым Деректерінің негізгі кемшіліктері төмендегідей:

- Деректердің шектен тыс көп болуы;
- Деректердің нашар бақылануы;
- Деректерді басқарудың жеткіліксіз мүмкіндіктері;
- программистердің еңбек шығындарының көптігі болып табылады. Деректердің шектен тыс көп болуы файлдық құрылым Деректерін пайдаланатын қосымшалардың тек өз Деректерімен ғана жұмыс істейтіндігімен байланысты. Сондықтан әртүрлі қосымшалардың Деректер файлдары бірдей ақпаратты қамтуы мүмкін. Сонымен қатар, әртүрлі қосымшалардағы Деректер арасындағы үйлесімділікті сақтау үшін файлдың біріндегі ақпаратты өзгерткен жағдайда басқа файлдардағы сәйкес ақпаратты да өзгерту қажет. Деректер файлдарындағы ақпарат тұтастай алғанда өзара байланыспаған, бұл Деректерді бақылау ісін қиындатады. Деректер файлдарында бақылаудың осылайша әлсіз болуы әртүрлі қосымшалардың бірдей Деректерге әртүрлі мағына беруінде жатыр. Файлдардағы Деректердің арасында байланыстардың болмауы Деректерді жеткіліксіз басқаруға және тіпті басқарудың мүмкін еместігіне әкеледі. Қосымшалар мен Деректер файлдарын пайдалану барысында оларды өзгертіп, түрлендіру қажеттілігі туындайды. Ол үшін программаны қайтадан жазып, Деректерді құрастыру қажет, бұл программалаушылардың біраз еңбек етуін талап етеді. Деректер қоры. Негізгі ұғымдар. Ақпараттық технологияларды Деректер қоры үлкен роль атқарады. Деректер қоры дегеніміз – белгілі бір сипаттамасы (құрылымы) бар, өзара байланыса сақталатын ақпараттар жиынтығы. Деректер қоры бірнеше ортақ қасиеттері бар элементтер жиынынан тұратын нысандарды сипаттауға арналған. Мысалы, нысан ретінде өндірістік ұжым бола алады. Бұл нысанның элементтері (адамдар) тегі, аты, әкесінің аты, туған жылы, мекен-жайы және т.б. сияқты қасиеттерге ие. Деректер қорларында келесі ұғымдар қолданылады: өріс – нысанның белгілі бір қасиетін сипаттайтын ең кіші ақпарат бірлігі; домен – өрістің қабылдайтын мәндерінің жиыны; Деректер қорының құрылымы – өрістердің жиынтығы; жазба – нысанның бір элементі үшін өріс мәндерінің жиынтығы; Деректер қорының файлы – сипатталатын нысанның элементтеріне сәйкес келетін жазбалар жиыны. Деректер қорының файлдық құрылым Деректерінен басты айырмашылығы – Деректерді ұйымдастыру тәсілінде. Әдеттегі файлдар өзара байланыспаған Деректер элементтерінің қандайда бір тізбегі ретінде физикалық тұрғыда ұйымдастырылса, ал Деректер қоры берілген сипаттамаға сәйкес ұйымдастырылады және жекелеген элементтерінің арасында байланыс болады. Деректер қоры мен файлдың тағы бір айырмашылығы – Деректер қорының сипаттамасы сол мәліметер қорымен бірге сақталатындығында. Файл күрделі құрылымды болуы мүмкін, алайда бұл құрылымды анықтау сол файлдың ішінде емес, сол Деректер файлы жасалынған және қолданылған программаларда орналасады. Деректердің сипаттамасы Деректер қорының өзінде сақталатын болғандықтан, ол Деректер қорымен бірге тасымалданып, осы Деректер қорын пайдаланатын программалардың қарамағына беріледі. Файлдағы Деректерді сұрау кезінде, жалпы жағдайда сұралып отырған ақпарат көлемі мен орнын беру қажет, яғни физикалық ұғымдармен

жұмыс істейсіз. Деректер қорында Деректер қорының құрылымына сай ақпараттың белгілі бір типі, мысалы мекен-жайы сұралады. Бақылау сұрақтары:

- Деректер қоры ненің нәтижесінде пайда болды?
- Файлдық құрылым Деректерінен Деректер қорының негізгі айырмашылығы қандай?
- Алғашқы (түйінді) кілт дегеніміз не?
- Басқа жазбадан сілтеме жасайтын жазба қалай аталады?
- Сыртқы кілт дегеніміз не?
- Басқа жазбаға сілтеме жасайтын жазба қалай аталады? Әдебиеттер:
- Бидайбеков Е.Ы., Елубаев К., Шекербекова Ш.Т. Деректер қоры және ақпараттық жүйелер. Алматы., 2010. 2-ДӘРІС.

Тақырыбы: ДББЖ құрамы және құрылымы Лекция жоспары:

- Деректер базаларының қасиеттері, артықшылығы, тұтастығы, пайдаланушылар қатынасының шектелуі, деректер тәуелсіздігі.
- ДҚБЖ компоненттерін құру жобасы және жоспары Құзыреттіліктер- Деректер базаларының қасиеттері олардан туындайтын салдарларды білуі, релятивистік емес кванттық механиканың математикалық аппаратын игеруі Лекция мазмұны Деректер базасы ақпараттық жүйелердің маңызды құрамдас бөлігі болып табылады. Ақпараттық жүйе өз алдында басқару функциясын орындау үшін әр алуан деңгейдегі жұмыстарды ақпаратпен жабдықтаған объект туралы ақпаратты жинау, тасымалдау, қайта өңдеу бойынша қатынас жүйесін білдіреді. Деректер базасы деп компьютер жадында сақтайтын, арнайы түрде ұйымдасқан, өзара байланысқан Деректер жиынтығын айтамыз. Ақпараттық жүйелердің мысалына: банктік жүйелер, кәсіпорындарда, автоматты түрде, басқару жүйелері, авиация немесе темір жол билеттерін, мейрамхана номерлерін алдын ала белгілеу және тағы басқа жатады. Пайдалану облысы бойынша келесі класстарды бөліп айтуға болады:
- Ғылыми зерттеуге арналған ;
- Автоматталған жобаларға арналған;
- Ұйымдастырушы басқарудағы ақпараттық жүйелер;
- Технологиялық процессорды баспаға арналған. Деректер қорын басқару жүйесін көптеген қолданушылар деректер қорын құру, енгізу, бірлісіп пайдалану және программалау құралдарының жиынтығын айтады. Деректер базасы дегеніміз Деректерді сақтауға арналған ұйымдасқан құрылым. Бұл деректер қорының құрылымына ақпаратпен қатар, оларды ұйымдастыруға, қолдануға арналған тәсілдер мен әдістер кіреді. Деректер қоры деректер қорын басқару жүйесімен тікелей байланысты. Бұл программалық құрылымдар кешені жаңа қордың құрылымын құруға, оны деректермен толтыруға, мазмұнын редактірлеуге және ақпаратты визуалдандыруға арналған. Қор ақпараттарын визуалдау дегеніміз берілген критерий бойынша экранға шығаратын Деректерді таңдап, оларды белгілі бір ретке келтіріп, безендіріп артынан баспаға немесе байланыс каналдарына беруді айтамыз. Әлемде деректер қорын басқаратын көптеген жүйелер бар. Деректер базасын құру жүйесі бұл программа. Ол арқылы ақпарат деректер базасына енгізіледі, сұрыпталынады, сүзбеу жүргізіледі, 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. ДҚБЖ ДҚ-ға енгізілетін және ДҚ-дан шығарылатын Деректерді өңдеу процедураларынан және экрандық және есептік формалар жиынтығынан тұрады. Windows ортасында ДБЖ бағдарламалау кезінде ДҚБЖ-ны қосымша деп атайды. Экрандық формалар тізімін анықтау үшін қолданушы жұмысының тәртібін анықтаймы, яғни біздің жүйеміздің қолданушысының тәртібін береміз. Қолданушы жұмысының тәртібін келесі тәснм арқылы берейік. Қолданушы жұмысының тәсімі. Қолданушы интерфейсі бас меню арқылы жүзеге асырылады. Сондықтан жүйеде бас меню болу қажет. Қорға ақпаратты енгізу үшін бөлек экрандық форма қажет. Қорды бөлек экрандық формада қарастыру керек. Сонымен қатар қолданушы қорда баспа түріндегі мәліметті алу керек, әр түрлі Деректер әртүрлі формада болу қажет. Осы барлық шығыс Деректер есептік немесе есептеу формалары деп аталады. Сондықтан қолданушыға есептік формаларды алу қажет. Осылайша есептерді алу үшін қолданушының толық жұмысына қажет есеп формаларын қарастыру қажет. Қарастырылған формалар негізінде қолданушы жұмысының тәртібі келесі түрде іске асырылады. Жүйені іске қосқан кезде экранда жүйенің бас менюі көрінуі керек. Менюда жүйе орындай алатын команда іс-әрекеттер тізімі берілген. Егер қолданушы түскен материалдар туралы Деректерді енгізуді жоспарласа, ол менюден ақпарат енгізу үшін форма шақырады. Қорды алдын-ала қарау үшін ол менюден қорды қарау формасын шығарады. Қолданушы үшін есепте Деректерді шығару үшін әртүрлі нұсқаулар қарастыруға болады. Қолданушы бас менюден қорды алдын-ала қару формасын шақыра алады және одан Деректер мен есеп формасын таңдай алады. Генерацияланған есеп нұсқауы баспа

құралына беріледі. Иілгіштікке жету үшін ақпаратты енгізу режимінде және қорды алдын-ала қарау режимінде есепті қалыптастыру нұсқауын қарастыру қажет. Осылайша жоғарыда көрсетілген “қолданушы –жүйе” жұмысының тәсімі нде жүйе жұмыс істеу үшін келесі формалар қажет екені көрініп тұр:

- Меню үшін форма;
- Қорға Деректер енгізу үшін форма;
- ДҚ-ны алдын-ала қарау үшін форма;
- №1-есеп үшін форма
- №2-есеп үшін форма
- №3-есеп үшін форма Жоғарыда ДБЖ-ның сыртқы формасының көрсетілуі берілді. ДБЖ-ның жұмыс мүмкіншілігін қамтамасыз ету үшін ДҚ формасымен байланысты қамтамасыздандыру керек: экрандық формада ДҚ Деректерін өңдеу. Ол келесі пунктерде жасалады. ДҚБЖ компоненттерін құру жобасы және жоспары Сыртқы көрініс формасына басқа ДҚБЖ-да ДҚ-ға қатынау қамтамасыздың құралдары және өңдеу процедуралары болу керек. ДҚБЖ-ны құрайтын барлық компоненттер жиынтығы құрулы және сақтаулы болу керек. ДҚБЖ-ны жобалаудың негізі форма болып табылады. Форма барлық қалған компоненттердің негізі және тасымалдаушысы болып Деректерді өңдеу процестері үшін фундамент қызметін атқарады. Бұл тәсілдің теориялық негізі объектіге бағытталған технология жобасы және бағдарламалау болып табылады. Алайда объектіге бағытталған технология жобасы және бағдарламалаудың теориялық сұрақтарына арнайы тоқталмаймыз. Енді ДҚБЖ жобасын құрайық, яғни ДҚБЖ компоненттерінің арасындағы байланысты, құрамын және тағайындалуын бекітеміз. Ол үшін қолданушы жұмысының жоғарыда көрсетілген тәсімінен шығамыз. Қолданушы жұмысы үшін үш есеп формасы қажет. Сондықтан әрбір экрандық және септік форма үшін бөлек форма берілу керек. Delphi-де ДҚ және ДҚБЖ формалады арасындағы ақпараттық байланыс анықталған визуалды емес компоненттер арқылы іске асырылады. Сондықтан ДҚ-мен экрандық форма арасындағы байланысты қамтамасыз ету үшін жобада арнайы визуалды емес компонент (контейнер) TdataModule қолданылады, ол бөлек құрылу керек. Осылайша эвристикалық алдын-ала сілтемелеуден шыға біздің форма үшін қажет тізімді берейік:
- Меню үшін форма;
- Қорға Деректер енгізу үшін форма;
- ДҚ-ны алдын-ала қарау үшін форма;
- №1-есеп үшін форма
- №2-есеп үшін форма
- №3 есеп үшін форма
- TDataModule компоненті Жобалаушы үшін ең ыңғайлы форманың және олардың компоненттерін қарастыру тәртібі келесі болып табылады.
- TDataModule компоненті
- Меню үшін форма;
- Қорға Деректер енгізу үшін форма;
- ДҚ-ны алдын-ала қарау үшін форма;
- №1-есеп үшін форма
- №2-есеп үшін форма
- №3 есеп үшін форма ДҚБЖ құру үшін формалардың екі әдіспен беру керек:
- Егер алдын-ала олардың саны белгілі болса, монитор экранында және сәйкесінше IDE Delphi жобасының қоймасында (repository) алдын-ала визуалды және визуалды емес формалардың қажет санын шығаруға болады. File/New form командасы көмегімен визуалды формалар қосылады, ал File/New data module командасымен визуалды емес форма қосылады. Біздің жүйе алты визуалды форма және бір TDataModule компонентін талап етеді.
- Егер жоба үшін қажет формалар саны белгісіз болса, онда монитор экранында және сәйкесінше IDE Delphi жобасының қоймасына (repository) ДҚБЖ құрудың ағымды деігейінде жоба үшін тек қажетті форма қосылады. Экранда формаларды қою келесі команда арқылы іске асырылады. Басқа сөзбен айтқанда әрбір форма қажеттілік молшері бойынша қосылады. Бұл жағдайда жоба қоймасына жаңа форманы қосу (File/ Add to project) командасын қосу арқылы іске асырылады. Осылайша жобадан шығатын болсақ, Деректерді өңдеу процедурасы, экрандық және есептік формалардан жоғарыда қойылған есепті автоматтандыру үшін ДҚБЖ құру бойынша жұмыс жоспарын ұсынайық. Алдымен ДҚБЖ-ның барлық формаларын құру үшін қажетті формаларды экранға шығару керек болсын деп есептейік. Егер форма жетіспесе оларды қажетті санға дейін жеткізейік. Бұл (File/New form) іс-әрекетін форма саны қанша талап

етсе сонша рет қайталау және бір контейнерді File/New data module командасын қайталау жолымен жасалады. Енді форманың жобалау тізбегін анықтайық. Жалпы жағдайда формаларды құру тізбегі ретсіз, бірақ сонда-да біз жоғарыда сипатталған тізбекті тағдаймыз. Формаларды хаттау және құру тізбегі ДҚБЖ құру процесін жеілдетеді. ДҚБЖ құруды ДҚ-мен байланыс үшін TDataModule контейнерін құрудан бастаймыз. ДҚБЖ формасын жобалаудың экрандық формадан бастаймыз, онда ДҚ құрамы көрсетіледі. ДҚБЖ жобалауы ДҚ-дан алынатын немесе экрандық формаға енгізілетін Деректерді өңдеу процедураларын құру және ДҚ-мен байланыс жасаудың бөлек жолы бойынша әрі форма бойынша әске асырылады. Сонымен қатар әртүрлі форма компоненттері арасында байланыс жасау керек. Бір экрандық формада жүйе жұмысын аяқтау командасын қарастыру қажет. Компоненттер жиыны және олардың формада орналасуын анықтағаннан кейін олардың ДҚ-ға қатынаусы жасалады және форманың әр компоненті үшін Деректерді өңдеу процедурасы құрылады. Жоғарыда көрсетілген анықтамадан көреміз: ДҚБЖ ДҚ сияқты көптеген компоненттерден тұрады. ДҚБЖ-ның барлық компоненттері қалыптастыруды және бағдарламалуды талап етеді. Сондықтан оларды ДҚ сияқты анықталған орында сақтау керек. Бұл мақсат үшін C:\Primer каталогында APP қосалқы каталогын құру қажет. Онда біз өңделген қосымшааларды сақтаймыз. Жалпы жағдайда қосымшаны кез-келген жерде сақтауға болады, алайда, ДҚ-мен бірге бір папкада бірақ бөлек қосалқы директорияда сақтау жобалаушы жұмысы үшін ыңғайлы. Деректер қоры (Ms Access).Сіздер өздеріңіз білесіздер, ақпараттарды реттеу үшін ақпараттардың тақырыптары, өзара қисынды біріктірілген арнайы анықтамалар, каталогтар, картотекалар, энциклопедиялар және т.б. бар.Сонымен қатар, ақпараттардың ағымы үнемі ұлғаюда және бүгінгі күнде ешқандай анықтамалар ақпаратты үнемі және өз уақытында жаңартып отырмақ түгел, оны сыйғыза да алмайды.Бұл мәселені шешу үшін компьютерде есептеуді жүргізетін графикалық бейнелерді , мәтіндерді құруға мүмкіндік беретін программалардан басқа, үлкен көлемді ақпараттарды сақтау, өңдеу және іздеу үшін арнайы программалар бар. Мұндай программалар деректер базасын басқару жүйесі (ДББЖ) деп аталады. Әуелі біз деректер базасы дегеніміз не ? Соған тоқталайық. ¶ Қандай да бір тасушыда, мысалы, қағазда, кинотаспада, магниттік дискіде сақталған арнайы ұйымдастырылған деректердің жиынтығын деректер базасы деп түсінуге болады. Мысал келтіре кетсем, сіздің мектебіңізде 1000-ға жуық немесе одан да көп оқушылар оқиды делік. Әрбір мектепте әрбір оқушының деректері: аты-жөні , туған жылы, ұлты, жынысы, бойы және т.б. тіркелген медициналық карточка жүргізетін медициналық кабинет бар. Осы карточкалар шкафта былыққан түрде жатыр делік. Сіз өзіңізге қажетті карточканы табу үшін қанша уақыт кетеді. Әрине көп уақыт . Егер осы карточкаларды сыныптың сатысы бойынша (1-ші, 2-ші, ..., 11 сынып), содан кейін әрбір сатыны алфавиттік ретте, ал әрбір сатының нақты сыныбының ішінде алфавиттік ретте жайып қоятын болсақ, яғни картотека ұйымдастырудың, онда қажетті карточканы іздеуге бірнеше секунд керек болады. Бұдан біз оқушының медициналық деректерін топтастырудың өзі деректер базасы екенін көреміз. Деректер базасы негізінде үш белгісі бойынша жіктеледі. 1-ші белгісі сақталған ақпараттың сипатына қарай деректер базасы фактографиялық және құжаттық болып бөлінеді. Фактографиялық деректер базасы қатаң анықталған пішінде ұсынылған, жазылған объектілер туралы қысқаша Деректерден. Мысалы: кітапханадағы кітаптар қорының деректер базасында әрбір кітап туралы деректер библиографиялық түрде сақталады: шығарылған жылы, авторы, аты және т.б. ¶ Құжаттық деректер базасы әр түрлі типтегі : мәтіндік, графикалық, дыбыстық, мультимедиялық кең көлемдегі информациялардың тұрады. Мысалы: заңдық актілердің құжаттық деректер базасы заңдардың мәндерінен тұрады. 2-ші белгісі- ақпараттарды сақтау әдісі бойынша деректер базасы орталықтанған және бөлектелінген болып бөлінеді. Орталықтанған деретер базасында барлық ақпараттар бір компьютерде сақталады. Бөлектелінген деректер базасы компьютердің жергілікті және ауқымды желілерінде қолданылады және ақпараттың әр түрлі бөліктері бөлек компьютерде сақталуы мүмкін. 3-ші белгісі – ақпаратты ұйымдастыру құрылымы бойынша деректер базасы реляциялық, иерархиялық және желілік болып бөлінеді. Реляциялық деректер базасы деп өзінің құрамды бөліктерінің өзара байланысынан құрастырылған деректер базасын атайды. Қарапайым жағдайда ол бір тікбұрышты кесте құрайды. Күрделі реляциялық деректер базасы көптеген тікбұрышты кестеден тұрады. Бақылау сұрақтары:

- Орталықтанған деретер дегеніміз не?
- Бөлектелінген деректер дегеніміз не?
- Реляциялық деректер дегеніміз не?
- Реляциялық деректер дегеніміз не?
- Фактографиялық дегеніміз не?
- Құжаттық дегеніміз не?
- Деректерді ұсынудың тұжырымдамалық, логикалық және физикалық деңгейлері дегеніміз не? Ұсынылатын әдебиеттер:
- Нейлор К.Как построить свою экспертную систему / К. Нейлор; пер. с англ. – М.:Энергоатомиздат, 1991-286 с:ил.
- Элти Дж. Экспертные системы: концепции и примеры / Дж. Элти, М.Кумбс: перс англ.-М.: Финасы и статистика, 2009-191 с:ил.

Тақырыбы: Базаларды басқару жүйесі және құрамдары, іске қосу және меню жүйесі, ДББЖ жұмыс істеу үшін баптау.

Құзыреттіліктер: Зерттелетін құбылыстың физикалық табиғатын ескеретін гипотезаны қалыптастыра білу Лекция жоспары

1. Деректер базасы.
2. Архитектурасының түрлері. Лекция мазмұны Деректерді интеграциялау концепциясына негізделген қазіргі ақпараттық жүйелер үлкен көлемді және құрылымы жағынан күрделі болып келеді. Олар көптеген пайдаланушылардың әртүрлі деңгейдегі талаптарын қанағаттандыруы тиіс. Кеңінен түсіндірер болсақ, деректер базасы дегеніміз – шынайы өмірдегі белгілі бір сала бойынша бар объектілер жайлы мағлұматтар 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.
3. Жергілікті архитектура – программа және деректер базасы бір компьютерде орналасқан.
4. 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.
5. 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.
6. 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.
7. Интернет-архитектура – деректер базасы және ДББЖ бір компьютерде орналасады, ал оларға рұқсат алу желілік протокол стандарты бойынша браузерлер арқылы жүзеге асады. Деректер базасы База көшірмесі ДБ-мен жұмыс үшін қосымша ДБ ядросы Желілік сервер Қолданушы компьютері Деректер базасы Деректер базасының сервері ДБ (Клиент)-мен жұмыс үшін қосымша ДБ ядросы Қашықтатылған сервер Қолданушы компьютері

1.1-СУРЕТ. «ФАЙЛ-СЕРВЕР» АРХИТЕКТУРАСЫ 1.2-СУРЕТ. «КЛИЕНТ-СЕРВЕР» АРХИТЕКТУРАСЫ

Берілгендер қоры – ақпаратты сақтауға және жинақтауға арналған ұйымдасқан құрылым. Бірақ қазіргі кезде берілгендер қорын басқару жүйелері өздерінің құрылымдарына тек Деректерді ғана емес, сонымен қатар олардың тұтынушыларынмен және басқа да ақпараттық – программалық кешендермен қарым-қатынасының әдістерін қамтиды. Берілгендер қорын басқару жүйесі дегеніміз – жаңа Деректер қорын құруға, оны Деректермен толтықтыруға, оның мазмұнын редакциялауға, оларды реттеуге, безендіруге және баспа құрылғыларына шығаруға және байланыс арналарына жөнелтуге программалық құралдар кешені. Берілгендер қорын құрғанда информациялардың бір-бірімен қарама-қайшы болмауы керек, сондықтан бірінші кілтті қолданамыз. Бірінші кілт – информациялардың бірегей болуын қамтамасыз етеді. Берілгендер қорын басқару жүйесі – Деректер қорын құрады және сұранысты өңдейді өздерінің мүмкіншіліктерімен ерекшеленетін және бір-бірімен бәсекелесетін берілгендер қорын басқару жүйелері: Paradox, dBase, Microsoft Access, FoxPro, Oracle, InterBase, SyBase, Microsoft SQL Server және тағы басқа. Олар әртүрлі ұйымдастырылған және берілгендер қорын сақтайды. Мысалы: Paradox және dBase-те әр таблицаға жеке файл қолданылады. Бұл жағдайда

берілгендер қорындағы кесте файл сақталынатын каталог. Microsoft Access және InterBase-те бірнеше файлдар бір файл түрінде сақталынады. Бұл жағдайда берілгендер қоры – файл аты. Клиент-сервер түріндегі жүйеге SyBase және Microsoft SQL серверлері жеке компьютерде Деректерді сақтайды және клиентпен арнайы тіл арқылы (SQL) байланысады. Берілгендер қорының қасиеттері әртүрлі болғандықтан, егер ол қосымшаға (приложения) каталогтарды, файлдарды, серверлерді көрсететін болсақ, онда онымен жұмыс істеу қиынға соғады. Бұл проблеманы шешу үшін берілгендер қорының псевдонимі қолданылады. Псевдоним (alias) – берілгендер қорына кіруді қамтамасыз етеді. Бұл псевдонимді берілгендер қорын құрғанда бір рет хабарлайды. Берілгендер қорының администраторының информациясының псевдонимімен сәйкес келсе жеткілікті. Microsoft Access программасының 6 өзіне ғана тән объектісі бар: кесте, сұраныс, форма, есеп, макрос, модуль. Кестелер – жазбалар мен өрістерден тұратын БҚ-ның негізгі объектісі. Кестелерде берілгендер сақталады. БҚ жасау алғашқы кестені құрудан басталады. Ал, кестеде оның өрістері және сол өрістердің типтері мен қасиеттері анықталады. Бұл БҚ терезесіндегі Құру түймесін басудан басталады. Сонда, экранда Жаңа кесте сұхбат терезесі ашылады. Бұл терезеде жаңа кесте құрудың бірнеше тәсілдері көрсетілген. Сұраныс – берілгендерді бір немесе бірнеше кестелерден қолданушының анықтаған шарты бойынша іріктеп алуға арналған құрал. Сұраныстар көмегімен берілгендерді зерттейді, іріктейді, сорттайды, таңдайды, өзгертеді, біріктіреді, яғни өңдейді. Access-те сұраныс құрудың бірнеше тәсілдері бар. Олар Сұраныс жапсырмасындағы Құру түймесі басылғанда пайда болатын Жаңа сұраныс терезесінде көрсетілген. Форма – кестелер мен сұраныстардағы берілгендерді ыңғайлы түрде экранда бейнелейтін және оларды басқаратын құрал. Формалар – берілгендерді кесте түрінде немесе оларды экран бетіне қалауынша орналастыруға мүмкіндік береді. Оларды берілгендерді іздеу үшін де қолдануға болады. Сондай-ақ, қандай да бір құжаттарды дайындау кезінде де форманы қолдану қажеттілігі туындайды. Формаларды құру тәсілдерінің жеңілдігі, әртүрлілігі және олардың сыртқы түрлері Access программасының басқа БҚБЖ-нен артық болуына әсер етеді. Бұған формалармен жұмыс істеу барысында көз жеткізуге болады. Есеп – кестелер мен сұраныстардағы берілгендерді ыңғайлы және көрнекі түрде экран бетіне басып шығару құралы. Есеп парақтары түзілгеннен кейін ондағы берілгендерді редакцияларға сәйкес қоюға болады. Макрос – макрокомандалар жиынтығы. Егер БҚ-да қандай да бір операциялар жиі орындалатын болса, онда олардың командаларын бір макроста жиналып, клавиштер комбинациясын сәйкес қоюға болады. Модуль – Visual Basic – тілінде жазылған – программалық процедура. Егер Access-тің стандартты құралдары қажетті қажетті талапты қанағаттандыра алмаса, онда программалаушы процедуралар көмегімен жүйенің мүмкіндігін кеңейте алады. Access программасында құрылған формалар, сұраныстар және есептер берілгендерді жылдам және эффектілі жаңартыға, әр түрлі сұрақтарға жауап алуға, қажетті берілгендерді іздеуге, берілгендерді талдауға, есептер мен диаграммаларды баспаға бағыға мүмкіндік береді. Біл кестедегі берілгендерді әр түрлі объектілерде көруге болады және де кестеге өзгерту енгізгенде сол берілгендер орналасқан басқа объектілердің барлығында да олар өзгереді. Контексті мәзір деп объектімен байланысқан мәзірді айтамыз. Кестені редакциялау үшін, терезенің әрбір элементтеріне сәйкес көрсетілген контексті мәзірлерді қолдануға болады. Оларды қолданушы өз қалауымен қолдана алады. Деректер қоры – бұл ең алдымен кестелер жиынтығы, алайда, біз соңынан Деректер қорына сондай – ақ процедуралармен бірқатар басқа объектілер жатуы мүмкін екенін көреміз. Таблицаны көптеген объектілердің сипаттамалары бар кәдімгі екі өлшемді таблица түрінде көруге болады. Таблицаның идентификатор аты болады. Кесте бағандары объектілердің қандайда бір сипаттамаларына – алаңдарға сәйкес. Әрбір алаң сақталатын Деректердің атымен және түрімен сипатталады. Алаң аты – бұл Деректермен пайдалану үшін, әртүрлі программаларда қолданылатын идентификатор. Ол кез келген идентификатор сияқты латын әріптерімен жазылады, бір сөзден тұрады және т.б. Алаң түрі алаңда сақталатын Деректердің түрін сипаттайды. Бұл жолдар, сандар, үлкен мәтіндер (мысалы, қызметкерлердің мінездемелері), суреттер (қызметкерлердің суреттері) және т.с.с. болуы мүмкін. Кестенің әрбір жолы объектілердің біреуіне сәйкес келеді. Ол жазба деп аталады және осы объектіні сипаттайтын барлық алаңдардың мазмұнын құрайды. Деректер қорының кестелерін құрған кезде ақпараттың қарама – қайшылықсыздығын қамтамасыз ету маңызды. Әдетте бұл негізгі алаңдарды енгізумен жасалады. Негізгі болып бір немесе бірнеше алаң болуы мүмкін. Кестемен жұмыс кезінде пайдаланушы немесе программа жазбалар үстінен курсормен жылжып отыратын сияқты. Әрбір уақыт сәтінде ағымдағы жазба бар, содан жұмыс жүргізіледі. Деректер қорының кестелерінде жазбалар қандай да бір тәртіпсіз, оларды енгізу (жаңа қызметкерлердің пайда болуы) ретіне қарай орналасуы мүмкін. Бірақ кесте Деректері пайдаланушыға ұсынылған кезде реттелуі керек. Пайдаланушы оларды алфавиттік тәртіпке, немесе бөлімдер бойынша реттелген, немесе туған жылының өсуіне қарай және т.б. қарауы мүмкін. Деректерді реттеу үшін индекс түсінігі қолданылады. Индекс кестені қандай реттілікте қараған жөн екенін көрсетеді. Ол пайдаланушы мен кесте арасында делдал болып табылады. Деректер қоры әдетте бір емес, көптеген кестелерден тұрады. Мысалы, кейбір ұйым туралы Деректер қорында әрбір бөлімшенің сипаттамасы бар кесте бар. Байланысқан кестелерде әдетте бір кесте басты болады, ал басқасы немесе бірнеше басқа кестелер – көмекші кестелер болады. Басты және көмекші кестелер бір – бірімен кілт арқылы байланысады. Кілт ретінде екі кестеде де болатын алаң шыға алады. Мысалы, жоғарыда келтірілген кестелерде басты болып Дер кестесі, көмекші болып Pers кестесі бола алады, ал олар болса екі кестеде де бар. Дер алаңы бойынша байланысады. Курсор басты кестенің индексі үстінен жүреді. Осылайша біздің мысалда егер басты кестенің Дер алаңындағы ағымдағы жазбада «бухгалтерия» деп жазылған болса, онда көмекші Pers кестесінде бухгалтерия қызметкерлерінің барлық жазбалары

шығады. Бақылау сұрақтары: 1.Деректер базасы не үшін қажет? 2.Деректер базасын басқару жүйесі дегеніміз не? 3.Деректер базасын құрастыру жолдары қандай? 4.MS Access бағдарламасын іске қосу жолы Ұсынылатын әдебиеттер: 1.Профессиональное руководство по SQL Server.Структура и реализация Чендерсон.- Изд. "Вильямс". 2006. 2.Кульфельд Дж. Access2002 для чайников М. Вильямс 2002 3.Руководство разработчика баз данных в Delphi Kylix. Э. Хармон. - Изд. Питер, 2008.

4-ДӘРІС

Тақырыбы: Ақпараттық жүйелер және Деректер қоры

Лекция мазмұны

- Ақпараттық жүйелердің архитектурасы.
- Ақпараттық жүйелердің классификациясы.
- Жергілікті ақпараттық жүйелердің классификациясы.
- Жергілікті ақпараттық жүйелер және компьютерлік желілердегі ақпараттық жүйелер.
- Ақпараттық жүйедегі Деректер қоры. CASE-технология. Құзыреттіліктер- Кестелер арасындағы байланысты орнатуды меңгерт

Лекция мазмұны Автоматтандырылған ақпараттық жүйелер (АЖ) деп техникалық құралдарды, атап айтқанда ЭЕМ қолданатын жүйелерді атайды. Қолданыстағы АЖ-лердің көпшілігі автоматтандырылған болып келеді, сол себептен оларды қысқаша АЖ-лар деп атайтын боламыз. Ақпараттық жүйе деп ақпаратты өңдеуге арналған есептеу кешені аталады. Ақпараттық жүйе келесілерден: компьютерлік құрал-жабдықтан; программалық қамтудан; Деректерден;

қызмет көрсетушілер құрамы (пайдаланушылар, әкімші) құралады.

Жергілікті ақпараттық жүйе деп бір компьютерде орнатылған және сонда жұмыс істейтін жүйе аталады. Клиент/сервер типіндегі компьютерлік желіде жұмыс істейтін ақпараттық жүйелер барған сайын кең таралуда. Бұл жүйелер Деректерді орталықтандырылған, сондай-ақ үлестірілген түрде өңделуді қамтамасыз етеді. Бұл жағдайда жалпы Деректер серверлерде орналастырылады, ал клиенттік компьютерлерде жеке клиенттердің Деректері орналасады. Деректер қорымен жұмыс істейтін ақпараттық жүйелерді программалық қамту төмендегілерден:

- жалпыға арналған программалық қамту;
- қолданбалы программалық қамту;
- Деректер қорын жобалауды автоматтандыру құралдарынан құралады. Деректер қорын ақпараттық жүйелерде қолданудың үлкен артықшылығы – Деректер қоры Деректердің клиенттік бағдарламалардан тәуелсіздігін қамтамасыз етеді, пайдаланушылар Деректерді тасымалдағыштарда физикалық орналастыру мен оларға қол жеткізу мүмкіндігін ұйымдастыру мәселесінен құтылады. Кең мағынада АЖ анықтамасына ақпаратты өңдеудің кез келген жүйесі жатады. Қолдану саласы бойынша АЖ-ларды өндірісте, білім беруде, денсаулықты сақтауда, ғылымда, әскери істе, әлеуметтік салада, сауда-саттықта және өзге де салаларда қолданылатын жүйелерге бөлуге болады. Мақсаттық қызметі бойынша АЖ-ларды шартты түрде келесі негізгі категорияларға бөлуге болады: басқарушы, ақпараттық-анықтамалық, шешімдерді қабылдауды қолдау. Кейбір жағдайларда АЖ ұғымы қандайда бір қолданбалы мәселені шешуге пайдаланылатын аппараттық-программалық құралдардың жиынтығы деген өте тар түсініктеме түрде қолданылады. Ұйымдастыруда, мысалы, өздеріне сәйкес келесі міндеттер жүктелген ақпараттық жүйелер болуы мүмкін: мамандар мен материалдық-техникалық құралдардың есебін жүргізу, жеткізушілер мен тапсырыс берушілермен есеп айыру, бухгалтерлік есеп және т.б. Деректер банкі АЖ-ның бір түрі болып табылады, мұнда бір немесе бірнеше Деректер қорына ұйымдастырылған ақпаратты орталықтандырылған түрде сақтау мен жинаудың қызметтері жүзеге асырылған. Деректер банкі (МБ) жалпы жағдайда келесі компоненттерден: Деректер қорынан (бірнеше қорынан), Деректер қорын басқару жүйесінен, Деректер сөздігінен, әкімшіден, есептеу жүйесі мен қызмет көрсетуші персоналдан тұрады. Аталған компоненттер мен солармен байланысты маңызды түсініктерді қысқаша түрде қарастырып кетейік. Деректер қоры (МБ) есептеу жүйесінің жадында сақталатын және объектілер мен қарастырылып отырған пәндік салада олардың өзара байланысының жағдайын бейнелейтін, арнайы түрде ұйымдастырылған Деректердің жиынтығы болып табылады. Қорда сақталатын Деректердің логикалық құрылымы Деректерді кейіптеудің моделі деп аталады. Деректерді кейіптеудің негізгі модельдеріне (Деректер моделіне) келесілер жатады: иерархиялық, желілік,

реляциялық, постреляциялық, көпөлшемді және объектілі-бағытталған. Деректер қорын басқару жүйесі (МҚБЖ) дегеніміз – бұл МҚ-ны құруға, енгізуге және оны көптеген пайдаланушылардың бірлесе отырып пайдалануына арналған тілдік және программалық құралдардың кешені. Әдетте МҚБЖ Деректердің қолданылатын моделі бойынша ажыратылады. Мысалы, реляциялық Деректер қорын пайдалануға негізделген МҚБЖ-лар реляциялық МҚБЖ-лар деп аталады. Алғашқы МҚБЖ-ларға келесілер жатады: IMS (IBM, 1968 ж.), IDMS (Cullinet, 1971 ж.), ADABAS (Software AG, 1969 ж.) және ИНЭС (КСРО АФ ВНИИСИ, 1976 ж.). Қазіргі заманғы Деректер қорын басқару жүйелері мыңдап саналады. Қосымша дегеніміз қолданбалы міндеттер үшін ақпаратты өңдеуді автоматтандыруды қамтамасыз ететін программа немесе программалар кешені болып табылады. Біз МҚ-н пайдаланатын қосымшаларды қарастырып отырмыз. Қосымшалар МҚБЖ ортасында немесе одан тыс ортада – МҚ-ға қол жеткізу құралдарын, мысалы, Delphi немесе C++ Builder пайдаланатын программалау жүйесінің көмегімен құрылуы мүмкін. МҚБЖ ортасында жасалған қосымшалар көбінесе МҚБЖ қосымшалары деп, ал МҚБЖ-дан тыс ортада жасалған қосымшалар сыртқы қосымшалар деп аталады. Деректер қорымен жұмыс істеу үшін көбінесе МҚБЖ құралдарын қолданған жеткілікті болады және өздерін құруға программалауды қажет ететін қосымшаларды пайдаланудың қажеті болмайды. Қосымшалар негізінен біліктілігі жоқ пайдаланушыларға МҚ-мен жұмыс істеу ыңғайлы болуы үшін немесе МҚБЖ интерфейсі пайдаланушыларды қанағаттандырмаған жағдайларда жасалады. Деректер сөздігі (МС) Деректердің құрылымы туралы, МҚ файлдарының өзара әрекеттері туралы, Деректердің типтері мен оларды ұсынудың форматтары туралы, Деректердің пайдаланушылардың меншігіне жатуы, қорғау және қол жеткізуді шектеу кодтары және т.б. туралы ақпаратты орталықтандырылған түрде сақтауға арналған МҚ ішкі жүйесі болып табылады. Функционалды түрде МС барлық МБ-ларда болады, алайда осы қызметтерді атқаратын компонент әрдайым дәл сондай атқа ие бола бермейді. Көбінесе МС қызметін МҚБЖ орындайды және олар жүйедегі негізгі менюден шақыртылып немесе соның утилиталары арқылы жүзеге асырылады. Деректер қорының әкімшісі (МҚӨ) дегеніміз – МҚ-ға қатысты талаптарды даярлау ісіне, оны жобалауға, құруға, тиімді түрде пайдалануға және ілесіп жүруге жауап беретін тұлға немесе тұлғалар тобы. Пайдалану барысында МҚӨ әдетте ақпараттық жүйенің жұмысын қадағалайды, рұқсат етілмеген қол жеткізуден қорғауды қамтамасыз етеді, МҚ-да сақталатын ақпараттың көптігін, қарама-қайшылықсыздығын, сақтанымдылығы мен дәйектілігін бақылайды. Бір пайдаланушыға арналған ақпараттық жүйелерде МҚӨ қызметтері, әдетте тікелей МҚ қосымшаларымен жұмыс істейтін тұлғаларға жүктеледі. МҚӨ есептеу желісінде, әдетте, желінің әкімгерімен өзара әрекеттеседі. Соңғысының міндеттеріне желінің аппараттық-программалық байланыс құралдарының жұмысын бақылау, желіні қайта құру, тоқтап қалулар мен істен шығулардан кейінгі программалық қамтуды қалпына келтіру, алдын-алу шаралары мен қол жеткізудің бөлініп шектелуін қамтамасыз ету ісі жатады. Есептеу жүйесі (ЕЖ) өзара байланысқан және келісілген түрде әрекет ететін ЭЕМ-дердің немесе процессорлар мен өзге де құрылғылардың жиынтығы болып табылады, ол ақпаратты қабылдау, өңдеу мен оны пайдаланушыларға ұсыну процестерін автоматтандыруды қамтамасыз етеді. МБ-ның негізгі қызметтеріне Деректерді сақтау мен өңдеу ісі жататын болғандықтан, қолданылатын ЕЖ орталық процессорлардың (ОП) оңтайлы қуаттылығымен қатар тікелей қол жеткізудің жеткілікті көлемдегі жедел және сыртқы жады болуы тиіс. Қызмет көрсетуші персонал техникалық және программалық құралдарды жұмысқа қабілетті қалпында ұстап отыру қызметін орындайды. Ол жоспар бойынша, сондай-ақ қажетінше алдын-алу, регламенттік, қалпына келтіру және өзге де жұмыстарды атқарады. Ақпараттық жүйенің архитектурасы. Ақпараттық жүйенің (АЖ) тиімді қызмет етуі көп жағдайда оның архитектурасына байланысты. Қазіргі уақытта клиент-сервер архитектурасының болашағы зор болып отыр. Оның жеткілікті түрде таралған нұсқасында компьютерлік желі мен үлестірілген Деректер қоры болады, соңғысында корпоративтік Деректер қоры (КМҚ) мен дербес Деректер қоры (ДМҚ) орналасады. КМҚ компьютер-серверде орналасады, ДМҚ корпоративтік МҚ клиенті болып табылатын бөлімдер қызметкерлердің компьютерлерінде орналасады. Компьютерлік желідегі белгілі бір ресурстың сервері деп осы ресурсты басқаратын компьютер (программа) аталады, ал клиент деп осы ресурсты пайдаланатын компьютер (программа) аталады. Компьютерлік желінің ресурсы ретінде мысалы, Деректер қоры, файлдық жүйелер, баспа қызметтері, пошта қызметтері алынады. Сервердің типі ол басқаратын ресурстың түрімен анықталады. Мысалы, егер басқарылатын ресурс Деректер қоры болып табылса, онда сәйкес сервер Деректер қорының сервері деп аталады. Ақпараттық жүйені клиент-сервер архитектурасы бойынша ұйымдастырудың артықшылығы пайдаланушылардың дербес ақпаратпен жеке түрде жұмыс істеуімен орталықтандырылған түрде сақтау, қызмет көрсету және жалпы корпоративтік ақпаратқа ұжымдық түрде қол жеткізуді сәтті түрде үйлестіру болып табылады. Клиент-сервер архитектурасында оны жүзеге асырудың түрлі нұсқаларына жол беріледі. Тарихи тұрғыдан алғашқы болып файл-серверін қолданумен үлестірілген АЖ-лар пайда болды (2.1-сурет). Мұндай АЖ-ларда пайдаланушылардың сұраныстары бойынша Деректер қорының файлдары дербес компьютерлерге (ДК) беріледі, ол жерде олар өңдеуден өткізіледі. Архитектураның бұл нұсқасының кемшілігі өңдеуден өткізілетін Деректерді берудің жоғары қарқындылығы. Оның үстіне көбінесе артық Деректер беріледі: пайдаланушыға Деректер қорындағы жазбалардың қаншасы қажет екендігіне тәуелсіз Деректер қорының файлдары тұтастай беріледі. Деректер қорының серверін пайдалану арқылы клиент-сервер архитектурасы бойынша құрылған үлестірілген АЖ-ның құрылымы 2.2-суретте келтірілген. Мұндай

архитектура болғанда Деректер қорының сервері Деректерді өңдеуден өткізудің басым бөлігінің орындалуын қамтамасыз етеді. Пайдаланушы немесе қосымша құрастыратын сұраныстар МҚ серверіне SQL тілінің нұсқаулықтары түрінде келіп түседі. Деректер қорының сервері керекті Деректерді іздестіріп, іріктеп алады, содан кейін олар пайдаланушының компьютеріне беріледі. Бұл тәсілдің алдыңғымен салыстырғандағы артықшылығы берілетін Деректер көлемінің едәуір мөлшерде азаюы.

КМҚ Файл-сервер

МҚБЖ ДМҚ

МҚБЖ ДМҚ

желілік орта 1 клиенттік ДҚ n клиенттің ДҚ . . .

2.1-СУРЕТ. ФАЙЛ-СЕРВЕРІ БАР АЖ ҚҰРЫЛЫМЫ

Клиент-сервер архитектурасы бойынша үлестірілген МҚ-ны құрудың негізгі нұсқалары «Клиент-сервер архитектурасы» бөлімінде қарастырылатын болады. Дербес МҚ-лар мен олармен жұмыс істейтін қосымшаларды құру және басқару үшін Microsoft фирмасының Access және Visual FoxPro, Borland фирмасының Paradox тәрізді МҚБЖ-лары қолданылады. Корпоративтік МҚ, мысалы Microsoft SQL Server немесе Oracle Server тәрізді МҚ серверінің басқаруымен құрылып, қолданылады, қызмет етеді. Шешілетін міндеттерді ұйымдастыру мөлшері мен ерекшеліктеріне қарай ақпараттық жүйе келесі конфигурациялардың біріне ие болуы мүмкін:

МҚ-нан мәліметтер беру КМҚ МҚ Сервері

МҚБЖ ДМҚ

МҚБЖ ДМҚ

желілік орта 1 клиенттік ДҚ n клиенттің ДҚ . . .

2.2-СУРЕТ. ДЕРЕКТЕР ҚОРЫ СЕРВЕРІ БАР АЖ ҚҰРЫЛЫМЫ

- корпоративтік және дербес қорлардан тұратын компьютер-сервер;
- ДМҚ бар компьютер-сервер мен дербес компьютерлер;
- ДМҚ бар бірнеше компьютер-серверлер мен дербес компьютерлер. Клиент-сервер архитектурасын пайдалану біріншіден, кәсіпорынның даму барысында, екіншіден, ақпараттық жүйенің өзінің дамуы барысында кәсіпорынның ақпараттық жүйесін біртіндеп ұлғайтуға мүмкіндік береді. Жалпы МҚ-ны корпоративтік МҚ мен дербес МҚ-ларға бөлу орталықтандырылған нұсқасымен салыстырғанда МҚ-ны жобалаудың күрделілігін төмендетуге, сол арқылы жобалау кезінде қателер жіберіп алудың ықтималдығы мен жобалаудың бағасын төмендетуге мүмкіндік береді. МҚ-ларды ақпараттық жүйелерде қолданудың аса маңызды артықшылығы – Деректердің қолданбалы программаларға тәуелсіздігін қамтамасыз ету. Бұл пайдаланушыларға физикалық деңгейде Деректерді кейіптеу мәселелерімен Деректерді жадыда орналастыру, оларға қол жеткізу әдістері және т.б. айналыспауға мүмкіндік береді. Мұндай тәуелсіздікке МҚБЖ-мен қолдау көрсетілетін МҚ-да логикалық (пайдаланушылық) және физикалық деңгейлерде Деректерді көп деңгейлік кейіптеу арқылы қол жеткізіледі. МҚБЖ мен Деректерді кейіптеудің логикалық деңгейінің болуы арқасында МҚ-ның тұғырнамалық моделін (түсініктік) оның ЭЕМ жадында физикалық кейіптеуден бөліп алу қамтамасыз етіледі. Жергілікті ақпараттық жүйелер. Ақпараттық жүйенің функционалдық бөлімдері бір немесе бірнеше компьютерлерде орналасуы мүмкін. Бір компьютерде АЖ-ны ұйымдастыру нұсқаларын қарастырайық. Сәйкес АЖ әдетте жергілікті немесе бір пайдаланушылық деп аталады (алайда соңғысы қатаң шарт болып табылмайды, себебі бір компьютерде кезекпен бірнеше пайдаланушы жұмыс істеуі мүмкін). Қандайда бір операциялық жүйенің (ӨЖ) ортасында бір компьютерде жергілікті АЖ-ның қызметін ұйымдастыру программалық құралдарды қолданудың келесі нұсқалары арқылы мүмкін болады: «толық» МҚБЖ; қосымша және «кесілген» (ядро) МҚБЖ; тәуелсіз қосымша. Алғашқы тәсіл әдетте компьютердің дискілік жадында бүкіл МҚБЖ орналасқан жағдайда

қолданылады және ол көбінесе қосымшаны аяғына дейін өңдеуден өткізу үшін қолданылады (2.3-сурет). Пайдаланушының МҚБЖ-мен өзара әрекеттесуі тікелей МҚБЖ-ның пайдаланушылық интерфейсі арқылы немесе қосымшаның көмегімен орындалады. Қосымша интерпретациялау режимінде орындалады. Пайдаланушы Қосымша

МҚБЖ МҚ

2.3-СУРЕТ. МҚБЖ ҚОЛДАНБАСЫН ПАЙДАЛАНУ.

Сұлбаның басты артықшылығы – дамытылған сәйкес жасау құралдары мен сервистік құралдар болған жағдайда МҚ мен қосымшаны жасау мен оларға ілесе жүрудің қарапайымдылығы. Бұл жүйенің кемшілігі – МҚБЖ программасын сақтауға жұмсалатын дискілік жадының шығындары.

МҚБЖ ядросы бар қосымша (2.4-сурет) келесі мақсаттарға жету үшін қолданылады:

Пайдаланушы Қосымша МҚБЖ ядросы МҚ

2.4-СУРЕТ. МҚБЖ ҚОСЫМШАСЫ МЕН ЯДРОСЫН ҚОЛДАНУ.

Бұл тәсілдің мысалы FoxBase+ жүйесінің FoxRun модулін пайдалану болып табылады. Қазіргі заманғы МҚБЖ-дың арасынан Microsoft Access-ті атап кетейік, оған Microsoft Access Developer's Toolkit қосымша пакеті кіреді. Оның көмегімен дискеталарда тасымалданатын, өзінде әзірлеу құралдары жоқ Microsoft Access-тың «қысқартылған» (run-time) версиясын құруға болады. МҚБЖ-ның толық версиясын қолданумен салыстырғанда МҚБЖ ядросын қолданудың артықшылығы: компьютер жадының ресурстарын аз мөлшерде қолдану, қосымша жұмысының үдеуі және қосымшаны модификациялаудан қорғау мүмкіндігі. Негізгі кемшіліктерге МҚБЖ ядросын сақтауға қажетті дискілік жадының әлі де елеулі көлемі, және қосымшалардың жұмыс істеу жылдамдығының жеткілікті дәрежеде жоғары еместігі жатады (қосымшаны орындау бұрынғысынан интерпретациялау арқылы жасалады). АЖ-ны ұйымдастырудың үшінші тәсіліне бастапқы программа алдымен компиляцияланады – орындалатын машиналық командалардың тізбегіне түрленіп өзгертіледі. Соның нәтижесінде өзінің қызмет етуі үшін не бүкіл МҚБЖ-ны, не оның ядросын қажет етпейтін орындауға дайын тәуелсіз программа пайда болады (2.5-сурет). Айта кететін нәрсе, Деректерді сақтау мен өңдеудің негізгі қызметтерін орындау тұрғысынан алып қарағанда бұл программа МҚБЖ-ның немесе оның ядросымен басқару арқылы орындалатын қосымшадан аздап қана ерекшеленеді.

Пайдаланушы Қосымша

МҚБЖ МҚ

2.5-СУРЕТ. ТӘУЕЛСІЗ ҚОСЫМШАНЫ ПАЙДАЛАНУ.

Бұл нұсқаның алдыңғы екеуімен салыстырғандағы негізгі артықшылығы – компьютердің сыртқы және жедел жадының үнемделуі, қосымшаны орындаудың үдеуі және қолданбалы модификациялаудан толықтай қорғалуы (дизассемблерлеу жағдайы мен өзінің кодын қою және сол тәрізді жағдайлар есепке алынбайды). Кемшіліктеріне қосымшаларды аяғына дейін өңдеудің көп еңбекті қажет ететіндігін және МҚ-ны күтімге алу жөніндегі МҚБЖ-ның стандартты құралдарын пайдалану мүмкіндігінің болмауын жатқызуға болады. Клиент-сервер архитектурасының модельдері. МҚ-мен жұмыс істейтін үлестірілген АЖ-ларды құрған кезде клиент-сервер архитектурасы кең түрде қолданылады. Оның негізін МҚ-ны басқару кезінде клиент мен сервердің өзара әрекеттесуін ұйымдастыру ұстанымдары құрайды. Клиент-сервер архитектурасының бір нұсқасы 1.2-ішкі тармағында қарастырылған. Жоғарғы екі деңгейге тоқталып кетейік: қолданбалы және өкілдік, оларға жасаушы мен пайдаланушы тарапынан аса үлкен көңіл бөлінеді. Қалған қызметтерді алғашқы екеуін жүзеге асыруға қажетті байланысқан қызметтер деп санайық. Бұл ретте МҚБЖ термині ретінде МҚ-дағы ақпараттың пайдаланатын барлық программалық жүйелердің кең түрдегі түсінігін ұстанатын боламыз. Пайдаланушымен интерфейсті қолдайтын программа ретінде жалпы жағдайда МҚБЖ келесі негізгі қызметтерді жүзеге асырады: Егер де жүйе бір ЭЕМ-де орналасса, онда барлық қызметтер бір программаға жинақталған және 1-бөлімде қарастырылған сұлбаға ұқсас түрдегі сұлба бойынша шақыртылады. МҚБЖ-ны желіде орналастыру кезінде қызметтерді тораптар бойынша үлестірудің түрлі нұсқалары болады. Өздерінің арасында МҚБЖ қызметтері үлестірілетін тораптардың санына байланысты екі буындық, үш буындық және т.б. модельдерге бөлуге болады. Екі буындық модельдер МҚБЖ қызметтерін желінің екі торабының арасында үлестірілуіне сәйкес келеді.. Компьютер-сервер мен компьютер-клиенттің арасында қызметтерді

үлестірудің аса типтік нұсқалары (2.6а, б-сурет) келесілер болып табылады: үлестірілген түрдегі ұсыну; қашықтан ұсыну; үлестірілген қызмет; Деректерге қашықтан қол жеткізу; үлестірілген МҚ. Үлестірілген Қашықтан ұсыну Үлестірілген түрдегі ұсыну қызмет Компьютер-сервер

Мәліметтерді басқару Өңдеу Ұсыну Компьютер-сервер

Мәліметтерді басқару Өңдеу Компьютер-сервер

Мәліметтерді басқару Өңдеу

Компьютер-клиент Ұсыну

Компьютер-клиент Ұсыну

Компьютер-клиент Ұсыну Өңдеу

2.6а-сурет. Екібуындық модельдер. Компьютер-сервер Компьютер-сервер Мәліметтерді басқару Мәліметтерді басқару

Компьютер-клиент

Мәліметтерді басқару

Өңдеу Өңдеу Ұсыну

Компьютер-клиент

Ұсыну

2.6б-сурет. Екібуындық модельдер. Клиент-сервер архитектурасы бар жүйелерде қызметтерін үлестірудің аталған әдістері әр түрлі нұсқаларды көрсетеді: қуатты сервер – бұл жағдайда барлық жұмыс сонда орындалады, қуатты клиент-қызметтердің басым бөлігі жұмыс станциясында орындалады, ал сервер желі арқылы соған келіп түсіп жатқан SQL-шақыртуларды өңдеуден өткізеді. Деректерге қашықтан қол жеткізу мен қашықтан ұсыну модельдерінде компьютер-клиент пен компьютер-сервердің арасында қызметтер қатаң түрде үлестіріледі. Басқа модельдерде бір мезгілде екі компьютерде Деректерді басқару (үлестірілген МҚ моделі), ақпаратты өңдеуден өткізу (үлестірілген қызмет моделі), ақпаратты ұсыну (үлестірілген ұсыну моделі) қызметтерінің біреуі орындалады. Алдымен аса кең тараған трүлері ретінде Деректерге қол жеткізу мен қашықтан ұсыну (МҚ серверін) модельдерін қарастырып кетейік. Деректерге қашықтан қол жеткізу моделінде (Remote Data Access – RDA), ақпаратты ұсыну қызметтері мен қолданбалы өңдеудің логикасын жүзеге асыратын программалар біріктірілген компьютер-клиенте орындалады. Деректерді басқару сервисіне жүгіну ісі операторлардың көмегімен SQL тілінің операторлары арқылы немесе API (Application Programming Interface – қолданбалы программалау интерфейсы) арнайы кітапхананың қызметтерін шақырту арқылы орындалады. RDA-моделінің негізгі артықшылығы – SQL-интерфейстері бар дайын МҚБЖ-лардың көптеп болуы және клиенттік бөлік программаларының жылдам түрде құрылуын қамтамасыз ететін аспаптық құралдардың болуы. Жасау құралдары көбінесе MS Windows-те пайдаланушының графикалық интерфейсін, ODBC интерфейсінің стандартты мен кодты автоматты түрде генерациялау құралдарын қолдайды. Жасау құралдарының басым көпшілігі төртінші буынның тілдерін пайдаланады. RDA-моделінің кемшіліктері біріншіден, барлық логика қосымшада шоғырланып, өңдеуден өткізілетін Деректер қашықтағы торапта орналасқандықтан Деректерді беру жүйесі жоғары дәрежеде жүктемеленгендігі болып табылады. Қосымшалардың жұмыс істеуі кезінде желі арқылы әдетте тұтастай МҚ-лар беріледі. Екіншіден, RDA моделіне негізделуімен құрылған жүйелер өздерін жасау, өзгерту және ілесе жүру жағынан ыңғайсыз болып келеді. Оның негізгі себебі – алынатын қосымшаларда қолданбалы қызметтер мен ұсыну қызметтері өзара тығыз байланысқан. Сол себептен жүйенің қызметі шамалы ғана өзгергеннің өзінде оның бүкіл қолданбалы бөлігін өзгерту қажет болады, бұл жүйені жасау мен модификациялауды қиындатады. МҚ серверінің моделінің (DataBase Server – DBS) алдыңғы модельден айырмашылығы – компьютер-клиенттің қызметтері ақпаратты кейіптеу қызметтерімен шектеледі, ал қолданбалы қызметтер компьютер-серверде орналасқан қосымшамен қамтамасыз етіледі. Бұл модель RDA-модельге қарағанда аса технологиялық болып табылады және Ingress мен Oracle тәрізді МҚБЖ-ларда қолданылады. Бұл жағдайда қосымшалар сақталатын процедуралар түрінде орындалады. Процедуралар әдетте МҚ сөздігінде сақталады және оларды бірнеше клиенттер бөліседі. Жалпы жағдайда сақталатын процедуралар компиляция мен интерпретациялау режимдерінде орындалады. DBS моделінің артықшылығы жасау, ілесе жүру және модификациялау кезеңдерінде қосымшаларды орталықтандырылған түрде жақсы әкімшілік ету мүмкіндігі, сондай-ақ есептеу және коммуникациялық ресурстарды тиімді пайдалану болып табылады. Соңғысына программаларды ұжымдық пайдалану режимінде орындау ісінің Деректерді желімен жіберуге едәуір аз шығындарды қажет ететінімен қол жеткізіледі. DBS моделі кемшіліктерінің бірі – сақталатын процедураларды жасау құралдарын шектеумен байланысты. Негізгі шектеу – операторлардың сақталатын процедураларды нақты МҚБЖ-ға күштілеп байланыстыруы. Сақталатын процедураларды жазу тілі SQL тілінің процедуралық кеңейтілуі болып табылады және өзінің мағыналық құралдары мен қызметтік мүмкіндіктері бойынша С және Pascal тәрізді үшінші буындағы дәстүрлі

тілдермен бәсекелесе алмайды. Сонымен қатар көптеген МҚБЖ-ларда сақталатын процедураларды баптап реттеу мен тестілеудің қанағаттанарлық құралдары жоқ, бұл олардың механизмін қауіпті құралға айналдырады – реттелмеген программалар МҚ-ның қателесуіне, жүйенің жұмысы кезінде серверлік және клиенттік программалардың текке тұрып қалуына және т.б. әкеп соқтыруы мүмкін. DBS-моделінің тағы бір кемшілігі – ЭЕМ-ның есептеу ресурстарын пайдалану тиімділігінің төмен болуы, себебі компьютер-сервердің программаларына қатысты сұраныстардың кіру ағынын басқару ісін ұйымдастыруды, сондай-ақ процедуралардың өзге компьютер-серверлерге ауысуды қамтамасыз ету мүмкін емес. Үлестірілген түрде ұсыну моделінде қуатты компьютер-сервер бар, ал жүйенің клиенттік бөлігі төмендеп кеткен. Клиенттік бөліктің қызметі жай ғана ақпаратты монитордың экранында көрсету және негізгі компьютермен жергілікті желі арқылы байланысы болып табылады. Осы тектес МҚБЖ-лар бұлайша айтқанда Х-терминалдардың жұмысын қолдайтын желілерде орын алуы мүмкін. Оларда негізгі компьютер (хост-машина) бірнеше Х-терминалдарға қызмет көрсету үшін жеткілікті мөлшердегі қуаттылыққа ие болуы тиіс. Х-терминалдың да жеткілікті дәрежеде жылдам жұмыс істейтін процессоры және жеткілікті мөлшердегі жедел жады болуы тиіс (дискілік жинақтағыштары жоқ). Көбінесе Х-терминалдарды RISC-компьютерлер (restricted [reduced] instruction set computer)-командалардың қысқартылған жинағынан тұратын компьютерлер негізінде құрылады. Бүкіл программалық қамтамасыз ету хост-машинада орналасқан. Ұсынуды басқару қызметтерімен желілік қызметтерді орындайтын Х-терминалды программалық қамтамасыз ету Х-терминал қосылған кезде желі арқылы серверден жүктеледі. Үлестірілген түрде ұсыну моделі кіші, орташа және үлкен ЭЕМ-дарда жұмыс істеген алғашқы буындағы МҚБЖ-ларға ие болған болатын. Х-терминалдар рөлінде дисплейлік станциялар мен абоненттік пунктер (жергілікті және қашықтағы) қызмет етті. Бұл жағдайда ақпаратты ұсыну қызметтерінің негізгі бөлігін МҚБЖ-лардың өздері орындады, ал пайдаланушының терминалдарында бейнелерді соңғы деңгейде құрастыру ісі шеткі құрылғыларда орындалды. Үлестірілген ұсыну моделі бойынша МҚ пайдаланушыларына гетерогендік (біртекті емес) ортада қызмет көрсету жүйелері құрылған. Мұндай жүйелердің серверлік бөлігі әдетте қандай да бірегейлендірілген интерфейсті қамтамасыз етеді, ал клиенттік бөліктері шеткі құрал-жабдықтың ерекшелігін немесе ақпаратты ұсынудың бір форматын өзге бір форматқа түрлендіру ерекшелігін ескеру қызметтерін орындайды. Үлестірілген ұсыну моделі есептеу ресурстарын басқарудың орталықтандырылған сызбасын жүзеге асырады. Осыдан оның басты артықшылықтары шығады – жүйеде қызмет көрсету мен соған қол жеткізуді басқару ісінің қарапайымдылығы және салыстырмалы түрде арзан (шеткі терминалдардың бағасы онша жоғары болмағандықтан) болуы. Модельдің кемшіліктері – орталық тораптың сенімділігі онша жоғары болмауынан жүйенің осалдығы, сондай-ақ клиенттердің саны көп болғанда сервердің өнімділігіне қатысты қойылатын жоғары талаптар. Үлестірілген қызмет моделінде Деректерді өңдеуден өткізудің логикасы екі торап бойынша таратылған. Мұндай модель өздерінде қолданбалы қызметтердің жалпы бөлігі компьютер-серверде, ал ақпаратты өңдеудің арнайы қызметтері компьютер-клиентте орындалатын АЖ-ларда болуы мүмкін. Жалпы сипаттағы қызметтерге Деректердің тұтастығын стандартты түрде, мысалы, сақталатын процедуралар түрінде қамтамасыз ету ісі жатуы мүмкін, ал қалған қолданбалы қызметтер арнайы қолданбалы өңдеуді орындайды. Осыған ұқсас модельге сонымен қатар бірнеше біртекті емес МҚ-лардағы ақпаратты қолданатын АЖ-лар ие болады. Үлестірілген МҚ моделі қуатты компьютер-клиентті пайдалануды қажет етеді, бұл ретте Деректер компьютер-клиентте және компьютер-серверде сақталады. Екі Деректер қорының өзара байланысы екі түрде болуы мүмкін: а) біртұтас МҚ-ның жекелеген бөліктері жергілікті және қашықтағы қорларда сақталады; б) жергілікті және қашықтағы МҚ өзара үйлестірілген көшірмелер болып табылады. Үлестірілген МҚ моделінің артықшылығы – соның негізінде құрылатын АЖ-лардың икемділігі, олар компьютер-клиентке жергілікті және қашықтағы МҚ-ларды өңдеуден өткізуге мүмкіндік береді. Көшірмелердің сәйкестігін үйлестіру механизмдері болған жағдайда жүйе тұтастай алғанда жоғары дәрежедегі өміршендігіне ие болады, себебі клиент пен сервердің өзара қосылысының үзілуі жүйенің күйреуіне әкелмейді, оның жұмысы қосылыс қайта жасалғаннан кейін қалпына келуі мүмкін. Модельдің кемшіліктеріне компьютер-клиенттерде саны көп бірдей қосымшаларды орындау кезіндегі артық шығындарды жатқызуға болады. Қызметтерді үлестірудің үш буындық моделі қосымшаның үш қызметінің әрқайсысы жекелеген компьютерде орындалатын типтік нұсқа болып табылады. Қосымшалардың қызметтерін компьютерлердің көптеген санына үлестірудің нұсқалары болуы мүмкін, алайда олар сирек қолданылатын болғандықтан, қарастырылмайды. Біз қарастырып отырған модель қосымшалар сервері немесе AS-модель (Application Server) деп аталады және 2.7-суретте көрсетілген. МҚ компьютер-сервері Қосымшалардың компьютер-сервері Компьютер-клиент Мәліметтерді басқару Қосымша Ұсыну

2.7-СУРЕТ. ҰШБУЫНДЫ МОДЕЛЬ

Ақырғы пайдаланушымен сұхбат жүргізуге жауап беретін үш буындық AS-модельге сәйкес, аталған процесс, әдетте, ақпаратты ұсыну қызметтерін орындайды және қосымшаның компонентімен DBS моделіндегіге ұқсас түрде әрекеттеседі. Қосымшаның компоненті жеке компьютерде орналасып өз кезегінде RDA моделіне ұқсас түрде Деректерді басқару компонентімен байланысты болады. AS-моделінің орталық буыны қосымшалар сервері болып табылады. Қосымшалар серверінде бірнеше қолданбалы қызметтер орындалады, олардың әрқайсысы оны қажет ететін программалардың барлығына қызмет көрсету қызметі ретінде рәсімделген. Қосымшалар серверлерінің бірнешеуі болуы мүмкін, олардың әрқайсысы өз қызметін ұсынады. Қосымшалар серверінен қызмет көрсетуін сұраған кез келген

программа ол үшін клиент болып табылады. Клиенттерден серверлерге келіп түскен сұраныстар кезекке тұрады, олар ол жерден қандай да бір пәнге байланысты, мысалы, басымдылықтары бойынша іріктеліп алынады. Ұсыну қызметтерін орындайтын және қосымшалар сервері үшін клиент болып табылатын компонент бұл модельге әдеттегіден кеңірек түсіндіріледі. Ол ақырға пайдаланушы бар интерфейсті ұйымдастыру үшін қызмет етуі, құрылғылардан, мысалы, сигнал бергіштерден Деректерді қабылдауды қамтамасыз етуі, немесе еркін түрдегі программа болуы мүмкін. AS-модельдің артықшылығы – қосымшаның қызметтері өзара тәуелсіз үш бөлікке бөлінгендіктен, оның икемділігі мен әмбебаптығы болып табылады. Көптеген жағдайларда бұл модель екі буындық модельмен салыстырғанда тиімдірек болады. Модельдің негізгі кемшілігі – қосымшаның компоненттерінің арасында ақпаратпен алмасуға жұмсалатын компьютердің ресурстары екі буындық модельмен салыстырғанда көбірек жұмсалатындығы. Қосымшалардың компьютер-серверлерінде қосымшалардың қызмет ету ортасын жүзеге асыратын программалық өнімдердің мысалы ретінде BEA WebLogic Server (BEA Systems Corp.), Inprise Application Server (Inprise Corp) және IBM WebSphere Application Server (IBM Corp.) атауға болады.

1. Деректер қоры және Интернет Интернет пен Интернеттегі ақпараттар көзі Деректер қорлары болған жағдайда Web-серверлер МҚБЖ компоненттерінің өзара әсері орын алады. Деректер қорларын пайдаланатын программалық компоненттердің жұмыс істеуін ұйымдастырудың үш түрін ажыратады: а) клиент жағында; б) Web-сервер жағында; с) Қосымшалар сервері жағында. Клиент жағындағы Деректер қорына қол жеткізу кезінде Java тілі клиент пен сервердің өзара әсерінің негізгі құралы болады. Бұл жағдайда HTML тілінің мүмкіндіктерін кеңейтетін JavaScript, Jscript, VBScript сценарийлер тілдері пайдалануы мүмкін. Өзара әсерлер схемалары келесі 2.8-суретте көрсетілген. Web-сервер Клиент МҚ сервер МҚ SQL сұраныс

2.8-СУРЕТ. КЛИЕНТ ПЕН СЕРВЕРДІҢ ӨЗАРА ӘСЕРЛЕР СХЕМАСЫ

МҚ серверіне жүгіну үшін Java – программаларының ішінен жұмыс принципі ODBC ұқсас JDBC хаттамасы пайдаланылады. Web-сервер жағындағы МҚ қол жеткізу SQL тілінде МҚ өзара әсер ететін Web-сервердің сыртқы программаларын шақыру арқылы жүзеге асырылады, ол келесі 2.9-суретте келтірілген.

Клиент Web-сервер МҚ сервері SQL сұраныс МҚ

2.9-СУРЕТ. WEB-СЕРВЕР ЖАҒЫНАН МҚ-НА ҚОЛ ЖЕТКІЗУ.

Келесі 2.10- суретте қосымшалар сервері жағындағы Деректер қорына қол жеткізу мүмкіндігі көрсетілген.

Клиент Web-сервер Қосымшалар сервері МҚ сервері МҚ SQL сұраныс

2.10-СУРЕТ. ҚОСЫМШАЛАР СЕРВЕРІ ЖАҒЫНАН ДЕРЕКТЕР ҚОРЫНА ҚОЛ ЖЕТКІЗУ.

Таратылған қосымшаларды жасаудың негізгі тілі – Java тілі болып табылады. 70-80 жылдардан бері қолданылып келе жатқан әзірленген жүйені және қабылданып жатқан техникалық шешімдерді сипаттаудың ресмиленген әдістері уақыт өткен сайын көп еңбекті қажет етеді. Соның нәтижесінде CASE-технологиясын жүзеге асыратын CASE-құралдары деп аталатын программалық-технологиялық құралдар пайда болды. CASE (Computer Aided Software Engineering) терминін компьютердің көмегімен программалық қамтуды жасау деп аударуға болады. CASE-құралдары дегеніміз – бұл ақпараттың жүйелерді құрастыру және оларға ілесе жүру үдерістерін қолдайтын программалық құралдар, оларға келесілер жатады: талаптарды талдаудан өткізу және тұжырымдау, Деректер қоры мен қосымшаларды жобалау, кодты генерациялау, тестілеу, сапаны қамтамасыз ету, конфигурацияны басқару және т.б. CASE-құралдарын келесі негізгі түрлерін атап көрсетуге болады:

- пәндік саланы құрастыру мен талдауға арналған талдау құралдары. Оларға Design/IDEF, Bpwin жатады;
- жобалық арналымдарды құруды қамтамасыз ететін талдау мен жобалау құралдары, мысалы, Vantage Team Builder, Silverrun, PRO-IV;
- Деректер қорын модельдеп, сұлбаларды жасауға мүмкіндік беретін Деректер қорын жобалау құралдары. Оларға Erwin, S-designot, DataBase Designer жатады;
- қосымшаларды жасау құралдары, мысалы, Uniface, JAM, PowerBuilder, Developer/2000, New Ero, SQL Windows. Кейбір CASE-құралдары нақты бір CASE-тің құрамына кірмейтін автономдық жүйелер түрінде жеткізіледі. Мұндай тәуелсіз CASE-жүйелердің қатарына S-Designer, Erwin, Silverrun жатады. МҚБЖ ішіне орнатылған CASE-құралы – Oracle құрамына кіретін Designer/2000 болып табылады. CASE-жүйелері мен құралдарының негізгі мақсаты – программалық қамтамасыз етудің жобалануын, оны кодтау мен кейінгі жасау кезеңдерінен бөліп алу, сондай-ақ программалық жүйелерді құрудың бүкіл үдерісін автоматтандыру болып табылады. CASE-технология ақпараттық жүйелерді жобалаудың әдіснамасы ретінде көрсетіледі. CASE-технологиясы дегеніміз – бұл сонымен қатар тақырыптық саланы үлгілеуге, оның үлгісін ақпараттық жүйені әзірлеу мен оған ілесе жүрудің барлық кезеңдерінде талдаудан өткізіп

отыруға және пайдаланушыларға арнап қосымшаны жасауға мүмкіндік беретін саймандық құралдар Бақылау сұрақтары

- МҚ әкімшісінің қызметіне не жатады?
- МҚ әкімшісі нені орындайды?
- Ақпараттық жүйе дегеніміз не?
- Деректер қоры серверінде не орналасады?
- Ақпараттық жүйеге не кірмейді?
- Қандай ақпараттық жүйе жергілікті деп аталады? Әдебиеттер:
- Бидайбеков Е.Ы., Елубаев К., Шекербекова Ш.Т. Деректер қоры және ақпараттық жүйелер. Алматы., 2010.
- Золотова С.И. Практикум по Access. Финансы и статистика, Москва, 2000.
- Джулия Келли. Самоучитель Access 97. Питер. Санкт-Петербург, 2000.
- Робинсон С. Access 2000. Учебный курс.

5-ДӘРІС.

Тақырыбы: Деректер қорының моделі. Лекция жоспары:

- Деректер қорының моделдері.
- Иерархиялық модель.
- Желілік модель.
- Реляциялық модель.
- Атрибут (өріс, бағана).
- Кортж (жазба, жол).
- Қатынас (кесте).
- Нормаль формалар.
- Алғашқы және сыртқы кілттер. Құзыреттіліктер- Деректер қорына иерархиялық модель және реляциялық модельдермен жұмыс жасай алуы менгеру Лекция мазмұны Деректер қорын пайдалану файлдық құрылым Деректерінде болған шектеулер мен кемшіліктерді жоюға мүмкіндік берді. Деректер қорының көмегімен шешілетін әралуан есептер Деректер қорын құрылымдау мен Деректердің арасындағы байланыстарды ұйымдастырудың түрлі әдістерін қалыптастыруға әкеледі. Соның нәтижесінде Деректер қорының бірнеше моделі пайда болды. Деректер қорының моделі деп Деректер қорының логикалық кейіптеуін атаймыз. Алғашқыда Деректер 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. Сонымен қатар Деректердің өзге модельдеріне негізделген белгілі модельдерді кеңейтетін алуан түрлі жүйелер жасалынууда. Олардың қатарында объектілі-реляциялық, дедуктивтік-объектілі-бағытталған, семантикалық, тұғырнамалық және бағытталған модельдерді атап кетуге болады. Бұл модельдердің кейбіреулері Деректер қорын, білім қоры мен программалау тілдерін интеграциялауға арналған. Кейбір МҚБЖ-лар бір мезгілде бірнеше Деректер модельдерімен жұмыс істейді. 3.1. Иерархиялық модель. Алғашқыда Деректер қорының иерархиялық модельдері пайда болды. Иерархиялық модельде Деректер арасындағы байланыстар реттелген граф (немесе бұтақ) түрінде берілген. Иерархиялық МҚ-ның құрылымын (сұлбасын) сипаттау үшін кейбір программалау тілінде «бұтақ» Деректер типі қолданылады. «Бұтақ» типі ПЛ/1 және Си программалау тілдерінің «құрылым» атты Деректер типімен және Паскаль тілінің «жазба» типімен ұқсас. Оларда типтердің бір-біріне салынуына жол беріледі, олардың әрқайсысы белгілі бір деңгейде орналасқан. «Бұтақ» типі құрамдас болып келеді. Оған ішкі типтер («кіші бұтақтар») кіреді, олардың әрқайсысы өз кезегінде «бұтақ» типі болып табылады. «Бұтақ» типінің әрқайсысы бір «түбірлік» типтен және бағыныңқы типтердің реттелген жиынтығынан (бос болуы да мүмкін) тұрады. «Бұтақ» типіне енгізілген қарапайым типтердің әрқайсысы «жазба» атты жай немесе құрама тип болып табылады. Қарапайым «жазба» бір типтен тұрады, мысалы, сандық типтегі, ал құрама «жазба» типтердің белгілі бір жиынтығын, мысалы, бүтін, символдар жолы және көрсеткіштерді (сілтеме) біріктіреді. Түбірлік деп бағыныңқы типтері бар және өзі ішкі тип болмайтын тип аталады.

Бағыныңқы тип (ішкі тип) өзі үшін ата тегі (ата-ана) рөлін атқаратын типке қатысты ұрпақ болып табылады. Бір типтің ұрпақтары бір-біріне қатысты егіздер болып табылады. Тұтастай алғанда «бұтақ» типі «жазба» типтерінің иерархиялық тұрғыдан ұйымдастырылған жиынтығын сипаттайды. Иерархиялық МҚ құрамында «жазба» (жазбалар) типіндегі Деректер даналарын (экземплярларын) қамтитын «бұтақ» типіндегі Деректер даналарының реттелген жиынтығы болып табылады. Көп жағдайда типтердің арасындағы туыстық қатынастарды жазбалардың арасындағы қатынастарға көшіреді. Жазбалар өрістері шындығында МҚ-ның негізгі мазмұнын құрайтын сандық немесе символдық мәндерді сақтайды. Иерархиялық МҚ-ғы барлық элементтерді қарап шығу әдетте жоғарыдан төмен қарай және солдан оңға қарай жүргізіледі. Иерархиялық модельдің Деректер қорлары бұтақ түріндегі құрылымы бар нысандарды сипаттауға ыңғайлы. Мысалы, белгілі бір кәсіпорынның құрылымын келесі түрде келіруге болады (1.2-сурет). Кәсіпорын Әкімшілік Бухгалтерия Бөлімдер Кадрлар бөлімі Өндірістік цехтар Шаруашылық бөлімі Жұмысшылар Лауазымдар

1.2-СУРЕТ. КӘСІПОРЫННЫҢ ИЕРАРХИЯЛЫҚ ҚҰРЫЛЫМЫ

Өзінен басқа жазбаға сілтеме жасалатын жазба «ата тегі» немесе бағындыратын жазба деп аталады. Сілтеме жасалатын жазба «ұрпақ» немесе бағыныңқы жазба деп аталады. Иерархиялық модельде әрбір ұрпақтың тек бір ғана ата тегі болады, басқаша айтқанда бір ұрпаққа екі және екіден артық ата тегі сілтеме жасай алмайды. Сілтемені нұсқау үшін ата тегі-жазбаның өрісі болуы тиіс, ол жерге тасымалдағыштағы ұрпақ-жазбаның физикалық мекен-жайы жазылады. Ұрпақ-жазба сақталатын тасымалдағыштағы осы физикалық мекен-жай көрсеткіш деп аталады. Жазбаларды байланыстырудың мысалы (1.3-сурет)

Бөлім нөміріЦех№1

Код Иванов

Инженер Ставка

1.3-СУРЕТ. ЖАЗБАЛАРДЫ БАЙЛАНЫСТЫРУ

Сонымен, иерархиялық модельдегі Деректер арасындағы байланысты ұйымдастыру үшін физикалық мекен-жайлардың көрсеткіштері қолданылады. Деректердің иерархиялық моделінің артықшылықтарына ЭЕМ жадын тиімді пайдалану мен Деректермен негізгі амалдарды орындау уақытының көрсеткіштері жатады. Деректердің иерархиялық моделі иерархиялық тұрғыдан реттелген ақпаратпен жұмыс істеу үшін ыңғайлы. Иерархиялық модельдің кемшілігіне күрделі логикалық байланыстары бар ақпаратты өңдеу үшін оның көлемінің тым орасан үлкен болуы, сондай-ақ қарапайым пайдаланушы үшін түсінуге қиындығы жатады.

3.2. Желілік модель. Желілік модельдерде Деректердің арасындағы өзара байланыстар еркін граф түрінде болады, ондағы әрбір ұрпақтың екі және одан да көп ата тегі болуы мүмкін. Желілік модельдің Деректер қоры жазбалар жинағы мен байланыстар жинағынан құралады. Байланыстардың жинағы физикалық көрсеткіштері бар өрістен тұрады. Осылайша, желілік МҚ сұлбасын сипаттау үшін типтердің екі тобы: «жазба» және «байланыс» қолданылады. «Байланыс» типі «жазба» типінің ата тегі және ұрпақ типтері үшін анықталады. «Байланыс» типінің айнымалылары байланыстардың даналары болып табылады. Желілік МҚ жазбалар жинағы мен сәйкес байланыстар жинағынан тұрады. Байланыстарды қалыптастыруға ерекше шектеулер қойылмайды. Егер иерархиялық құрылымдарда ұрпақ-жазбаның бір ғана ата тегі –жазбасы болса, ал Деректердің желілік моделінде ұрпақ-жазба ата тегі-жазбалардың (өгей ата-аналар) кез келген санына ие бола алады. Келесі суретте (1.4-сурет) инженер лауазымына сәйкес келетін жазбаға екі жазба сілтеме жасайды. Код Иванов Инженер Ставка

Код Петров

1.4-СУРЕТ. ЖЕЛІЛІК МОДЕЛЬДЕГІ ЖАЗБАЛАРДЫ БАЙЛАНЫСТЫРУ

Иерархиялық модельді желілік модельдің дербес жағдайы ретінде қарастыруға болады. Деректердің желілік моделінің артықшылығы жадының шығындалуы мен жеделдік көрсеткіштері бойынша тиімді түрде жүзеге асыру мүмкіндігі болып табылады. Желілік-модель иерархиялық модельмен салыстырғанда еркін түрдегі байланыстардың құрылу тұрғысынан үлкен мүмкіндіктер ұсынады. Деректердің желілік моделінің кемшілігі – оның негізінде құрылған МҚ сұлбасының жоғары дәрежедегі күрделілігі мен қатқылдығы, сондай-ақ қарапайым пайдаланушы үшін МҚ-дағы ақпаратты өңдеуді түсіну мен орындаудың қиындығы. Бұған қоса, Деректердің желілік моделінде жазбалардың арасында еркін түрдегі байланыстардың орнау мүмкіндігінің салдарынан байланыстардың біртұтастығына қатысты бақылау әлсірейді. Деректер қорының иерархиялық және желілік модельдерінде физикалық көрсеткіштерді пайдалану, Деректерді өңдеудің жылдамдығын едәуір арттыра түсті. Сонымен бірге, бұл модельдерде Деректердің өзара байланыстарының біртұтастығын сақтауға қатысты елеулі кемшіліктер анықталды. Байланыстардың біртұтастығының сақталуын бақылау

ата-тегісіз ұрпақ болмайды деген ұстанымды сақтауға негізделген. Деректер қорына өзгертулер енгізген жағдайда байланыстарды, физикалық көрсеткіштерді қайтадан анықтау қажеттігі туындайды, бұл үлкен қиындықтармен, пайдаланушының уақыты мен күш-жігерінің шығындарымен байланысты болады. Бақылау сұрақтары:

- Деректер қорының моделдері.
- Иерархиялық модель.
- Желілік модель.
- Реляциялық модель.
- Атрибут (өріс, бағана).
- Кортёж (жазба, жол). Әдебиеттер: 1.Добротворский И.Н. Теория электрических цепей. 2012 г. 2.Лосев А.К. Теория линейных электрических цепей. 2010 г.

6-ДӘРІС.

Тақырыбы: Реляциялық модель.Реляциялық алгебра Лекция жоспары:

- Реляциялық модель.
- Атрибут (өріс, бағана).
- Кортёж (жазба, жол).
- Қатынас (кесте).
- Нормаль формалар.
- Алғашқы және сыртқы кілттер.
- Реляциялық алгебраның амалдары.
- Реляциялық есептеу.
- Қалыптандыру. Құзыреттіліктер- Деректер қорына атрибут,кестелерді ендіру және олармен жұмыс жасай алуды меңгеру Лекция мазмұны Деректердің реляциялық моделі. Реляциялық модельдің анықтамасы. Иерархиялық және желілік модельдерде жазбаларды байланыстыру үшін физикалық көрсеткіштерді пайдалану Деректерді манипуляциялау мүмкіндіктерін едәуір шектейді және осы модельдердің Деректер қорларын өңдеу күрделі және көп еңбекті қажет ететін процесс болып табылады. 1970 жылы IBM фирмасының қызметкері Эдгар Кодд реляциялық модель деп аталған Деректер қорының жаңа моделін ұсынған бірқатар еңбектерін жариялаған болатын. Сонымен қатар, реляциялық Деректер қорларын өңдеу үшін Кодд Деректерді өңдеудің тілдерін – реляциялық алгебра мен реляциялық есептеуді әзірлеп шығарды, олар бір командамен бүкіл Деректер қорын өңдеуден өткізуге мүмкіндік береді, ал иерархиялық және желілік модельдерде бір жолы тек бір ғана жазба өңдеуден өткізіледі. Физикалық көрсеткіштердің орнына Кодд Деректердің тасымалдағышта физикалық орналасуына тәуелсіз оларды өздерінің ішкі логикалық қарым-қатынастарына сәйкес байланыстыру идеясын ұсынды. Реляциялық модельдің мәні – ол Деректерді реляциялар немесе қатынастар деп аталатын екі өлшемді кестелер түрінде беруді ұйымдастырады және ұсынады. Бір кесте бір нысанды, мысалы, сауда фирмасының клиенттерін сипаттайды. Кестенің бағандары Деректер қорының өрістеріне сәйкес келеді және атрибуттар деп аталады. Сол арқылы атрибут нысанның жеке қасиетін сипаттайды. Кестенің жолдары жазбаларға сәйкес келеді және кортеж деп аталады. Сонымен, кортеж нысанның бір элементіне арналған атрибуттар мәндерінің жиынтығын, ал қатынас – кортеждердің жиынын сипаттайды. Деректер қорының құрылымын әзірлеу және сипаттау кезінде әрбір кестедегі атрибуттарға атаулар беріледі және олардың типтері анықталады, мысалы, атауы Name, типі CHAR немесе атауы Amount, ал типі REAL. Сонымен қатар INTEGER типі, DATA типі және басқа да типтер болуы мүмкін. Егер кестенің элементі әлі анықталмаған болса немесе ешқандай мәнге ие болмаса, онда оған Null - мәні меншіктеледі. Деректердің арасындағы логикалық байланыс бастапқы және сыртқы кілттердің көмегімен жүзеге асырылады. Бастапқы кілт деп кестенің әрбір жолын бір мәнді анықтайтын кестедегі атрибуттардың ең аз жиынтығы аталады. Мысалы, Salespeople кестесінде бастапқы кілт Snum бағаны, ал Customersкестесінде Snum бағаны болып табылады. Осылайша, бастапқы кілтті құрайтын атрибуттар мәндерінің комбинациясы әр жазба үшін бірегей, қайталанбайтын болып табылады. Сыртқы кілт деп мәндерінің комбинациясы кейбір жазба үшін бастапқы кілттің жиынтығы аталады. Сыртқы кілттер бір Деректер қорының кестелеріндегі Деректердің арасындағы қажетті байланыстарды қамтамасыз етеді. Мысалы, Orion Деректер қорында Orders кестесінің Snum және Snum бағандары сыртқы кілттер болып табылады. Деректер арасындағы байланыстарды келесі сурет түрінде көрсетуге болады (1.5-сурет) Orders Onum Amount Date Cnum Snum ... 0813006 1098.16 10.03.2005 2008 1007 ... Customers Cnum Cname City ... 2008 Pereira 1004 ...

1.5-СУРЕТ. РЕЛЯЦИЯЛЫҚ МОДЕЛЬДЕГІ ЖАЗБАЛАРДЫ БАЙЛАНЫСТЫРУ

Сонымен реляциялық кестенің анықтамасынан келесілер шығады:

- бастапқы кілттің және оған сәйкес келетін сыртқы кілттің атрибуттарының атаулары бірдей болуы міндетті емес, бірақ та бір типті болуы тиіс;
- Деректер қорының әрбір кестесінің бірнеше сыртқы кілттері болуы мүмкін, ал бастапқы кілт тек біреу ғана болады;
- жалпы жағдайда кестеде екі және одан да көп бірдей кортеждер болуы мүмкін емес;
- кестеде кортеждер реттелмеген, еркін түрде болуы тиіс;
- атрибуттар да реттелмеген, еркін түрде болуы тиісті. 2 қосымшада адамзат қызметінің әр түрлі саласындағы Деректер қорының мысалдары келтірілген. Деректердің реляциялық моделінің артықшылығы оны ЭЕМ-де физикалық жүзеге асырудың қарапайым, түсінікті және ыңғайлы болуында. Пайдаланушы үшін дәл осы қарапайымдылығы мен түсініктілігі оларды кең түрде қолдануға негізгі себеп болды. Осы типтегі Деректерді өңдеу тиімділігінің мәселелері техникалық тұрғыдан шешілетін болып шықты. Реляциялық модельдің негізгі кемшіліктері:
- жеке жазбаларды идентификациялаудың стандарттық құралдарының болмауы; иерархиялық және желілік байланыстарды сипаттаудың күрделілігі жатады. Реляциялық алгебра Реляциялық алгебра – бұл реляциялық кестелерді өңдеу тілі. Реляциялық алгебра келесі амалдардан тұрады:
- біріктіру;
- қиылысу;
- айыру;
- көбейту;
- таңдау;
- проекцияны құру;
- проекция;
- қосу (жалғау);
- бөлу;
- меншіктеу.
- Біріктіру амалы екі кестенің Деректерін үйлестіруге, екі кестенің жазбалары (кортеждері) тек бір-ақ рет пайда болатын кестені құруға мүмкіндік береді. Біріктіру амалы $\cup$ әрпімен беріледі. Мысалы: $a1\ a2\ a3c1\ c2\ c3a1\ a2\ a3\ b1\ b2\ b3\ U\ f1\ f2\ f3 = b1\ b2\ b3\ c1\ c2\ c3\ a1\ a2\ a3\ c1\ c2\ c3\ d1\ d2\ d3d1\ d2\ d3\ f1\ f2\ f3$ Екі кестені біріктіру кезінде олар бағандардың (атрибуттардың) саны бойынша, сондай-ақ олардың типі бойынша сәйкес келуі тиіс.
- Қиылысу амалы екі кестенің ортақ жолдарын бөліп алуға мүмкіндік береді. $\wedge$ символымен белгіленеді. Мысалы, $a1\ a2\ a3\ c1\ c2\ c3\ a1\ a2\ a3\ b1\ b2\ b3\ \wedge\ f1\ f2\ f3 = c1\ c2\ c3\ c1\ c2\ c3\ a1\ a2\ a3\ d1\ d2\ d3$
- Айыру амалы бір кестеден басқа кестеде жоқ жазбаларды (жолдарды) бөліп алуға мүмкіндік береді, минус белгісімен белгіленеді. Мысалы: $a1\ a2\ a3\ c1\ c2\ c3b1\ b2\ b3\ b1\ b2\ b3\ -\ f1\ f2\ f3 = d1\ d2\ d3\ c1\ c2\ c3\ a1\ a2\ a3\ d1\ d2\ d3$
- Көбейту амалы екі кестенің декарттық көбейтіндісін құрайды. $*$ символымен белгіленеді. Мысалы: $x1\ y1\ z1\ * \ a1\ a2\ \times1\ y1\ z1\ a1\ a2\ \times2\ y2\ z2b1\ b2 = x1\ y1\ z1\ b1\ b2\ c1\ c2 \times1\ y1\ z1\ c1\ c2\ \times2\ y2\ z2\ a1\ a2\ \times2\ y2\ z2\ b1\ b2\ \times2\ y2\ z2\ c1\ c2$
- Таңдау амалы бір кестеден көрсетілген шартқа сәйкес келетін жазбаларды іріктеп алып, екінші кестені құруға қолданылады. Таңдау әрекетін белгілеу үшін SELECT сөзі қолданылады. Мысалы: `SELECT (Orders:'03/10/2000' <Data<'05/10/2000')` Шарт дегеніміз өзінде $=$, $>$, $<$, $>=$, $<=$ қатынастық әрекеттер мен «және», «немесе», «емес» логикалық амалдар қолданылатын логикалық өрнек болып табылады.
- Проекцияны құру амалы кестеден жаңа кестені құру үшін қажетті бағандарды таңдап алуға мүмкіндік береді. Проекцияны құру амалы келесі тәртіппен жазылады. Кестенің атауы. [баған, баған, . . . , баған]
- Қосу (жалғау) амалы кестелердің арасындағы Деректерді байланыстыру үшін қолданылады. Бұл амал келесі түрде жазылады: JOIN (кесте, кесте) Жалғау үшін кестелердің бағандары бірдей болуы тиіс. Жалғау амалын орындау кезінде келесі әрекеттер орындалады:
- Бастапқы кестелердің көбейтіндісі анықталады;
- Алынған көбейтіндіден ортақ бағандардың мәндері бірдей болатын жолдар таңдап алынады;
- Жобалау кезінде алғашқы кестенің ортақ бағандары алынып тасталады.

Жалғау мысалы: Ортақ бағандар Жалғаудың үш түрі болады:

- табиғи (қарапайым) жалғау;

- тэта – жалғау;
- сыртқы жалғау. Қарапайым жалғауда бастапқы кестелердің көбейтіндісінен, ортақ атрибуттардың мәндері өзара сәйкес келетін жазбалар іріктеп алынады. Тэта – жалғауда бастапқы кестелердің көбейтіндісінен берілген атрибуттар белгілі бір қатынаста болатын жазбалар іріктеп алынады, мысалы, артық амалы бойынша. Тэта – жалғау жалпы алғанда келесі түрде беріледі. JOIN (A,B: A.атрибут Θ B.атрибут) Осылайша, қарапайым жалғауда және тэта-жалғауда қорытынды кестеде бастапқы кестелердің бір-біріне сәйкес келетін жазбалары ғана беріледі. Сыртқы жалғауда бастапқы кестелердің әрбір жазбасы қорытынды кестеде ең болмағанда бір рет берілетін болады. Алғашқы (сол) немесе екінші (оң) бастапқы кестенің жазбалары қорытынды кестеде тым болмағанда бір рет берілетініне байланысты, сәйкесінше сол жақтағы, оң жақтағы және толық жалғануды айырады.
- Бөлу амалы бір кестенің ішінен екінші кестенің әрбір жолына сәйкес келетін жолдарды таңдап алу арқылы жаңа кестені құруға мүмкіндік береді. Бөлу амалы келесі түрде жазылады: 1-кесте/2-кесте ху a b b c d / e = a a e f a f
- Меншіктеу амалы реляциялық алгебраның алдыңғы амалының нәтижесіне қандай да бір атау меншіктейді. Мысалы:

C:= JOIN(A, B)

Дейт [3] реляциялық алгебраны келесі амалдармен толықтырады:

- қайтадан атау беру. RENAME <кесте><атрибуттың ескі аты> AS <атрибуттың жаңа аты>, <атрибуттың ескі аты> AS <атрибуттың жаңа аты>, . . . кестедегі атрибут атын өзгертуге мүмкіндік береді; кеңейту EXTEND <кесте> ADD <өрнек> AS <жаңа атрибут> берілген өрнекті есептеу арқылы мәні табылатын жаңа атрибутты кестеге қосады; қорытындылар жасау SUMMARIZE <кесте> BY (<атрибуттар тізімі>) ADD <өрнек> AS <жаңа атрибут> берілген тізіммен топтастырылған кортеждер үшін, берілген өрнек бойынша вертикальды топтық есептеулерді орындайды және есептеулер нәтижелері оның кортеждер саны бастапқы кестенің топтар санына тең болатын, қалыптастырылып отырған кестедегі жаңа бағанды құрайды; кіргізу INSERT <1 кестеден таңдау> INTO <2 кестеге қосу> 2 кестеге 1 кестеден таңдап алынған кортеждерді қосу; жаңарту UPDATE <бастапқы кестеден таңдау><элементтер тізімі> таңдап алынған кортеждер үшін көрсетілген элементтердің атрибут:=өрнек түріндегі мәндерін өзгертулер орындалады; жою DELETE <бастапқы кестеден таңдау> бастапқы кестеден таңдап алынған кортеждерді жою орындалады. Сонымен, реляциялық алгебра дегеніміз – бұл реляциялық кестелерді өңдеудің процедуралық тілі, басқаша айтқанда реляциялық алгебра есептің қадамдық шешімін қамтамасыз етеді. Реляциялық есептеу. Реляциялық есептеу, реляциялық алгебраға қарағанда, кестені бір сұраныстың көмегімен анықтайды. Реляциялық есептеуде сұраныс келесі түрде беріледі: {мақсаттық тізім: анықтаушы өрнек} Мақсаттық тізім реляциялық есептеудегі сұраныстың шешімі болып табылатын кестенің атрибуттарын (бағандарын) анықтайды. Анықтаушы өрнек шешу кестесінің элементтері қанағаттандыруға тиіс болатын шарттарды береді. Фигуралы жақшалар шешу кестесіндегі жолдардың көптігін білдіреді. Еркін түрдегі жолды белгілеу үшін латын алфавитінің кіші әріптері қолданылады. Мысалы, «Орион» фирмасының сауда агенттерінің қайсысы Лондонда тұрады деген сұраныс келесі түрде беріледі: {r.Summ, r.Sname:=r IN Salespeople AND r.City="London"} Сұраныс нәтижесінде келесі кесте пайда болады. Summ Sname 1001 Pal 1004 Smith Бұл мысалда реляциялық есептеудің сұранысы реляциялық алгебраның келесі екі амалына – таңдау және проекцияны құруға теңбе-тең болады. A=select(Salespeople: City=London), B=A[Summ, Sname] Тұтастай алғанда жоғарыда көрсетілген мысалдар тәрізді сұраныстарды, жалғау және бөлу амалдарынан басқа, реляциялық алгебраның амалдарымен орындауға болады. Реляциялық есептеуде қосу және бөлу амалдарына ұқсас келетін әрекеттерді орындау үшін сәйкесінше бар болу кванторы мен жалпыға бірдей кванторы қолданылады. Реляциялық есептеуде болмыс кванторы дегеніміз – бұл берілген шартты қанағаттандыратын ең болмағанда бір жолдың болуын білдіретін өрнек. Егер де, мысалы, 1002 агентіне сұраныс жасаған клиенттерді атап шығу қажет етілсе, онда келесідегідей сұраныс жасау қажет: {r.Cname:=r IN Customers AND EXISTS s IN Orders (s.Cnum=r.Cnum and s.Snum=1002)} Мұндай сұраныс реляциялық алгебраның қосу және проекцияны құру амалдары орындайтын әрекетті орындайды. Sname Lewe Grass A=Join (Customers Orders: Customers.Cnum=Orders. Cnum and Orders.snum=1002) B=[Cname] Реляциялық есептеудегі жалпыға бірдей кванторы – бұл белгілі типтегі әрбір жолда қолданылатын шартты білдіретін өрнек. Мысалы, әрбір клиенттен қай агент тапсырыс алды? Сұраныс келесі түрде болады. {r.Snum, r.Sname:

r IN Salespeople AND s IN Orders AND THERE EXISTS c IN Customers (r.Snum = s.Snum) AND (c.Cnum = s.Cnum)}

Реляциялық есептеу – қалай қол жеткізуге болатынын емес, нені орындау қажет екендігін тұжырымдауға мүмкіндік беретін процедуралық емес тіл. Реляциялық есептеу, реляциялық алгебра сияқты теориялық тіл болып табылады және логикалық сипаттамаларға негізделген реляциялық Деректер қорының кестелерін, олардың физикалық құрылымына қатыссыз манипуляциялауға арналған. Реляциялық есептеу лексикалық тұрғыдан реляциялық алгебраға барабар, яғни ол реляциялық алгебра шешетін есепті шешуге мүмкіндік береді. Қалыптандыру. Деректер қорын әзірлеу, жобалау кезінде кестелер белгілі бір талаптарды қанағаттандыруы, яғни стандарттық түрге ие болуы тиіс. Бұл талаптар қалыпты форма деп, ал қалыпты түрге келтіру үдерісі – қалпына келтіру деп аталады. Қалпына келтірілмеген кестелерді реляциялық алгебраның әрекеттері дұрыс өңдемейді. Деректер қорының кестелері 4 қалыпты түрге сәйкес келуі тиіс. Алғашқы қалыпты форма атрибуттардың барлық мәндері жиындар емес, қарапайым шамалар болуға тиіс болатынын ескереді. Мысалы, келесі түрдегі жазбаға

```
0023003 {607.1, 160.09} 10/03/2000 2001 1001
```

жол берілмейді, себебі ол алғашқы қалыпты формаға сәйкес келмейді. Екінші қалыпты форма. Егер де ешбір кілттік емес атрибуттар кілттің бөлігіне тәуелді болмаса, реляциялық кесте екінші қалыпты формада болады. Осылайша, екінші қалыпты форма бастапқы кілт құрамдас болған жағдайда, былайша айтқанда, бірнеше атрибуттардың жиыны болған жағдайда ғана бұзылуы мүмкін. Мысалы, келесі кесте,

```
Clerk Office Name 1004 312 Smith 1001 312 Pol 1004 515 Smith 1001 460 Pol 1001 435 Pol
```

ондағы Clerk пен Office бастапқы кілт, екінші қалыпты түрде емес, себебі Name атрибуты Clerk атрибутына тәуелді. Кесте екінші қалыпты формада болу үшін оны келесі түрде екі кестеге бөлу керек

```
Clerk OfficeClerk Name 1004 312 1001 Paul 1001 312 1004 Smith 1004 515 1001 460 1001 435
```

Үшінші қалыпты форма. Егер кез келген детерминант кілт болып табылса, онда реляциялық кесте үшінші қалыпты формада болады. Детерминант деп кестенің әрбір жолындағы басқа атрибуттың мәнін анықтайтын атрибут аталады. Кестені қарастырайық.

```
Kod Skill Bonus 1004 Manager 3.5 1001 Clerk 3.0 1007 Clerk 3.0
```

Бұл кестеде Kod бағаны кілт болып табылады және Skill мен Bonus бағандарын анықтайды. Сонымен қатар, Skill (біліктілігі, разряды) бағаны кілт болмаса да, Bonus бағанында анықтайтынын көріп отырмыз. Бұл деген сөз, кесте үшінші қалыпты формада бола алмайды. Бұл мәселені шешу үшін кестені келесі түрде екі кестеге бөлу керек.

```
Kod SkillSkill Bonus 1004 Manager Manager 3.5 1001 Clerk Clerk 3.0 1007 Clerk
```

Төртінші қалыпты форма өзге атрибутқа (көпмәнді байланыстылық) тәуелді болатын атрибут мәнінің көп рет қайталану мүмкіндігіне жол бермейді. Мысалы, егер кестеде агент Smith-тің 1001 және 1007 қызметкерлерінің менеджері болып табылатынын, және 2003, 2001, 2004 клиенттерінен тапсырыс алғанын көрсету керек болса, онда келесі кесте пайда болады.

```
Name Snum Cnum Smith 1001 2003 Smith 1001 2001 Smith 1001 2004 Smith 1007 2003 Smith 1007 2001 Smith 1007 2004
```

Алынған кесте төртінші қалыпты формада емес. Төртінші қалыпты формаға келтіру, кестені әрбір тәуелді атрибут жеке орналасқан екі кестеге бөлу арқылы орындалады.

```
Name SnumNameCnum Smith 1001 Smith 2003 Smith 1007Smith 2001 Smith 2004
```

Қарастырылған төрт қалыпты формадан өзге, негізінен теориялық сипаттамасы бар әрі практикалық маңызы жоқ басқа да формалар болады. Бұлар біріккен тәуелділіктері болмайтын бесінші қалыпты форма мен облыс/кілт қалыпты формасы. Бесінші қалыпты формада біріккен тәуелділіктер болмайды. Мысал қарастырайық.

```
Kod Office Project 1001 02 010 1001 05 010 1003 01 008 1004 03 008 1004 03 007
```

Берілген кестеде бірнеше бөлімдерде жұмыс істейтін және бірнеше жобаларды орындайтын қызметкерді алып тастау немесе жобаның нөмірін өзгерту, кестеде атрибуттардың біріккен тәуелділіктері болғандықтан қиын болады. Сондықтан берілген кестені төмендегідей үш кесте түрінде ұсыну қажет.

```
Kod Office Kod Project Office Project 1001 02 1001 010 02 010 1001 05 1003 008 05 010 1003 01 1004 008 01 008 1004 03 1004 007 03 008
```

Реляциялық кесте, егер кез келген шектеу шарты бар болу (доменнің) облысының анықтамасы мен кілттерден шығатын болса облыс/кілт қалыпты формасына ие болады. Бақылау сұрақтары:

- Иерархиялық және желілік Деректер қорының тұтастығы нені білдіреді?
- Деректер қорының желілік моделі дегеніміз нені білдіреді?
- Деректер қорының желілік моделі дегеніміз нені білдіреді?
- Реляциялық Деректер қоры дегеніміз не?
- Қандай Деректер қорларында Деректерді байланыстыру үшін физикалық көрсеткіштер қолданылады?
- Нысанның бір элементі үшін өрістер мәндерінің жиынтығы не болып табылады?
- Деректер қорында өріс деп нені айтамыз?
- Кортж дегеніміз не? Әдебиеттер:
- Бидайбеков Е.Ы., Елубаев К., Шекербекова Ш.Т. Деректер қоры және ақпараттық жүйелер. Алматы., 2010.
- Золотова С.И. Практикум по Access. Финансы и статистика, Москва, 2000.
- Джулия Келли. Самоучитель Access 97. Питер. Санкт-Петербург, 2000.
- Робинсон С. Access 2000. Учебный курс.

7-ДӘРІС.

Тақырыбы: Деректер қорының басқа модельдері Лекция жоспары:

- Постреляциялық модель.
- Көпөлшемді модель.
- Нысанды-бағытталған модель. Құзыреттіліктер- Сұраныстар құруды меңгерту Лекция мазмұны Деректер қорының басқа модельдері. Постреляциялық модель. Реляциялық модельге қарағанда постреляциялық модельде көпмәнді өрістерге жол беріледі, яғни онда алғашқы қалыпты түр қоятын шектеудің күші жойылады. Көп мәнді өрістердің мәндер жиыны өзіндік кесте болып саналады. Мысалы, Деректер қорының реляциялық моделінің келесі кестелері.

Тапсырыстар Cnum Jnum 2001 2002 2003 0023003 0073009 7003005

Тауарлар Onum Products Summa 0023003 0023003 0073009 0073009 0073009 7003005 Шырын Конфеттер Лимонад Шырын Печенье Йогурт 302,50 464,69 271,83 409,67 1031,63 5160,45

келесі постреляциялық кестедегідей беріледі

Cnum Jnum Products Summa 2001

2002

2003 023003

0073009

7003005 Шырын Конфеттер Лимонад Шырын Печенье Йогурт 302,50 464,69 271,83 409,67 1031,63 5160,45

Осылайша, постреляциялық модель өзара байланысқан реляциялық кестелердің жиынтығын бір постреляциялық кесте ретінде көрсетуге мүмкіндік береді. Өрістердің бір-біріне салынуын қамтамасыз етумен қатар постреляциялық модель ассоциацияланған көпмәнді өрістерді (көптеген топтарды) қолдайды. Ассоциацияланған өрістердің жиынтығы ассоциация деп аталады. Бұл ретте ассоциацияның бір бағанының жолындағы алғашқы мәні ассоциацияның барлық өзге бағандарының алғашқы мәндеріне сәйкес келеді. Осыған ұқсас түрде бағандардың барлық екінші мәндері және т.б. байланысқан. Өрістердің ұзындығына және кестелерде қалыпқа келтірілмеген Деректер сақталатын болғандықтан, Деректердің тұтастығы мен қарама-қайшылықсыздығын қамтамасыз ету мәселесі туындайды. Аталған мәселе МҚБЖ-ға клиент-серверлік жүйелерде сақталатын процедураларға ұқсас механизмдерді енгізу арқылы шешіледі. Өрістердегі мәндерді бақылау қызметін сипаттау үшін Деректерге жүгінуге дейін немесе содан кейін автоматты түрде шақырылатын процедураларды (конверсия кодтары мен корреляция кодтары) құру мүмкіндігі бар. Корреляция кодтары Деректерді оқығаннан кейін, бірден оларды өңдеудің алдында орындалады. Конверсия кодтары, керісінше, Деректерді өңдеуден өткізгеннен кейін орындалады. Постреляциялық модельдің артықшылығы сол – ол байланысқан реляциялық кестелердің жиынтығын бір ғана постреляциялық кесте түрінде беру мүмкіндігі. Бұл ақпаратты ұсынудың жоғары түрдегі көрнекілігін және оны өңдеу тиімділігінің артуын қамтамасыз етеді. Сонымен қатар, постреляциялық модель Деректердің шектен тыс көп болуына жол бермейді. Постреляциялық модельдің кемшілігі сақталатын Деректердің тұтастығы мен қарама-

қайшылықсыздығын қамтамасыз ету мәселесін шешудің қиындығы болып отыр. Деректердің постреляциялық үлгісіне негізделген МҚБЖ-лардың қатарына Bubba және Dasdb жүйелері жатады. Көпөлшемді модель.Деректер қорының көпөлшемді моделі ақпаратты интерактивтік аналитикалық өңдеуден өткізуге арналған. Деректердің тиімді жедел түрде өңдеуден өткізілуін қамтамасыз ететін реляциялық модельге қарағанда, көпөлшемді модель талдау жүргізу мен шешім қабылдау барысында Деректерді жедел түрде өңдеуге мүмкіндік береді. Сол арқылы ол жасанды интеллект жүйелерінде, атап айтқанда эксперттік жүйелерде қолданыста болады. Модельдің көпөлшемділігі Деректерді сипаттау және манипуляциялау кезінде Деректер құрылымын көпөлшемдік логикалық тұрғыдан ұсынуды білдіреді. Мысалы, жоғарыда келтірілген Тапсырыстар мен Тауарлар атты реляциялық кестені гиперкуб деп аталатын келесі көпөлшемді кесте түрінде көрсетуге болады.

2001 2002 2003 Конфеттер Лимонад Шырындар Печенье 464,69

302,50

271,83 409,67 1031,63

Йогурт

5160,45

Көпөлшемді модельдер үшін Деректердің агрегирлігі, тарихилығы мен болжамдылығы тән болып келеді. Деректердің агрегирлігі ақпаратты оны жалпылаудың әр түрлі деңгейлерінде қарастыру дегенді білдіреді. Ақпараттық жүйелерде пайдаланушы үшін ақпаратты ұсынудың егжей-тегжейлік дәрежесі оның дегейіне тәуелді: талдаушы, пайдаланушы-оператор, басқарушы, жетекші болады. Деректердің тарихилығы Деректер мен олардың өзара байланыстарының өзгермейтіндігін жоғары деңгейін қамтамасыз етуді, сондай-ақ Деректерді міндетті түрде уақытқа байланыстыруды білдіреді. Деректердің статикалығы оларды өңдеу кезінде арнайы жүктемелеу, сақтау, индексациялау мен таңдау әдістерін қолдануға мүмкіндік береді. Деректерді уақытша байланыстыру, таңдау құрамында уақыт пен ай-күн мәндері бар сұраныстарды жиі орындау үшін қажет. Деректерді өңдеу мен пайдаланушыға ұсыну кезінде Деректерді уақыт бойынша ретке келтіру қажеттілігі ақпаратты сақтау мен оған қол жеткізу механизмдеріне талаптар қояды. Сұраныстарды өңдеуден өткізу уақытын азайту үшін Деректер әрдайым өздері жиі сұралатын тәртіппен сұрыпталған күйде болғаны дұрыс. Деректердің болжамдалуы болжамдау қызметтерін ұсыну мен оларды түрлі аралықтарында қолдануды білдіреді. Деректер моделінің көпөлшемділігі цифрлық Деректерді көрсетудің көпөлшемділігін емес, сипаттау кезінде және Деректерді манипуляциялау әрекеттерінде ақпараттың құрылымын көпөлшемдік логикалық кейіптеуді білдіреді. Реляциялық модельмен салыстырғанда Деректердің көпөлшемді ұйымдастырылуы барынша жоғары көрнекілік пен ақпараттылықты береді. Егер екіден артық өлшемділікке ие модель туралы сөз болса, онда ақпараттың көпөлшемдік нысандар (үш, төрт және одан да артық өлшемдік гиперкубтар) түрінде ұсынылуы міндетті емес. Пайдаланушыға бұл жағдайларда да екі өлшемдік кестелерді немесе графиктерді пайдаланған ыңғайлы болады. Бұл ретте Деректер түрлі дәрежедегі егжей-тегжейлігімен орындалған көпөлшемдік Деректер қоймасынан алынған «қиындықтар» түрінде болып келеді. Көпөлшемдік модельдің негізгі ұғымдарына өлшем мен ұяшық жатады. Өлшем дегеніміз – бұл гиперкуб қырларының бірін құрайтын бір типтік Деректердің жиыны. Көптеген жағдайларда келесі уақыттық өлшемдер: күндер, айлар, тоқсандар, жылдар қолданылады. Сонымен қатар географиялық өлшемдер: қалалар, аудандар, аймақтар, елдер және т.б. пайдаланылады. Ұяшық дегеніміз – өлшем мәндерінің белгіленген жиыны бір мәнді анықталған өріс. Деректер қорының көпөлшемдік моделінде бірақатар арнайы амалдар қолданылады: қиындыны құрастыру, айналдыру, агрегациялау және детализациялау. Қиынды бір немесе бірнеше өлшеулерді белгілеу арқылы алынатын гиперкубтың ішкі жиыны. Қиындыны құрастырған соң біз сол арқылы гиперкубтың белгілі бір шектелген бөлігіне ауысамыз. Айналдыру өлшеулердің орналасу тәртібін өзгерту үшін қолданылады. Агрегациялау мен детализациялау сәйкесінше пайдаланушыға Деректердің көпөлшемді моделінің басты артықшылығы сол уақытпен байланысты Деректердің үлкен көлемдерін талдамалық өңдеудің ыңғайлы және тиімді болуы. Ұқсас Деректерді реляциялық модельге негізделген өңдеуді ұйымдастыру барысында МҚ өлшемділігіне байланысты әрекеттердің еңбек сыйымдылығы сызықтық емес түрде өседі және индексацияға жұмсалатын жедел жады шығыны елеулі түрде артады. Деректердің көпөлшемдік моделінің кемшілігі – ол ақпаратты әдеттегі жедел өңдеудің қарапайым міндеттері үшін тым орасан зор болып келуінде. Деректердің көпөлшемді модельдерін қолдайтын жүйелердің мысалдары: Essbase (Arbor Software), Media Multi-matrix (Speedware), Oracle Express Server (Oracle) және Cache (InterSystems). Кейбір программалық өнімдер, мысалы, Media/MR (Speedware) бір мезгілде көпөлшемді және реляциялық МҚ-лармен жұмыс істеуге мүмкіндік береді. Деректердің ішкі моделі көпөлшемді модель болып табылатын Cache МҚБЖ-да Деректерге қол жеткізудің үш тәсілі жүзеге асырылған: тікелей (көпөлшемді массивтердің тораптары деңгейінде), объектілі және реляциялық. Объектілі-бағытталған модель. Бұл модельдің негізін объектілі-бағытталған программалаудың идеялары мен принциптері (ұстанымдары) құрайды. Объектілі-бағытталған МҚ-ның логикалық құрылымы сырттай иерархиялық МҚ-ның құрылымына ұқсас болып келеді. Олардың бір-бірінен негізгі айырмашылығы – Деректерді манипуляциялау әдістерінде. Деректер қорының объектілі-

бағытталған моделі өзара күштің байланысқан Деректермен жұмыс істеуге мүмкіндік береді. Деректер мен оларды өңдеуден өткізу әдістерінің арасында программалаудың объектілі-бағытталған тілдеріндегі сәйкес құралдарға ұқсас механизмдердің көмегімен өзара байланыстар орнайды. Объектілердің қасиеттері тек real, char және сол сияқты стандартты типтермен ғана емес, сондай-ақ пайдаланушылардың өздері жасайтын типтермен де сипатталуы мүмкін. Класстар деп аталатын бұл типтер көптеген объектілерді құруға арналған шаблон (қимаулігі) болып табылады. Class типінің мәні болып табылатын объектілер осы класстың даналары деп аталады, мысалы, келесі суретте көрсетілгендей (1.6-сурет).

Orders Onum Integer Amount Real Data Date Customers Class Salespeople Class

Customers Cnum Integer Cname Char City Char

Salespeople Snum Integer Sname Char City Char Manager Integer

1.6-СУРЕТ. ОБЪЕКТІЛІ-БАҒЫТТАЛҒАН МОДЕЛЬДЕГІ ЖАЗБАЛАРДЫ БАЙЛАНЫСТЫРУ

Класстардың негізінде инкапсуляция, мұра етіп алу және полиморфизм деген үш іргелі ұстаным жатыр. Инкапсуляция нысанды өзге класстардың объектілерінен оқшаулауы, оның қасиеттерінің көрінуін шектейді. Қасиеттің мәні сол инкапсуляцияланған нысанмен анықталады. Мұра етіп алу, керісінше, аталық объектінің қасиеттерін оның барлық ұрпақтарына таратады. Полиморфизм әртүрлі кластағы объектілерде атаулары бірдей қызметтер мен өңдеу процедураларын (әдістерін) қолдануға жол ашады. Мәліметтермен әрекеттерді орындау үшін қарастырылып отырған МҚ моделінде инкапсуляция, мұра ету және полиморфизмнің объектілі-бағытталған механизмдерімен күшейтілген логикалық амалдар қолданылады. SQL командаларына (мысалы, МҚ құру үшін) ұқсас амалдар шектелген түрде қолданылуы мүмкін. МҚ құру және модификациялау Деректерді жылдам іздестіруге арналған ақпаратты қамтыған индекстерді (индекстік кестелерді) автоматтты түрде қалыптастыру мен кейінгі түзетулерді жүзеге асырумен ілесе жүргізіледі. Деректердің объектілі-бағытталған моделінің реляциялық модельмен салыстырғандағы негізгі артықшылығы – объектілердің күрделі өзара байланыстары туралы ақпаратты бейнелеу мүмкіндігі. Деректердің объектілі-бағытталған моделі Деректер қорының жеке жазбасын идентификациялап, оларды өңдеуден өткізудің қызметтерін анықтауға мүмкіндік береді. Объектілі-бағытталған модельдің кемшілігі – түсінудің жоғары деңгейдегі қиындығы, Деректерді өңдеуден өткізудің ыңғайсыздығы және сұраныстарды орындау жылдамдығының төмендігі. 90-шы жылдары объектілі-бағытталған Деректер қорын басқару жүйелерінің тәжірибелік түп нұсқалары болған. Қазіргі кезде мұндай жүйелер кең таралымға ие болды, атап айтқанда оларға келесі МҚБЖ-лар: POET (POET Software), Jasmine (Computer Associates), Versant (Versant Technologies), O2(Ardent Software), ODB-Jupiter («Интелтек Плюс» ғылыми-өндірістік орталығы), сондай-ақ Iris, Orion және Postgres жатады. Бақылау сұрақтары:

- Көпөлшемді модельдің қандай амалы гиперкубтың қандай да бір шектелген бөлігіне өтуді береді?
- Көпөлшемді модельдің қандай амалы өлшемдер ретін өзгерту үшін қызмет етеді?
- Көпөлшемді модельдің қандай амалы ақпараттардың барынша нақты түрде берілу үшін қызмет етеді?
- Көпөлшемді модельдің қандай амалы ақпараттардың жалпы берілулері үшін қызмет етеді?
- Көпөлшемді модельде гиперкубтың қырын құрайтын, Деректердің бір типті жиыны қалай аталады?
- Көпөлшемді модельде өлшемнің мәндерінің бекітілген жиыны қалай аталады? Әдебиеттер:
- Бидайбеков Е.Ы., Елубаев К., Шекербекова Ш.Т. Деректер қоры және ақпараттық жүйелер. Алматы., 2010.
- Глушаков С.В., Ломотько Д.В. Базы данных.-Харков: Фолио; М.:ООО «Издательство АСТ», 2002.
- Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных. Учебник для высших учебных заведений. Под редакцией проф. А. Д. Хомоненко. – Санкт-Петербург: «КОРОНА принт», 2000 г

8-ДӘРІС.

Тақырыбы: Деректер қорының үш деңгейлі архитектурасы Лекция жоспары:

- Деректер қорын берудің деңгейлері:қолданушылық деңгей, концептуальдық деңгей, физикалық деңгей.
- Деректер қорының физикалық және логикалық тәуелсіздігі.
- Деректер қорына басшылық жасау
- Деректер қорын жоспарлау, жобалау, жасау және іске қосу кезеңдеріндегі Деректер қоры әкімшілігінің негізгі қызметтері. Құзыреттіліктер- Деректерді ендіру, толтыру,реттеу,іздеу дағдыларын меңгеру Лекция мазмұны 1978 жылы ANSI/SPARC комитеті Деректер қоры жүйелерінің жалпылама құрылымын ұсынған болатын. Ол құрылым үш деңгейлі архитектура деген атауға ие болды, оған сәйкес Деректер қорын ұсынудың үш деңгейі болады: сыртқы, тұғырнамалық және ішкі. Сыртқы деңгей Деректердің пайдаланушылық ұсынылуын анықтайды. Сондықтан бұл деңгей

пайдаланушылық деңгей деп те аталады. Сыртқы деңгейде әрбір пайдаланушы топ Деректер қорындағы Деректер туралы өз кейіптемесін алады, онда Деректер элементтері мен олардың өзара қарым-қатынастарының пайдаланушыға бағытталған сипаттамасы беріледі. Түрлі кестелер, байланыстар, есеп берулер, формалар және т.б. өздері шешетін міндеттерге сәйкес Деректер қорының пайдаланушылық кейіптемесіне сәйкес келеді. Кейіптеменің тұғырнамалық немесе логикалық деңгейі дегеніміз – бұл Деректер қорын жасаудың деңгейі. Бұл деңгейде талап етілген міндетті шешуге қажетті объектілер анықталады және олардың арасындағы байланыстар орнатылады. Тұғырнамалық жобалаудың нәтижесінде объектілердің, олардың элементтерінің және олардың арасындағы қарым-қатынастарының тұғырнамалық моделі, сұлбасы, объектінің логикалық сипаттамасы қалыптасады. Мысалы, Orion оқу Деректер қорына арналған тұғырнамалық сұлба келесі түрде болады (1.7-сурет).

Salespeople Customers Orders 1:M 1:M

1.7-СУРЕТ. «ORION» ОҚУ ДЕРЕКТЕР ҚОРЫНЫҢ ТҰҒЫРНАМАЛЫҚ СҰЛБАСЫ

Жасалған тұғырнамалық сұлбаны жүзеге асыру үшін Деректер қорының ыңғайлы моделі таңдалып алынады, Деректердің құрылымы сипатталады. Ішкі деңгей Деректер қорына физикалық көзқарасты қамтамасыз етеді. Бұл деңгей үшін жүйелік программалаушылар, Деректердің физикалық қорын жобалаушылар жауап береді, олар Деректерді кодтау мен компьютерде және тасымалдағыштарда кейіптеу мәселелерін шешеді, Деректерді шұғыл түрде оқу мен оларды өзгерту мақсатында Деректерге қол жеткізу әдістерін анықтайды. Деректерге қол жеткізудің негізгі әдістерін индексстеу, хэш-адресстеу және сығу болып табылады. Хэш-адресстеу дегеніміз – қандай да бір алгоритм (хэш-қызметі) бойынша тасымалдағыштағы мекен-жайды кестенің қандай да бір өрісінің, көбінесе басты өрісінің негізінде есептеп шығу. Деректерді жазу кезінде хэш-қызметінің көмегімен мекен-жайы есептеп шығарылады және содан кейін Деректер осы есептелген мекен-жай бойынша тасымалдағышқа жазылады. Деректерді оқу кезінде хэш-қызметі алдымен Деректер орналасқан мекен-жайды есептеп шығады, содан кейін оқиды. Деректер қорының физикалық құрылымын жасаушылар Деректерді орналастыру тәсілдерін қолда бар нақты құрал-жабдықтарды, құрылғыларды ескере отырып анықтайды. Осылайша, үш деңгейлік архитектура Деректер қорын пайдаланушылық кейіптеменің физикалық кейіптемеге қатысты логикалық тәуелсіздігін көрсетеді. Келесі суретте (1.8-суретте) Деректер қорының үш деңгейлік архитектурасы келтірілген.

Мәліметтер қорын әкімшіліктендіру Мәліметтер қорының тұғырнамалық моделін жасау Мәліметтер қорының физикалық құрылымын жасау Ақырғы пайдаланушы Пайдаланушының кейіптемесі

Логикалық (тұғырнамалық) деңгей

Физикалық деңгей

1.8-СУРЕТ. ДЕРЕКТЕР ҚОРЫНЫҢ ҮШ ДЕҢГЕЙЛІК АРХИТЕКТУРАСЫ

Әдетте, Деректер қорының логикалық моделі мен физикалық құрылымын жасаушылар Деректер қорының әкімшісі (администраторы) деп аталатын топты құрайды. Деректер қоры әкімшісінің қызметтеріне төмендегілер жатады:

- Деректер қорына қатысты талаптарды әзірлеп шығару;
- Деректер қорын жобалау;
- Деректер қорын қалыптастыру;
- Деректер қорын тиімді пайдалану және оған ілесе жүру. Деректер қоры өзі арналып құралған, шешілетін міндетке сәйкес келуі тиіс. Деректер қорын жобалауға логикалық модельді жасауды, шифрлеуді, кодтауды және өзге де қорғау құралдарын қамтитын тасымалдағыштардағы Деректерді сақтау тәсілдері жатады. Деректер қорын тұғырнамалық жобалаудың ұстанымдары Пайдаланушылардың Деректер қорының көмегімен шешетін міндеттері, әдетте, қоршаған болмыспен, осы болмыстың нақты нысандарымен байланысты болады. Сондықтан Деректер қорын әзірлеуге кіріскен кезде пайдаланушылардың жасалып жатқан Деректер қорына қоятын талаптарына қатысты нысандардың табиғатын зерттеп шығу қажет. Пайдаланушылардың жасалып жатқан Деректер қорына қоятын талаптарын көрсететін объектілердің жиынтығын модель деп атайық. Осылайша, модель қойылған міндетті шешу үшін болмыстың қажетті кейіптемесін береді. Әрине, модель пайдаланушылардың талаптарын дәл бейнелеуі тиіс, онда қойылған міндетті шешуге қажетті объектілер ғана болуы тиіс, онда сол міндетке керек емес нысандар болмауы керек. Сондықтан Деректер қорын жобалау мен жасау кезінде модельді қалыптастыру процесі маңызды роль атқарады. Егер бұл процесс тиісті деңгейде орындалатын болса, онда модель міндеттерді дұрыс шешу көзіне айналады. Әзірленген модель тұғырнамалық модель деп аталады. Деректердің тұғырнамалық моделі сонымен қатар семантикалық модель деп те аталады, себебі ол нақты әлемдегі заттардың мәндерін белгілеп отырады. Тұғырнамалық модельдің басты

элементтері объектілер мен қатынастар болып табылады. Модельдегі міндетті шешу үшін қажетті нақты әлемнің категорияларын бейнелейтін нысандар нақты заттар (физикалық нысандар), сондай-ақ абстрактілі (тұғырнамалық) заттар болуы мүмкін. Нақты нысандардың мысалдары адамдар, машиналар, ағаштар, кітаптар, құрылыстар және т.б. болып табылады. Компаниялар, іскерлік әрекеттер, білімдер, дағдылар және т.б. – бұл абстрактілі нысандар. Бір нысанға көптеген бір типті элементтер жиыны біріктіріледі, ол объектілі жиын деп аталады. Жобалау барысында объектілі жиындарды нақтылау мен қорытындылау қажеттілігі туындайды. Мысалы, келесі суретте (1.9-сурет) «Заңды тұлға» және «Жеке тұлға» деген объектілі жиындар «Клиент» объектілі жиынының нақтылануы болып табылады.

Клиент

Заңды тұлға Жеке тұлға нақтылау

1.9-СУРЕТ. ОБЪЕКТІЛІ ЖИЫНДЫ НАҚТЫЛАУ

Жалпы жағдайда, нақтылау дегеніміз – бұл басқа объектілі жиынның ішкі жиыны болып табылатын объектілі жиын. Қорытындылау дегеніміз – бұл нақтылауға кері келетін процесс және ол өз құрамында өзге объектілі жиындарды қамтитын объектілі жиын болып табылады. Тұғырнамалық модельдің басқа элементі қатынас болып табылады. Қатынас екі объектілі жиынды байланыстырады. Қатынас бір объектілі жиынның элементі мен басқа объектілі жиынның элементінен тұратын жұптарды құрайды. Осылайша, қатынас бастапқы объектілі жиындардың сәйкес элементтерінен көптеген жұптарды құрайды. Қатынастың өзі құрамдас деп аталатын объектілі жиын болып табылады. Қатынас көбінесе етістік арқылы беріледі. Мысалы, «еркек» және «әйел» деген объектілі жиындар «неке құрған» деген қатынаспен байланыстырылуы мүмкін және сол арқылы «жанұялық жұп» деген құрамдас жиынды құрайды. Құрамдас жиын келесі суретте (1.10-сурет) көрсетілгендей өзге объектілі жиынмен байланыстырылуы мүмкін.

еркек әйел неке құрған жанұялық жұп адрес тұрады

1.10-СУРЕТ. ҚҰРАМДАС ЖИЫНДЫ БАЙЛАНЫСТЫРУ

Қатынас бір объектілі жиынның басқа объектілі жиынның бір элементімен байланысқан элементтерінің санын көрсететін қуаттылығымен сипатталады. Ең төмен және ең жоғары қуаттылықтар болады. Мысалы, «неке құрған» қатынасының ең жоғары қуаттылығы әрбір бағытта 1-ге тең, басқаша айтқанда 1.11-суретте көрсетілгендей әрбір еркек тек бір ғана әйелмен неке құра алады және, керісінше, әрбір әйел тек бір ғана еркекпен неке құра алады (1.11-сурет). еркек әйел 1 неке құрған 1

1.11-СУРЕТ. БІРДІҢ-БІРГЕ ҚАТЫНАСЫ

Ең төмен қуаттылық бұл мысалда 0-ге тең. Қуаттылығы екі бағытта да 1-ге тең болатын қатынас бірдің-бірге қатынасы деп аталады. Егер бір бағыттағы қатынастың қуаттылығы 1-ге тең болса, ал екіншісінде – көп болса, онда мұндай қатынас бірдің-көпке қатынасы деп аталады. Келесі суретте (1.12-сурет) бірдің-көпке қатынасы көрсетілген.

қызметкерлер бөлім М жұмыс істейді 1

1.12-СУРЕТ. БІРДІҢ-КӨПКЕ ҚАТЫНАСЫ

Келесі суретте (1.13-сурет) көптің-көпке қатынасы көрсетілген,

студенттер курс М тыңдайды М

1.13-СУРЕТ. КӨПТІҢ-КӨПКЕ ҚАТЫНАСЫ

Бұл жағдайда студент көп курстарға қатысады, және бір курсқа көптеген студенттер қатысады. Бинарлық және n-дік қатынастарға бөлінеді. Бинарлық қатынастар дегеніміз – бұл екі объектілі жиындардың арасындағы қатынастар. Жоғары ретті қатынастар немесе n-дік қатынастар дегеніміз – бұл үш және одан да көп объектілі жиындардың арасындағы қатынастар. Мысалы, келесі суретте (1.14-сурет)

Тұрмыстық аспаптар заводы шаңсорғыш Кір жуу машиналары өндіреді

1.14-СУРЕТ. N-ДІК ҚАТЫНАСТАР

«тұрмыстық аспаптар заводы», «шаңсорғыш» және «кір жуу машиналар» деген объектілі жиындар «өндіреді» деген қатынаспен байланысқан. Объектілі жиынның элементтері атрибуттардың қандайда бір саңына ие болады. Мысалы, адамның есімі, туған күні, жынысы, мекен-жайы және т.б. Деректер болады. Жалпы жағдайда атрибут бір объектінің екінші объектіге функционалдық қатынасын анықтайды, яғни олар өзара атрибуттар жиынтығымен ажыратылады. Атрибуттың мәні объектілі жиынның әрбір элементі үшін бір мәнді анықталған. Егер ол анықталмаған болса, онда сол

элементке арналған атрибут бос мәнге ие болады. Бір элементке арналған атрибуттардың мәндері, мысалы, жасы, мекен-жайы және т.б. өзгеріп отыруы мүмкін. Сол себептен мәндері өзгермейтін атрибуттарды білген маңызды, себебі оларды кілт ретінде пайдалануға болады. Егер де нысан басқа нысанның нақтыламасы болып табылса, осы нақтыланған нысан жалпылама нысанның барлық атрибуттары мен қатынастарын мұраға алады. Сонымен қатар, нақтыланған нысанның өз атрибуттары пайда болуы мүмкін. Мысалы, келесі суретте (1.15-сурет) нақтыланған нысан «үйленген», компанияда жұмыс істейді, яғни «жұмыс істейді» қатынасын мұраға алады, сонымен қатар «адам» жалпылама объектісінің барлық атрибуттарын (есімі, мекен-жайы, туған жылы) мұраға алады, және өзіндік «зайыбы» атрибуты болады.

компания адам Жұмыс істейді

бойдақ үйленген нақтылау Есімі Мекен-жайы Туған жылы зайыбы

1.15-СУРЕТ. АТРИБУТТАР МЕН ҚАТЫНАСТАРДЫ МҰРАҒА АЛУ

Қомақты Деректердің қорын жобалау кезінде жасаушылардың әртүрлі топтары қатыстырылуы мүмкін, олар Деректердің жалпы қорының шектелген бөлігінің кейіптемесі болып табылатын Деректердің бірнеше моделін құрайды. Бұл модельдерді Деректердің бір ғана тұғырнамалық моделіне біріктіру қажеттілігі туындайды. Модельдерді біріктіру процесіне артық, қайталана беретін объектілі жиындарды, қатынастар мен атрибуттарды жою, сондай-ақ Деректердің кейіптемесін анықтайтын жаңа қатынастарды анықтау арқылы анықтау жатады. Бақылау сұрақтары:

- Деректер қорының құрылымы Деректердің берілуінің қай деңгейінде сипатталады?
- Деректердің берілуінің қандай деңгейі тасымалдағыштағы Деректердің орналасу тәсіліне сәйкес?
- МҚ берілуінің қанша деңгейі бар?
- Пайдаланушы деңгейінде Деректер қоры қандай түрге ие болады? Әдебиеттер:
- Бидайбеков Е.Ы., Елубаев К., Шекербекова Ш.Т. Деректер қоры және ақпараттық жүйелер. Алматы., 2010.
- Глушаков С.В., Ломотько Д.В. Базы данных. -Харков: Фолио; М.:ООО «Издательство АСТ», 2002.
- Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных. Учебник для высших учебных заведений. Под редакцией проф. А. Д. Хомоненко. – Санкт-Петербург: «КОРОНА принт», 2000 г

9-ДӘРІС.

Тақырыбы: Деректер қорын жобалау. Лекция жоспары:

- Деректер қоры құрылымына әкімшілік жасаудың модельдері, әдістері және программалық құралдары.
- Деректер тұтастығын бақылау.
- Деректер қорында ақпараттарды қорғау. Құзыреттіліктер- Деректер қорында формалармен жұмыс істеуді меңгеру
Лекция мазмұны Деректер қорымен жұмыс істейтін ақпараттық жүйелердегі жалпыға арналған программалық қамтамасыз ету Деректер қорын басқару жүйесі (МҚБЖ) деп аталады. Деректер қорын басқару жүйелері – бұл Деректер қорын құруға, енгізуге және өңдеуден өткізуге арналған тілдік және программалық құралдардың кешені. Әлемде МҚБЖ-ның 1000-нан астам түрі бар. Деректер қорын басқарудың әр алуан жүйелері Деректердің әр түрлі модельдерін пайдаланады. Деректер қорының әр түрлі модельдеріне негізделген МҚБЖ-лардың мысалдарына төмендегілер жатады: Иерархиялық модель – IMS PC/Focus, Team-UP, Data Edge; Желілік модель- IDMS, db-Vista III; Реляциялық модель - Dbase, DB2(IBM), FoxBase және FoxPro (Fox Software), Paradox, Dbase for Windows және InterBase (Borland), Visual FoxPro және Access (Microsoft), Clarion (Clarion Software), Ingress (ASK Computer Systems), Oracle (Oracle), Informix; Постреляциялық модель - Bubba, DasDb; Көпөлшемді модель - EssBase (Ardor Software), Media Multi-matrix (Speedware), Oracle Express Server (Oracle), Cache (InterSystems); Объектілі бағытталған модель – POET(POET Sofyware), Jasmine (Computer Associates), Versant (Versant Technologies), O2 (Ardent software), Iris, Orion, PostgreSQL, ODB-Jupiter. Деректер қорының сервері ретінде келесі МҚБЖ-лар болуы мүмкін: NetWare SQL (Novell), MS SQL Server (Microsoft), InterBase (Borland), SQLBase Serve (Gupta), Intelligent DataBase (Ingress), Oracle, Informix көптеген пайдаланушыларға арналған МҚБЖ болып табылады, ол текті емес ортада жұмыс істейді (әртүрлі ЭЕМ мен ЖЖ). МҚБЖ-ның негізгі қызметіне келесілер жатады:
- Деректерді құру және бақылау;
- Деректерді қорғау және бақылау;
- көптеген пайдаланушылардың Деректерге қол жеткізуі;
- пайдаланушыға Деректерді манипуляциялау мүмкіндігін ұсыну;

- қолданбалы программаларды құру мүмкіндігі. Деректерді құру мен бақылау Деректер сөздігінің көмегімен орындалады. Деректердің сөздік/каталогы Деректердің құрылымы, Деректердің типтері, оларды ұсыну форматтары, Деректердің өзара байланысу сұлбалары, пайдаланушылар, Деректерді қорғау мен оларға қол жеткізу кодтары туралы ақпаратты орталықтандырылған түрде сақтауға арналған. Көбінесе сөздіктің қызметін МҚБЖ-лар орындайды және олар жүйелік мәзірден шақыртылады немесе утилиттері арқылы орындалады. Мысалы, Orion атты оқу Деректер қорының келесі түрдегі сөздігі болуы мүмкін:

```
SYSTABLES NAMECREATORCOLCOUNT Salespeople AGU 4 Customers AGU 3 Orders AGU 5 SYSCOLUMNS NAMETBNAME
COLTYPE Snum Salespeople Integer Sname Salespeople Char City Salespeople Char Manager Salespeople Integer Cnum
Customers Integer Cname Customers Char City Customers Char Onum Orders Char Amount Orders Real Date Orders Data
Cnum Orders Integer Snum Orders Integer
```

Реляциялық МҚБЖ-ларда Деректер сөздігі реляциялық кестелер жинағы түрінде

сақталады. Сөздік/каталог Деректердің анықтамаларын Деректердің өзінен бөлек түрде қолдауға мүмкіндік береді, сол арқылы Деректердің тұтастығын қамтамасыз етеді. Деректерді қорғау Деректерге рұқсатсыз қол жеткізуді болдырмау мақсатында парольдауды, кодтауды (шифрлеу) және Деректердің кейіптеулерін құруды қамтамасыз етеді. Деректерді деңгейіндегі тұтастық мен қарама-қайшылықсыздық Деректер элементтерінің мәніне шектеулер қоюмен, сондай-ақ Деректер қорына өзгерістер енгізу кезінде резервтік көшірмелерді автоматты түрде құру арқылы қамтамасыз етіледі.

Сілтемелер деңгейіндегі тұтастық сыртқы кілттердің мәндері үшін әрдайым соған сәйкес бастапқы кілттердің мәндері болуы тиіс дегенді білдіреді. Бастапқы кілті бар кестелердегі, осы бастапқы кілтке сілтемелер (сыртқы кілттер) бар кестелердегі жазбаларды өзгерткен кезде немесе жойғанда МҚБЖ сыртқы кілттер мәндеріне «ауада» тұрып қалмас үшін сәйкес өзгертулер жасалуын бақылап отырады. Бұл жағдайда сыртқы кілттердің сәйкес мәндері бар жазбалар жойылады, немесе оларға Null-мәндер не келісім бойынша мәндер беріледі, немесе олар өзгеріссіз қалады. Бұл амалдар CREATE TABLE, DROP TABLE және ALTER TABLE командаларында қарастырылған. Сілтемелер деңгейіндегі Деректер тұтастығын сақтау 3.1-суретте көрсетілгендей Деректер қорындағы каскадты өзгеріске алып келуі мүмкін.

Бастапқы кілт ... Басқа өрістер ...

1005

...

Кесте A

Бастапқы кілт Сыртқы кілт ... Басқа өрістер ...

2003 1005

...

2010 1005

...

Кесте B

Бастапқы кілт Сыртқы кілт ... Басқа өрістер ...

3004 2003

...

Кесте C

МҚБЖ-дағы Деректерді тұтастығын сақтау үшін транзакциялар қолданылады. Транзакция деп Деректер қорындағы МҚБЖ басынан аяғына дейін қадағалайтын, Деректермен жүргізілетін амалдардың қандай да бір тізбегі аталады. Транзакция сәтті аяқталған жағдайда амалдардың нәтижелері Деректер қорында сақталады. Құрал-жабдықтың немесе программалық қамтудың жұмысы тоқтаған кезде, сондай-ақ пайдаланушы транзакцияны аяқтаудан бас тартқан кезде транзакция аяқталмаған (үзілген) күйінде қалады. Бұл жағдайда Деректермен орындалған амалдардың нәтижелері Деректер қорына ауыстырылмайды. Деректерге қол жеткізу мүмкіндігі көптеген пайдаланушыларға Деректер қорына жылдам және тиімді түрде жүгінуге мүмкіндік береді. МҚБЖ көптеген пайдаланушылар бір мезгілде қол жеткізуге ие болған жағдайда пайдаланушының бірі Деректерге жүгінсе, енді бірі сол кезде оларға өзгертулер енгізіп жататындай

келіспеушілік оқиғалардың туындамауын бақылап отырады. МҚБЖ пайдаланушыларға Деректерді манипуляциялауға, Деректер қорынан тікелей есеп берулерді сұрауға және алуға, сондай-ақ мерзімді есеп берулерді құрастыру үшін сұраныстарды сақтаудың мол мүмкіндіктерін ұсынады. Көптеген МҚБЖ-ларда қосымшаларда жасауға арналған ішіне орнатылған программалау тілдері болады. Бұл қосымшалар ехе-файлдар тәрізді Деректер қорымен дербес түрде, немесе сол МҚБЖ-ның құрамында жұмыс істеуі мүмкін. МҚБЖ-ның Деректермен жұмыс істеу қызметтері екі тілде қамтамасыз етіледі:

- Деректерді сипаттау тілі;
- Деректерді манипуляциялау тілі. Бұл тілдер көптеген МҚБЖ-ларда екі түрде QBE (Query by Example) үлгісі бойынша сұраныс жасау тілі ретінде және SQL (Structured Query Language) тілі ретінде жүзеге асырылған. QBE және SQL тілдерінің теориялық негізі реляциялық есептеу болып табылады. QBE тілі пайдаланушыларға Деректер қорын құру мен манипуляциялау үшін сауалнама формаларын ұсынады, оларды толтыру жұмысты жеңілдетеді. Тағы да бір айта кететін нәрсе, физикалық деңгейде әр түрлі МҚБЖ-лар Деректерді ұсынудың әр түрлі форматтары болады. Әр алуан МҚБЖ-лардың Деректері өзара үйлесімділігі үшін арнайы интерфейстер, драйверлер әзірленеді. Мысалы, Microsoft фирмасы ODBC (Open Database Connectivity) драйверін әзірлеп шығарды, ол аса кең тараған МҚБЖ-лар форматтарында Деректер қорымен жұмыс жүргізуге мүмкіндік береді. Көптеген МҚБЖ-лардың басқа форматта ұсынылған Деректер қорын оқуға арналған қоса орнатылған интерфейстері бар. МҚБЖ-ға программалардың келесі негізгі түрлері жатады:
- толық қызметті МҚБЖ;
- МҚ серверлері;
- МҚ клиенттері;
- МҚ-мен жұмыс істеу бағдарламаларын жасау құралдары. Толық қызметті МҚБЖ (ТҚМҚБЖ) алдымен үлкен машиналарға арналып, кейіннен мини-машиналар мен ДЭЕМ-дерге арналып пайда болған дәстүрлі МҚБЖ болып табылады. Барлық МҚБЖ-лардың арасында қазіргі заманғы толық қызметті МҚБЖ саны жағынан көп және өзінің мүмкіндіктері бойынша аса қуатты болып табылады. Толық қызметті МҚБЖ-ларға, мысалы, Clarion Database Developer, DataBase, Dataplex, dBase IV, Microsoft Access, Microsoft FoxPro және Paradox R: BASE сияқты пакеттер жатады. Әдетте ТҚМҚБЖ-лардың меню командалары арқылы МҚ-мен негізгі әрекеттерді орындауға мүмкіндік беретін дамыған интерфейсі болады: кестелердің құрылымын құрып, түрлендіруге, Деректерді енгізуге, сұраныстарды құрастыруға, есеп берулерді әзірлеуге, оларды басып шығаруға және т.б. сұраныстар мен есеп берулерді құрастыру үшін программалаудың қажеті жоқ, оның орнына QBE (Query by Example – үлгі бойынша сұраныстарды құрастыру) тілін қолданған ыңғайлы. Көптеген ТҚМҚБЖ-ларға кәсіби құрастырушыларға арналған программалау құралдары кіреді. Кейбір жүйелердің қосалқы құралы ретінде МҚ сұлбаларын жобалаудың қосымша құралдарында немесе CASE-ішкі жүйелерінде болады. Басқа МҚ-ларға немесе SQL-серверлердің Деректеріне қол жеткізуді қамтамасыз ету үшін толық қызметтік МҚБЖ-ларды факультативтік модульдер болады. МҚ серверлері ЭЕМ желілерінде Деректерді өңдеу болып орталықтарын құруға арналған. Қазіргі кезде МҚ-лардың бұл тобының аз болып табылады алайда олардың саны біртіндеп артуда. МҚ серверлері басқа программалар (клиенттер) сұратқан Деректерді басқару қызметтерін SQL операторларының көмегімен жүзеге асырады. Келесі бағдарламалар МҚ серверлерінің мысалы болып табылады: NetWare SQL (Novell), MS SQL Server (Microsoft), InterBase (Borland), SQLBase Server (Gupta), Intelligent Database (Ingress). Жалпы жағдайда МҚ серверлері үшін клиенттік программалар ретінде әртүрлі программалар қолданылуы мүмкін: ТҚМҚБЖ, электронды кестелер, мәтіндік процессорлар, электронды пошта программалары және т.б. Бұл жағдайда «клиент – сервер» жұбының элементтері бір немесе бірнеше программалық қамтамасыз етудің өндірушілеріне жатуы мүмкін. Клиенттік және серверлік бөліктер бір фирмада жасалған жағдайда олардың арасындағы қызметтердің үлестірілуі ұтымды түрде орындалған деп күту табиғи нәрсе. Қалған жағдайларда, әдетте, Деректерге «қандай жолмен болсын» қол жеткізуді қамтамасыз ету мақсаты көзделеді. Келесі жағдай осындай қосылыстың мысалы болып табылады: көп қызметтік МҚБЖ-лардың бірі сервер рөлін, ал екінші МҚБЖ (өзге фирма өндірген) – клиент рөлін атқарады. SQL Server (Microsoft) МҚ сервері үшін клиенттік (ауқымдық) программалар ретінде көптеген МҚБЖ-лар болуы мүмкін, мысалы: dBASE IV, Blyth Software, Paradox, DataEase, Focus,1-2-3, MDBS III, Revelation және басқалары сияқты. МҚ-лармен жұмыс істеу программаларын жасау құралдары келесі бағдарламалардың әралуан түрлерін:
- клиенттік программаларды;
- МҚ серверлері мен олардың жекелеген компоненттерін;
- пайдаланушылардың қосымшаларын құрастыруға қолданылуы мүмкін. Бірінші және екінші түрдегі программалар саны жағынан аздау болып келеді, себебі олар негізінен жүйелі программалаушыларға арналған. Үшінші түрдегі пакеттер көптеп саналуда, алайда олардың саны толық қызметтік МҚБЖ-ларға қарағанда аз. Пайдаланушы қосымшаларын жасау құралдарын программалау жүйелері жатады, мысалы Clipper, түрлі программалау тілдеріне

арналған әртүрлі программалау кітапханалары, сондай-ақ әзірлемелерді (соның ішінде клиент-сервер типіндегі жүйелерді) автоматтандыру пакеттері. Аса кең тараған құралдар ретінде келесі құрал-жабдықтық жүйелерді атап кетуге болады: Delphi және Power Builder (Borland), Visual Basic (Microsoft), SILVERRUN (Computer Advisers Inc.), S-Designor (SDP мен Powersoft) және ERwin (LogicWorks). Жоғарыда аталған құралдармен қатар Деректерді басқару мен МҚ қызмет көрсетуді ұйымдастыруға арнап түрлі қосымша құралдар, мысалы, транзакциялар мониторлары қолданылады (4.2 ішкі бөлімін қараңыз). МҚБЖ-ларды қолдану сипаты бойынша дербес және көп пайдаланушыға арналған деп бөлінеді. Дербес МҚБЖ-лар әдетте, дербес МҚ-лары мен олармен жұмыс істейтін онша қымбат емес қосымшаларды жасау мүмкіндігін қамтамасыз етеді. Дербес МҚБЖ-лар немесе олардың көмегімен жасалған қосымшалар көбінесе көп пайдаланушыға арналған МҚБЖ-ның клиенттік бөлігі ретінде қызмет атқаруы мүмкін. Дербес МҚБЖ-ларға, мысалы, Visual FoxPro, Paradox, Clipper, dBase, Access және т.б. жатады. Көп пайдаланушыға арналған МҚБЖ-ларға МҚ сервері мен клиенттік бөлігі кіреді және, әдетте, бір текті емес (ЭЕМ-дер мен амалдық жүйелері әртүрлі болып келетін) есептеу жүйесінде қызмет ете алады. Көп пайдаланушыға арналған МҚБЖ-ларға, мысалы Oracle мен Informix МҚБЖ-лар жатады. Қолданылатын Деректер моделі бойынша МҚБЖ-лар (МҚ тәрізді) иерархиялық, желілік, реляциялық, объектілі-бағытталған және басқа түрлерге бөлінеді. Кейбір МҚБЖ-лар бір мезгілде бірнеше Деректер моделіне қолдау көрсете алады. Пайдаланушы тұрғысынан МҚБЖ ақпаратты сақтау, өзгерту (толықтыру, редакциялау және алып тастау) және өңдеу қызметтерін сондай-ақ түрлі баспа құжаттары жасау мен алу қызметтерін жүзеге асырады. Деректер қорында сақталған ақпаратпен жұмыс істейтін МҚБЖ программалар мен пайдаланушыларға келесі екі тілі ұсынады:

- Деректерді сипаттау тілі – Деректердің логикалық құрылымын сипаттауға арналған декларативтік типтегі жоғары деңгейлі процедуралық емес тіл;
- Деректерді манипуляциялау тілі – Деректерімен негізгі амалдарды: енгізу, модификациялау және сұраныс бойынша Деректерді таңдауды, орындауды қамтамасыз ететін құрылымдардың жиынтығы. Аталған тілдердің түрлі МҚБЖ-ларда өзара айырмашылықтары болуы мүмкін. Стандартталған екі тіл аса кең таратылған: QBE (Query By Example) – үлгі бойынша сұраныс тілі және SQL (Structured Query Language) – сұраныстардың құрылымдалған тілі. QBE негізінен Деректерді манипуляциялау тілінің қасиеттерінен тұрады, ал SQL өзінде екі түрдегі тілдердің қасиеттерін – Деректерді сипаттау және манипуляциялау қасиеттерін жинақтаған. МҚБЖ-ның жоғарыда аталған қызметтері, өз кезегінде, одан төмен деңгейдегі келесі негізгі қызметтерді қолданады:
- сыртқы жадыдағы Деректерді басқару;
- жедел жадының буферлерін басқару;
- транзакцияларды басқару;
- МҚ-дағы өзгертулер журналын жүргізу;
- МҚ-ның тұтастығы мен қауіпсіздігін қамтамасыз ету. Оларды төмен деңгейлі деп атайық: Қазіргі МҚБЖ-ларда аталған қызметтерді жүзеге асырудың қажеттілігі мен ерекшеліктеріне қысқаша сипаттама беріп кетейік. Әртүрлі жүйелерде сыртқы жадыдағы Деректерді басқару қызметтерінің жүзеге асырылуы ресурстарды басқару деңгейінде де (ОЖ файлдық жүйелерін немесе ДЭЕМ құрылғыларын тікелей басқаруды қолдану арқылы), Деректерді басқару алгоритмдердің логикасы бойынша да ажыратылуы мүмкін. Деректерді басқару әдістері мен алгоритмдері негізінен МҚБЖ-ның «ішкі шарасы» болып табылады және оларды пайдаланушыға тікелей қатысы жоқ. Осы қызметін жүзеге асырылу сапасы арнайы АЖ-лардың мысалы, орасан зор МҚ-лармен, күрделі сұраныстармен, Деректерді өңдеуде өткізудің үлкен мөлшерімен жұмыс істеуінің тиімділігіне аса зор әсерін тигізеді. Деректерді буферлеу және соның салдарынан жедел жады буферлерін басқару қызметін жүзеге асырудың қажеттілігі жедел жады көлемінің сыртқы жады көлеміне қарағанда біз болуымен негізделген. Буферлер сыртқы және жедел жадының арасындағы алмасуды жеделдетуге арналған жедел жадыны саласы болып табылады. Буферлерде МҚ үзінділері уақытша сақталады, олардағы Деректер МҚБЖ-ға жүгінген кезінде пайдаланылады немесе өңдеуден өткізгеннен кейін Деректер қорына жасыру жоспарланады. Транзакциялардың механизмі МҚБЖ-да қордағы Деректердің тұтастығын қамтамасыз ету үшін қолданылады. Транзакция деп МҚБЖ басынан аяғына дейін қадағалап отыратын МҚ-дан Деректерімен жүргізілетін амалдардың қандай да бір бөлінбейтін тізбегі аталады. Егер де қайсыбір себептермен (құрал-жабдықтың істен шығуы және тоқтап қалуы, қосымшаны қоса алғанда, программалық қамтудың қателіктерін) транзакция аяқталмай қалса, онда оның жұмысы тоқтатылады. 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). Транзакцияның мысалы ретінде банктік жүйеде бір есепшоттағы ақшаны екіншісіне аудару амалын айтуға болады. Бұл жерде, ең болмағанда екі қадамдық үдеріс қажет. Алдымен бір есепшоттағы ақша алынады, сосын олар екінші есепшотқа салынады. Егер әрекеттердің біреуі сәтсіз орындалса, әрекеттің нәтижесі дұрыс болмайды және

шоттардың арасындағы теңгерім бұзылады. Транзакцияның қатар түрде орындалатын бір пайдаланушыға арналған және көп пайдаланушыға арналған МҚБЖ-ларда транзакцияларды бақылау өте маңызды. Соңғы жағдайда транзакциялардың топтамалануы туралы айтылады. Қатар орындалатын транзакциялардың топтамалануы деп транзакцияларды жүзеге асырудың қосынды әсері оларды біртіндеп орындаудың әсеріне тепе-тең болатындай етіп құруды айтады. Оларды орындау жоспарын (сериялық жоспарды) құруды түсінеді. Транзакциялар қоспасын қатар орындаған кезде келіспеушіліктер (блоктаулар) туындауы мүмкін, олардың шешу МҚБЖ-ның қызметіне жатады. Мұндай жағдайлар табылған кезде әдетте бір немесе бірнеше транзакциямен орындалған өзгерістердің күшін жою арқылы «қайта шегіндіру» жүргізіледі. МҚ-ғы өзгертулер журналын жүргізу (өзгертулерді журналдау) МҚБЖ ақпараттық істен шығулар мен тұрып қалулар, сондай-ақ программалық қамтуда қателер болған жағдайда қордағы Деректердің сақталу сенімділігін қамтамасыз ету үшін орындалады. МҚБЖ журналы дегеніміз – бұл пайдаланушының тікелей қолы жетпейтін және Деректер қорындағы барлық өзгерістер туралы ақпаратты жазу үшін қолданылатын ерекше МҚ немесе негізгі МҚ-ның бөлігі. Өртүрлі МҚБЖ-ларда журналға өртүрлі деңгейлердегі МҚБЖ-дағы өзгерістерге сәйкес келетін жазбалар, яғни сыртқы жадының бетін модификациялаудың ішкі ең кіші амалынан бастап МҚ модификациялаудың логикалық амалдарына (мысалы, жазбаны салу, бағанды жою, өрістегі мәнді өзгерту) және транзакцияларына дейін енгізілуі мүмкін. МҚ-ға өзгертулер журналын жүргізу қызметін тиімді орындау үшін журналдың өзін жұмыс қалпында ұстау және сақтау сенімділігін қамтамасыз ету қажет. Кейбір жағдайларда жүйеде журналдың бірнеше көшірмелері сақталады. МҚ-ның тұтастығын қамтамасыз ету, әсіресе МҚ-ны желілерде пайдалану үшін, МҚ-ның табысты түрде қызмет етуінің қажетті шарты болып табылады. МҚ-ның тұтастығы дегеніміз – Деректер қорында пәндік саланы толық қарама-қайшылықсыз және дұрыс көрсететін ақпарат бар екендігін білдіретін Деректер қорының қасиеті. МҚ-ның тұтастығын қолдауға оның бүтіндігін тексеру және Деректер қорында қарама-қайшылықтар анықталған жағдайда оның қалпына келтірулер жатады. МҚ-ның тұтас жағдайы бүтіндігін шектеу арқылы қорда сақталып жатқан Деректер қанағаттандыруы тиіс болатын шарттар түрінде сипатталады. Аталған шарттарының мысалы өздері туралы Деректер МҚ-да сақталатын объектілер атрибуттарының мүмкін болатын мәндерінің диапазондарын шектеу немесе реляциялық МҚ-лар кестелерінде қайталанып отыратын жазбалардың болмауы болып табылуы мүмкін. Қауіпсіздікті қамтамасыз етуге МҚБЖ-да қолданбалы программаларды, Деректерді шифрлеумен, парольмен қорғаумен, Деректер қорына және оның жекелеген элементтеріне (кестелерге, формаларға, есеп берулерге және т.б.) қол жеткізу деңгейлерін қолдаумен қол жеткізіледі.

МҚ-ны қорғау құралдары. Өр түрлі МҚБЖ-ларда МҚ-ны қорғау құралдары бір-бірінен біршама өзгеше болады. Borland және Microsoft фирмаларының қазіргі кездегі МҚБЖ –ларына талдау жүргізу негізінде МҚ-ларды қорғау құралдары шартты түрде екі: негізгі және қосымша топқа бөлінеді деп тұжырымдауға болады. Ақпарат қорғаудың негізгі құралдарына келесі құралдарды жатқызуға болады:

- парольдік қорғау;
- Деректер мен программаларды шифрлеу;
- МҚ объектілеріне қол жеткізу құқықтарын орнату;
- МҚ кестелерінің өрістері мен жазбаларын қорғау. Парольдік қорғау МҚ-ны рұқсат етілмеген қол жеткізуден қорғаудың қарапайым және тиімді әдісі болып табылады. Парольдерді соңғы пайдаланушылар немесе МҚ әкімшілері орнатады. Парольдерді есепке алу мен сақтауды МҚБЖ-ның өзі жүргізетін. Әдетте парольдер МҚБЖ-ның белгілі бір жүйелік файлдарында шифрленген түрінде сақталады. Сол себептен парольді табу және анықтау мүмкін емес. Парольді енгізгеннен кейін МҚБЖ пайдаланушысына қорғалған МҚ-мен жұмыс жүргізудің барлық мүмкіндіктерін ұсынылады. МҚБЖ-ның өзін парольмен қорғаудың қажеті жоқ. Деректерді (бүкіл қорды немесе жекелеген кестелерді) шифрлеу әдісі «сол МҚБЖ-дағы МҚ-ның форматын білетін» өзге программалар Деректерді оқи алмауы үшін қолданылады. Мұндай шифрлеу (Microsoft Access-те қолданылатын), онша да көп нәрсе бермейді, себебі кез келген тұлға МҚ-ны «туған» МҚБЖ көмегімен шифрін анықтауы мүмкін. Егер шифрлеу мен шифрді шешу парольді сұраса, онда пароль дұрыс енгізілген жағдайда шифрді шешіп алу мүмкін болады. Access-тің жасаушылары осыны келесі нұсқаларында ескеретініне сенімдіміз. Программалардың бастапқы мәтіндерін шифрлеу тиісті алгоритмдердің сипаттамасын өзіне оларға қол жеткізу рұқсат етілмеген пайдаланушыдан жасырып қалуға мүмкіндік береді. МҚБЖ-ның негізгі ресурстарын пайдалануды бақылау мақсатында көптеген жүйелерде МҚ объектілеріне қол жеткізу құқықтарын орнату құралдары болады. Қол жеткізу құқықтары объектілерге қатысты мүмкін болар әрекеттерді анықтайды. Нысанның иегері (нысанды құрастырған пайдаланушы), сондай-ақ МҚ әкімшісі барлық құқықтарға ие болады. Қалған пайдаланушылар өртүрлі нысандарға түрлі қол жеткізу деңгейлеріне ие бола алады. Кестелерге қатысты жалпы жағдайда келесі қол жеткізу құқықтары қарастырылуы мүмкін:
- Деректерді қарап шығу (оқу);
- Деректерді өзгерту (редакциялау);
- жаңа жазбаларды қосып жазу;

- Деректерді қосып жазу және алып тастау;
- барлық амалдар, соның ішінде кестенің құрылымын өзгерту. Кестедегі Деректерге, жекелеген өрістер мен жекелеген жазбаларға қатысты қорғау шаралары қолданылуы мүмкін. Өзімізге мәлім реляциялық МҚБЖ-ларда жекелеген жазбалар арнайы қорғалмайды, әйтсе де іс-тәжірибеден бұл қажет болған жағдайлардың мысалын келтіруге болады. Қол жеткізу құқықтарын бақылау өздерінде жекелеген жазбалардың идентификация бар объектілі – бағытталған МҚБЖ-ларда болуы тиіс сияқты (объектілі-бағытталған моделдің реляциялық моделден бір айырмашылығы). Кестелердің өрістеріндегі Деректерді қатысты қол жеткізудің келесі құқық деңгейлерін атап өтуге болады:
- қол жеткізуге толығымен тыйым салу;
- тек оқу;
- барлық адамдарға рұқсат ету (қарап шығу, жаңа мәндерді енгізу, алып тастау және өзгерту). Формаларға қатысты негізгі екі амал ескерілуі мүмкін: жұмыс істеу үшін шақырту және жасау (Құрастырушыны шақырту). Құрастырушыны шақыртуға тыйым салуды соңғы пайдаланушы байқаусызда қосымшаны бұзып алмауы үшін, экрандық формалар үшін дайын қосымшаларды дайындаған дұрыс. Экрандық формалардың өзінде бөлек элементтер де қорғалуы мүмкін. Мысалы, бастапқы кестенің кейбір өрістері мүлде болмауы немесе пайдаланушыдан жасырылуы мүмкін, ал кейбір өрістерді – қарап шығу үшін қол жеткізуге болады. Келесілерді ескермегенде есеп берулер көбінесе экрандық формаларға ұқсас болып келеді. Біріншіден, олар кестелердегі Деректерді өзгертуге мүмкіндік бермейді, ал екіншіден, олардың негізгі қызметі – ақпаратты қағазға басып шығару. Экрандық формалар тәрізді, есеп берулерге де оларды жасау құралдарын шақыртуға тыйым салу орнатылуы мүмкін. МҚБЖ қосымшаларында қолданылатын программалардың мәтіндерін (кездейсоқ және әдейі) қарау мен өзгертуге жол бермеу үшін шифрлеумен қатар, оларды парольмен қорғауды қолдануға болады. МҚ-ны қосымша қорғау құралдарына өздерін тікелей қорғау құралдарына жатқызуға болмайтын, әйтсе де Деректердің қауіпсіздігіне тікелей әсер ететіндерін жатқызуға болады. Оларға келесі құралдар жатады:
- типіне байланысты Деректердің мәндерін бақылаудың ішіне орнатылған құралдары;
- енгізілген Деректердің дәйектілігін арттыру құралдары;
- кестелердің байланысының біртұтастығын қамтамасыз ету құралдары;
- МҚ нысандарын желіде бірлескен түрде қолдануды ұйымдастыру құралдары. МҚ-да редакциялай отырып, пайдаланушы мәндер енгізілетін жолдың типіне сәйкес келмейтін мәндерді кездейсоқ енгізіп қоюы мүмкін. Мысалы, нысандық өріске мәтіндік ақпаратты енгізуге тырысады. Бұл жағдайдан МҚБЖ мәндерді бақылау құралдарының көмегімен енгізуге тосқауыл қояды және пайдаланушыға қате туралы дыбыстық сигнал беру арқылы, енгізілген символдың түсін өзгерту арқылы немесе өзге әдіспен хабарлайды. МҚБЖ-ға енгізілетін мәндердің дәйектілігін арттыру құралдары, өңдеуден өткізілген Деректердің семантикасымен байланысты барынша терең түрдегі бақылау жасауға қызмет етеді. Олар әдетте кестені құру кезінде мәндерге қатысты келесі шектеулерді көрсету мүмкіндігін қамтамасыз етеді: ең аз және ең көп мәндер; үнсіз қабылданатын мән (егер енгізу болмаса), міндетті түрде енгізуді талап ету; енгізудің қимаұлгісін (енгізу шаблону) тапсыру; енгізілген мәндер бақыланатын қосымша салыстыру кестесін көрсету және т.б. МҚ-дағы ақпараттың дәйектілігін бақылауды ұйымдастырудың аса дамыған түрі сақталынатын процедураларды жасау болып табылады. Сақталынатын процедуралар механизмі, серверде орналасқан МҚ-ларда қолданылады. Сақталып тұрған процедуралардың өзі алгоритмдері Деректермен қандай да бір қызметтерді (оның ішінде бақыланатыны да бар) орындауды қарастыратын программалар болып табылады. Процедуралар Деректермен бірге сақталады және қажет болған жағдайда қосымшалардан немесе МҚ-да қандай да бір оқиғалар орын алған жағдайда шақыртылады. Қолданбалы есептегі шешу, әдетте, кестелерден бірнешеуіндегі ақпаратты қажет етеді. Кестелердің өздері ақпаратты өңдеу ыңғайлы болу үшін және оны қайталануын болдырмау мақсатында қандай да бір әдіспен байланыстырылады (3.3 ішкі бөлім). Байланысқан кестелердің логикалық тұтастығын қолдау қызметтерін МҚБЖ міндетіне алады (3.4 ішкі бөлім). Өкінішке орай МҚБЖ-лардың барлығы бірдей бұл қызметтерді (толықтай) жүзеге асыра бермейді, бұл жағдайда байланыстардың нақтылығына жауапкершілік қосымша жүктеледі. МҚБЖ-ның кестелердің тұтастығын бақылауға қатысты мүмкін болар әрекеттерінің мысалын келтірейік. Екі кестенің арасында 1:М түріндегі байланыс бар делік, және соған сәйкес негізгі кестенің бір жазбасына қосалқы кестенің бірнеше жазбалары сәйкес келуі мүмкін. Жазбаларды қосалқы кестеге енгізген кесте жүйе негізгі кестедегі байланыс өрісіндегі сәйкес мәндердің бар болуын бақылайды. Егер енгізілетін мән негізгі кестеде болмаса, МҚБЖ жаңа жазбамен жұмысқа уақытша тосқауыл қояды және мәнді өзгертуді немесе оны тұтастай алып тастауды ұсынады. Қосымша кестелердің жазбаларын алып тастау «ауырсынбайтындай» орындалады, ал негізгі кестенің жазбалары туралы олай деп айтуға болмайды. Негізгі кестенің жазбасы қосымша кестенің бірнеше жазбаларымен байланысқан жағдайда, екі түрлі әрекет жасауға болады: бағыныңқы жазбаның ең болмаса біреуі қалса да негізгі жазбаны алып тастамау (жазбаларды пайдаланушы алып тастауы тиіс), немесе негізгі жазбаны және барлық бағыныңқы жазбаларды түгелдей алып тастау (каскадтық жою). Көп терезелі жүйелерде (қазіргі программалардың

түгелге жуығы) және Деректер қорымен жұмыс істейтін үлестірілген ақпараттық жүйелерде бірдей нысандармен (МҚ нысандарын бірлесіп пайдалану) түрлі амалдарды орындаудың арасындағы келіспеушіліктерді шешу мәселесі туындайды. Мысалы, жергілікті желіні пайдаланушылардың бірі МҚ-ға өзгертулер енгізіп, ал екіншісі оның құрылымын өзгерткісі келгенде не істеу керек? Мұндай жағдайлар үшін МҚБЖ-да келіспеушіліктерді шешу механизмдері ескерілуі тиіс. Әдетте желінің бірнеше пайдаланушылары бір мезгілде жұмыс істеген кезде, сондай-ақ бір компьютерде бірнеше қосымша жұмыс істеген кезде немесе МҚБЖ-ның бірнеше терезелерінде жұмыс істеген кезде тосқауылдар қолданылады. Тосқауылдар МҚ-ның әртүрлі нысандарына және нысандардың жекелеген элементтеріне әсер ете алады. МҚ нысандарын тосқауылдаудың нақты жағдайы бір мезгілде нысанды пайдалану және осы нысанды жасау режиміне енуге тырысуы болып табылады. Деректер қорының кестелеріне қатысты қосымша тосқауылдар жекелеген жазбалармен немесе өрістерімен жұмыс жүргізу кезінде пайда болуы мүмкін. Тосқауылдар айқын және айқын емес болады. Айқын түрдегі тосқауылдарды пайдаланушы немесе командалар көмегімен қосымша орнатады. айқын емес тосқауылдарды орын алуы мүмкін болатын келіспеушіліктерді болдырмас үшін жүйенің өзі құрады. Мысалы, ақпаратты редакциялау кезінде МҚ құрылымының өзгертуге тырысқан жағдайда Деректерді редакциялау аяқталғанша МҚ-ны қайта құрылымдауға тыйым салынады. Бақылау сұрақтары:

- МҚБЖ дегеніміз не?
- Қандай МҚБЖ иерархиялық деп аталады, барынша танымалдарын атап шығыңыздар?
- Қандай МҚБЖ реляциялық деп аталады, кең тараған түрлерін атап шығыңыздар?
- Деректер қорының сервері болатын, танымал МҚБЖ атап шығыңыздар?
- МҚБЖ-ның негізгі қызметтерін атаңыздар? Әдебиеттер:
- Бидайбеков Е.Ы., Елубаев К., Шекербекова Ш.Т. Деректер қоры және ақпараттық жүйелер. Алматы., 2010.
- Глушаков С.В., Ломотько Д.В. Базы данных. -Харковь: Фолио; М.:ООО «Издательство АСТ», 2002.
- Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных. Учебник для высших учебных заведений. Под редакцией проф. А. Д. Хомоненко. – Санкт-Петербург: «КОРОНА принт», 2000 г

10-ДӘРІС.

Тақырыбы: Деректер қорын программалық жабдықтау. Лекция жоспары:

- Жүйелік және пайдаланушылық құралдар.
- Деректер қорын басқару жүйесі (МҚБЖ).
- МҚБЖ тілдік құралдары.
- SQL тілі.
- Жоғарғы деңгейдегі программалау тілдерін пайдаланып Деректер қорын программалау. Құзыреттіліктер- Деректер қорында есептермен жұмыс істеуді меңгеру Лекция мазмұны Қазіргі уақыттағы МҚБЖ-лар Деректер қорымен жұмыс істеу бойынша ауқымды есептерді қосымшаны жасамай-ақ шешуге мүмкіндік береді. Дегенмен де, қосымшаны жасау дұрыс болатын жағдайлар да бар. Мысалы, егер Деректермен манипуляциялауды автоматтандыру қажет етілсе, МҚБЖ-ның терминалдық интерфейсі жеткілікті түрде дамымаған болса, немесе МҚБЖ-дағы ақпаратты өңдеуден өткізу жөніндегі стандарттық қызметтер пайдаланушының қажеттерін қанағаттандырмайтын болса. Деректер қорын қолданбалы программалық қамтамасыз ету дегеніміз – бұл нақты мәселелерді шешу үшін пайдаланушылардың өздері құрастыратын программалар мен жүйелер. Қазіргі уақыттағы программалаудың алгоритмдік тілдерінде – Delphi, Cu, Visual Basic және т.б. – SQL тілін пайдалана отырып Деректер қорына қол жеткізетін программалау құралдары бар. Сонымен қатар, көптеген МҚБЖ-ларда жүйе ішіне орнатылған программалау тілі бар. Dbase форматындағы Деректер қорын программалауға арналған арнайы Clipper алгоритмдік тілі бар. Қосымшалады жасау үшін МҚБЖ –ның программалық интерфейсі болуы тиіс, оның негізін сәйкес программалау тілінің функциялары және/немесе процедуралар құрайды. Қолданыстағы МҚБЖ –лар қосымшаларды жасаудың келесі технологияларына (және олардың комбинацияларын) қолдау жасайды:
- программаларды қолмен кодтау (Clipper, FoxPro, Paradox);
- генераторлардың көмегімен қосымшалардың мәтіндерін құрастыру (FoxPro-де FoxApp, Paradox-та Personal Programmer);
- визуалды программалау әдістерімен дайын қосымшаны автоматты түрде генерациялау (Delphi, Access, Paradox for Windows). Қолмен кодтау кезінде программалаушылар қосымшалар программаларын мәтінін қолмен тереді, сосын олардың жұмысын жүргізіп баптайды. Генераторларды пайдалану қосымшаларды жасауды жеңілдетеді, себебі бұл жағдайда программалық кодты қолмен термей-ақ алуға болады. Қосымшалардың генераторлары қосымшалардың

негізгі элементтерін (мәзір, экрандық форма, сұраныстар және т.б.) жасауды жеңілдетеді, алайда көбінесе қолмен кодтауды жоққа шығара алмайды. Қосымшаларды визуалды программалау құралдары қосымшалардың генераторларын қолдану идеясының дамуы боып табылады. Бұл ретте қосымша ыңғайлы біріктірілген ортаның көмегімен дайын түрдегі «құрылыс блоктарынан» құрылады. Қажет болған жағдайда жасаушы қосымшаға өзінің кодын оңай енгізеді. Біріктірілген орта, әдетте, қосымшаларды құрастыру, баптау мен өзгертудің күшті құралдарын ұсынады. Визуалды программалау құралдарын пайдалану қысқа мерзім ішінде алғашқы екі әдіспен алынған қосымшалардан да сенімді, тартымды және тиімді қосымшаларды құрастыруға мүмкіндік береді. Жасалған қосымша әдетте амалдар жүйесінің бір немесе бірнеше файлдарынан құралады. Егер де қосымшаның негізгі файлы орындалатын файлы (мысалы, exe-файл) болса, онда бұл қосымша МҚБЖ ортасынан дербес түрде орындалатын тәуелсіз қосымша болып табылады. Іс-жүзінде тәуелсіз қосымшаны алу, мәтінді қолмен теру сияқты, сондай-ақ қосымшаның генераторының немесе визуалды программалау ортасының көмегі сияқты тәсілдерімен алынған программаның бастапқы мәндерін компиляциялау жолымен жүзеге асырылады. Тәуелсіз қосымшаларды мысалы, FoxPro МҚБЖ-лары және Delphi визуальді программалау жүйесін алуға мүмкіндік береді. Айта кететін нәрсе, Delphi құралдарының көмегімен әдетте, тәуелсіз қосымшаларды жасамайды, себебі бұл өте көп еңбекті қажет ететін үдеріс, оның орнына МҚБЖ ядросының рөлін атқаратын BDE (Borland DataBase Engine) Деректер қорының процессорын пайдаланады. Дербес ЭЕМ-дарға арналған қосымшаларды жасаудың ең алғашқы құралының бірі «таза түрдегі компилятор» болып саналатын Clirreg жүйесі болып табылады. Көп жағдайларда қосымша МҚБЖ ортасы болмағанда орындалмайды. Қосымшаны орындау МҚБЖ қосымша файлдарының құрамдас бөлігін (жеке жағдайда бұл бастапқы программаның мәтіні) талдайды және орындалатын қажетті машиналық командаларды автоматты түрде құруына негізделген. Былайша айтқанда, қосымша интерпретациялау әдісі арқылы орындалады. Интерпретациялау режимі қазіргі заманғы көптеген МҚБЖ-ларда, мысалы, Access, Visual FoxPro және Paradox сияқты сондай-ақ, FoxBase және FoxPro тәрізді бұрынғы МҚБЖ-ларда жүзеге асырылған. Сонымен қатар, компиляция мен интерпретацияның аралық нұсқасын – былайша айтқанда (псевдокомпиляция) жалған компиляцияны пайдаланатын жүйелер болады. Мұндай жүйелерде бастапқы программа компиляция арқылы аралық кодқа (псевдокодқа) өзгереді және дискіде жазылады. Бұл түрде оны кейбір жүйелерде тіпті редакциялауға рұқсат етіледі, алайда жалған компиляцияның басты мақсаты – программаны оны интерпретациялау үдерісін жылдамдатуға мүмкіндік беретін түрге өзгерту. Мұндай тәсіл DOS басқаруымен жұмыс істейтін МҚБЖ-ларда, мысалы, Foxbase+ және Paradox 4.0/4.5 for DOS-та кеңінен қолданылған. Windows басқаруымен жұмыс істейтін МҚБЖ-ларда (псевдокод) жалған код көбінесе қосымшаны модификациялауға тыйым салу үшін қолданылады. Бұл жұмыс істеп жатқан программаны кездейсоқ немесе әдейі бұзудан сақтау үшін пайдалы болады. Мысалы, мұндай тәсіл Paradox for Windows МҚБЖ-да қолданылған, онда әзірленген экрандық формалар мен есеп берулерді редакциялауға сәйкес нысандарға өзгертуге жол беріледі. Кейбір МҚБЖ-лар пайдаланушыға қосымшаны жасау нұсқасын таңдап алу мүмкіндігін ұсынады: МҚБЖ-мен интерпретацияланатын программалық код ретінде немесе тәуелсіз программа ретінде. Тәуелсіз қосымшаларда қолданудың артықшылығы – машиналық программаны орындау уақыты, әдетте интерпретация кезіндегі уақытпен салыстырғанда қысқалау болады. Мұндай қосымшаларды әлсіз машиналарда және қосымшаны пайдаланушылардың тарапынан өзгертуге қарсы жабу қажет болатын, жүйелерді, «қолдануға даяр» етіп орнатқан жағдайда пайдаланған дұрыс. Интерпретацияланатын жүйелердің тағы бір үлкен артықшылығы – жақсы МҚБЖ-ларды әдетте Деректердің бүтіндігін бақылау мен оларға рұқсатсыз қол жеткізуден қорғауға арналған күшті құралдар болады, ал компиляциялаушы түрдегі жүйелер туралы олай деп айта алмаймыз. Сонғыларында аталған қызметтерді қолмен программалауға, немесе әкімшелердің құзырына қалдыруға болады. Қосымша жасау құралдарын таңдау кезінде келесі негізгі үш факторды: компьютердің ресурстарын, қосымшаның ерекшеліктерін (программаның қызметтерін модификациялаудың қажеттігі, жасауға жұмсалатын уақыт, қол жеткізуді бақылаудың қажеттілігі және ақпараттың тұтастығын қамтамасыз ету) және жасаудың мақсатын (шеттетілетін программалық өнім немесе өзінің күнделікті іс-әрекетін автоматтандыру жүйесі) ескерген дұрыс. Қазіргі заманғы компьютері бар және онша күрделі емес қосымшаны құруды жоспарлап отырған пайдаланушы үшін интерпретациялау типіндегі МҚБЖ көбірек сай болады. Еске сала кейік, мұндай жүйелер жеткілікті дәрежеде күшті болып келеді, олардың жоғары деңгейлі құралы бар, олар жасау мен баптауға ыңғайлы, жасау жұмысын жылдам түрде орындауға мүмкіндік береді және қосымшаға еріп жүру мен оны модификациялаудың ыңғайлы болуын қамтамасыз етеді. Сипаттамалары нашар компьютерді пайдаланған кезде тәуелсіз қосымшаларды жасау құралдары бар жүйені таңдаған дұрыс. Бұл ретте ескертетін нәрсе, қосымшадағы кішігірім өзгеріс программалау, программаны компиляциялау және баптау кезеңдерін циклдік түрде қайталануына әкеледі. Тәуелсіз қосымшаны орындау мен қосымшаны интерпретация режиміндегі орындаудан айырмашылығы тәуелсіз қосымшаның пайдасына миллисекунд айналасында тербелуінде болады. Сонымен қатар, қосымшаны оны пайдалануға дайындау уақытындағы айырмашылық әдетте интерпретациялау жүйелерінің пайдасына минуттар-сағаттар дәрежесіндегі шама құрайды.

SQL және Деректер қорының қауіпсіздігі. МҚБЖ-нің қызметтерінің біріне Деректер қорының қауіпсіздігін қамтамасыз ету жатады. Реляциялық МҚБЖ-лардың қауіпсіздік жүйесінің негізі SQL тілі болып табылады. Деректерді қорғаудың негізгі үш принципі қолданылады:

- Деректерді оқуға, енгізуге және жаңартуға арналған кез келген SQL-команда қандай да бір пайдаланушының атынан орындалады;
- қорғау объектілері негізінен кестелер мен бейнелеу болады;
- әрбір пайдаланушыға қандай да бір әрекеттерді жасауға арналған белгілі бір артықшылықтар (құқықтар) беріледі. Реляциялық Деректер қорының әрбір пайдаланушысына идентификатор – пайдаланушыны бірімәнді анықтайтын қысқаша атау меншіктеледі. Коммерциялық, өнеркәсіптік Деректер қорларында пайдаланушыға атауды Деректер қорының әкімшісі береді. Дербес ЭЕМ-дарда Деректер қорында Деректер қорын құрған пайдаланушының тек бір ғана идентификаторы болуы мүмкін. Көптеген МҚБЖ-лар үшін идентификатор ретінде операциялық жүйеде тіркелген пайдаланушылардың атаулары қолданылады. Әдетте, пайдаланушы жұмыс алдында идентификаторды және сонымен байланысты парольді енгізуі тиіс. Өртүрлі пайдаланушылардың бірдей идентификаторлары болуы мүмкін.

Артықшылықтар. SQL стандарттарында төмендегідей артықшылықтар анықталған:

- оқуға (SELECT);
- жазбаларды енгізуге (INSERT);
- жазбаларды жоюға (DELETE);
- жазбаларды өзгертуге (UPDATE).

CREATE TABLE және CREATE VIEW операторлары көмегімен кестені немесе бейнелеуді құрған пайдаланушы автоматты түрде осы объектілердің иесіне айналады және осыларға қатысты жоғарыда аталған барлық артықшылықтарға ие болады. Кестенің немесе көрсетімнің иесі басқа пайдаланушыларға осы Деректерге қол жеткізуге рұқсат ете алады. Ол үшін GRANT операторы қолданылады, оның синтаксистік диаграммасы төмендегідей (4.35–сурет)

```
GRANT SELECT INSERT DELETE UPDATE ( баған атауы ) , , ALL PRIVILEGES ON кесте атауы TO пайдаланушы атауы PUBLIC WITH GRANT OPTION
```

4.35-СУРЕТ. GRANT ОПЕРАТОРЫНЫҢ СИНТАКСИСТІК ДИАГРАММАСЫ

Командада PUBLIC кілттік сөзі артықшылықтарды әрбір пайдаланушыға жекелеп берудің орнына, бірден барлық пайдаланушыларға артықшылықтар беру үшін қолданылады. Егер GRANT операторында WITH GRANT OPTION сөйлемі көрсетілсе, онда артықшылықтарға ие болатын пайдаланушылар өз кезегінде оларды басқа пайдаланушыларға беруі мүмкін. Мысал Smith агентіне өз артықшылықтарын басқа пайдаланушыларға ұсыну мүмкіндігін беріп Orders кестесіне ақпаратты қосуға және жоюға рұқса ету GRANT INSERT, DELETE ON Orders TO Smith WITH GRANT OPTION

REVOKE SELECT

```
INSERT DELETE UPDATE ( баған атауы ) , , ALL PRIVILEGES ON кесте атауы FROM пайдаланушыатауы PUBLIC ,
```

өмендегідей (4.36–сурет)

4.36-СУРЕТ. REVOKE ОПЕРАТОРЫНЫҢ СИНТАКСИСТІК ДИАГРАММАСЫ

Мысал Smith, Pol агенттерінің Orders кестесіне сұраныстар мен өзгерістер енгізуге қатысты артықшылықтарын алып тастау.

REVOKE SELECT, UPDATE ON ORDERS

FROM Smith, Pol Бақылау сұрақтары:

- SQL тілі қандай Деректерді өңдеу құралы болып табылады?
- SQL қандай тіл болып табылады?
- 1992 жылы SQL тілінің қандай стандарты пайда болды?

- SQL тілінде белгісіз немесе анықталмаған Деректер қалай белгіленеді?
- SQL тілінің теориялық негізі не болып табылады? Әдебиеттер:
- Бидайбеков Е.Ы., Елубаев К., Шекербекова Ш.Т. Деректер қоры және ақпараттық жүйелер. Алматы., 2010.
- Глушаков С.В., Ломотько Д.В. Базы данных.-Харковь: Фолио; М.:ООО «Издательство АСТ», 2002.
- Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных. Учебник для высших учебных заведений. Под редакцией проф. А. Д. Хомоненко. – Санкт-Петербург: «КОРОН

11-ДӘРІС.

Тақырыбы: Деректер базасын жобалаудың физикалық моделі Лекция жоспары:

1.Ақпарат ұғымы.

- Деректердің негізгі құрылымы.
- Деректерге қолданылатын амалдар.
- Деректердің негізгі құрылымы Құзыреттіліктер- Деректер базаларын әзірлеу құрылымын, құрамы мен тәсілдерін, деректер базаларын басқару жүйелерінің бірін, сарапшыларды қыңқтеу әдістері мен критерийлерін, топтық сараптама әдістерін, сараптық жүйелерді құру негіздеріне зерттеулер жүргізе білу Лекция мазмұны Деректер базасында жобалау кезеңдері. Деректер базасында белгілі – бір пәндік облыс туралы ақпарат көрінеді. Пәндік облыс (ПО) деп нақты зертеу үшін қызығұшылық білдіретін реалды әлемнің бөлігі аталады. Автоматтандырылған ақпараттық жүйелердепәндік облыстың көрінісі Деректердің бірнеше деңгейлерінің модельдерімен ұсынылады. Модельдердің деңгейлерінің саны МББЖ – нің ерекшеліктеріне тәуелді болады. Әрі қарай біздер МББЖ – ге деректер базаларын жобалауда сұрақтарын қарастырамыз, құрылымдық деректер модельдерін қолдаймыз. Модельдер логикалық және физикалық деңгейде бөлек түрінде қолданатына қарамастан, методологиялық көзқарастан модельдердің деңгейлерін және деректер базаларының жобалау кезеңдерін бәрі – бір белгілеуге болады. Деректер базасының даталогиялық моделі (ДМ). Даталогиялық модель логикалық деңгейдің моделі болып табылады және ол деректер элементтерінің логикалық арасындағы байланысты бейнелеп, олардың сақтау сыртқы ортаға да мазмұнына тәуелді болмайды. Осы модель информациялық бірліктер терминдеріне құрылады, нақты МББЖ рұқсат етілген, біздер деректер базасын жобалайтын ортада.ДЛМ құру кезеңі даталогиялық жобалау болып аталады. МББЖ тілімен деректер базасының логикалық құрылымын суреттеуді сызбанұсқа д.а.ДБ физикалық моделі . Даталогиялық модельдің сақталатын орнына байланыстыру үшін физикалық деңгейдің деректер моделі қолданылады(көп жағдайда физикалық моделі д.а.). Осы модель есте сақтау құралғыларды анықтайды және деректерді сақтайтын жерде физикалық түрін ұйымдастырады. Физикалық деңгейдегі модель мүмкіндік нәтижелілігін есептеп құрылады, МББЖ суреттеуі бойынша. Деректер базасының физикалық құрылымын суреттеуді сақтау сызбанұсқасы д.а. Соған сәйкес ДБ жобалау деңгейі физикалық жобалау д.а. Бақылау сұрақтары:
- Деректер базасын жобалаудың физикалық моделі дегеніміз не?
- Физикалық параметрлерді анықтау дегеніміз не?
- Жад пен уақытты бағалау дегеніміз не?
- Деректер базаларын модельдеу дегеніміз не? Ұсынылатын әдебиеттер: 1.Базы данных. Модели и языки. Кузнецов.- М.2008. 2.ТарасоваР.Н.Конспект лекций по СУБД в электронномвиде.Шымкент,2007. 3.М.Т.Шоманбаева, МУ и контрольные задания по СУБД- FoxPro, Шымкент 2002.

12-ДӘРІС.

Тақырыбы: Деректер базасының тұжырымдамалық моделі Лекция жоспары:

- Деректерді жинау.
- Талдау және редакциялау Құзыреттіліктер- тұжырымдамалық моделі: деректерді жинау, талдау және редакциялау элементтер туралы қабілеттерін, оқуға деген саналы көзқарасты, білімді жалғастыру және кәсіптік саналы тандау дайындықты қалыптастыру Лекция мазмұны Деректер базасы түсінігі. Компьютерде мәтін, графикалық кескін жасауға, есептеу жүргізуге мүмкіндік беретін программалардан басқа көлемі үлкен ақпаратты сақтауға, өңдеуге және іздеуге арналған арнайы программалар да бар. Мұндай программалар деректер базасын басқару жүйесі (ДББЖ) деп аталады. Мынадай мысалды алайық, сендердің мектептеріңде 1000- ға жуық оқушы оқиды делік (мүмкін көбірек шығар). Әр мектепте медициналық кабинет болады, онда оқушының аты-жөні, туған жылы, ұлты, жынысы, бойы, салмағы, мекен-жайы, телефон нөмірі, дәрігерлік екпелері туралы тіркелген медициналық карталар (кітапшалар) бар. Барлық карта шкафта ретсіз араласып жатыр делік. Қандай да болсын бір тасушыда, мысалы,

қағазда, кинотаспада, магниттік дискіде сақталатын, арнайы ұйымдастырылған деректер жиынтығы деректер базасы деп түсіну керек. Бір базадағы деректер өзара қисынды байланысты болуы тиіс. Бұл деректерді белгілі параметрлері, қасиеттері, яғни бір-біріне қатысты белгілері бойынша топтастыру арқылы жүзеге асырылады. Деректер базасын жіктеу. Деректер базасы, негізінен, үш белгісі бойынша жіктеледі. Бірінші белгісі - сақтаулы ақпараттың сипатына қарай деректер базасы: фактографиялық және құжаттық болып бөлінеді. Фактографиялық деректер базасында сипатталатын объектілер туралы қысқаша Деректер нақты белгіленген пішімде беріледі, құжаттық деректер базасы әртүрлі типтегі мәтіндік графикалық дыбыстық мультимедиялық кең көлемді ақпараттардан тұрады. Екінші белгісі - ақпараттарды сақтау әдісі бойынша деректер базасы орталықтандырылған және үлестірілген болып бөлінеді. Орталықтандырылған деректер базасында бар ақпарат бір компьютерде сақталады. Үлестірілген деректер базасы жергілікті және ауқымды компьютерлік желілерде қолданылады және ақпараттың әр түрлі бөліктері бөлек компьютерлерде сақталуы мүмкін. Үшінші белгісі - ақпаратты ұйымдастыру құрылымы бойынша деректер реляциялық, иерархиялық және желілік болып бөлінеді. Біз сендермен фактографиялық және реляциялық деректер базасын ғана қарастырамыз. Реляциялық деректер базасының құрылымы Реляциялық ДБ деп оны құрайтын бөліктерінің өзара байланысымен құрастырылған деректер базасын атайтын. Қарапайым жағдайда ол бір тіктөртбұрышты кесте болып табылады. Күрделі реляциялық деректер базасы көптеген тіктөртбұрыш кестеден тұрады. Деректері өзара әртүрлі қатынастармен байланысқан бірнеше кесте құрайық. Кестелердегі деректердің арасында қатынастың өте кең тарағаны және әсіресе жиі қолданылатыны «объект - қасиет» типті болып табылады. Мысалы: «Сыныптағы оқушылар тізімі» деректер базасы кесте түрінде көрсетілген болады. Сыныптағы оқушылар тізімі: Кесте элементтері ұяшық, жол және баған болған бұрын қарастырылған кестелерден айырмашылығы - кестені деректер базасы немесе деректер базасының объектісі ретінде қарастырғанда жазба және өріс түсініктері енгізіледі. Реляциялық деректер базасындағы жолдар – жазбалар деп, ал бағандар – өрістер деп аталады. Кестенің әрбір өрісінің аты болады, мысалы, тегі (фамилиясы), аты, туған жылы; жынысы; ұлты; көзінің түсі; шашының түсі. Өрістер типтері Реляциялық деректер базасында өрістердің негізгі төрт типі пайдаланылады: сандық, символдық, даталық, логикалық. Сандық типі мәндері тек сандар: бүтін, бөлшек, ондық (масса, адам саны, арақашықтық және т.б) болатын өрістерден тұрады. Символдық типте символдық ақпараты бар өрістер болады, бірақ символдық ақпарат цифрлардан да тұруы мүмкін, мысалы, реттік нөмір, телефон нөмірлері және т.б сонда бұл өрістерге символдық тип меншіктеуге болады (сендердің кестелеріңдегі барлық өрістердің типі-символдық). Бақылау сұрақтары:

- Бірінші белгісі қандай?
- Екінші белгісі қандай?
- Үшінші белгісі қандай? Ұсынылатын әдебиеттер: 1.База данных п/р Хомоненко А.Д. СПб Корона – принт 2000, 2004. 2.Дунаев В., Базы данных. Язык SQL для студ., СПб –БХВ, 2006. 3.Омельченко Д., Самоучитель Visual FoxPro 9.0, СПб –БХВ, 2005. 4.Гусева О.Л. Практикум по Visual Basic, Москва, Финансы и Статистика, 2006.

13-ДӨРІС.

Тақырыбы: ДББЖ: деректер базаларын жобалау Лекция жоспары:

- Деректер базаларын жобалауға арналған деректер базаларын құруға реляциялық амал.
- Деректер базасын басқару жүйесі Құзыреттіліктер- Деректер базаларын және өнім сапасын сараптық жүйелерін құруда зерттеулер жүргізе білу Лекция мазмұны База ақпаратын көрсетушілік ретінде берілген критерийге сәйкес бейнелейтін кез келген МББЖ Деректермен төрт қарапайым операцияны орындауға
 - 1.Кестеге бір немесе бірнеше жазбаны қосу; █
 - 2.Кестеден бір немесе бірнеше жазбаны жою; █
 - 3.Кейбір өрістердің бір немесе бірнеше жазбаларындағы мәндерді жаңарту; █
 - 4.Берілген шартты қанағаттандыратын бір немесе бірнеше жазбаларды табу. █
 МББЖ-ның тағы бір функциясы-Деректерді басқару. █
 - 1.1 Деректер базасы туралы жалпы түсінік █
 ЭЕМ (электронды есептеуіш машиналардың) маңызды ерекшеліктерінің █
 бірі- ақпараттың мол көлемін сақтау және өңдеу,сонымен бірге █
 және графикалық құжаттар (суреттер,сызулар,фотосуреттер,географиялық карталар) ғана емес,жаһандық жүйе,дыбыстық ретінде қарастырылады. █
 Деректер базасы(МБ) дегеніміз – ол арнайы орталықтандырылған программалық арқаулы ауқым ретінде нақты зерттеуші үшін қызықты нақты █
 Қарапайым МБ мысалы ретінде телефон анықтамалығын,поездар қозғалысы кестесін,кәсіпорын электронды Деректер базасының басты қасиеті-ақпараттық тез іздестіру мен Деректер базасын басқару теориясы дербес пән ретінде қарастырылады. █

Объект деп - мағлұматтары Деректер базасында сақталатын ақпараттық жүйесінің

Атрибут-объектінің қасиеттерін ақпараттық бейнелеу.Әрбір объект атрибуттардың кейбір жиынтығымен

Деректердің басты элементі деп Деректердің басқа элементтерінің мәндерін

Алғашқы кілт- объектiнiң(жазбаның) әрбір данасын бірегей түрде ұқсастарын

Екінші ретті кілт-бірнеше жазбалар үшін мәні қайталанатын

(немесе атрибуттар тобы).Ең алдымен,екінші реттегі кілттер жазбаларды іздестіру

Деректер базасының ерекшелігі –ол базада Деректердің өзінің сипаттамаларымен

МетаДеректердің орталықтандырылған жиыны Деректер сөздігі. МетаДеректер құрамына Деректер

Деректер базасынан бір мезгілде параллельді түрде бірнеше қолданушылар

Құрылған немесе бар Деректер базасының файлына жаңа ақпараттарды

Деректер базасына жаңа бос файлды қосу;

Деректер базасы файлындығы ақпаратты өзгерту (модификациялау);

Деректер базасынан ақпаратты іздестіру;

Деректер базасыдағы файлдан ақпаратты жою;

Деректер базасында файлдарды өшіру.

Деректер базасын басқару жүйесінің архитектурасы

МБ теориясындағы ең негізгі

Сыртқы модельдер деңгейі- ең жоғарғы деңгей.

Концептуалдық деңгей – бұл орталық басқарушы

Физикалық деңгей- файлдарда орналасқан немесе сыртқы ақпарат

Деректер базасын жобалау технологиясы

Деректер базасын жобалау заттық образды талдаудан басталады және

Деректер базасын жобалау барысында келесі мақсаттар орындалуы қажет:

Деректер базасын жобалау барысында кедергілерге жолықпау үшін келесі

Деректер базасының мазмұны

Базаны ұйымдастырудың тиімді жолы

Құрылатын жүйені басқарудың құралдарын анықтау керек.

Жобалаушының негізгі мақсаты МБ-ң құрылымын кез келген МББЖ

қамтитын ақпараттық қолдану саласы жағдайына сәйкес келуі және

Деректер базасын жобалау процесі – бұл қолдану саласының

Деректер базасын жобалау

Қосымшаларды жобалау

Деректер базасын жүзеге асыру

Деректер базасын басқарудың

арнайы әдістерін қалыптастыру

Деректер базасын қолдану

Бұл схемада МБ-ң өмірлік циклының кезеңдері көрсетілген.

Қолдану саласын талдауды аяқтағанан кейін МБ-н жобалаудың негізгі

Айта кететін жайт, жобалау процесі өте күрделі процесс.

Модельді құрғанда болашақ Деректер базаларын нақты бір МББЖ

XX ғасырдың жетпісінші жылдарында семантикалық модель ретінде

Қазіргі кезде көптеген case құралдарда ER моделі

Инфологиялық модель қолдану саласының құрылымын және динамикасын, жүйені

Жүйелік талдау және қолдану саласының ақпараттық нысандарын қарапайым

Қолдану саласының инфологиялық моделін жобалау, яғни берілген бір

Деректер базаларының даталогиялық немесе логикалық жобалау, яғни Деректер

Деректер базаларын физикалық жобалау, яғни қосымшалардың тиімді қызмет Бақылау сұрақтары: 1.Деректер базасын жүзеге асыру дегеніміз не?

2.Деректер базасын басқарудың арнайы әдістерін қалыптастыру дегеніміз не?

3.Деректер базасын қолдану дегеніміз не? Ұсынылатын әдебиеттер: 1.Базы данных: основы, проектирование, использование. Учеб.для ВУЗов (2издание).Малыхина М.П.-ВНУ-СПб, 2008. 2.Ауелова 3. Деректер қоры жүйесі. Лекциялар жинағы. 2007.

Тақырыбы: Деректер базаларының иерархиялық моделі. Лекция жоспары:

- Деректер моделі.
- Негізгі концепциялары Құзыреттіліктер- Деректер базаларының иерархиялық моделін бақылай алу біліктілігін қалыптастыру Лекция мазмұны

Деректер моделі. Негізгі концепциялары. Ескертіліп кеткендей инфологиялық модель, өмірді адамға түсінікті деректерді сақтау ортасының параметрлеріне толығымен тәуелсіз концепциясымен бейнелейді. Осындай модель құрудың көптеген әдістері бар: графтік модель, семантикалық желілер, "мән-байланыс" моделі және т.б. Осылардың ішіндегі ең әйгілі болғаны, екінші бөлімде қарастырылатын "мән-байланыс" моделі. Инфологиялық модель ДҚБЖ "түсінікті" компьютерлі-бағдарланған даталогиялық модельге бейнеленуі керек. Деректер қорының теориялық өркендеу және практикалық қолдану процесінде, сол сияқты есептеу техникасының құралдарында әртүрлі даталогиялық модельді ұстанатын ДҚБЖ құрылды. Бастапқыда иерархиялық даталогиялық модельдер құрылды. Ұйымдастырудың қарапайымдылығы мынада, мәндер арасындағы алдын-ала берілген байланыстың бар болуы және деректердің физикалық модельдерінің ұқсастығы, шектелген жады көлемі бар ақырын жұмыс істейтін ЭЕМ –дағы иерархиялық ДҚБЖ жеткілікті өнім шығаруына мүмкіндік берді. Бірақ, егер деректердің ағаш тәрізді құрылымы болмаса, онда иерархиялық модель құрған кезде және қажетті өнімге жету үшін көптеген қиыншылықтар пайда болды. Желілік модельдер сол сияқты аз ресурсты ЭЕМ үшін құрылды. Бұл «жиыннан» тұратын өте күрделі құрылымдар – екі деңгейлі ағашқа аты ауыстырылған. "Жиындар" тізім құра отырып, "жазба-түйін" көмегімен бірігеді. Желілік модельдерді құру кезінде ДҚБЖ өнімділігін жоғарлатуға мүмкіндік беретін, көптеген "аз ғана қулықтар" ойлап табылды. Қолданбалы программалаушы терминдерді білуі керек, ДҚБЖ бірнеше ішкі тілдерін оқуы керек, әр түрлі экзemplар, жиын, жазба арасында навигация жүргізу үшін деректер қорын логикалық құрылымын көз алдына бөлшек түрінде елестетуі керек. UNIX операциялық жүйесінің бір құрушысы былай деп айтқан "Желілік қор – бұл деректерді жоғалтудың дұрыс тәсілі". Иерархиялық және желілік ДҚБЖ практикалық қолданудың күрделілігі деректерді көрсетудің басқа тәсілдерін іздеуді талап етті. 60 жылдардың аяғында деректерді манипуляция жасайтын ыңғайлы тілдері бар және ұйымдастыру жағынан қарапайымдылығымен ерекшелінетін, инвертирленген файл негізіндегі ДҚБЖ пайда болды. Бірақ мұндай ДҚБЖ деректерді сақтау үшін файлдар санына, олардың арасындағы байланыс санына, жазба ұзындығына және өріс санына шектеу қояды. Бүгінгі күні көп тарағаны реляциялық модельдер, ол жөнінде үшінші бөлімде қарастырылады. Деректердің физикалық ұйымдастырылуы ДҚ эксплуатациялық сипаттамасына жалпы әсер етеді. ДҚБЖ құрушылары деректердің физикалық моделін құруға тырысады, пайдаланушыға нақты бір ДҚ құралын ұсынады. Заман талабына сай өнімді ДҚБЖ физикалық моделдерін реттеудің әртүрлі тәсілдері осы бөлімде оны қарауға мүмкіндік бермейді. Локальды көріністі модельдеу, модель элементтерінің агрегациясы және жалпыламасы. Инфологиялық модельдеудің мақсаты - құрылатын деректер қорында сақтауға ұсынылатын, адамға керекті сол ақпаратты жинау және көрсету тәсілдерімен қамтамасыз ету. Сондықтан деректердің инфологиялық моделін нақты тіл аналогиясы бойынша тұрғызуға тырысады (соңғысы мәтінді компьютерлік өңдеу күрделілігінен таза түрінде қолданылуы мүмкін емес). Инфологиялық модельдің негізгі жоба элементтері ол мән, оның қасиеттері және арасындағы байланыс болып табылады (атрибуттары). Мән - деректер қорында сақталу жөніндегі ақпаратты ерекшелейтін кез келген (бір бірінен айырмашылығы бар объект). Мән дегеніміз ол адам, орынар, тік ұшақтар, дәм, түс және т.б. Мынадай түсініктерді айыра білу керек, мән типі және мән экзemplары сияқты. Мән типі түсінігіне тұтас болатын бір текті тұлға, зат, жағдай немесе идея жатады. Мән экзemplарына жиындағы нақты зат жатады. Мысалы, мән типі ҚАЛА болуы мүмкін, ал экзemplар – Москва, Киев және т.б. Атрибут - мәннің аты өзгертілген сипаттамасы. Оның аталуы мәннің нақты типі үшін уникальды, бірақ мәннің әртүрлі типтері үшін бірдей болуы керек (мысалы, ТҮС көптеген мән үшін анықталады: ИТ, АВТОМОБИЛЬ, ТҮТІН және т.б.). Атрибуттар мән жөнінде қандай ақпарат жинау керектігін анықтау үшін қолданылады. АВТОМОБИЛЬ мәні үшін мысалы атрибут ТИП, МАРКА, НӨМІР БЕЛГІСІ, ТҮС және т.б. болады. Мұнда тағы тип және экзemplар арасындағы ерекшелік бар: Қызыл, Көк, Банан түсті, Ақ түн және т.б., бірақ мәннің әр экзemplарына атрибуттың бір мәні меншіктеледі. Атрибут және мән типтері арасында ерекшелік жоқ. Атрибут тек мән типімен байланыста болады. Басқа контексте атрибут өз бетінше мән бола алады. Мысалы, автомобиль заводы үшін түс – бұл тек өнім шығарушы атрибуты, ал бояу фабрикасындағы түс – мән типі. Кілт - атрибуттың минимальды жиыны, олардың мәні бойынша керекті мән экзemplарын табуға болады. Минимальдық білдіреді, кез келген атрибут жиынынан шектеу, қалғандары бойынша мәнді идентификациялауға мүмкіндік береді. Мән үшін Кесте кілті болып Рейс_нөмірі атрибуты немесе жиын: Шығу_пункті, Ұшу_уақыты және Бару_пункті (шарт бойынша, бір пункттен екінші пунктке әр уақыт сайын бір тік ұшақ ұшады). Байланыс - екі немесе одан көп мәндерді ассоциациялау. Егер деректер қорын міндеті өзара бір бірімен байланыспаған тек сақтау болса, онда оның құрылымы қарапайым болады. Бірақ деректер қорын ұйымдастыруға қойылатын талаптардың негізгі біреуі – бұл бір мәнді басқаның мәнімен іздеу мүмкіндігімен қамтамасыз ету. Сол сияқты нақты деректер қорында жүздеген немесе мыңдаған мәндер болуы мүмкін, онда олардың арасында теория жүзінде миллион байланыс орнатылуы мүмкін. Мұндай көптеген байланыстың бар болуы инфологиялық модель күрделілігін анықтайды. Байланыс сипаттамасы және модельдеу тілі. Инфологиялық модель құру кезінде ER-диаграмм (ағылшынша Entity-Relationship, т.с. мән-байланыс) тілін қолдануға болады. Мұнда мәндер белгіленген

тікбұрышпен, ассоциациялар - белгіленген ромб немесе алты бұрышпен, атрибуттар - белгіленген овалмен, ал олардың арасындағы байланыс - байланыс деңгейі қойылатын бағытталмаған қабырғамен кескінделеді («көп» сөзін ауыстыратын, 1 немесе әріп қойылады") және қажетті түсініктеме келтіріледі. Екі мән арасында, мысалы А және В арасында төрт түрлі байланыс болуы мүмкін. Бірінші типі – БІРГЕ-БІР (1:1) байланысы: әр уақыт сайын әр А мән экземплярна В мән экземплярның 1 немесе 0 тиісті болады. Студент стипендия алады, соның ішінде жай немесе жоғары стипендия алуы мүмкін. Иерархиялық байланыс көмегімен жасалса, онда ол деректердің ден жазбаларының құрылымы тармақты құрылыммен жасалған. Иерархиялық ЭЕМ және де негізгі операцияларды орындауының уақыт көрсеткіші жаман емес. Иерархиялық модель иерархиялы кемшілігі болып көлемі жағынан үлкендігі, логикалық байланыстары бойынша күрделілігі, сонымен қатар жай қолданушыға түсіну қиындығы. Иерархиялық ішінде IMS, PC/Focus, Team-Up, Data Edge және де Ока, ИНЭК, МИРИС желілік байланыстар анықталуы мүмкін. Пәндік саланың жалпы түрдегі деректерін модельді – деректердің желілік моделі деп модельге қарағанда еркін байланыс орнату мүмкіндіктері жатады. Кемшілігі болып – жүйелер кең таралмаған. Олардың ішіндегі ең атақтыларына: IDMS,db_Vistalll, Сеть, Сетор және Компас ДББЖ жатады. Реляциялық модельдерде Деректердің деректер базаларының құрылу принциптері. Егер деректер базасын басқару реляциялық алгебраны қолданумен жүзеге асырылса, онда бұл деректер деп анықтауға, басқаруға және оларға әр түрлі операциялар арнайы тілді қолданылады(SQL). Реляция (Relation) – қатынас деген кестелерде сақталады. Доктор И.Ф. Кодд реляциялық модельдің авторы. Ол белгілі тізім жасаған және реляциялық модель осы. Бұл жие деп атайды. Осы тізімнің реляциялық жүйелерге арналған кейбір ережелерін айтып кетуге болады:

- олар түрінде сақтау керек;
- деректердің логикалық құрлымын ұстану керек;
- құрлымды сұраныстарды орындау және деректер базасындағы ақпаратты өзгерту үшін операцияларды (таңдау, жобалау, біріктіру) және де теоретикалы-жиынды операцияларды (біріктіру, қиылыстыру, қосу) сияқты операторларды қолдау керек;
- кестелерде белгісіз айнымалыларды айырып тану (nulls), нөлдік мәндер және деректердегі бос орындар;
- механизмдерді қамтамасыздандыру. Реляциялық деректер базасындағы барлық ақпара жол мен бағандардан тұрады. Барлық Деректер кесте түрінде жолы жоқ. кестелердің жиынтығы деректер базасын құрайды. Реляциялық базаларда кестелер бөлінген, бірақ құқықтары тең. Деректердің жол мен табу үшін кестенің атауын, бағанын және алғашқы кілттің мәнін немесе идентификаторын білу – бұл және .объектілер. Кестеде атауы болады – идентификатор. Кесте бағандары объект қасиеттеріне сай келіп өрістер деп. Әр атауы және типімен анықталады. Өріс атауы – идентификатор болып табылады және әр түрлі программаларда үшін . Ол кез сияқты құрылады, яғни латын әріптерімен жазылады және бір сөзден тұрады. Өріс типі бұл - Олар жол, сан, болуы мүмкін. Кестенің әр бір жолы – жазба деп .объектті сипаттайтын барлық өрістердің нәтижелерін көрсетуге болады. Кесте құрғанда маңызды болып табылады. Бұндай жағдай болмау үшін кілттік өріс өріс бір көрсетеді. Бір немесе бірнеше өрістер кілттік бола алады. Кестедегі жазбалар ретсіз немесе енгізілген реті реті бойынша орналастыра алады. Мысалы: алфавит бойынша жылы үшін қолданылады, кестені қолданушыға ыңғайлы реті бойынша көрсетуін қамтамасыздандырады. Индекстер екі түрлі болып бірінші ретті екінші ретті болып бөлінеді. Бірінші ретті индекс деп деректер базасы құрылғандағы өрістерді айтамыз. Екінші ретті және құрылған кезіндегі индекстерді айтамыз. Екінші ретті индекстерге , яғни ыңғайлы болады. Егер индекс бірнеше өріс ал бірінші өрісте бірдей нәтижелері бар үшін – өріс қызметкерлер деректер базасын отделдер бойынша, ал әр бір – алфавит бойынша реттеуге болады. Деректер базалары көптеген кестелерден тұрады. Мысалы бір мекеменің деректер әр бір мүмкін. Бөлек, бөлек кестелер болған жақсы бірақ кестелер жиынтығынан көбірек қарастыру керек. Байланысқан кестелерде біреуі басты кесте болып, ал екіншісі немесе басқалары – бағынышты және кілт арқылы байланысады. Кілт ретінде екі кестенінде құрамына кіретін өрістер бола алады. Бақылау сұрақтары:

- Бірінші ретті индекс деп?
- Екінші ретті индекс деп?
- механизмдерді қамтамасыздандыру дегеніміз не? Ұсынылатын әдебиеттер:
- Наумов А.Н. Системы управления базами данных и знаний / А.Н. Наумов – М.: Финансы, 2004-231 с. 2. Бешелов С.Д. Математико-статистические методы экспертных оценок / С.Д. Бешелов, Ф.Г. Гурвич .

15-ДӨРІС.

Тақырыбы: Деректер базалары типтері Лекция жоспары:

- Ақпараттың көлемі бойынша және байланыстарды көрсету тәсілдері бойынша жіктелуі. деректер базаларының
- Реляциялық моделі. Реляциялық деректер базаларының құрамы және құрылымы.
- Деректер базасының желілік моделі. деректер базаларын әкімшіліктендіру. Лекция мазмұны Реляциялық модельде файлдар арасында байланыс нақты түрде қарастырылмайды. Олар орнатылады динамикалық түрде деректердің

өңдеу кезіндегі сәйкес жолдардың тендік мағыналарда. Желілік және иерархиялық модельдерде жазу құрылымы кез – келген болуы мүмкін. Реляциялық модельдерде жазу құрылымы сызықтық болуы қажет. Аңықталуы бойынша әрбір қатынастың өзіндік кілті болады атрибут (қарапайым кілт) немесе атрибуттардың жиынтығын (құрама кілт) сәйкестіретін кортеж. Кейбір жағдайларда қатынастарда бірнеше мүмкін болатын кілттер болады. Өкінішке орай реляциялық МББЖ – дің ішінен барлығы кілттік концепциясын қолданбайды, себебі осы жағдайда көптеген проблемалар (көбінесе, кілттің уникалды болуын тексеру және кейбір шектеулерін қадағалауды ескеру) қолдануға жүктеледі. Тәуілдігіне қарамастан МББЖ жазуда кілттердің сипатталуын талаптардың бар немесе жоқ болуына қарамастан, деректер базасының жобалаушы түсіну керек, қандай кілт әрбір қатынастың болады. Бірнеше мүмкін болатын кілттердің ішінен біреуі таңдалады да бірінші кілт болып сипатталады. Атрибут немесе атрибуттар группасы қарастыратын қатынастарда кілт болатыны сипаттайтын, ал басқа қатынастарда керісінше кілт саналатын сыртқы кілт болып аталады. Егерде кез – келген таблица өзінің құрамында сыртқы кілтті ұстайтын болса, онда ол а) құрамына бірінші кілтті ұстайтын таблицамен логикалық түрде байланысқан. б) сол қатынастың өзіндік белгісі "біреудің – көбілерге" (құрамында сыртқы кілтті ұстайтын таблица осы қатынастың "көбілер" жағында орналасады). Өзінше түсініктері бойынша "ата - ана" – "бала" иерархиялық модельде файл "иеленуші" – файл "мүше" желілік модельдерде және байланыс "кілт" – "сыртқы кілт" реляциялық модельдерде біртүрлі қатынасты береді – жазудың ішінде сәйкес файлдарда 1 : М қатынастың болуы. Реляциялық МББЖ – де көбінде "көзқарас" (view) мағынасы қолданылады. Ол виртуалдық таблицаны сипаттайды. Бірнеше жолдармен бағаналардың жалпы мағыналар логикалық қосылу нәтижесінде және мүмкін жолдардың біріккендігін көптірлігін кірістіреді. Берілген немесе таңдалған шартпен болады. Осы мағына деректер банкінің дәстүрлі "сызбанұсқа" мағынасы кеңейтеді. Айыру және жалпы модельдердің әртүрлі класстардың түсінігі деректер базасының құрылымын жобалауда жалпы қатынасты қолдануға мүмкіндік береді, модельдердің бір түрден екінші түрге өзгертілуі мүмкіндігінің, құралдарын қолдануға көбінесе тілдік бір модельдің классымен басқаларымен жұмыс істеуге арналған. Деректер базасында жобалау кезеңдері. Деректер базасында белгілі – бір пәндік облыс туралы ақпарат көрінеді. Пәндік облыс (ПО) деп нақты зертеу үшін қызығушылық білдіретін реалды әлемнің бөлігі аталады. Автоматтандырылған ақпараттық жүелерде пәндік облыстың көрінісі Деректердің бірнеше деңгейлерінің модельдерімен ұсынылады. Модельдердің деңгейлерінің саны МББЖ – нің ерекшеліктеріне тәуелді болады. Әрі қарай біздер МББЖ – ге деректер базаларын жобалауда сұрақтарын қарастырамыз, құрылымдық деректер модельдерін қолдаймыз. Модельдер логикалық және физикалық деңгейде бөлек түрінде қолданатына қарамастан, методологиялық көзқарастан модельдердің деңгейлерін және деректер базаларының жобалау кезеңдерін бәрі – бір белгілеуге болады. Деректер қорын семантикалық жобалау Программалық өнім жасаудың, сонымен қатар деректер қорын жасаудың бірнеше кезеңдерге бөлуге болады. Алғашқыда пәндік аймақ зерттелінеді және осы зерттеу негізінде инфологиялық модель құрылады. Жалпы жағдайда инфологиялық модель пәндік аймақта болып жатқан процестерді және осы процестерге қолданатын деректерді сипаттайды. Инфологиялық модельде болашақтағы деректер базасының сипаттамасы, математикалық формулалар, кестелер, графикалар және т.с.с сипатталыныды. Инфологиялық модельді құру кезінде негізгі түсініктерді бөле білу қажет. Мысалы пәндік аймақ ретінде Деканаты алатын болсақ онда түсініктер ретінде "Баға", "емтихан", " сынақ" және т.с.с болады. Пәндік аймақтың инфологиялық моделі жазу арқылы сипаттауға болады. Бірақ программаны үшін көріністі және ұтымды ретінде көрсету үшін арнайы графикалық нотациялардың көмегімен жаған дұрыс. Бүгінгі таңда пәндік аймақты сипаттаудың әдістемелері көп. Оларға мысалы, IDEFO әдістемесіне негізделген SADT құрылымды талдау әдістемесі, Гена-Сарсона деректер ағымының диаграммасы, объектіге бағытталған талдау UML және т.с.с. Реляциялық деректер қорына үшін инфологиялық модельді құрудың құралы қазіргі уақытта ER – диаграммасы болып табылады. ER- диаграммасы бойынша деректер қорының моделін жасау бұл семантикалық модель болып табылады. Мұнда деректер құрылымы деректердің мазмұнына қарай құрылады. Деректер қоры құрудың екінші кезеңі даталогиялық модель болып табылады. Бұл модельде пәндік аймақтағы негізгі түсініктердің қасиеттері, олардың арасындағы байланыстар және деректерге қойылған шектеулер жазылады. Даталогиялық модель болашақ деректер қорының бастапқы прототипі болып табылады. Деректер қоры құрудың үшінші кезеңі физикалық модель болып табылады. Физикалық модель арнайы ДҚБЖ программасында деректерді сипаттайды. Даталогиялық модельдерде құрылған қатынастар кестелерге, атрибуттар бағандар, ал кортеждер жазбаларға айналады. Деректер қорын жобалау Деректер қорын құрылымын жобалау кезде тапсырыс беруші бағдарламашыға қолданатын формалар мен бланктерді береді. Егер қағаз бетіндегі деректерді кестелерге түсірсек онда кейбір проблемалар туы мүмкін. Сондықтан деректер қорын жобаламас бұрын жобалаудың мақсатын бөліп алғанымыз жөн. Бұл мақсаттарға келесі жатады:

- Деректер қорын сақтаудың қажеттілігі;
- Артық деректерден бас тарту;
- Кестелер санын минимумға дейін алып бару. Деректер қорын жобалаудың негізгі мақсаты артық деректерден бас тарту болып табылады. Бұл жадыны экономды пайдалануға және қарама-қайшы Деректерді жинамауға алып келеді Нормальдау және оның қажеттілігі Әмбебап кестені пайдаланғанда келесі проблемалар шығады: Деректердің артық

болуы проблемасы. Барлық бағандардағы деректер бірнеше рет қайталанады. Деректерді жаңарту проблемасы . Деректердің артық болуынан тапсырыс берушінің мекен жайын бір қатарда өзгертуімізге болады, ал қалған қатарларда өзгеріссіз қала береді . Сондықтан екі адрес пайда болады. Ал бәрін өзгерту үшін бүкіл кестені қарап шығуымызға болады. Деректерді қосу проблемасы. Жаңа дерек енгізу деректер жазылып тұрсада барлық бағандарға деректерді енгізуді талап етеді. Бұл деректердің артықшылығына алып келеді. Деректерді жою проблемасы. Қатарды жойған кезде қандай да бір керекті информация жойылуы мүмкін.

Бақылау сұрақтары:

- Деректердің артық болуы проблемасы қандай?
- Деректерді жаңарту проблемасы қандай?
- Деректерді қосу проблемасы қандай?
- Деректерді жою проблемасы қандай?
- Деректер базалары типтерінің ақпараттың көлемі бойынша және байланыстарды көрсету тәсілдері бойынша жіктелуі қандай?
- Деректер базаларының реляциялық моделі қандай? Ұсынылатын әдебиеттер:
- Работа в Microsoft Access примерах / С.А. Каратыгин, А.Д.тихонов, Л.Н. Тихонова –М: БИНОМ, 2010-512 с:ил
- Джексон Г. Проектирование реляционных баз данных для использования с микро ЭВМ / Г. Джексон: пер. с англ.-М.: Мир, 2010.252с.