



**Коммерциялық емес
акционерлік қоғамы**

**АЛМАТЫ
ЭНЕРГЕТИКА
ЖӘНЕ БАЙЛАНЫС
УНИВЕРСИТЕТІ**

Инженерлік кибернетика
кафедрасы

МӘЛІМЕТТЕР ҚОРЛАРЫН БАСҚАРУ ЖҮЙЕЛЕРІ

5В070200 - Автоматтандыру және басқару мамандығы студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Алматы 2016

ҚҰРАСТЫРУШЫЛАР: Утепбергенов И.Т., Саунова К.Т., Сагындыкова Ш.Н. Мәліметтер қорларын басқару жүйелері: 5B070200 - Автоматтандыру және басқару мамандығы студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар.– Алматы: АЭЖБУ, 2016. – 61 б

Әдістемелік нұсқауларда MS SQL Server ортасында мәліметтер қорларын өңдеудің сұрақтары қарастырылған: мәліметтер қорының объектілерін жасау, мәліметтермен әрекет жасау және мәліметтерді табу үшін әр түрлі күрделі сұраныстарды құрастыру.

SQL кестелермен жұмыс жасауға негізделген сондықтан оның күрделі қолданбалы бағдарламаларды жасауға құралдары жоқ. Сондықтан онымен бірге жоғары деңгейлі бағдарламалық тілдер қолданылады.

5B070200 - Автоматтандыру және басқару мамандығы студенттері үшін зертханалық жұмысты орындауға арналған әдістемелік нұсқауы.

Пікір беруші: т.ғ.к.,аға оқытушысы Г.Д. Мусапирова

«Алматы энергетика және байланыс университеті» Коммерциялық емес акционерлік қоғамының жоспары бойынша 2015 жылы басылып шығарылады.

© «Алматы энергетика және байланыс университеті» КАЕК, 2016 ж.

Введение

В процессе выполнения лабораторных работ по дисциплине «Системы управления базами данных» студенты должны выработать навыки физического проектирования баз данных, а также навыки разработки клиентских приложений для работы с базами данных, расположенных на сервере.

Изучение этих вопросов предусмотрено действующим государственным образовательным стандартом и в данном учебно-методическом пособии базируется на решении задач, актуальных для студентов специальности «Автоматизация и управления» (АУ).

В серии лабораторных работ используются Microsoft SQL Server, Microsoft Visual C Express Edition. Важной составной частью работ является освоение SQL стандарта. Для того чтобы приступить к выполнению этих работ, студенту уже нужно иметь минимальные навыки обращения с используемыми в них программными средствами. Эти необходимые навыки даются студентам на установочных занятиях, в начале семестра, вместе с выдачей заданий и электронных материалов.

Результаты выполнения работ рекомендуется сохранять в личных папках, так как лабораторные работы взаимосвязаны.

Кіріспе

«Мәліметтер қорын басқару жүйесі» пәні бойынша зертханалық жұмыстарды орындау үрдісінде студенттер мәліметтер қорын физикалық жобалау дағдыларын, сонымен қатар серверде орналасқан мәліметтер қорымен жұмыс істеуге арналған клиенттік қосымшаларды құру дағдыларын қалыптастырулары керек.

Осы сұрақтарды меңгеру қолданыстағы мемлекеттік білім беру стандарттарымен қарастырылған және берілген оқу-әдістемелік құралында

«Автоматтандыру және басқару» (АБ) мамандығының студенттері үшін өзекті тапсырмаларды шешуге негізделген.

Зертханалық жұмыстар топтамасында Microsoft SQL Server, Microsoft Visual C Express Edition қолданылады. Жұмыстың маңызды құрамдас бөлігі SQL стандартын меңгеру болып табылады. Осы жұмыстарды орындауға кірісу үшін студентте онда қолданылатын бағдарламалармен жұмыс істеудің кішкене дағдысы болуы керек. Бұл қажетті дағдылар студентке орнату сабақтарында беріледі, семестр басында, тапсырмаларды тарату мен электронды материалдарды беру кезінде.

Жұмыстардың орындалу нәтижесін жеке папкада сақтауға кеңес беріледі, өйткені зертханалық жұмыстар өзара байланысқан.

Зертханалық жұмыс №1

Тақырыбы: Microsoft SQL Server Management Studio МҚБЖ-нің интерфейсі. MS SQL Server-де мәліметтер қорын құру. MS SQL МҚБЖ құрамына кіретін көмекші бағдарламалар.

Жұмыстың мақсаты: MS SQL МҚБЖ құрамына кіретін көмекші бағдарламалармен танысу, олармен жұмыс істеп үйрену, Transact SQL тілі операторларының көмегімен мәліметтер қорын және осы мәліметтер қорына жататын байланысқан кестелер жиынын құруды үйрену.

Жұмыстың барысы:

1) Microsoft SQL Server құрамына кіретін көмекші бағдарламалар жинағымен танысу.

2) SQL Server Management Studio көмекші бағдарламалардың жұмысымен танысу.

3) Келтірілген операторлардың көмегімен «Кітаптар» мәліметтер қорының үлгісін құру.

4) Байланысқан кестелердің жиынтығы бар жеке мәліметтер қорын үлгі бойынша құру.

Жұмыстың орындалуына түсініктеме.

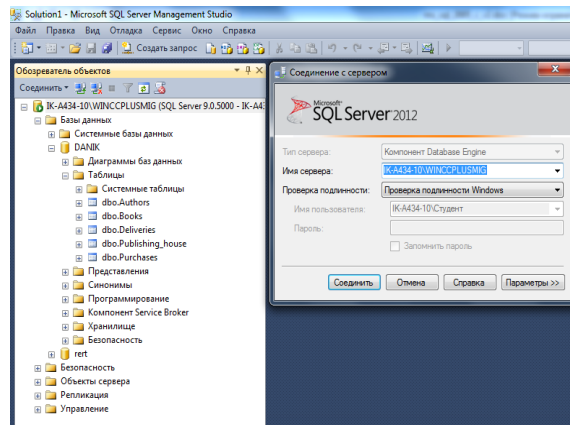
MS SQL Server 2012-ны іске қосу үшін *SQL Server Management Studio* белгісін таңдаңыз және оны іске қосыңыз. Қосылу кезінде «Сервермен байланыс» терезесінде келесі ақпаратты көрсету керек:

1) *Сервер типі*. Мұнда міндетті түрде қандай қызметке қосылу қажеттілігін таңдау абзал. «Database Engine компоненті» нұсқасын қалдырыңыз.

2) *Сервер аты*. Қосылу қай серверге жүзеге асырылатынын нұсқауға мүмкіндік береді. Өзгерту енгізусіз SQL Server аты компьютердің атымен сәйкес келеді. Өзіңіздің жеке компьютеріңізді таңдаңыз.

3) *Жалғандыққа тексеру*. Аутентификация мүмкіншілігі бойынша «Windows жалғандығын тексеруді» немесе «SQL Server жалғандығын тексеруді» таңдауға болады. Бірінші әдіс ағымдағы қолданушы Windows-қа кіруді осы арқылы жүзеге асырған тіркелу жазбасын қолданады. SQL Server нұсқасы өзінің жеке қауіпсіздік жүйесін қолданады. Windows жалғандығын тексеру нұсқасын қалдырыңыз.

«SQL Server Management Studio» құру ортасын іске қосамыз. Ол үшін «Пуск» мәзірінде «Все программы\ Microsoft SQL Server 2012\ SQL Server Management Studio» пунктін таңдаймыз (1 сурет).



1 сурет - MS SQL Server-ді іске қосу

Management Studio терезесінің құрылымы келесідей:

1) Терезе мәзірі серверді басқаруға және әртүрлі операциялар орындауға арналған бұйрықтардың толық жиынын құрайды.

2) Саймандар тақтасы – жиі туындайтын операцияларды орындауға арналған батырмалардан құралған.

3) «Нысандардың шолушысы» тақтасы. Бұл құрылымы сервердің барлық нысандарын көрсететін ағаш текті тақта, сонымен қатар сервермен және оның мәліметтер қорымен де әртүрлі операцияларды жүргізуге мүмкіндік береді. Нысандардың шолушысы мәліметтер қорын құрудың негізгі нысаны болып табылады.

4) Жұмыс аймағы. Жұмыс аймағында мәліметтер қорына қатысты барлық іс-әрекеттер жүргізіледі.

5) Жеке жұмыстық мәліметтер қорыңызды құрудан бұрын SQL Server-ді орнату кезінде автоматты түрде құрылатын оның қызметтік мәліметтер қорларын қарастырайық. Егер нысандардың шолушысында «Мәліметтер қорлары – Жүйелік мәліметтер қорлары» тармағын ашсақ, онда қызметтік мәліметтер қорларының келесі жиынтығын көреміз:

б) master. Бүкіл сервердің басты қызметтік мәліметтер қоры. Мұнда сервердің ортақ қызметтік ақпараты сақталады: жұмысының баптаулары, әрбір мәліметтер қорының және оның файлдарының баптауы туралы ақпараты бар сервердегі мәліметтер қорының тізімі, қолданушылардың тіркелу жазбасы, серверлік рөлдер туралы ақпарат және т.б.

7) msdb. Бұл мәліметтер негізінен қоры SQL Server Agent қызметінің ақпаратын сақтауға арналған (пакеттік тапсырмаларды, ескертулерді және т.б.), бірақ оған басқа да қызметтік ақпарат жазылады (мысалы, резервті көшірудің тарихы).

8) model. Бұл мәліметтер қоры SQL Server-де жаңа мәліметтер қорын құруға арналған үлгі болып табылады. Егер онда өзгерту енгізсе, мысалы, кестелер жиынтығын құрса, онда бұл кестелер барлық құрылатын мәліметтер қорына қатысады.

9) tempdb. Бұл қолданушымен және SQL Server-дің өзімен құрылатын уақытша кестелерге және сақталатын рәсімдерге арналған мәліметтер қоры. Бұл мәліметтер қоры SQL Server-дің әрбір қосылуында құрылады.

Microsoft SQL Server-де мәліметтер қоры екі бөліктен тұрады: - мәліметтер файлы – mdf кеңейтілуі бар және барлық кестелер мен сұраныстар орналасқан файл; - мәмілелер журналының файлы – ldf кеңейтілуі бар, МҚ-н барлық әрекеттерін тіркеп бекітетін журналы бар файл. Берілген файл МҚ істен шыққан жағдайда қайта қалпына келтіруге арналған. Оларды құрудың екі әдісі бар:

- сыбалық түрі SQL Server Management Studio көмегімен;
- Transact-SQL коды арқылы.

SQL Server Management Studio-да мәліметтер қорын құру.

Берілген көмекші бағдарламаларды қолдану мәліметтер қорын құрудың ең қарапайым әдісі болып табылады. «Кітаптар» Books мәліметтер қорын құрайық, кейін оны кестелермен толтырамыз.

Кестелер барлық мәліметтерді сақтау үшін қолданылатын мәліметтер қорының нысаны болып табылады. Бір кестеде тек нақты бір типтегі мәндер (мысалы, клиенттер, тауарлар, тапсырыстар және т.б.) туралы ақпараттар сақталуы керек. Бұл мәліметтер қорын ұйымдастырудағы ең басты ереже болып табылады.

Кестелердегі мәліметтер аймағы және жазбасы бойынша ұйымдастырылған. Аймақ (немесе кестенің бағаны) белгілі типтегі ақпараттан тұрады, мысалы, тегі, мекен-жайы, телефон номері. Жазба (немесе кестенің жолдары) – мәннің жеке түпнұсқалары туралы ақпараттан тұратын байланысқан аймақтардың тобы.

Кестенің кез келген аймағы кем дегенде үш міндетті ерекшеліктермен сипатталады:

- бағанның атауы. Кестедегі нақты аймаққа сұраныс жасауды жүзеге асырады. Аймақтарға мағыналы атауларды берген жөн;
- мәліметтер типі. Берілген аймақта қандай типтегі ақпарат сақталатынын анықтайды;
- Null мәніне рұқсат беру. Берілген аймақ үшін нақты мәліметтердің болмауы рұқсат па екенін анықтайды. Ол үшін null бос мәннің маркері қолданылады.

Мәліметтердің типтері.

Баған үшін мәліметтердің типін таңдау кезінде осы бағанға кез келген мүмкін мәндерді сақтауға және бұл кезде дискте ең аз орын алуға мүмкіндік беретін типке мән берген жөн. MS SQL Server-дегі мәліметтер типін сегіз санатқа бөлуге болады:

Бүтін санды мәліметтер:

- bit (1 байт). Тек 0,1 немесе null (мәліметтердің жоқ болуын хабарлайтын бос орын) мәндерін сақтай алады. Оны күй индикаторы ретінде қолданған ыңғайлы – қосулы/өшірулі, ия/жоқ, шын/жалған;

- tinyint (1 байт). 0 ден 255 дейінгі бүтін мәндер;
- smallint (2 байт). Мәндерінің аралығы -215 (-32768) тен 215 (3767) дейін;

- int (4 байт). -231 (-2147483648) ден 231 (2147483647) дейінгі бүтін сандардан тұруы мүмкін;

- bigint (8 байт). -263 (3223372036854775808) тен 263 (9223372036854775807) дейінгі мәндерден тұрады. int мәліметтер типіне сыймайтын өте үлкен сандарды сақтауға ыңғайлы.

Мәтіндік мәліметтер:

- char. 8000 белгіге дейін бекітілген ұзындықтағы сандық Unicode емес мәліметтерден тұрады;

- varchar. 8000 белгіге дейін айнымалы ұзындықтағы сандық Unicode емес мәліметтерден тұрады;

- nchar. 4000 белгіге дейін бекітілген ұзындықтағы Unicode мәліметтерден тұрады; Барлық Unicode мәліметтер типіне ұқсас оны әртүрлі тілдегі клиенттермен оқылатын мәтіннің шағын бөліктерін сақтау үшін қолдануға ыңғайлы;

- nvarchar. 4000 белгіге дейін айнымалы ұзындықтағы Unicode мәліметтерден тұрады;

Ондық мәліметтер:

- decimal. -1038-1 ден 1038-1 дейінгі бекітілген дәлдіктегі сандардан тұрады. Ол екі параметрді қолданады: дәлдік және дәреже. Дәлдік деп аймақта сақталатын белгілердің барлық санын айтады, ал дәреже – ондық үтірдің оң жағындағы белгілердің саны;

- numeric. Бұл decimal мәліметтер типінің синонимы – олар ұқсас.

Мәліметтердің ақшалық типі:

- money (8 байт). Ақша бірлігінен ондық дәлдікпен -263 тен 263 дейінгі ақша мәндерінен тұрады. 214768,3647-ден асатын ақша көлемін сақтауға ыңғайлы;

- smallmoney (4 байт). Ондық дәлдікпен -214748,3648 ден 214748,3647 дейінгі мәндерден тұрады.

Ауыспалы нүктесі бар мәліметтер:

- float. -1,79E+38 ден 1,79E+38 дейінгі ауыспалы нүктесі бар сандардан тұрады;

- real. -3,40E+38 ден 3,40E+38 дейінгі ауыспалы нүктесі бар сандардан тұрады.

Күн және уақыт мәліметтер типі:

- datetime (8 байт). 3,33 мс дәлдікпен 1753 жылдың 1 қаңтарынан 9999 жылдың 31 желтоқсанына дейінгі аралықтағы күн мен уақыттан тұрады;

- smalldatetime (4 байт). 1 мин-ке дейінгі дәлдікпен 1900 жылдың 1 қаңтарынан 2049 жылдың 6 маусымына дейінгі аралықтағы күн мен уақыттан тұрады;

Мәліметтердің екілік типі:

- binary. 8000 байтқа дейін бекітілген ұзындықтағы екілік мәліметтерден тұрады;

- varbinary. 8000 байтқа дейін айнымалы ұзындықтағы екілік мәліметтерден тұрады.

Мамандандырылған мәліметтер қоры

1) sql_variant. Мәліметтер типі әртүрлі мәндерді сақтау үшін қолданылады.

2) timestamp. Сәйкесінше жаңарып отыратын жазбалардың уақытша белгілерін орнату үшін қолданылады. Мәліметтердегі өзгерістерді қадағалау үшін ыңғайлы.

3) uniqueidentifier. Ғаламдық әмбебап идентификатор.

4) xml. Толық құжаттарды немесе XML бөліктерін сақтау үшін қолданылады.

Мәліметтердің қолданушы типін құру.

SQL Server мәліметтердің жүйелік типтерінің негізінде алдын ала берілген параметрлермен, барлық шектеулер мен үндемеушіліктерді қоса қолданушы типтерін құруға рұқсат береді. Мысал ретінде Deliveries кестесінде жеткізушінің телефон номерлерін сақтау үшін қолданылатын phone мәліметтер типін құрайық. Оны құру үшін Management Studio көмекші бағдарламалар графикалық интерфейсін қолданамыз.

Нысандарды көрсету ағашында «Мәліметтер қоры – DB Books – Бағдарламалау - Типтер» папкасын ашыңыз. «Қолданушымен анықталатын мәліметтер типі» тармағының мәтіндік мәзірінде «Қолданушымен анықталатын мәліметтер типін құруды» таңдаңыз.

Пайда болған терезенің «Аты» мәтіндік аймағына phone енгізіңіз. «Мәліметтер типі» ашылатын тізімде nchar таңдаңыз. Ұзындығы ретінде 10 енгізіңіз. Жаңа жеткізушіні енгізетін кезде телефон номерін көрсетпеу мүмкіндігі болу үшін «null мәніне рұқсат беру» параметрін белгілеңіз.

«Бекіту» бөлімінде бос мәндерді қалдырыңыз және ok батырмасын басыңыз. Құрылған мәліметтердің қолданушы типі нысандарды көрсету ағашында пайда болуы керек.

DB_Books мәліметтер қорында бес кесте құрайық. Бірінші кесте «Books», кітаптар туралы ақпаратты сақтайды, екінші кесте – авторлардың анықтамасы, үшіншісі, баспалардың анықтамасы, төртіншісі, жеткізушілердің анықтамасы, бесінші сатып алу туралы толық ақпаратты қамтиды. Төменде осы кестелердің барлық аймақтары және олардың негізгі қасиеттері көрсетілген.

DB_Books.

1 кесте – Кітаптар анықтамасы (кестенің атауы Books)

Аймақтың атауы	Аймақ типі	Аймақ сипаты
code_book	int	Кітаптың коды
title_book	char	Кітаптың атауы
code_author	int	Автордың коды

pages	int	Парақтардың саны
code_publish	int	Баспалардың саны

2 кесте – Авторлардың анықтамасы (кестенің атауы Authors)

Аймақтың атауы	Аймақ типі	Аймақ сипаты
code_author	int	Автордың коды
name_author	char	Автордың тегі, аты, әкесінің аты
birthday	date	Туылған күні

3 кесте – Баспалардың анықтамасы (кестенің атауы Publishing house)

Аймақтың атауы	Аймақ типі	Аймақ сипаты
code_publish	int	Баспаның коды
publish	char	Баспа
city	char	Қала

2 кесте – Жеткізушілердің анықтамасы (кестенің атауы Deliveries)

Аймақтың атауы	Аймақ типі	Аймақ сипаты
code_delivery	Int	Жеткізушінің коды
name_delivery	Char	Тегі, аты, әкесінің аты
Name_company	Char	Жеткізуші компанияның атауы
address	Char	Занды мекен-жайы
phone	Numeric	Байланыс телефоны
inn	Char	ИНН

5 кесте – Сатып алулар (кестенің атауы Purchases)

Аймақтың атауы	Аймақ типі	Аймақ сипаты
code_book	int	Кітаптың жеке коды
Date_order	date	Кітапқа тапсырыс беру күні
code_delivery	Int	Жеткізушінің коды
Type_purchase	bit	Сатып алу типі (көтерме/бөлшек сауда)
cost	money	Бір тауардың бағасы
amount	int	үлгілердің саны
Code_purchase	int	Сатып алу коды

Кестелерді қарапайым графикалық әдіспен құрайық. Алдымен Books кестесін құрамыз.

DB_Books мәліметтер қорының нысандарды көрсететін ағаш текті құрылымында «Кестелер» тармағының мәтіндік мәзірінен «Кесте құру...» командасын таңдаңыз. Жұмыс аймағында кестелерді құрастырушысы бар қосымша терезе пайда болуы керек.

«Бағанның аты» бағанының бірінші жолына code_book енгізіңіз, «Мәліметтер типі» бағанында int таңдаңыз. «Null мәніне рұқсат» параметрінің өшірулі екеніне көз жеткізіңіз.

Экранның төменгі бөлігіндегі «Бағандар қасиеттері» бөлімінде аймақтың сипаттамасын және кітаптың нөмірі автоматты түрде құрылуы үшін «Идентификатордың ерекшеліктері/(Идентификатор)» параметрінің мәнін «Ия» мәніне өзгертіңіз. Int мәліметтер типімен бірге қолданылатын «Сәйкестендіруші баған» (Identity) қасиеті әрбір жаңа жазбаны қосқан кезде мәнін бір бірлікке автоматты түрде өсіруге арналған. Мысалы, кестеге бірінші қосылған кітап идентификатордың 1 мәніне ие болады, екіншісі – 2, үшіншісі – 3, және т.б.

Осыған ұқсас қалған барлық аймақтардың сипаттамаларын енгізіңіз және кестелер құрастырушының терезесін жабыңыз. Кестенің аты ретінде Books енгізіңіз. Қайта құрылған кесте нысандарды көрсететін ағаш текті құрылымда «Кестелер» папкасында пайда болуы керек.

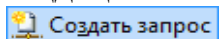
Өзіндік жұмысқа арналған тапсырма: жоғарыда келтірілген сипаттамаға сәйкес қалған төрт кестені құрыңыз: Authors, Publishing house, Deliveries және Purchases.

Transact – SQL кодының қатысуымен.

SQL Server Management Studio ортасында бағдарлама кодын жазу үшін «Стандартты» («Standart») құрал-саймандар тақтасында «Сұраныс құру» («New query») батырмасын басып керек.

Жаңа мәліметтер қорын құрудың құрылымы қолданыстағы мәліметтер қорының негізінде туындауы мүмкін. Ол үшін SQL Server Management Studio ортасында «DB_Books» торабының мәтіндік мәзірінен «Мәліметтер қоры үшін құрылым құру – CREATE қолдану арқылы – Алмасу буфері» командасын таңдаңыз. Нәтижесінде алмасу буферінде Management Studio-да DB_Books мәліметтер қорын құру кезінде көрсетілген параметрлері бар жаңа мәліметтер қорын құруға арналған сұраныстың мәтіні сақталады. Жаңа мәліметтер қорын құруға арналған алдын ала құрылған сұраныстарды қолдану үшін келесі қадамдарды орындаңыз:

1) DB_Books қорының мәтіндік мәзірінде «Сұраныс құру» командасын таңдаңыз немесе саймандар тақтасындағы сәйкес батырманы таңдаңыз



2) SQL редакторының ашылған терезесінде алмасу буферінен құрылған сұранысты қойыңыз.

3) Сұранысты орындауға жіберу үшін саймандар тақтасында  батырмасын басыңыз немесе F5 батырмасын басыңыз.

4) «Мәліметтер қоры» торбының мәтіндік мәзірінен «Жаңарту» командасымен нысандарды көрсету ағашының құрамын жаңартыңыз. Sales мәліметтер қоры қол жетімділердің тізімінде пайда болуы керек.

Мәліметтер қорын ажырату және жалғау.

Мәліметтер қорын басқа серверге көшіру үшін оны қолданыстағы серверден ажырату қажет. Ол үшін DB_Books мәліметтер қорының мәтіндік мәзірінен «Тапсырмалар – Ажырату...» командасын таңдаңыз. «Мәліметтер қорын ажырату» диалогті терезесінде «ОК» батырмасын басыңыз және нысандарды көрсету ағашындағы мәліметтер қоры тізімінен DB_Books жоғалғанына көз жеткізіңіз. Енді мәліметтер қорының файлы басқа серверге көшіріледі.

Мәліметтер қорын серверге жалғау үшін «Мәліметтер қоры» торабының мәтіндік мәзірінен «Жалғау...» командасын таңдаңыз. «Мәліметтер қорын жалғау» диалогті терезесіндегі «Қосу...» батырмасының көмегімен алдыңғы санаттарда құрылған DB_Books.mdf файлын таңдаңыз (ldf файлы автоматты түрде жүйемен анықталады), қожайынын sa-ға өзгертіңіз және «ОК» бастырмасын басыңыз. DB_Books мәліметтер қоры нысандарды көрсету ағашында пайда болуы керек.

DB_Books есімімен жаңа мәліметтер қорын команда көмегімен құру.

Create database DB_Books

Команданың орындалуы үшін F5 басыңыз.

«Нысандарды көрсеткіш» тақтасында DB_Books МҚ-ң болуын тексеру, егер оны Databases бөлімінде көрмесеңіз, онда жаңарту үшін F5 басыңыз.



2 сурет – МҚ құрудың нәтижесі

Мұнда келесі командалардың көмегімен аталған кестелерді құрыңыз (SQL Server Management Studio-да код үшін жаңа парақша құру үшін «Сұраныс құру» батырмасын басыңыз).

Use DB_Books.

Create table Authors(Code_author INT PRIMATY KEY, name_author CHAR(30), Birthday DATETIME)

Create table Publishing_house(Code_publish INT PRIMATY KEY, Publish CHAR(30), City CHAR(20))

Create table Books(Code_book INT PRIMATY KEY, Title_book CHAR(40), Code_author INT FOREIGN KEY REFERENCES Authors(Code_author), Pages INT, Code_publish INT FOREIGN KEY REFERENCES Publishing_house(Code_publish))

Create table Deliveries(Code_delivery INT PRIMATY KEY, name_delivery CHAR(30), Name_company CHAR(20), Address VARCHAR(100), Phone BIGINT, INN CHAR(13))

Create table Purchases(Code_purchase INT PRIMATY KEY, code_book INT FOREIGN KEY REFERENCES Books(Code_book), Date_order SMALLDATETIME, Code_delivery INT FOREIGN KEY REFERENCES Deliveries(Code_delivery), Type_purchase BIT, Cost FLOAT, Amount INT)

Орындалуға F5 бастырмасымен жіберіңіз.

SQL Server Management Studio утилитында DB_Books МҚ-ң және ондағы кестелердің бар екендігін тексеру.

Шектеулерді құру.

Кестелермен жұмысты бастамастан бұрын мәліметтердің бүтіндігін қамтамасыз ету мақсатында онда енгізілетін мәліметтерді шектеу керек, яғни мәліметтер қорына қате немесе қарсы келетін мәліметтердің енуін шектеу, енгізілген жағдайда қандай да бір жазбаны өзгерту немесе өшіру, мысалы, теріс мәнді бағаны немесе тауар санын енгізу. Мәліметтер бүтіндігінің төрт типі қолданыста: доменді, қажетті, сілтемелік және қолданбалы (немесе бизнес-ереже). Осыларды жүзеге асыру үшін SQL Server-де ұсынылған негізгі құрал-саймандарды қарастырайық.

1) Доменді бүтіндікті қамтамасыз ету. Аймаққа қолданушымен енгізілетін мәліметтердің арлығын шектеу. Тексерудің шектеулері және әдеттегідей мәндер доменді бүтіндіктің негізгі құралдары болып табылады.

2) Тексеруші шектеулерді қолдану.

3) Аймаққа енгізілетін мәліметтерді олардың дұрыс типті болғанының өзінде шектеу үшін тексеруге шектеулер қолданылады. Шектеулерді құруды қарастырайық:

- Books кестесінің «Шектеулер» папкасының мәтіндік мәзірінен «Шектеу құру» командасын таңдаңыз;

4) Ашылған «Тексеруші шектеулер» терезесінде келесі аймақтарды толтырыңыз:

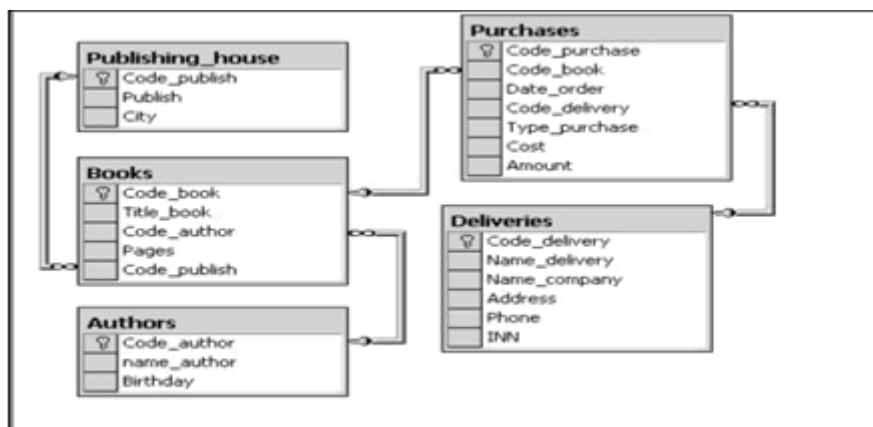
- аты: CK_Pages

- өрнегі: ([Pages]>0). Берілген теңсіздік Pages бағанының мәндері үшін шектеуді сипаттайды;

- сипаттамасы: бағанның мәндеріне шектеу – парақтар саны.

5) «Жабу» батырмасын басыңыз және кестелер құрастырушысын өзгертулерді сақтау арқылы жабыңыз (ол шектеулерді құруды бастағанда ашық болды).

Диаграммалар бөлімінде жаңа диаграмма құрыңыз, онда тізімдегі бес кестені қосып, олардың арасындағы байланысты тексеру керек.



3 сурет – Диаграмма құрудың нәтижесі

Пайдаланылған операторлар:

- 1) PRIMARY KEY – негізгі өрісті құрудың белгісі.
- 2) FOREIGN KEY ... REFERENCES... - басқа кестемен байланысу өрісін құрудың белгісі.
- 3) CREATE TABLE – қолданыстағы мәліметтер қорында кесте құрудың командасы. USE – бәсекелес МҚ-ны белсенді ету.
- 4) CREATE DATABASE – жаңа МҚ құрудың командасы.

Зертханалық жұмыс №1 арналған тапсырмалардың нұсқалары

SQL Server Management Studio утилитында Create Database операторының көмегімен жаңа мәліметтер қоры құрылады, МҚ атауы тақырыптық аймаққа байланысты анықталады. Операторға түсініктеме беру (- - – бір жолды түсініктеме, /* */ – көп жолды түсініктеме). Use операторының көмегімен құрылған МҚ-ны бағдарламалау арқылы белсенді ету. Атап өтілген кестелерді Create table операторының көмегімен құру, және де өз бетінше кестелердің типін анықтау (басыңқы және бағыныңқы), аймақтардың типін және олардың өлшемдерін, Primary key және Foreign key типті аймақтарды табу. Бағдарлама файлын Студенттің Тегі_ЗМ_1_нұсқа № атауымен сақтаңыз. SQL Server Management Studioда құрылған МҚ-ң диаграммалар бөлімінде жаңа диаграмма құрастыру, кестелер арасындағы байланыстарды тексеру.

МҚ-ң нұсқалары	1 кесте	2 кесте	3 кесте
Берілген жаңа жылдық сылықтарды есепке алу	Қызметкердің коды	Қызметкердің коды	Баланың коды
	Тегі	Баланың аты	Сыйлықтың бағасы
	Аты	Туған күні	Сыйлықты беру күні
	Әкесінің аты	Баланың коды	Беру коды
	Лауазымы		Баланың коды
	Бөлімше		
	Жұмысқа		

	қабылданған күні		
Жөндеу жұмыстары	Жөндеудегі аспаптың коды	Аспаптың коды	Мастердің коды
	Аспап атауы	Мастердың коды	Мастердің тегі
	Аспап типі	Аспап иесінің аты-жөні	Мастердің аты
	Құрастырылған күні	Жөндеуге қабылданған күні	Мастердің әкесінің аты
		Бұзылудың түрі	Мастердің санаты
		Жөндеудің құны	Жұмысқа қабылданған күні
		Жөндеу коды	
Гүл сату	Гүлдің коды	Гүлдің коды	Сатушының коды
	Гүлдің атауы	Сатылым күні	Тегі
	Гүлдің сорты	Сатылым бағасы	Аты
	Орташа биіктігі	Сатушының коды	Әкесінің аты
	Парақтың типі	Сатылым коды	Санаты
	Гулденген		Айлық
	Қосымша ақпарат		Жұмысқа қабылданған күні
Дәрілік құралдардың түсімі	Дәрінің коды	Дәрінің коды	Жеткізушінің коды
	Дәрінің атауы	Жеткізушінің коды	Қысқартылған атауы
	Қолданудың нұсқамасы	Жеткізілген күні	Толық атауы
	Өлшем бірлігі	Бір данасының бағасы	Заңды мекен-жайы
	Қораптағы саны	Саны	Телефон
	Өндірушінің атауы	Түсірілген күні	Жетекшінің
Құрылғыларды қолданыстан шығару	Құрылғының коды	Құрылғының коды	Қызметкердің коды
	Құрылғының атауы	Қолданыстан шығару себебі	Тегі
	Құрылғының типі	Қолданыстан шығару күні	Аты
	Қабылданған күні	Қызметкердің коды	Әкесінің аты

	Жауапты адамның ТАӘ	Қолданыстан шығару коды	Лауазымы
	Орнатылған орыны		Бөлімшесі
			Жұмысқа қабылданған күні
6. Аспаздық кітап	Тағам коды	Тағам коды	Өнімнің коды
	Тағам типі	Өнімнің коды	Өнімнің атауы
	Тағам салмағы	Өнімнің көлемі	Өлшем бірлігі
	Дауындау реті		
	Калориялар саны		
	Көмірсутегілердің саны		
7. Кіріс құжаттаманы тіркеу	Тіркеушінің коды	Құжаттың коды	Жіберуші ұйымның коды
	Тегі	Құжаттың нөмірі	Қысқартылған атауы
	Аты	Тіркелген күні	Толық атауы
	Әкесінің аты	Құжаттың қысқаша мазмұны	Заңды мекен-жайы
	Лауазымы	Құжаттың типі	Телефон
	Жұмысқа қабылданған күні	Жіберуші ұйымның коды	Жетекшінің ТАӘ
		Тіркеушінің коды	
8. Қызметкердің жұмыстан шығуы	Қызметкердің коды	Құжаттың коды	Жұмыстан шығу бабының коды
	Тегі	Құжаттың нөмірі	Жұмыстан шығу бабының атауы
	Аты	Тіркелген күні	Жұмыстан шығу себебі
	Әкесінің аты	Жұмыстан шығу күні	Жұмыстан шығу бабының нөмірі
	Лауазымы	Жұмыстан шығу бабының коды	Жұмыстан шығу тармағының/тармақшасының нөмірі
	Бөлімшесі	Қызметкердің коды	
	Жұмысқа қабылданған	Қаржылай өтемақы	

	күні		
9. Демалыс	Қызметкердің коды	Құжаттың коды	Демалыс коды
	Тегі	Құжаттың нөмірі	Демалыс түрі
	Аты	Тіркелген күні	Демалыстаң төлемі
	Әкесінің аты	Демалыстың басталған күні	Демалыс бойынша жеңілдіктер
	Лауазымы	Демалыстың аяқталған күні	
	Бөлімшесі	Қызметкердің коды	
	Жұмысқа қабылданған күні	Демалыс коды	
10. Шығыс құжаттаманың тіркелуі	Жіберушінің коды	Құжаттың коды	Қабылдаушы ұйымның коды
	Тегі	Құжаттың нөмірі	Қысқартылған атауы
	Аты	Тіркелген күні	Толық атауы
	Әкесінің аты	Құжаттың қысқаша мазмұны	Заңды мекен-жайы
	Лауазымы	Құжаттың типі	Телефон
	Жұмысқа қабылданған күні	Қабылдаушы ұйымның коды	Жетекшінің ТАӘ
		Жіберушінің коды	
11. Лауазым	Қызметкердің коды	Құжаттың коды	Лауазым коды
	Тегі	Құжаттың нөмірі	Лауазым атауы
	Аты	Тіркелу күні	Лауазым бойынша жеңілдіктер
	Әкесінің аты	Тағайындалу күні	Біліктілікке қойылатын талаптар
	Жұмысқа қабылданған күні	Қызметкердің коды	
	Туған күні	Лауазымының коды	
	Жынысы		

12. Құрылғыларды жалға беру	Клиенттің коды	Берілген коды	Құрылғының коды
	Тегі	Құжаттың коды	Құрылғының атауы
	Аты	Жалдаудың басталған күні	Құрылғының типі
	Әкесінің аты	Жалдаудың аяқталған күні	Жалдауға қабылданған күні
	Мекен-жайы	Құрылғының коды	
	Телефон	Клиенттің коды	
	Төлқұжаттың нөмірі мен сериясы	Бағасы	
13. Жалдаудағы құрылғыларды қолданыстан шығару	Құрылғының коды	Құрылғының коды	Қызметкердің коды
	Құрылғының атауы	Қолданыстан шығару себебі	Тегі
	Құрылғының типі	Қолданыстан шығару күні	Аты
	Жалдауға қабылданған күні	Қызметкердің коды	Әкесінің аты
		Құжаттың нөмірі	Лауазымы
		Тіркелу күні	Жұмысқа қабылданған күні
		Қолданыстан шығару коды	
14. Гүл сататын бутик	Гүлдің коды	Гүлдің коды	Жеткізушінің коды
	Гүлдің атауы	Қабылданған күні	Қысқартылған атауы
	Гүлдің сорты	Бір дананың құны	Толық атауы
	Орташа биіктігі	Жеткізушінің коды	Заңды мекен-жайы
	Қағаз типі	Қабылдау коды	Телефон
	Гүлденетін	Саны	Жетекшінің ТАӘ
	Қосымша мәлімет		
15. Қонақжай клиенттерін тіркеу	Бөлме коды	Тіркеу коды	Клиенттің коды
	Бөлме типі	Бөлме коды	Тегі
	Жайлылық деңгейлері	Орналасу күні	Аты
	Бір күнге	Шығу күні	Әкесінің аты

	бағасы		
		Бағасы	Мекен-жайы
		Клиенттің коды	Телефон
			Төлқұжатының нөмірі
16. Құрылғыны жалға беру қызметіне қайтару	Клиенттің коды	Қайтару коды	Құрылғының коды
	Тегі	Құжаттың нөмірі	Құрылғының атауы
	Аты	Қайтару күні	Құрылғының типі
	Әкесінің аты	Құрылғының күйі	Жалдауға қабылданған күні
	Мекен-жайы	Құрылғының коды	
	Телефон	Клиенттің коды	
	Төлқұжатының нөмірі	Айыппұл	
17. Кәсіпорындағы материалдық құндылықтардың есебі	Құндылықтың коды	Есепке қою коды	Материалды жауапты адамның коды
	Құндылықтың атауы	Құндылықтың коды	Тегі
	Құндылықтың типі	Материалды жауапты адамның коды	Аты
	Сатып алу бағасы	Есепке қою күні	Әкесінің аты
	Кепілдеме мерзімі	Құндылықтың орналасқан орыны	Лауазымы
	Кепілдеменің басталу күні		Жұмысқа қабылданған күні
			Бөлімшесі
18. Жөндеу жұмыстары	Жөндеу жұмысының коды	Жөндеу жұмысының коды	Мастердің коды
	Жұмыс кезеңінің коды	Мастердің коды	Мастердің тегі
	Жұмыс кезеңінің атауы	Жөндеудің құны	Мастердің аты
	Кезеңнің бағасы	Жөндеу күндерінің саны	Мастердің әкесінің аты
		Жөндеу жұмысының	Мастердің санаты

		атауы	
			Жұмысқа қабылданған күні
19.Дәріхана	Дәрінің коды	Чектің нөмірі	Чек нөмірі
	Дәрінің атауы	Бір дананың құны	Сатылым күні
	Қолдануға арналған көрсетілімдер	Саны	Сумма
	Өлшем бірлігі	Дәрінің коды	Кассирдің ТАӘ
	Қораптағы саны	Чектердегі жазбаның коды	
	Өндірушінің атауы		
20.Құжатайнал ым	Құжаттың коды	Орындаушының коды	Құжаттың коды
	Орындалуға тағайындалған күні	Тегі	Құжаттың нөмірі
	Күн бойынша орындалу мерзімі	Аты	Тіркелу күні
	Нәтиже типі	Әкесінің аты	Қысқаша мазмұны
	Орындаушының коды	Лауазымы	Құжаттың типі
	Орындалу дерегі	Бөлімшесі	Жіберуші мекеме
		Жұмысқа қабылданған күні	Орындаушының коды

Бақылау сұрақтары.

1. SQL Server Management Studio –ның іске қосылуы кезінде қандай мәліметтерді көрсету қажет?
2. Management Studio терезесінің компоненттерін көрсетіңіз?
3. MS SQL сервердің қызметтік мәліметтік қорларын және олардың қызметтерін атап шығыңыз?
4. Microsoft SQL Server мәліметтер қоры қандай бөліктерден тұрады?
5. SQL Server Management Studio көмегімен мәліметтер қорын құрудың графикалық әдісін сипаттаңыз?
6. Transact – SQL кодының қатысымен мәліметте қорын құруды сипаттаңыз?
7. MS SQL Server қандай мәліметтер типтерін қолдайды?

8. Мәліметтер қорын серверге қалай жалғайды және ажыратады?

Зертханалық жұмыс №2

Тақырыбы: Мәліметтермен айла-тәсілдер жасайтын Microsoft SQL Server операторлары.

Жұмыстың мақсаты: Select, Insert, Update, Delete операторларын қолдануды үйрену.

Жұмыстың барысы:

1) Келтірілген операторлардың көмегімен өткен зертханалық жұмыста сипатталған «Кітаптар» мәліметтер қоры мысалын құрыңыз (егер МҚ серверде болмаса).

2) Кестелерді мәліметтермен толтыру үшін Insert операторының көмегімен SQL Server Management Studio-да «Сұраныс құру» арқылы бағдарлама құру.

3) Select операторының көмегімен тапсырмалар бойынша МҚ-ға сұраныстарды орындау.

Жұмыстың орындалуына түсініктеме.

Мәліметтерді SQL-командалардың қатысымен Query Analyzer терезесінде немесе Enterprise Manager көмегімен енгізуге болады. Екі жағдайда да нәтиже бірдей болғанымен, қолданушының көзқарасы тарапынан бұл екі әдістің бір бірінен көп айырмашылығы бар. Қажет жағдайда MS SQL Server 2012 анықтама дереккөздерімен Book OnLine көмекші қосымшасын қосу арқылы қолдануға болады.

SQL тілі операторлар жиынынан тұрады, оларды келесідей төрт топқа бөлуге болады:

- DDL (Data Definition Language) – мәліметтерді анықтау операторлары;
- DML (Data Manipulation Language) – мәліметтермен айла-тәсілдер жасау операторлары;
- DCL (Data Control Language) – мәліметтерге қолжетімділікті анықтайтын операторлар;
- TCL (Transaction Control Language) – транзакцияларды басқару операторлары.

SELECT – жиі қолданылатын SQL оператор. Ол кестеден ақпаратты көшіріп алуға арналған. SELECT операторының көмегімен кестедегі мәліметтерді көшіріп алу үшін кем дегенде екі нәрсені көрсетуіңіз керек – не таңдайтыныңызды және қайдан алатыныңызды.

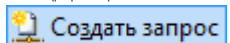
Жеке бағандарды көшіру.

```
SELECT [code_book]  
FROM Books
```

Жоғарыда келтірілген операторда Books кестесінен code_book бағанын көшіріп алу үшін SELECT операторы қолданылған. Ізделініп отырған бағанның атауы SELECT сөзінен кейін бірден көрсетіледі, ал FROM кілттік сөзі мәліметтері таңдалатын кестенің атауына сілтейді.


Берілген сұранысты Management Studio-да құру және сынау үшін келесі қадамдарды орындаңыз:

1) DB_Books қорының мәтіндік мәзірінде «Сұраныс құру» командасын таңдаңыз немесе құрал-саймандар тақтасындағы сәйкес батырманы басыңыз



2) Ашылған жаңа сұраныстар құру терезесіне жоғарыда көрсетілген SQL нұсқауларын енгізіңіз.

3) Сұранысты орындалуға жіберу үшін құрал-саймандар тақтасында  **Выполнить** батырмасын басыңыз немесе F5 батырмасын басыңыз.

4) Management Studio SQL пакеттерін сақтауға рұқсат береді. Бұл алдағы уақытта қайта іске қосылатын күрделі сұраныстарды сақтауға қолайлы. Ол үшін құрал-саймандар тақтасында  батырмасын басыңыз. Сұраныстар файлдары өздігінен .sql кеңейтілімінде сақталады. Мұнан былай сақталған сұраныс «Файлды ашу» командасымен ашыла алады.

Бірнеше бағанды таңдау.

Кестеден бірнеше бағанды таңдау үшін де SELECT операторы қолданылады. Айырмашылығы тек мұнда, SELECT кілттік сөзінен кейін бірнеше бағанның атауын үтір арқылы бөліп көрсету керек.

```
SELECT [code_book], Title_Book  
FROM Books
```

Барлық бағандарды таңдау.

Бірнеше белгілі бағандарды (біреу немесе бірнеше) таңдауды жүзеге асыру мүмкіндігімен қатар, SELECT операторының көмегімен әр бағанды атап өтпей-ақ, барлық бағандарды сұрауға болады. Ол үшін бағандардың атауының орынына “жұлдызша” топтық белгісі қойылады. Бұл келесідей орындалады.

```
SELECT *  
FROM Books
```

Мәліметтерді сұрыптау.

Таңдауға арналған сұраныстың орындалуының нәтижесінде мәліметтер кестеде орналасқан ретімен шығарылады. SELECT операторының көмегімен таңдалған мәліметтерді дәл сұрыптау үшін ORDER BY операторы қолданылады. Бұл операторда нәтижелерін сұрыптау қажет бір немесе бірнеше бағанның атауы көрсетіледі. Келесі мысалды қараңыз.

```
SELECT [code_book], Title_Book  
FROM Books  
ORDER BY Title_Book
```

МҚБЖ-ға Title_Book бағанының мәндерін өсу ретімен сұраптауды сілтейтін ORDER BY сөйлемін қарастырмағанда, бұл өрнек алдыңғыға ұқсайды.

Бірнеше баған бойынша сұрыптау.

Бірнеше баған бойынша сұрыптауды жүзеге асыру үшін олардың атауларын үтір арқылы көрсетіңіз. Келесі кодта үш баған таңдалады, ал нәтижесі екеуі бойынша сұрыпталады – алдымен саны бойынша, кейін атауы бойынша.

```
SELECT code_book, Title_Book  
FROM Books  
ORDER BY code_book, Title_Book
```

Бірнеше баған бойынша сұрыптау кезінде сұрыптау реті сұраныста көрсетілген ретпен жүзеге асырылатынын білу маңызды. Басқаша айтқанда, жоғарыда көрсетілген мысалда кітаптар Title_Book бағаны бойынша сұрыпталады, тек егер code_book мәні бар бірдей жатық жолдары бар болса. Егер code_book бағанының ешқандай мәні сәйкес келмесе, онда Title_Book бағаны бойынша мәліметтер сұрыпталмайды.

Сұрыптаудың бағытын көрсету.

ORDER BY сөйлемінде кему ретімен сұрыптауды да қолдануға болады. Ол үшін DESC кілттік сөзін көрсету керек. Келесі мысалда өнім саны бойынша кему ретімен, қосымша атауы бойынша сұрыпталады.

```
SELECT code_book, Title_Book  
FROM Books  
ORDER BY code_book DESC, Title_Book
```

DESC кілттік сөзі өзінің алдында көрсетілген бағанға ғана қолданылады. Өткен мысалда DESC кілттік сөзі code_book бағаны үшін көрсетілген болатын, Title_Book үшін емес. Осылайша, code_book бағаны кему ретімен сұрыпталды, ал Title_Book бағаны өсу ретімен (үнсіз келісім бойынша).

Мәліметтердің фильтрациясы.

Қажет мәліметтерді ғана таңдау фильтрация атауымен белгілі іздеу шартынан тұрады. SELECT операторында мәліметтер WHERE сөйлемінде іздеу шартын көрсету арқылы сұрыпталады. WHERE сөйлемі бағанның атауынан кейін (FROM сөйлемі) бірден келесідей көрсетіледі:

```
SELECT Title_Book
FROM Books
WHERE code_book =11
```

Бұл оператор кітаптар кестесінен кітаптың атауы – мәнін шығарады, бірақ барлық жолдарды көрсетпейді, тек code_book бағанының 11-ге тең мәндерін (кітап коды) шығарады.

ORDER BY мен WHERE өрнектерін бірге қолданған кезде, ORDER BY сөйлемі WHERE сөйлемінен кейін жүреді.

Өткен мысалда теңдікке тексеру жүргізілді, яғни көрсетілген мәнің бағанда бар болуы анықталды. SQL келесі кестеде көрсетілген барлық шартты (логикалық) операцияларды қолдайды.

1 кесте

Операция	Сипаттамасы
=	Теңдік
<>	Теңсіздік
!=	Теңсіздік
<	Кем
<=	Кем немесе тең
!<	Кем емес
>	Артық
>=	Артық немесе тең
!>	Артық емес
BETWEEN	Көрсетілген екі мәнің арасында
IS NULL	NULL мәні

Келесі мысалда барлық жеткізуші компаниялардың атауларын, байланыс телефондары көрсетілмеген жауапты адамдарымен бірге таңдау жүзеге асырылады.

```
SELECT name_delivery, name_company, phone
FROM Deliveries
WHERE phone is NULL
```

Мәндер аралығын іздеу үшін BETWEEN операциясын қолдануға болады. Оның жүйесі WHERE сөйлемінің басқа операцияларынан біраз ерекшеленеді, өйткені оған екі мән қажет: бастапқы және соңғы. Мысалы, BETWEEN операциясын дана саны 5 пен 10 аралығындағы кітаптарды іздеуге қолдануға болады.

```
SELECT code_book
FROM Purchases
```


WHERE amount BETWEEN 5 AND 10

WHERE сөйлеміне бірнеше шартты біріктіру үшін AND және (немесе) OR логикалық операторларын қолдану қажет. AND операторы екі операцияның қатар орындалуын талап етеді. Өткен мысалды екі салыстыру операциясын AND операторымен біріктірілген түрінде жазайық.

```
SELECT code_book
FROM Purchases
WHERE (amount >= 5) AND (amount <= 10)
```

AND кілттік сөзі МҚБЖ-ға таңдаудың аталған шарттарын қанағаттандыратын жолдарын ғана қайтаруды көрсетеді. Берілген жағдайда дана саны 5 пен 10 аралығында жатқан кітаптар ғана таңдалынады.

OR операторы МҚБЖ-ға кем дегенде бір шартты қанағаттандыратын жолдарды таңдауды көрсетеді.

```
SELECT publish
FROM Publishing house
WHERE (CityName = «Алматы») OR (CityName = «Астана»)
```

Осы SQL сұранысқа қатысты баспалардың анықтама кітабынан Алматы мен Астанада орналасқандар ғана таңдалынады. OR кілттік сөзі МҚБЖ-ға бірден екеуін емес, белгілі тек бір шартты қолдануды көрсетеді. Егер мұнда AND кілттік сөзі қолданылғанда ешқандай мәлімет алынбас еді.

Өткен WHERE сөйлемінің өрнегін мұқият қарайтын болсаңыз, онда қаланың атаулары салыстырылатын мәндер бірлік тыршақшаға алынғанын байқайсыз. Бірлік тырнақшалар жолдың шекарасын анықтау үшін қолданылады (жатық жол тұрақтысы). Жатық жол тұрақтыларымен жұмыс істеу кезінде оларды үнемі бірлік тырнақшаға алу керек.

WHERE сөйлемі AND және OR операторларының кез келген санынан тұруы мүмкін. Оларды біріктіру арқылы күрделі фильтрларды құруға болады. Бірақ AND және OR кілттік сөздерін біріктіру кезінде AND операторы OR операторынан бұрын орындалатынын білу қажет, яғни жоғары басымдылығы бар. Басымдылықты доғал жақшалардың көмегімен өзгертуге болады.

Келесі мысалда клиенттер кестесінен байланыс телефондары көрсетілмеген барлық Ивановтар мен Петровтар таңдалынады.

```
SELECT name_company, address, phone
FROM Deliveries
WHERE (name_delivery = «Куралбаев А.С») OR name_delivery =
«Федоренко И.А») AND phone is NULL
```

Жақшалар болмаған жағдайда нәтиже дұрыс болмас еді, атап айтқанда байланыс телефондары жоқ барлық Куралбаев А.С-ларды және ешқандай шектеусіз барлық Федоренко И.А шығарушы еді.

Салыстырылып отырған мәннің белгілі бір берілген көптікке кіретінін анықтау үшін IN операторын қолдануға болады. Сонымен бірге жақшаға алынған барлық рұқсат етілген мәндер үтір арқылы атап өтіледі. Сондай-ақ IN операторын қолдану арқылы өткен мысал ықшам түрде жазылуы мүмкін.

```
SELECT name_company, address, phone
FROM Deliveries
WHERE name_delivery IN («Куралбаев А.С»,»Федоренко И.А») AND
phone is NULL
```

Қандай да бір шартты жоққа шығару үшін NOT логикалық операторы қолданылады. NOT өздігінен қолданылмайтындығынан (тек басқа логикалық операторлармен бірге), оның сөйлем жүйесі басқа операторлардан ерекшеленеді. Олардан қарағанда NOT сұрыптау қажет баған атауының алдына қойылады, кейін емес. Келесі мысалда барлық кітап жеткізушілер таңдалынады, олардың байланыс телефондары туралы мәлімет бар.

```
SELECT name_company, address, phone
FROM Deliveries
WHERE NOT phone is NULL
```

Белгілі бір таңбалық жолдың берілген үлгімен сәйкестігі шарты бойынша мәліметтерді сұрыптау үшін LIKE опертары қолданылады. Үлгі қарапайым таңбалардын және үлгі-таңбалардан тұруы мүмкін. Үлгімен сәйкестендірген кезде оның қарапайым таңбалары жолда көрсетілген таңбалармен дәл келуі маңызды. Таңба-үлгілер таңбалық жолдың еркін элементтерімен дәл келуі мүмкін. Таңба-үлгілерді LIKE опертарымен қолдану қарапайым салыстыру операторларын қолданудан қарағанда көбірек мүмкіндіктер ұсынады. Үлгі келесі таңба-үлгілерден тұруы мүмкін.

2 кесте

Таңба-үлгі	Сипаттамасы	Мысал
%	Таңбалардың кез келген саны	WHERE name_company LIKE «A%» нұсқаулығы аты «А» әріпінен басталатын барлық жеткізушілерді іздейді және шығарады.
—	Кез келген жалғыз таңба	WHERE name_author LIKE «_акатаев» нұсқаулығы аты сегіз әріптен тұратын және «акатаев» (Макатаев, Такатаев және т.б.)

		сөзімен аяқталатын барлық авторларды іздейді және шығарады.
[]	Тік жақшада көрсетілген кез келген таңба	WHERE name_author LIKE «[Л-С]омов» нұсқаулығы тегі «омов»-ға аяқталатын және «Л» ден «С»-ға дейінгі аралықта кез келген әріпке басталатын авторларды іздейді және шығарады, мысалы, Ломов, Ромов, Сомов және т.б.
[^]	Тік жақшада көрсетілгендерден басқа таңбалардың кез келгені	WHERE name_author LIKE «ив[^а]» нұсқаулығы тегі «Та»-ға басталатын және үшінші әріпі «а»-дан басқа авторларды іздейді және шығарады.

Келесі мысалда атауы Т әріпінен басталатын барлық кітаптарды таңдау жүзеге асырылады.

```
SELECT *
FROM Books
WHERE title_book] LIKE «Т%»
```

Есептелетін өрістерді құру.

SELECT құрылысы кесте бағандарының атауларынан бөлек есептелетін өрістерден де тұрады. Өткенде таңдалған барлық бағандардан айырықша, есептелетін өрістер іс жүзінде МҚ-ның кестелерінде жоқ. Олар SELECT SQL-операторымен “ұшып бара жатып” құрылады. Келесі мысалды қарастырайық.

```
SELECT code_delivery AS «Жеткізушінің нөмірі», name_company+»
«+inn AS «Компания атауы және ЖСН»
FROM Deliveries
```

Мұнда AS кілттік сөзінің көмегімен «Компания атауы және ЖСН» лақап аты берілген есептелетін өріс құрылады. Ол + операторының көмегімен Компания атауын, бос орын және ЖСН-ды бір өріске (бағанға) біріктіруге (конкатенация жүргізуге) мүмкіндік береді. Лақап ат кестенің қарапайым бағанына да берілуі мүмкін. Сонымен қатар code_delivery бағанына «Жеткізушінің нөмірі» лақап аты берілген.

Есептелетін өрістерді қолданудың тағы бір әдісі ретінде таңдалған мәліметтерге математикалық операцияларды орындау болып табылады.

Мысалды қарастырайық.

```
SELECT code_purchases, cost, amount, cost * amount AS «Стоимость»
FROM Purchases
```

WHERE code_ purchases=1

Мұнда көбейту операторының * көмегімен коды 1 тапсырысындағы кітаптардың сатылымының ортақ бағасы санын бағасына көбейту арқылы есептеледі.

Қосарлама жазбаларды шеттету.

Таңдаудың нәтижесінен қосарлама жолдарды шеттету үшін SELECT командасынан кейін бірден көрсетілетін DISTINCT кілттік сөзі қолданылады. Келесі мысалда барлық кітап авторларының тектерін шығару жүзеге асырылады. Егер арасында тегі ұқсастары болса да, әр тегі бір рет қана шығарылады.

```
SELECT DISTINCT name_author  
FROM Authors
```

Агрегаттық функцияларды қолдану.

SQL-де әртүрлі санаттағы көптеген орнатылған функциялары анықталған, олардың ішінде агрегаттық функциялар ерекше орын алады, көптіктің жол мен бағандарының мәндерін пайдаланады және бір мән қайтарады. Кесте бағандары мен оларға қолданылған амалдардың мәндері де агрегаттық функцияның аргументы бола алады. Агрегаттық функциялардың өздері де басқа арифметикалық өрнектерге қосыла алады. Келесі кестеде жиі қолданылатын стандартты унарлы агрегаттық функциялар келтірілген.

3 кесте

Функция	Қайтарылатын мәні
COUNT	Кестенің жолдары мен бағандарындағы мәндердің саны
SUM	Сумма
AVG	Орташа мәні
MIN	Минимум
MAX	Максимум

Унарлы агрегаттық функцияның түрі келесідей:

функцияның_аты ([ALL|DISTINCT] өрнегі)

мұнда DISTINCT функция аргументтің тек әртүрлі мәндерін қарастыруы керектігін көрсетеді, ал ALL – қайталанатын мәндермен қоса (бұл нұсқа әдеттегі қалпы бойынша қолданылады), барлығын.

Агрегаттық функциялар SELECT және HAVING сөздерінде қолданылады. Мұнда, олардың SELECT сөзінде қолданылуын қарастырамыз. Бұл жағдайда функция аргументіндегі өрнек SELECT сөзінің барлық кіріс кестелерінің жолдарына қолданылады. Сонымен қатар, SELECT сөзінде GROUP BY сөзін қолданбай, агрегаттық функциялар мен кесте бағандарын (немесе олар қатысқан өрнектерді) қатар қолдануға болмайды.

COUNT функциясының екі үлгісі бар. Бірінші жағдайда кіріс кестенің жолдарының саны қайтарылады, екінші жағдайда – кіріс кестедегі аргумент мәндерінің саны қайтарылады:

- COUNT(*);
- COUNT([DISTINCT|ALL] өрнегі).

Осы функцияны қолданудың ең қарапайым түрі – кестедегі жолдардың санын санау (берілген шартты қанағаттандыратындарды немесе барлығын). Ол үшін сөйлем құрылысының бірінші нұсқасы қолданылады.

Сұраныс: Қайталанатын мәндері жоқ авторлар саны.

```
SELECT COUNT(distinct name_author AS «Авторлар саны»)
FROM Authors
```

Басқа унарлы агрегаттық функцияларды қолдану осы COUNT қолданылуына ұқсайды, бірақ MIN және MAX функциялары үшін DISTINCT және ALL кілттік сөздерін қолдану мағынасыз. COUNT, MIN және MAX функцияларымен бірге сандық аймақпен қатар таңбалық аймақтар да қолданылады. Егер агрегаттық функцияның аргументінің мәндері болмаса, онда COUNT функциясы 0 қайтарады, ал басқалары – NULL мәнін.

Жолдарды топтау сұранысы.

Жоғарыда сипатталған агрегаттық функциялар барлық кестелерге қолданылды. Бірақ та есеп құру кезінде аралық қорытында мәндерді пішімдеу қажеттілігі жиі туындайды, яғни кестеге қатысты барлық мәліметтерді емес, тек оның бөлігін. Ол үшін GROUP BY сөзі арналған. Ол кестенің жолдарының жиынын бір немесе бірнеше бағандар мәндерінің (немесе олар қатысқан өрнектердің) теңдігі белгісі бойынша топтарға бөлуге мүмкіндік береді. GROUP BY сөзі WHERE сөзінен кейін орналасуы керек (егер ол жоқ болса, онда FROM сөзінен кейін).

GROUP BY сөзінің қатысуымен SELECT сөзі топтау сөзімен құрылған әр топқа қолданылады. Бұл жағдайда SELECT сөзінде көрсетілген агрегаттық функциялардың әрекеті де бүкіл қорытынды кестеге емес, тек әр топтың жолдары бойынша таралады. SELECT сөзінің тізіміндегі әр өрнек топ үшін тек жалғыз мән қабылдауы керек, яғни ол болуы мүмкін:

- тұрақты сан;
- топ бойынша аргументтің барлық мәндерін пайдаланатын және оларды бір мәнге (мысалы, суммаға) біріктіретін агрегаттық функциялар;
- GROUP BY сөзінде орналасқан сөзге ұқсас өрнек;
- жоғарыда келтірілген нұсқаларды біріктіретін өрнек.

GROUP BY сөзін қолданудың ең қарапайым түрі бір бағанның мәндері бойынша топтау болып табылады.

Сұраныс: баспалардың қалалар бойынша саны.

```
SELECT Publishing house, COUNT(*) AS «Баспалардың саны»
```

```
FROM Publishing house
GROUP BY Publishing house
```

Егер сұраныста WHERE мен GROUP BY сөздері қатар қолданылса, онда WHERE сөзінің шарттарын қанағаттандырмайтын жолдар топтау орындалғанша дейін шеттетіледі. Сондықтан топтау тек шартты қанағаттандыратын жолдар бойынша орындалады.

Сұраныс: «Глобус» атауымен жеткізушілердің қалалар бойынша саны.

```
SELECT Publishing house, COUNT(*) AS «Клиенттер саны»
FROM Publishing house
WHERE publish = «Глобус»
GROUP BY Publishing house
```

SQL кестенің жолдарын және бағандарының бірнешеуін топтауға мүмкіндік береді. Бұл жағдайда бағандардың атаулары GROUP BY сөзінде үтір арқылы атап өтіледі.

Алынған топтардың ішінен жолдарды іріктеп алу үшін HAVING сөзі қолданылады. Ол WHERE қолданыстағы кестелер үшін атқаратын қызметті топтар үшін атқарады, және GROUP BY сөзінің қатысуымен ғана қолданылады. SELECT сөйлемінде WHERE, GROUP BY және HAVING сөздері келесі ретпен өңделеді.

– WHERE сөзімен онда көрсетілген шартты қанағаттандыратын жолдар іріктеледі;

– GROUP BY сөзі іріктелген жолдарды топтайды;

– HAVING сөзімен онда көрсетілген шартты қанағаттандыратын топтар іріктеледі.

HAVING сөзінде көрсетілетін шарт әр топтың барлық жолдарына әмбебап болуы керек. Сондықтан HAVING сөзінде бағандар мен агрегаттық функциялардың атауларын қолдану ережелері, GROUP BY сөзінің қатысуымен SELECT сөзі үшін қолдану сияқты. Бұл HAVING сөзінде салыстыру операнды ретінде тек топтастырылатын немесе агрегаттық функцияларды қолдану мүмкіндігін білдіреді.

Сұраныс: Баспалар саны 6-дан көп қалалар саны.

```
SELECT Publishing house
FROM Publishing house
GROUP BY Publishing house
HAVING COUNT(*)>6
```

Зертханалық жұмыс №2 арналған тапсырмалардың нұсқалары

Жалпы ережелер.

Create Database операторының көмегімен DB_Books деп аталатын жаңа мәліметтер қорын құру, онда №1 зертханалық жұмыстың үлгісі бойынша

Create table операторының көмегімен атап өтілген кестелерді құру. Бағдарлама файлы StudentTegі_3M_1_DB_Books атауымен сақтаңыз. SQL Server Management Studio утилитында «Сұраныс құру» батырмасының көмегімен әр сұраныс бойынша жеке бағдарлама құру, оларды дискте келесідей сақтау: StudentTegі_3M_2_тапсырма№_. Бағдарламаның өзіне тапсырма мәтінін түсіндірме түрінде көшіру. Барлық орындалған сұраныстарды бір файлда сақтауға болады. SELECT операторының жұмысын тексеру үшін алдын ала INSERT операторының көмегімен DB_Books МҚ-ның барлық кестесін бірнеше жазбамен толтыратын бағдарлама құрыңыз, бағдарламаларды StudentTegі_3M_2_Insert атауымен сақтаңыз.

Тапсырмалар нұсқаларының тізімі

Нұсқа	Жаттығулар нөмірлерінің тізімі												
1	1	6	11	16	21	26	31	36	41	46	51	56	61
2	2	7	12	17	22	27	32	37	42	47	52	57	62
3	3	8	13	18	23	28	33	38	43	48	53	58	63
4	4	9	14	19	24	29	34	39	44	49	54	59	64
5	5	10	15	20	25	30	35	40	45	50	55	60	65
6	6	11	16	21	26	31	36	41	46	51	56	61	1
7	7	12	17	22	27	32	37	42	47	52	57	62	2
8	8	13	18	23	28	33	38	43	48	53	58	63	3
9	9	14	19	24	29	34	39	44	49	54	59	64	4
10	10	15	20	25	30	35	40	45	50	55	60	65	5
11	2	6	12	16	22	26	32	36	42	46	52	56	62
12	1	5	11	15	21	25	31	35	41	45	51	55	61
13	3	7	13	17	23	27	33	37	43	47	53	57	63
14	4	8	14	18	24	28	34	38	44	48	54	58	64
15	5	9	15	19	25	29	35	39	45	49	55	59	65
16	6	13	26	33	36	43	46	53	56	63	1	11	17
17	7	18	27	28	37	38	47	48	57	58	2	4	10
18	8	9	18	19	28	29	38	39	48	49	58	59	60
19	9	14	29	24	39	34	49	44	59	54	61	65	3
20	10	12	20	22	30	32	40	42	50	52	60	62	8

Іріктеу.

1. Books кестесінен кітаптар туралы барлық мәліметтерді таңдау және кітаптың коды бойынша нәтижені іріктеу (Code_book өрісі).

2. Books кестесінен кітаптар кодын, атауын және беттер санын таңдау (Code_book, Title_book және Pages өрістері), нәтижені кітаптар атауы

бойынша (Title_book өрісі өсу ретімен) және Pages өрісі бойынша (кему ретімен) іріктеу.

3. Deliveries кестесінен жеткізушілер тізімін таңдау (Name_Delivery, Phone және INN өрістері), нәтижесін INN өрісі бойынша іріктеу (кему реті бойынша).

Өрістердің ретін өзгерту.

1. Бағандар реті келесідей болатындай Deliveries кестесінен барлық өрістерді таңдау: Name_Delivery, INN, Phone, Address, Code_delivery.

2. Бағандар реті келесідей болатындай Publishing house кестесінен барлық өрістерді таңдау: Publish, City, Code_publish.

Кейбір өрістерді екі кестеден таңдау.

1. Books кестесінен кітаптар атауын және беттер санын таңдау (Title_book және Pages өрістері), ал Authors кестесінен кітаптың сәйкес авторын таңдау (Name_author өрісі).

2. Books кестесінен кітаптар атауын және беттер санын таңдау (Title_book және Pages өрістері), ал Deliveries кестесінен кітаптың сәйкес жеткізушісін таңдау (Name_delivery өрісі).

3. Books кестесінен кітаптар атауын және беттер санын таңдау (Title_book және Pages өрістері), ал Publishing house кестесінен кітаптың сәйкес баспа орталығын және басып шығарылған қаласын (Publish және City өрістері).

Нақты сәйкес келмеудің шарты.

1. Жеткізушілер анықтамасынан (Deliveries кестесі) компания атын, телефонын және ЖСН таңдау (Name_company, Phone және INN өрістері), компания атауы “ТОО” деп басталатындарын (Name_company өрісі).

2. Books кестесінен кітаптар атауын және беттер санын таңдау (Title_book және Pages өрістері), ал Authors кестесінен атауы «Мемуарлар» сөзінен басталатын кітаптың сәйкес авторын таңдау (Name_author өрісі).

3. Authors кестесінен «Мом» мәнінен басталатын авторлардың тегін, атын, әкесінің атын таңдау (Name_author өрісі).

Бір өріс мәндерінің нақты сәйкес келмеуі.

1. Publishing house кестесінен «Мәскеу» қаласында орналаспаған (City өріс бойынша шарт) баспалардың атауларының тізімін таңдау (Publish өрісі).

2. Books кестесінен «Питер-софт» баспасынан басқа (Publishing house кестесіндегі Publish өрісі) кез келген баспалармен шығарылған кітаптар атауын таңдау (Title_book өрісі).

Мәндер аралығы бойынша жазбаларды таңдау (Between).

1. Authors кестесінен туған күндері 01.01.1940 – 01.06.1990 аралығында жатқан (Birthday өрісі) авторлардың тегін, атын, әкесінің атын таңдау (Name_author өрісі).

2. 12.03.2003 – 15.06.2003 аралығында сатып алынған (Purchases кестесіндегі Date_order өрісі бойынша шарт) кітаптардың атауларын (Books кестесіндегі Title_book өрісі) және санын (Purchases кестесіндегі Amount өрісі) шығару.

3. Books кестесінен беттерінің саны 200-300 аралығында жатқан (Pages өрісі бойынша шарт) кітаптардың атауларының тізімін (Title_book өрісі) және беттер санын (Pages өрісі) шығару.

4. Authors кестесінен тегі «В» және «Г» аралығындағы әріптерге басталатын (Name_author өрісі бойынша шарт) авторлардың тегін, атын, әкесінің атын шығару (Name_author өрісі).

(In) мәнінің аралығы бойынша жазбаларды таңдау.

1. Кодтары 3, 7, 9, 11 жеткізушілермен жеткізілген (Purchases кестесіндегі Code_Delivery өрісі бойынша шарт) кітаптардың атауларын (Books кестесіндегі Title_book өрісі) және санын (Purchases кестесіндегі Amount өрісі) шығару.

2. Books кестесінен «Питер-софт», «Алматы кітап», «Наука» баспаларынан шығарылған (Publishing house кестесіндегі Publish өрісі бойынша шарт) кітаптар атауларының тізімін шығару (Title_book өрісі).

3. Books кестесінен «Толстой Л.Н.», «Достоевский Ф.М.», «Пушкин А.С.» авторларымен жазылған (Authors кестесіндегі Name_author өрісі бойынша шарт) кітаптар атауларының тізімін шығару (Title_book өрісі).

Like қолдану арқылы жазбаларды таңдау

4. Authors кестесінен «Т» әріпіне басталатын авторлардың тізімін (Name_author өрісі) шығару.

5. Publishing house кестесінен атауында «софт» тіркесі бар баспалардың атауларын (Publish өрісі) шығару.

6. Deliveries кестесінен мәні «ский» болып аяқталатын компаниялар атауларын таңдау (Name_company өрісі).

Бірнеше шарт бойынша жазбаларды таңдау.

1. Жеткізушілердің кодтарын таңдау (Code_delivery өрісі), тапсырыс күнін (Date_order өрісі) және кітаптардың атауын (Title_book өрісі), егер тапсырыстағы кітаптар саны (Amount өрісі) 100 ден асса немесе бір кітаптың бағасы (Cost өрісі) 200-300 тг аралығында болса.

2. Авторлар кодтарын таңдау (Code_author өрісі), авторлардың атауларын (Name_author өрісі), сәйкес кітаптардың атауларын (Title_book өрісі), егер баспа коды (Code_Publish өрісі) 10 мен 25 аралығында және кітаптағы беттер саны (Pages өрісі) 120 дан артық болса.

3. Publishing house кестесінен атауы (Title_book өрісі) «Естеліктер» сөзінен басталатын және баспа қаласы (City өрісі) – «Алматы» болатын кітаптар шығарылған баспалардың тізімін (Publish өрісі) шығару.

Көпкестелік сұраныстар (JOIN қолдану арқылы екі кестеден таңдау, үш кестеден таңдау).

1. Жеткізуші компаниялардың атауларының тізімін (Name_company өрісі) және 01.01.2012 мен 31.12.2013 аралығында (Date_order өрісі бойынша шарт) жеткізген кітаптарының атауларын (Title_book өрісі) шығару.

2. «Мир» баспасында (Publish өрісі бойынша шарт) шығарылған кітаптардың авторларының тізімен (Name_author өрісі) шығару.

3. «Алатау» баспасының (Publish өрісі бойынша шарт) кітаптарын жеткізетін жеткізушілердің (Name_comprany өрісі) тізімін шығару.

4. «Атамекен» жеткізушісімен (Name_comprany өрісі бойынша шарт) жеткізілген кітаптардың авторларының (Name_author өрісі) тізімін және атауларын (Title_book өрісі) шығару.

Есептеу.

1. Аттас кітаптар топтамасының жиынтық бағасын (Amount және Cost өрістерін қолдану) және әр жеткізудегі кітаптардың атауларын (Title_book өрісі) шығару.

2. Әр кітаптың бір бетін басып шығару құнын (Cost және Pages өрістерін қолдану) және сәйкес кітаптардың атауларын (Title_book өрісі) шығару.

3. Авторлардың туған күнінен бастап жасын (Birthday өрісін қолдану) және сәйкес авторлардың атын (Name_author өрісі) шығару.

Агрегаттық функцияларды қолдану арқылы қорытынды мәндерді шығару.

4. «Мектеп» орындаған (Name_comprany өрісі бойынша шарт) кітаптардың жеткізілуінің (Cost өрісін қолдану) ортақ бағасын шығару.

5. 01.01.2014 – 01.02.2015 аралығында (Date_order өрісі бойынша шарт) орындалған барлық жеткізулердің толық санын (Purchases кестесінен кез келген өрісті қолдану) шығару.

6. Бір жеткізудегі авторы «Пушкин» болып табылатын (Name_author өрісі бойынша шарт) кітаптардың орташа дана санын (Amount өрісін қолдану) және орташа бағасын (Cost өрісін қолдану) шығару.

7. Жеткізу бойынша барлық мәліметтерді (Purchases кестесінің барлық өрістері), сонымен қатар ортақ бағасы ең төмен (Amount және Cost өрістерін қолдану) кітаптың атауын (Title_book өрісі) шығару.

8. Жеткізу бойынша барлық мәліметтерді (Purchases кестесінің барлық өрістері), сонымен қатар ортақ бағасы ең жоғары (Amount және Cost өрістерін қолдану) кітаптың атауын (Title_book өрісі) шығару.

Өрістердің атауларын өзгерту.

1. Кітаптың атауын (Title_book өрісі), аттас кітаптар топтамасының (Amount және Cost өрістерін қолдану) жиынтық бағасын шығару, оларды 01.01.2002 – 01.06.2002 аралығындағы жеткізуге (Date_order өрісі бойынша шарт) жататын Itogo атаулы өрістің нәтижесіне енгізу.

2. Әр кітаптың бір бетін басып шығару құнын (Cost және Pages өрістерін қолдану) шығару, нәтижесін One_page атаулы өріске енгізу және сәйкес кітаптардың атауларын (Title_book өрісі) шығару.

3. «ТОО Луч» (Name_comprany өрісі бойынша шарт) орындаған кітап жеткізудің ортақ бағасын (Cost өрісін қолдану) шығару және нәтижесін Sum_cost атаулы өріске енгізу.

Айнымалыларды шарттарда қолдану.

1. Соңғы айдағы (Date_order өрісі бойынша шарт) келісімшарттардың тізімін (Purchases кестесінің барлық өрістері) шығару.

2. Қолданушы көрсеткен (Birthday өрісін қолданатын шарт) жастан аз жастағы автордардың (Name_author өрісі) тізімін шығару.

3. Қолданушы сұранысында көрсетілген саннан (Amount өрісін қолданатын шарт) кем сатып алынған кітаптардың (Title_book өрісі) тізімін шығару.

Кесте атауларының орынына айнымалыларды қолдану.

1. Жеткізуші компаниялардың атауларын (Name_company өрісі) және олар жеткізген кітаптардың атауларын (Title_book өрісі) шығару.

2. «Просвещение», «Питер софт», «Мектеп» баспаларында (Publish өрісі бойынша шарт) шығарылған кітаптардың авторларының (Name_author өрісі) тізімін шығару.

3. 1050 тг құнымен (Cost өрісі) жеткізілген кітаптардың баспаларының (Name_company өрісі) тізімін шығару.

Нәтижені нұсқамаға таңдау.

1. Кітаптар атауларының (Title_book өрісі) тізімін және әр кітаптағы беттер санын (Pages өрісі) шығару, нәтижесін Temp1 атаулы нұсқамасына орнату.

2. Жеткізуші компаниялардың атауларының (Name_company өрісі) тізімін шығару және нәтижесін Temp2 атаулы нұсқамасына орнату.

3. Авторлар тізімін (Name_author өрісі) шығару және нәтижесін Temp3 атаулы нұсқамасына орнату.

Сұраныспен бірге функцияларды қолдану.

1. Беттерінің саны (Pages өрісі) кестедегі барлық кітаптар беттерінің орташа санынан артық кітаптардың (Title_book өрісі) тізімін шығару.

2. Жасы (Pages өрісі) кестедегі барлық авторлардың орташа жасынан кіші авторлардың (Name_author өрісі) тізімін шығару.

3. Беттерінің саны (Pages өрісі) кестедегі беттері ең аз кітаптарға тең кітаптардың (Title_book өрісі) тізімін шығару.

Сұраныстарда бар болу кванторын қолдану.

1. Кітаптары көтерме саудамен («опт» Type_Purchases өрісінен) алынған баспалардың (Publish өрісі) тізімін шығару.

2. Кітаптары Books кестесінде жоқ авторлардың (Name_author өрісі) тізімін шығару.

3. «ТОО Глобус» жеткізушісімен (Name_company өрісі бойынша шарт) жеткізілген кітаптардың (Title_book өрісі) тізімін шығару.

Update мәліметтерін өңдеу операторы.

1. Books кестесінде Pages өрісінің мәнін 300-ге өзгерту, егер автор коды (Code_author өрісі) =56 және кітаптың атауы (Title_book өрісі) = «Абай жолы» болса.

2. Deliveries кестесінде Address өрісінің мәнін «мәлімет жоқ» мәніне өзгерту, егер өрістің мәні бос болса.

3. Purchases кестесінде бағаны (Cost өрісі) 20%-ға өсіру, егер тапсырыстар соңғы айдың (Date_order өрісі бойынша шарт) ішінде жасалған болса.

Insert мәліметтерді өңдеу операторы.

1. Purchases кестесіне жаңа жазба қосу, сатылым коды (Code_purchase өрісі) автоматты түрде бір бірлікке өсуі керек, ал сатып алу түріне (Type_purchase өрісі) «опт» мәні енгізілуі керек.

2. Books кестесіне жаңа жазба қосу, мұнда кілттік өрістің орынына кестедегі максимал кодтан автоматты түрде бір бірлікке ұлғайған код (Code_book өрісі) қою керек, кітап атауының орынына (Title_book өрісі) «Ғылым. Техника. Инновация» деп жазу.

3. Publish house кестесіне жаңа жазба қосу, мұнда кілттік өрістің орынына кестедегі максимал кодтан автоматты түрде бір бірлікке ұлғайған код (Code_publish өрісі) қою керек, қала атауының (City өрісі) орынына – «Қарағанды», баспа (Publish өрісі) орынына «Наука» деп жазу керек.

Delete мәліметтерді өңдеу операторы

4. Purchases кестесінен тапсырыстағы кітаптарының саны (Amount өрісі) 0-ге тең барлық жазбаларды өшіру.

5. Authors кестесінен Name_author өрісінде авторлардың аты жоқ жазбаларды өшіру.

6. Deliveries кестесінен ЖСН көрсетілмеген (INN өрісі бос) барлық жазбаларды өшіру.

Зертханалық жұмыс №3

Тақырыбы: T-SQL орнатылған тілінің көмегімен бағдарламалауды меңгеру

Жұмыс мақсаты: Transact SQL орнатылған тілінің көмегімен MS SQL-дегі бағдарламалаудың негізгі принциптерімен танысу.

Жұмыстың мазмұны:

1) MS SQL Server анықтамалық жүйесіндегі (Books Online утилиті) командалардың синтаксисін белгілеудің ережелерімен танысу.

2) Transact SQL-де бағдарламаларды жазу ережелерін үйрену.

3) Идентификаторларды құру ережелерін, айнымалыларды және олардың типтерін жариялау ережелерін үйрену.

4) Циклдармен және тармақтармен жұмыс істеуді үйрену.

5) Table және Cursor типті айнымалылармен жұмыс істеуді үйрену.

6) Барлық мысалдарды қарастыру, нәтижелерді анализдеу және оларды орындау.

7) Нұсқа бойынша индивидуалдық тапсырмаларды орындау.

Жұмысты орындауға түсініктеме.

Бағдарламалауды меңгеру үшін, №1-2 зертханалық жұмысында жасалған, DB_Books атымен сақталынған мәліметтер қорының мысалын қолданамыз. Мысалдар мен тапсырмаларды орындау барысында ДҚ, кестелердің және жобаның басқа да нысандарының аттарының сәйкестігіне назар аударыңыз.

1 кесте – Арнайы белгілеулер және қарапайым операторлар

Белгі	Арналуы	Белгі	Арналуы	Белгі	Арналуы
*	Көбейту белгісі	" "	Бұларда жолақтық мән жазылады егер SET_QUOTED_IDENTIFIER OFF	!>, !<	Қарағанда артық емес, кем емес
-	Кеміту белгісі	« «	Бұларда жолақтық мән жазылады	<, >	кем, артық
%	Екі санды бөлуден қалған қалдық	<>	Тең емес	.	Ата-аналық және бағынышты нысандарды айырады
+	Қосу немесе конкатенация (екі жолды бір жолға біріктіруі) белгісі	[]	Жақшаның аналогы, бұларға идентификаторлардың атын жазуға болады, егер құрамында пробел кездессе	<=, >=	Кем немесе тең, артық немесе тең
=	Теңдік немесе салыстыру белгісі	/	Бөлу белгісі	@	Айнымалының атының алдына қойылады
!=	Тең емес	@@	Жүйелік функцияға нұсқайды	-- /* */	Коментарий

Идентификаторлар - бұл T-SQL тілінде жазылған бағдарлама ішінде сілтенуге болатын нысандардың аттары. Бірінші символ ағылшын тілінің әріптерінен немесе “_”, “@”, “#” тұра алады. Қалғандары қосымша сандардан және «\$».

Идентификатордың аты резервтелген сөзбен сәйкес келмеуі қажет.

SET_QUOTED_IDENTIFIER ON параметрі орнатылған кезде идентификаторларды шектеу үшін квадраттық сонымен қатар бірлік жақшаны да қолдануға, ал жолдық мәндерді тек бірлік жақшада қолдануға болады (үнсіз келісім бойынша).

Егер орнатылған параметрді SET_QUOTED_IDENTIFIER OFF режимінде қолдансақ, онда идентификаторларды шектеуші ретінде квадраттық жақшаны қолдануға, ал жолдық мәндерді бірлік жақшада немесе қос тырнақша ішінде көрсетуге болады.

Айнымалылар аралық мәндерді сақталынған процедурада және функцияда сақтау үшін қолданылады. Барлық айнымалылар локалды болып саналады. Айнымалы аты @ басталуы қажет.

Айнымалыны жариялау.

MS SQL Server белгіленуіндегі синтаксис:

```
DECLARE @аты_айнымалы1 айнымалы_типi, ... , @аты_айнымалыN  
айнымалы_типi
```

Егер айнымалы типі көлемді көрсетуді болжайтын болса, айнымалыларды жариялау үшін келесідей синтаксис қолданылады:

```
DECLARE @аты_айнымалы1 айнымалы_типi (көлемi), ... ,  
@аты_айнымалыN айнымалы_типi (көлемi)
```

Мысал:

```
DECLARE @a INT, @b numeric(10,2)
```

```
DECLARE @str CHAR(20)
```

Айнымалыларға мәндерді тағайындау және мәндерді экранға шығару

```
SET көмегімен тағайындау – қарапайым тағайындау, синтаксис SET  
@аты_айнымалы = мән.
```

Мысал:

```
DECLARE @a INT, @b numeric(10,2)
```

```
SET @a = 20
```

```
SET @b = (@a+@a)/15
```

```
SELECT @b -- экранға нәтижені шығару
```

SELECT көмегімен тағайындау – сұраныс нәтижесін айнымалыға орнату. Егер сұраныстың атқарылу нәтижесінде бірде-бір жол қайтарылмаса, онда айнымалының мәні өзгермейді, яғни ескі күйі сақталады.

Мысал:

```
DECLARE @a INT
```

```
SELECT @a = COUNT(*) FROM Authors
```

Мысал:

```
DECLARE @str CHAR(30)
```

```
SELECT @str = name FROM Authors
```

Бұл мысалда айнымалыға сұраныс нәтижесінің соңғы мәні орнатылады.

SET және SELECT кілттік сөздерінің үйлесімі

Мысал:

```
DECLARE @a INT
```

```
SET @a = (SELECT COUNT(*) FROM Authors)
```

Уақытпен және датамен жұмыс

Оператор SET DATEFORMAT dmy | ymd | mdy дата компоненттерінің орналасу ретін береді.

Мысал:

```
SET DATEFORMAT dmy
```

```
DECLARE @d DateTime
```

```
SET @d = «31.01.2005 13:23:15»
```

```
SET @d = @d+1
```

```
SELECT @d
```

Уақыттық кестені TABLE типті айнымалы арқылы құру

DECLARE арқылы жақша ішінде кесте бағандарын, олардың типтерін, өлшемдерін, үнсіз келісім бойынша қойылатын мәндерді, сонымен қатар тип индекстерін көрсете отырып жарияланады.

PRIMARY KEY немесе UNIQUE

Мысал:

```
DECLARE @mytable TABLE(id INT, myname CHAR(20) DEFAULT «Есімді еңгізіңіз»)
```

```
INSERT INTO @mytable(id) VALUES (1)
```

```
SELECT * FROM @mytable
```

Мысал:

```
DECLARE @mytable TABLE(id INT, myname CHAR(20) DEFAULT «Есімді еңгізіңіз»)
```

```
INSERT @mytable SELECT Code_publish, City FROM Publishing_house
```

```
SELECT * FROM @mytable
```

Айнымалылардың типін түрлендіру

CAST функциясы көрсетілген типке түрлендірілген мәнді қайтарады:

CAST(@айнымалы немесе мән AS талап етілген мәліметтер типі)

Мысал:

```
DECLARE @d DateTime, @str char(20)
```

```
SET @d = «31.01.2005 13:23:15»
```

```
SET @str = CAST(@d AS Char(20))
```

```
SELECT 2str
```

CONVERT функциясы берілген формат бойынша көрсетілген типке түрлендірілген мәнді қайтарады. Өз қалауыңыз бойынша қосымша үйрену.

Операторлық жақшалар

```
BEGIN
```

/* бұларға ДҚ нысандарының құрылымын өзгертетін бұйрықтарды орналастыруға болмайды. Операторлық жақшаларда кем дегенде бір оператор

болуы қажет. Поливарианттық тармақтарды, шарттық және циклдық конструкцияларды құру үшін қажет.

```
*/  
END
```

IF шарттық операторы

Синтаксис:

IF шарт

Операторлар жиыны1

ELSE

Операторлар жиыны2

Мысал:

```
DECLARE @a INT  
DECLARE @str CHAR(30)  
SET @a = (SELECT COUNT(*) FROM Authors)  
IF @a >10  
BEGIN  
SET @str = «Количество авторов больше 10»  
SELECT @str
```

```
END
```

```
ELSE
```

```
BEGIN
```

```
SET @str = «Количество авторов = « + str(@a)
```

```
SELECT @str
```

```
END
```

WHILE циклы

Синтаксис:

WHILE Шарт Операторлар жиыны1

BREAK

Операторлар жиыны2

CONTINUE

BREAK және CONTINUE конструкциясы міндетті емес.

Циклды мәжбүрлі түрде тоқтатуға болады, егер оның денесінде BREAK бұйрығын орындаса. Егер дененің құрамындағы барлық бұйрықтардың орындалуын күтпестен циклды басынан бастау қажет болса, CONTINUE бұйрығын орындау керек.

Мысал:

```
DECLARE @a INT
```

```
SET @a = 1
```

```
WHILE @a <100
```



```

BEGIN
PRINT @a – айнымалы мәнін экранға шығару
IF (@a>40) AND (@a<50)
BREAK—циклдан кейінгі бірінші команданы орындау және шығу
ELSE
SET @a = @a+rand()*10
CONTINUE
END
PRINT @a

```

Курсорды жариялау

CURSOR- бұл сұраныстың орындалуының нәтижесі болып табылатын жолақтар жиыны. Бір уақыт кезеңінде тек бір ғана жолаққа (ағымдағы) қол жетімді, курсор бойынша орына ауыстыруға және қарапайым мәліметтерге қол жеткізуге болады. Курсорды жариялау кезінде мәліметтердің уақытша көшірмесі жасалады, ол өз кезегінде tempdb ДҚ сақталады.

Динамикалық курсор – курсордағы мәліметтерді өзгертуге болады.

Статикалық курсор - курсордағы мәліметтерді өзгертуге болмайды.

Курсорды жариялаудың стандартты жолы, келесідей синтаксис

MS SQL Server:

```

DECLARE cursor_name [ INSENSITIVE ] [ SCROLL ] CURSOR FOR
select_statement
[ FOR { READ ONLY | UPDATE [ OF column_name [ ,...n ] ] } ]

```

Курсорды жариялаудың мысалы:

```

DECLARE MyCursor1 CURSOR FOR (select*from Authors) /* MyCursor1
атты курсорды жарияладық, ол өз кезегінде авторлар жайлы барлық
ақпаратты қамтиды. Курсорда бірінші жазбадан соңғы жазбаға дейін төмен
қарай қозғалу қажет. Курсор динамикалық болып табылады. */

```

```

DECLARE MyCursor1 INSENSITIVE CURSOR FOR (select*from
Authors) /* MyCursor1 атты курсорды жарияладық, ол өз кезегінде авторлар
жайлы барлық ақпаратты қамтиды. Курсорда бірінші жазбадан соңғы жазбаға
дейін тек төмен қарай қозғалу қажет. Курсор статикалық болып табылады. */

```

```

DECLARE MyCursor1 SCROLL CURSOR FOR (select*from Authors) /*
MyCursor1 атты курсорды жарияладық, ол өз кезегінде авторлар жайлы
барлық ақпаратты қамтиды. Курсорда кез келген бағытта қозғалуға болады.
Курсор динамикалық болып табылады. */

```

```

DECLARE MyCursor1 INSENSITIVE SCROLL CURSOR FOR
(select*from Authors) /* MyCursor1 атты курсорды жарияладық, ол өз кезегінде
авторлар жайлы барлық ақпаратты қамтиды. Курсорда кез келген бағытта
қозғалуға болады. Курсор статикалық болып табылады. */

```

```

DECLARE MyCursor1 CURSOR FOR (select*from Authors) FOR READ
ONLY

```

```

/* MyCursor1 атты курсорды жарияладық, ол өз кезегінде авторлар
жайлы барлық ақпаратты қамтиды. Курсорда бірінші жазбадан соңғы жазбаға

```

дейін тек төмен қарай қозғалу қажет. Курсор динамикалық болып табылады. Мәліметтер тек оқылу үшін ғана қол жетімді */

```
DECLARE MyCursor1 CURSOR FOR (select*from Authors) FOR  
UPDATE /* MyCursor1 атты курсорды жарияладық, ол өз кезегінде авторлар  
жайлы барлық ақпаратты қамтиды. Курсорда бірінші жазбадан соңғы жазбаға  
дейін тек төмен қарай қозғалу қажет. Курсор динамикалық болып табылады.  
Курсордың мәліметтерін өзгертуге болады */
```

Курсормен жұмыс үшін арналған операторлар
Курсордың мәліметтеріне бармас бұрын, оны жариялағаннан кейін ашу қажет.

OPEN операторының MS SQL Server-дегі синтаксисі:

```
OPEN{ {[GLOBAL] cursor_name} | cursor_variable_name }
```

Мысал:

```
DECLARE MyCursor1 CURSOR FOR (select*from Authors) OPEN  
MyCursor1
```

Курсормен жұмыс аяқталған соң, оны жабу қажет. Курсор өзі жасалған триггер немесе процедура аймағында ары қарайғы қолдануға қол жетімді болады.

CLOSE операторының MS SQL Server-дегі синтаксисі:

```
CLOSE { { [ GLOBAL ] cursor_name } | cursor_variable_name }
```

Мысал:

```
DECLARE MyCursor1 CURSOR FOR (select * from Authors) OPEN  
MyCursor1
```

-- мұнда курсормен жұмыс жасау операторлары

```
CLOSE MyCursor1
```

Егер де курсормен жұмыс жасалмайтын болса, оны жою және айнымалыны босату қажет.

DEALLOCATE операторының MS SQL Server-дегі синтаксисі:

```
DEALLOCATE { { [ GLOBAL ] cursor_name } | @cursor_variable_name }
```

Мысал:

```
DECLARE MyCursor1 CURSOR FOR (select * from Authors) OPEN  
MyCursor1
```

-- мұнда курсормен жұмыс жасау операторлары

```
CLOSE MyCursor1
```

```
DEALLOCATE MyCursor1
```

FETCH- курсор жазбалары бойынша қозғалыс және ағымдағы жазбадағы мәліметтерді көрсетілген айнымалыларға ауыстыру операторы.

FETCH операторының MS SQL Server-дегі синтаксисі:

Fetch

```
[ [ Next | Prior | First | Last | Absolute { n | @nvar } ]
```

```

| Relative { n | @nvar }
]
from
]
{ { [ Global ] cursor_name } | @cursor_variable_name } [ INTO
@variable_name [ ,...n ] ]

```

Мысал:

```

DECLARE MyCursor1 SCROLL CURSOR FOR (select * from Authors)
DECLARE @i bigint, @s char(20), @d smalldatetime
OPEN MyCursor1
FETCH FIRST FROM MyCursor1 INTO @i, @s, @d PRINT @i
PRINT @s PRINT @d CLOSE MyCursor1
DEALLOCATE MyCursor1

```

@@FETCH_STATUS – берілген функция ағымдағы курсордың басының немесе соңының белгісін анықтайды. Функция келесідей мәндердің біреуін қабылдайды: 0 – курсордың аралығындамыз, соңында емес; 1 – бірінші жазбаның аймағынан жоғары қарай (в никуда) шығу әрекеті; 2 - соңғы жазбаның аймағынан төмен қарай (в никуда) шығу әрекеті.

Мысал:

```

DECLARE MyCursor1 SCROLL CURSOR FOR (select * from Authors)
DECLARE @i bigint, @s char(20), @d smalldatetime
OPEN MyCursor1
FETCH FIRST FROM MyCursor1 INTO @i, @s, @d WHILE
@@FETCH_STATUS = 0
BEGIN
FETCH NEXT FROM MyCursor1 INTO @i, @s, @d PRINT @i
PRINT @s PRINT @d
END
CLOSE MyCursor1 DEALLOCATE MyCursor1

```

Өздігінен орнатылған функциялар

SQL-мен жұмыс кезінде қолданушыға қол жетімді өздігінен орнатылған функцияларды келесідей топтарға бөлуге болады:

- математикалық функция;
- жолақтық функция;
- уақытпен және мерзіммен жұмыс жасау функциясы;
- конфигурациялау функциясы;
- қауіпсіздік жүйесінің функциясы;
- метамәліметтерді басқару функциясы;
- статистикалық функциялар.

2 кесте - Жолақтық айнымалылармен жұмыс жасау үшін функцияларды қолдану

Функция атауы	Функция атқаратын әрекет
ASCII	Жолақтың сол жақ символының ASCII кодын қайтарады
CHAR	ASCII коды бойынша символды қайтарады
CHARINDEX	Жолақшаның жолаққа енуі басталатын символдың реттік номерін анықтайды.
DIFFERENCE	Жолақтардың сәйкестік көрсеткішін қайтарады
LEFT	Жолақтың басынан берілген символдар санын қайтарады
LEN	Жолақтың ұзындығын қайтарады
LOWER	Жолақтың барлық символдарын төменгі регистрға ауыстырады
LTRIM	Жолақтың басындағы бос орындарды өшіреді
NCHAR	Код бойынша Unicode символын қайтарады
PATINDEX	Көрсетілген шаблон бойынша жолақтағы жолақшаны іздеуді орындайды
REPLACE	Жолақшаның берілген мәнге енуін ауыстырады
QUOTENAME	Жолақты Unicode форматына конвертациялайды
REPLICATE	Жолақты белгілі бір мөлшерге тирождауды орындайды
REVERSE	Кері ретте жазылған символдары бар жолақты қайтарады
RIGHT	Берілген мөлшердегі символдарды жолақтың соңынан қайтарады
RTRIM	Жолақтың соңындағы бос орындарды өшіреді
SOUNDEX	Жолақтың дыбысының кодын қайтарады
SPACE	Көрсетілген бос орындар санын қайтарады
STR	Сандық типтегі мәндерді символдың форматқа конвертацияны орындайды
STUFF	Берілген мөлшердегі символдарды өшіріп, жаңа жолақшамен алмастырады
SUBSTRING	Берілген символдан көрсетілген ұзындықтағы жолақшаны жолақ үшін қайтарады
UNICODE	Жолақтың сол жағының Unicode –кодын қайтарады
UPPER	Жолақтың барлық символдарын жоғарғы регистрға ауыстырады

Сандармен жұмыс үшін функцияны қолдану.

3 кесте - Математикалық функцияларға қысқаша шолу

Функцияның атауы	Функциямен атқарылатын әрекет
ABS	Санның абсолюттік мәнін анықтайды
ACOS	Арккосинусты есептеп шығарады

ASIN	Арксинусты есептейді
ATAN	Арктангенсты есептейді
ATN2	Квадраттарды есептей отырып арктангенсты есептейді
CEILING	Жоғары қарай домалақтауды орындайды
COS	Бұрыштың косинусын есептейді
COT	Бұрыштың котангенсын қайтарады
DEGREES	Бұрыштың мәнін радианнан градусқа түрлендіреді
EXP	Экспонентаны қайтарады
FLOOR	Төмен қарай домалақтауды орындайды
LOG	Натуралды логарифмді есептейді
LOG10	Ондық логарифмді есептейді
PI	«Пи» мәнін қайтарады
POWER	Санды дәрежеге шығарады
RADIANS	Бұрыштың мәнін градустан радианға түрлендіреді
RAND	Кездейсоқ санды қайтарады
ROUND	Домалақтауды берілген дәлдікпен орындайды
SIGN	Санның таңбасын анықтайды
SIN	Бұрыштың синусын есептейді
SQUARE	Санды квадратқа шығаруды орындайды
SQRT	Квадраттық түбірді есептейді
TAN	

Уақыт/Мерзім типтерімен жұмыс жасау үшін функцияны қолдану.

4 кесте - Уақытпен және мерзіммен жұмыс жасауын функцияларға қысқаша шолу

Функция атауы	Функциямен атқарылатын қызметтер
DATEADD	Мерзімге берілген күнді, айды, сағаттарды қосады және т.б
DATEDIFF	Көрсетілген екі мерзім арасындағы айырымды қайтарады
DATENAME	Мерзімнің көрсетілген бөлігін ерекшелейді және оны символдық форматта қайтарады
DATEPART	Мерзімнің көрсетілген бөлігін ерекшелейді және оны сандық форматта қайтарады
DAY	Көрсетілген мерзімнен санды қайтарады
GETDATE	Ағымдық жүйелік уақытты қайтарады
ISDATE	Мерзімді енгізуге болатын екі түрлі форматқа сәйкестігін тексереді

MONTH	Көрсетілген мерзімнен айдың мәнін қайтарады
YEAR	Көрсетілген мерзімнен жылдың мәнін қайтарады
MINUTE	Көрсетілген мерзімнен/уақыттан минуттың мәнін қайтарады
HOURL	Көрсетілген мерзімнен/уақыттан сағаттың мәнін қайтарады
SECOND	Көрсетілген мерзімнен/уақыттан секундтың мәнін қайтарады

Зертханалық жұмыс №3 арналған тапсырмалардың нұсқасы

Жалпы мәліметтер.

Операторлардың немесе функциялардың жұмысы жайлы көбірек ақпарат алу үшін MS SQL Server құрамындағы Books Online утилитін қосып, «Указатель» бөлімінен ізделініп отырған элементтің кілттік сөзін тереміз.

Тапсырманы орындау үшін екінші зертханалық жұмыстағы нұсқалар мен нөмірлердің тізіміне назар аудару қажет.

Арнайы белгілеулер мен Transact SQL –дегі қарапайым операторлар.

1. Суреттелген ET QUOTED_IDENTIFIER қондырғысының жұмысын тексеру

2. Суреттелген SET DATEFIRST қондырғысының жұмысын тексеру.

Айнымалыларды жариялау.

1. Ақша типті Perem1 айнымалысын және сандық типті Perem2 айнымалысын 8 бүтін және 2 бөлшек бөлігімен жариялау.

3. Ұзындығы 100-ге тең жолақ типті Perem1 айнымалысын және ұзын бүтін типті Perem2 айнымалысын жариялау.

4. Максималды ұзындығы 1000-ға тең динамикалық жолақ типті Perem1 айнымалысын және бүтін типті Perem2 айнымалысын жариялау.

5. ұзындығы 30-ға тең жолақ типті Perem1 айнымалысын және сандық типті, бүтін бөлігі 10-ға және бөлшек бөлігі 3-ке тең Perem2 айнымалысын жариялау.

6. Уақыт/мерзім типті Perem1 айнымалысын және 0-ден 255-ке дейінгі диапазондағы сандық типті Perem2 айнымалысын жариялау.

Айнымалыларға мән беру және мәндерді экранға шығару.

1. Сатып алынған кітаптардың орташа бағасын есептеп шығару (SELECT сұранысының көмегімен) және нәтижені 123,34-ке көбейту, ол мәнді өз кезегінде жеке айнымалыға сақтап, айнымалы мәнін экранға шығару.

2. Сатып алынған барлық кітаптардың суммарлық бағасын есептеу және нәтижені айнымалыға сақтап, мәнін экранға шығару.

3. Ақпарат кітапшасындағы кітаптардың санын есептеу және нәтижені айнымалыға сақтап, мәнін экранға шығару.

4. Авторлар жайлы ақпарат кітапшасындағы туылған мерзімі ең кіші авторды анықтау және нәтижені айнымалыға сақтап, мәнін экранға шығару.

SET және SELECT кілттік сөздерінің үйлесімі.

1. Кітаптарды тасымалдаушылардың санын анықтау және нәтижені айнымалыға орналастыру.

2. Кітаптарды сатып алуға кеткен сумманы анықтау және нәтижені айнымалыға орналастыру.

3. Кітаптарды сатып алатын кестедегі кітаптың орта бағасын анықтау және нәтижені айнымалыға орналастыру.

5. Тізімдегі ең қымбат кітаптың бағасын анықтау және нәтижені айнымалыға орналастыру.

Уақытпен және мерзіммен жұмыс жасау.

Уақыт/мерзім типті Date1 айнымалысын анықтау. Оған dd.mm.yyyy форматында 31.12.2006 мерзімді орнату.

TABLE типті айнымалы арқылы уақыттық кестені құру.

1. TEMP атты және ұзын бүтін, уақыт/мерзім, жолақ типті аймақтары бар локалдық кестені құру. Оған екі жазбаны енгізі және нәтижені экранға шығару.

2. TEMP атты және ұзын бүтін, жолақ және «кез келген нәрсені еңгізіңіз» үнсіз келісім бойынша, қаржы типті аймақтары бар локалдық кестені құру. Оған екі жазбаны енгізі және нәтижені экранға шығару.

3. TEMP атты және бүтін, динамикалық жолақ, үнсіз келісім бойынша мәні «1» тең бит типті аймақтары бар локалдық кестені құру. Оған екі жазбаны енгізі және нәтижені экранға шығару.

4. TEMP атты және уақыт/мерзім, ұзын бүтін, жолақ типті аймақтары бар локалдық кестені құру. Оған екі жазбаны енгізі және нәтижені экранға шығару.

5. TEMP атты және уақыт/мерзім, автоөсімі бар ұзын бүтін, динамикалық жолақ типті аймақтары бар локалдық кестені құру. Оған екі жазбаны енгізі және нәтижені экранға шығару.

Айнымалы типін түрлендіру.

1. FLOAT, CHAR, TINYINT типті айнымалыны жариялау. Сәйкесінше типтерге мәнді беру. FLOAT, CHAR, TINYINT типті айнымалыларын сәйкесінше INT, DATETIME, BIT-ке түрлендіру және нәтижені экранға шығару.

2. INT, DATETIME, BIT типті айнымалыны жариялау. Сәйкесінше типтерге мәнді беру. INT, DATETIME, BIT типті айнымалыларын сәйкесінше FLOAT, CHAR, TINYINT -ке түрлендіру және нәтижені экранға шығару.

3. NUMERIC, VARCHAR, DATETIME типті айнымалыны жариялау. Сәйкесінше типтерге мәнді беру. NUMERIC, VARCHAR, DATETIME типті айнымалыларын сәйкесінше FLOAT, CHAR, BIGINT -ке түрлендіру және нәтижені экранға шығару.

4. BIT, NVARCHAR, DATETIME типті айнымалыны жариялау. Сәйкесінше типтерге мәнді беру. BIT, NVARCHAR, DATETIME типті айнымалыларын сәйкесінше FLOAT, INT, BIGINT -ке түрлендіру және нәтижені экранға шығару.

IF шарттық конструкциясы.

1. Deliveries кестесіндегі тасымалдаушылардың санын анықтау. Егер де кестеде 2 ден 5 ке дейін болса, ештемені хабарламалау, басқа жағдайда «Кестеде ... тасымалдаушы» хабарламасын шығару (көп нүкте орнына нақты санын қою).

2. Сатылымдар кестесіндегі сатылған кітаптардың суммасын анықтау. Егер де кестеде 1000 нан 5000 ға дейін болса, ештемені хабарламалау, басқа жағдайда «Сатылым суммасы = ...» хабарламасын шығару (көп нүкте орнына сумма санын қою).

3. Сатылымдар кестесіндегі сатылған кітаптардың орташа бағасын анықтау. Егер де кестеде 1000 нан 5000 ға дейін болса, ештемені хабарламалау, басқа жағдайда «сауданың орташа бағасы = ...» хабарламасын шығару (көп нүкте орнына сауданың орташа бағасын қою).

4. Сатылымдар кестесіндегі сатылған кітаптардың минималды бағасын анықтау. Егер де кестеде 200 ден 300 ге дейін болса, ештемені хабарламалау, басқа жағдайда «сауданың минималды бағасы = ...» хабарламасын шығару (көп нүкте орнына сауданың минималды бағасы қою).

WHILE циклі.

1. Authors кестесіндегі жазбалар санын анықтау. Жазбалар 15-тен кем болған кезге дейін, кестеге жазбаны циклдық түрде қосуды кілттік жолақтың автоматтық өсуімен орындау, ал автордың есімінің орнына «Автор белгісіз» мәнін қою.

2. Баспалар кестесіндегі жазбалар санын анықтау. Жазбалар 20-дан кем болған кезге дейін, кестеге жазбаны циклдық түрде қосуды кілттік жолақтың автоматтық өсуімен орындау, ал баспа атының орнына «белгісіз» мәнін қою.

3. Тасымалдаушылар кестесіндегі жазбалар санын анықтау. Жазбалар 17-ден кем болған кезге дейін, кестеге жазбаны циклдық түрде қосуды кілттік жолақтың автоматтық өсуімен орындау, ал тасымалдаушы атының орнына «белгісіз» мәнін қою.

Курсорды жариялау.

1. Books кестесінің деректері бойынша Good_book, Title_book өрістері бар статикалық курсорын құру.

2. Тасымалдаушылар кестесінің (Deliveries кестесі) деректері бойынша Name_delivery, Name_company өрістері бар динамикалық курсорын құру.

3. Books және Authors кестелерінің деректері бойынша Good_book, Title_book, Name_author өрістері бар статикалық курсорын құру.

4. Books және Publishing_house кестелерінің деректері бойынша Good_book, Title_book, Publish өрістері бар статикалық курсорын құру.

Курсормен жұмыс жасау үшін қажет операторлар.

1. Deliveries кестесінің деректері бойынша Code_delivery, Name_delivery, өрістері бар динамикалық курсорын құру. 3-ші жазбаның мәндерін шығару.

2. Ағымдағы db_books ДҚ құру. Курсорға Purchases кестесінің мәліметтерін орналастыру. Purchases кестесінің барлық жазбаларын қойып

шығу. Cost және Amount өрістерінің мәндерінің көбейтінділерін суммалау және нәтижені Sum_table айнымалысына сақтау, нәтижені экранға шығару.

3. Authors және Books кестесінің мәліметтері бойынша статикалық курсорды жариялау. 5-ші жазбаны шығару.

Функцияларды жолақтық айнымалылармен жұмыс үшін қолдану.

1. Базалық мәтін жеке файлда нұсқа бойынша берілген. Бұл тапсырмалар блогын орындау үшін, сіз құрған бағдарламаның басында varchar типті айнымалыны жариялаңыз және оның мәні ретінде тапсырмада түзетілетін немесе анализденетін кез келген базалық тексты бар жолақты беріңіз.

2. Мәтіндегі артық бос орындарды өшіріңіз. Артық дегеніміз бос орыннан (пробелдан) кейін орналасқан орындар. Түзетілімдер санын анықтаңыз.

3. Базалық мәтіндегі келесідей әріптердің қанша рет кездескенін анықтаңыз: «а», «в», «и», «п».

4. «е», «о» әріптерінің кездесуінің проценттік үлесін анықтаңыз, егер де әрбір осы әріптің кездесулерінің суммарлық үлесі 100 % немесе е% + о% кездесулерінің қосындысы 100 %.

5. Машиналық тексттарды жазудың жалпы ережесі бойынша .,!?:; белгіленулерінің алдына бос орын қойылмайды, алайды олардан кейін міндетті түрде бос орын болуы қажет. Артық бос орындарды өшіріңіз. Түзетілімдер санын анықтаңыз.

6. Машиналық тексттарды жазудың жалпы ережесі бойынша .,!?:; белгіленулерінің алдына бос орын қойылмайды, алайды олардан кейін міндетті түрде бос орын болуы қажет. Жетпейтін бос орындарды қойыңыз. Түзетілімдер санын анықтаңыз.

7. Бастапқы мәтіннен екінші сөйлемді табыңыз және оны Perem айнымалысына қайтару, сонымен қатар экранға бастапқы мәтінді және табылған сөйлемді шығару.

8. Бастапқы мәтіннен 2, 4, 6, 8 сөздерін өшіру.

9. Бастапқы мәтіннен 3, 5, 7, 10 сөздерін өшіру.

10. Бастапқы мәтіндегі «а» әрпінің орнына «АА» қою.

11. Бастапқы мәтіндегі «е» және «о» әріптерінің орнына «ББ» қою

12. Базалық мәтіндегі бірінші және соңғы сөздердің орындарын ауыстырыңыз.

Сандармен жұмыс үшін функцияны қолдану.

Формуласының (1) мәнін шығару, айнымалыларына кез келген мәнді беруге болады.

Уақыт/мерзім типтерімен жұмыс жасайтын функциялар.

Экранға ағымдағы айдың атын және уақытты шығару. Purchases кестесінің Date_order өрісіне бірдей жеткізілген тауарлардың келу датасын жазу. Дата 12.03.2000 тең.

Ағымдағы датаны және уақытты жеке-жеке бөліктерге бөлу және экранға кезек-кезегімен келесідей ретпен шығарыңыз: «Бүгін: Күн = ..., Ай = ..., Жыл = ..., Сағат = ..., Минут = ..., Секунд = ...»

Perem айнымалысында сақталынған бастапқы мәтінде «уақыт» сөзінен кейін ағымдағы уақытты қойыңыз. Нәтижені қайтадан сол Perem айнымалысына сақтаңыз және экранға шығарыңыз.

Зертханалық жұмыс №4

Жұмыс тақырыбы: MICROSOFT SQL SERVER-де сақталынатын процедураны құру.

Жұмыс мақсаты: – ДҚ серверінде сақталынатын процедураны құруды және қолдануды үйрену.

Жұмыстың мазмұны:

1) Барлық мысалдарды қарастыру, SQL Server Management Studio утилитіндегі орындалуының нәтижелерін анализдеу. Ағымдағы ДҚ-да жасалған процедураның бар-жоқтығын тексеру.

2) Зертханалық жұмыс барысында барлық мысалдар мен тапсырмаларды орындау.

3) Нұсқа бойынша индивидуалдық тапсырмаларды орындау.

Зертханалық жұмысты орындауға түсініктеме

Сақталынатын процедураларды бағдарламалауды меңгеру үшін DB_Books атты, №1 зертханалық жұмыста құрылған деректер базасын мысал ретінде қолданамыз. Мысалдар мен тапсырмаларды орындау барысында ДҚ, кестелердің және жобаның басқа да нысандарының аттарының сәйкестігіне назар аударыңыз.

Сақталынатын процедуралар бір немесе бірнеше SQL операторынан немесе функциядан тұратын және базада компиляцияланған күйде сақталынатын бұйрықтардың жиынынан тұрады.

Сақталынатын процедуралардың типтері.

Жүйелік *сақталынатын процедуралар* әртүрлі административтік қызметтерді атқаруға арналған. Серверді административтеуге қатысты әрекеттердің барлығы дерлі олардың қызметі арқылы іске асырылады. Жүйелік *сақталынатын процедуралар* жүйелік кестелермен жұмыс жасауды қамтамасыз ететін интерфейс болып табылады деуге болады. Жүйелік *сақталынатын процедуралар* sp_ префиксіне ие, жүйелік деректер қорында сақталады және кез келген басқа деректер қорының контекстында шақырыла алады.

Қолданушылық *сақталынатын процедуралар* белгілі бір әрекетті іске асырады. *Сақталынатын процедуралар*- деректер қорының толық бір мүшесі.

Осының нәтижесінде әрбір *сақталынатын процедуралар* белгілі бір нақты деректер қорында сақталады, және сол жерде іске асырылады.

Уақытша *сақталынатын процедуралар* белгілі бір уақытқа ғана пайда болады, кейін сервермен автоматты түрде өшіріледі. Олар локалдық және глобалдық болып бөлінеді. Локалдық уақытша *сақталынатын процедуралар* тек өздері жасалынған байланыстан ғана шақырыла алады. Мұндай процедураны жасау кезінде оған # символымен басталатын ат беру қажет. Барлық уақытша нысандар сияқты, осы типті *сақталынатын процедуралар* қолданушы өшкен кезде, қайтадан қосу немесе сервердің тоқтауы кезінде автоматты түрде өшіріледі. Глобалдық уақытша *сақталынатын процедуралар* дәл осындай процедурасы бар сервердің кез келген байланысы үшін қол жетімді. Оны анықтау үшін бар болғаны ## символынан басталынатын ат берсе жеткілікті. Бұл процедуралар қайтадан қосу немесе сервердің тоқтауы кезінде, сонымен қатар өздері құрылған контекстінің байланысының жабылуы кезінде өшіріледі.

Сақталынатын процедураларды құру, өзгерту.

Сақталынатын процедураны құру келесідей қиындықтарды шешуді қарастырады: ену рұқсатын жобалау. Сақталынатын процедураны құру кезінде ол нысандарға оны құрған қолданушы сияқты ену құқығына ие болатынын ескеру қажет; сақталынатын процедуралар кірістік және шығыстық параметрлерге ие бола алады; сақталынатын процедураның кодын жобалау. Процедура коды SQL-дің кез келген бұйрықтардың ретін қамти алады, сонымен қатар басқа да ; сақталынатын процедураны шақыру.

SQL SERVER-дегі белгілену бойынша жаңа сақталынатын процедураны құру немесе құрылған процедураны өзгерту операторының синтаксисі:

```
{CREATE | ALTER } PROC[EDURE] процедура_аты [нөмір]
[{@параметр_аты мәліметтер_типi } [VARYING ] [=default][OUTPUT] ][...n]
[WITH { RECOMPILE | ENCRYPTION | RECOMPILE,
ENCRYPTION }}
[FOR REPLICATION]
```

AS

sql_оператор [...n]

Берілген команданың параметрлерін қарастырайық.

sp_, #, ## префикстерін қолдана отырып жасалғалы жатқан процедураның жүйелік немесе уақытша екендігін беруге болады. Команданың синтаксисінен көріп отырғанымыздай жасалынып жатқан процедураның иесінің атын, сонымен қатар ол орналастырылуы қажет деректер базасының атын көрсетуге рұқсат берілмейді. Осылайша, жасалып жатқан сақталынатын процедураны нақты бір деректер базасында орналастыру үшін, осы деректер қорының контекстінде CREATE PROCEDURE командасын орындау қажет.

Сақталынатын процедура денесінен сол деректер қорының басқа нысандарына сұраныс жасау үшін қысқартылған есімдерді қолдануға болады, яғни деректер қорының атын көрсетпестен. Ал егер басқа деректер қорындағы нысандарға сұраныс жасау үшін міндетті түрде деректер атын көрсету қажет.

Жасалынып жатқан сақталынатын процедурада кірістік және шығыстық мәліметтерді жіберу үшін, параметрлердің аты @ символынан басталуы қажет. Бір сақталынатын процедура ішінде үтірмен бөле отырып, бірнеше параметрді көрсетуге болады. Процедура денесінде аттары осы процедураның параметрлерінің аттарымен сәйкес келетін локалдық айнымалылар қолданылмауы қажет.

Сақталынатын процедураның параметрлерінің мәліметтер типін анықтау үшін SQL-дің кез келген мәліметтер типі жарай береді, соның ішінде қолданушымен анықталғандар да. Алайда CURSOR мәліметтер типі сақталынатын процедураның тек шығыстық параметрі ретінде қолданыла алады, яғни OUTPUT кілттік сөзін көрсете отырып.

OUTPUT кілттік сөзінің жазылуы, сәйкесінше параметр сақталынатын процедурадан мәліметтерді қайтаруға арналғандығын білдіреді. Алайда, параметр мәліметтерді сақталынатын процедураға жіберуге сәйкес келмейді деген сөз емес. OUTPUT кілттік сөзінің көрсетілуі, серверге сақталынған процедурадан шығар кезде, процедураны параметр ретінде шақырған кезде көрсетілген ағымдағы локалдық айнымалының мәнін қабылдауды міндеттейді. Айта кетерлік жайт, OUTPUT кілттік сөзінің көрсетілуі кезінде сәйкесінше параметрдің мәні процедураны шақыру кезінде тек локалдық айнымалы көмегімен ғана берілуі мүмкін. Қарапайым параметрлерге қолдануға болатын кез келген өрнектерді және константаларды қолдану рұқсат етілмейді.

VARYING кілттік сөзі CURSOR типті OUTPUT параметрімен бірге қолданылады. Ол шығыстық параметр ретінде нәтижелендіруші көбейтінді болатынын анықтайды.

DEFAULT кілттік сөзі сәйкесінше параметр үнсіз келісім бойынша қабылдайтын мәнді білдіреді. Осылайша, процедураны шақыру кезінде сәйкесінше параметрдің нақты мәнін көрсетпеуге де болады.

Сервер сұраныстың орындалу жоспарын және компиляцияланған кодты кәштейтіндіктен, процедураны келесі рет шақырғанда дайып тұрған мәндер қолданылады. Алайда кейбір жағдайларда процедураның кодын қайтадан компиляциялауға тура келеді. RECOMPILE кілттік сөзінің қолданылуы жүйеден, процедураны әрбір шақырған сайын оның орындалу жоспарын құруды талап етеді.

FOR REPLICATION параметрі мәліметтерді репликациялаған кезде және құрылып отырған сақталынатын процедураны статья ретінде публикацияға қосатын кезде қажет.

ENCRYPTION кілттік сөзінің қолданылуы серверден сақталынатын процедураның кодын шифрлауды талап етеді, ол өз кезегінде сақталынатын

процедураның жұмысын іске асырып отырған авторлық алгоритмдердің қолданылуынан қорғанысты қамтамасыз етеді.

AS кілттік сөзі сақталынатын процедураның өз денесінің басында орналастырылады. Процедура денесінде SQL-дің барлық дерлік командалары қолданыла, транзакциялары жариялана, блокировкалар орнатылып және басқа сақталынатын процедуралар шақырыла алады. Сақталынатын процедурадан RETURN командасының көмегімен іске асыруға болады.

Сақталынатын процедураны өшіру

DROP PROCEDURE {процедура_аты} [...n]

Сақталынатын процедураны іске асыру.

Сақталынатын процедураны іске асыру үшін келесідей команда іске асырылады: [[EXEC [UTE] процедура_аты [;нөмір] [[@параметр_аты=]{мәні | @айнымалы_аты}][OUTPUT]][DEFAULT]][...n]

Егер де сақталынатын процедураны шақыру пакеттегі жалғыз команда болмаса, онда EXECUTE командасының жазылуы міндетті. Сонымен қатар, бұл команда басқа процедура денесінен немесе триггерден процедураны шақыру үшін қажет.

Процедураны шақыру кезінде OUTPUT кілттік сөзін қолдануға тек OUTPUT кілттік сөзі бар процедураны құру кезінде жарияланған параметрлер үшін ғана рұқсат етіледі.

Егер де процедураны шақыру кезінде параметр үшін DEFAULT кілттік сөзі қолданылатын болса, онда үнсіз келісім бойынша берілген мәндер орнатылады. DEFAULT сөзі үнсіз келісім бойынша мәндері берілген параметрлер үшін ғана рұқсат етіледі.

EXECUTE командасының синтаксисінен көріп отырғанымыздай, процедураны шақыру кезінде параметрлер аты көрсетілмеуі мүмкін. Бірақ бұл жағдайда, қолданушы параметрлер үшін мәндерді олардың процедураны құрған кездегі ретіне сәйкес қылып нұсқауы қажет. Параметрге үнсіз келісімді мәнді орнату үшін, ретін көрсету кезінде бос орын қалдырып кетуге болмайды. Ал егер, үнсіз келісім бойынша мәні берілген параметрді алып тастау қажет болса, процедураны шақыру кезінде параметр атын анық көрсету жеткілікті. Сонымен қатар, осы тәсілмен параметрлерді және олардың мәндерін еркін ретпен көрсетуге болады.

Айта кетер жайт, процедураны шақыру кезінде мәндерімен бірге параметр аттары, немесе тек мәндері параметр атынсыз көрсетіледі. Оларды комбинациялауға болмайды.

Сақталынатын процедурада RETURN-ды қолдану.

Процедурадан көрсетілген шарт бойынша кез келген нүктеден шығуға, сонымен қатар процедураның орындалу нәтижесін процедураның орындалу сапасы мен оның орындалу дұрыстығын бағалай алатындай санмен жіберуге мүмкіндік береді.

Процедураны параметрсіз құру үлгісі.

```
CREATE PROCEDURE Count_Books AS
Select count(Code_book) from Books
```

Go

Тапсырма 1. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны EXEC Count_Books командасының көмегімен іске қосыңыз. Нәтижесін тексеріңіз.

Кірістік параметрі бар процедураны құру мысалы:

```
CREATE PROCEDURE Count_Books_Pages @Count_pages as Int
AS
Select count(Code_book) from Books WHERE Pages>=@Count_pages
Go
```

Тапсырма 2. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны EXEC Count_Books_Pages 100 командасының көмегімен іске қосыңыз. Нәтижесін тексеріңіз.

Кірістік параметрлері бар процедураны құру мысалы:

```
CREATE PROCEDURE Count_Books_Title @Count_pages as Int, @Title
AS
Char(10)
AS
Select count(Code_book) from Books WHERE Pages>=@Count_pages
AND Title_book LIKE @Title
Go
```

Тапсырма 3. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны EXEC Count_Books_Title 100, «П%» командасының көмегімен іске қосыңыз. Нәтижесін тексеріңіз.

Кірістік параметрлері және шығыстық параметрі бар процедураны құру мысалы:

```
CREATE PROCEDURE Count_Books_Itogo @Count_pages Int, @Title
Char(10) , @Itogo Int OUTPUT
AS
Select @Itogo = count(Code_book) from Books WHERE
Pages>=@Count_pages AND Title_book LIKE @Title
Go
```

Тапсырма 4. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны келесідей командалар жиыны арқылы іске қосыңыз:

```
Declare @q As int
EXEC Count_Books_Itogo 100, «П%», @q output select @q
```

Нәтижесін тексеріңіз.

Кірістік параметрлері және RETURN бар процедураны құру мысалы:

```
CREATE PROCEDURE checkname @param int
```

```
AS
```

```
IF (SELECT Name_author FROM authors WHERE Code_author = @param)  
= «Пушкин А.С.»
```

```
RETURN 1
```

```
ELSE
```

```
RETURN 2
```

Тапсырма 5. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны келесідей командалар жиыны арқылы іске қосыңыз:

```
DECLARE @return_status int
```

```
EXEC @return_status = checkname 1
```

```
SELECT «Return Status» = @return_status
```

Нәтижесін тексеріңіз.

Purchases кестесінде кілттік жолдың мәнін 2 есе үлкейтуге арналған параметрлерсіз процедураны құру мысалы:

```
CREATE PROC update_proc
```

```
AS
```

```
UPDATE Purchases SET Code_purchase = Code_purchase*2
```

Процедура ешқандай мәлімет қайтармайды.

Тапсырма 6. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны келесідей команда арқылы іске қосыңыз:

```
EXEC update_proc
```

Белгілі бір автор туралы барлық ақпаратты алу үшін кірістік параметрі бар процедура мысалы:

```
CREATE PROC select_author @k CHAR(30)
```

```
AS
```

```
SELECT * FROM Authors WHERE name_author=@k
```

Тапсырма 7. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны келесідей командалар арқылы іске қосыңыз:

```
EXEC select_author «Пушкин А.С.» немесе select_author @k=«Пушкин  
А.С.» немесе EXEC select_author @k=«Пушкин А.С.»
```

Purchases кестесінде кілттік жолдың мәнін қажетті есе (үнсіз келісім бойынша 2 есе) үлкейтуге арналған кірістік параметрі және үнсіз келісім бойынша мәні бар процедураны құру мысалы:

```
CREATE PROC update_proc @p INT = 2
```

```
AS
```

```
UPDATE Purchases SET Code_purchase = Code_purchase *@p
```

Процедура ешқандай мәлімет қайтармайды.

Тапсырма 8. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны келесідей командалар арқылы іске қосыңыз:

```
EXEC update_proc 4 немесе  
EXEC update_proc @p = 4 немесе  
EXEC update_proc -- үнсіз келісім бойынша орнатылған мән  
қолданылатын болады.
```

Кірістік және шығыстық параметрлері бар процедураны құру мысалы. Белгілі бір уақыт аралығындағы жасалған тапсырыстар санын анықтауға арналған процедураны құру.

```
CREATE PROC count_purchases  
@d1 SMALLDATETIME, @d2 SMALLDATETIME, @c INT OUTPUT  
AS  
SELECT @c=count(Code_purchase) from Purchases WHERE Date_order  
BETWEEN @d1 AND @d2  
SET @c = ISNULL(@c,0)
```

Тапсырма 9. Берілген процедураны DB_Books деректер қорының Stored Procedures бөлімінде SQL server Management Studio утилиті арқылы құрыңыз. Оны келесідей командалар арқылы іске қосыңыз:

```
DECLARE @c2 INT  
EXEC count_purchases «01-jun-2006», «01-jul-2006», @c2 OUTPUT  
SELECT @c2
```

Зертханалық жұмыс №4 арналған тапсырмалар нұсқасы

Жалпы мағлұмат.

SQL Server Management Studio утилитінде кодқа арналған («Сұраныс құру» батырмасы) жаңа парақшаны құру. Use операторының көмегімен құрылған DB_Books деректер қорын бағдарламалы түрде активті ету. Create procedure операторларының көмегімен сақталулы процедураны құру, сонымен қатар өз-бетінше процедура атын анықтау. Әрбір процедура екінші зертханалық жұмыста жасалған бір-бір SQL сұранысты орындайты болады. SQL сұраныстардың кодын, олардың ішінде сұраныс орындалып отырған аймақтардың мәндерін жібере алатындай етіп өзгерту қажет.

Мысалы, №2 зертханалық жұмыстағы бастапқы тапсырма мен сұраныс:
/* Жеткізушілер анықтамасынан (Deliveries кестесі) компания аты
(Name_company жолы) «ОАО МИР» болатын компаниялардың аттарын,
телефондарын және ИИН-дерін таңдау.

```
SELECT Name_company, Phone, INN FROM Deliveries
```



```

WHERE Name_company = «ОАО МИР»
*/ --Бұл жұмыста келесідеу процедура құрылатын болады:
CREATE PROC select_name_company @comp CHAR(30)
AS
SELECT Name_company, Phone, INN FROM Deliveries WHERE
Name_company = @comp
--процедураны іске қосу үшін келесідей команда қолданылады:
EXEC select_name_company «ООО МИР»

```

Бағдарлама файлын ФамилияСтудента_ЛАб_4 атымен сақтаңыз. SQL Server Management Studio-да сақталулы процедуралар бөлімінде DB_Books ДҚ процедуралардың бар-жоқтығын тексеру.

Тапсырмалар тізімі.

SQL Server Management Studio утилитінде жаңа бағдарламаны құру. №1 зертханалық жұмыста жасалған индивидуалды деректер қорын Use операторын қолдана отырып бағдарламалы түрде активті ету. Create procedure операторының көмегімен сақталулы процедураны құру, сонымен қатар процедура атын өз бетінше анықтау. Әрбір процедура бір-бір SQL сұранысты орындайды, олар өз кезегінде нұсқа бойынша тапсырмалар түрінде келтірілген.

Бағдарлама файлын Студен тегі_Зерт_4_№ нұсқасы атымен сақтаңыз. SQL Server Management Studio-ның индивидуалды ДҚ-ның сақталулы процедуралар бөлімінде процедуралардың бар-жоқтығын тексеру.

Нұсқа 1

1. Тым болмағанда бір баласы бар қызметкерлер тізімін шығару.
2. Көрсетілген уақыт мерзімінде сыйлық алған балалар тізімін шығару.
3. Кәмелеттік жасқа жетпеген балалары бар ата-аналар тізімін шығару.
4. Көрсетілген бағадан қымбат тұратын сыйлықтар туралы ақпаратты шығару, мерзім бойынша сорттау.

Нұсқа 2

1. Көрсетілген типтегі құралдар тізімін шығару.
2. Жөндеуден өткен құралдар санын және көрсетілген шеберде жөнделген құралдардың жалпы жөнделу соммасын шығару.
3. Құралдардың иелерінің тізімін және олардың қанша рет шебер көмегіне жүгінгендігін шығару, сонымен қатар кемуі бойынша сорттау.
4. Разрядтары көрсетілен мөлшерден артық немесе жұмысқа тұрған мерзімі белгілі бір мөлшерден кем шеберлер тізімін шығару.

Нұсқа 3

1. Жапырақтары көрсетілген типке жататын гүлдер тізімін шығару.
2. Гүлдерді белгілі бір мөлшерден артық соммаға сатқан сатылымдардың кодының тізімін шығару.

3. Көрсетілген сатылым коды бойынша сатылым датасын, соммасын, сатушының және гүлдің атын шығару.

4. Биіктігі көрсетілген мөлшерден артық немесе гүлдер тұрған гүлдердің сортын және гүлдердің аттарының тізімін шығару.

Нұсқа 4

1. Көрсетілген қолдану тәсіліне сәйкес келетін дәрілер тізімін шығару.

2. Белгілі бір дәрінің сатылымы көрсетілген мөлшерден артық болған датаны шығару.

3. Көрсетілген мөлшерден артық жеткізілу коды бойынша жеткізілу датасын, соммасын, жеткізушінің ТАЖ (ФИО) және дәрі атын шығару.

Нұсқа 5

1. Көрсетілген лауазымдағы қызметкерлер тізімін шығару.

2. Көрсетілген себеп бойынша қолданыстан алынған құралдар тізімін шығару.

3. Көрсетілген уақыт аралығында қолданыстан алынған құралдардың жеткізілу уақытын, құралдардың атын, жауапты адамның ФИО және қолданыстан алыну датасын шығару.

4. Көрсетілген типтегі құралдардың немесе жеткізілу уақыты белгілі бір мөлшерден артық құралдардың тізімін шығару.

Нұсқа 6

1. Салмағы көрсетілген мөлшерден артық болатын тағамдар тізімін шығару.

2. Аттарында көрсетілген сөздің фрагменті бар азық-түлік (продукт) тізімін шығару.

3. Тағам коды бойынша қажетті азық-түлік көлемін және аттарын, тағамның атын шығару.

4. Құрамындағы көмірсутек мөлшері немесе калориясын белгілі бір мөлшерден артық тағамның атын және дайындалу кезеңдерін шығару.

Нұсқа 7

1. Көрсетілген лауазымдағы қызметкерлер санын шығару.

2. Құрамында көрсетілген сөздің фрагменті бар құжаттар тізімін шығару.

3. Көрсетілген уақыт периодында тіркелген тіркелушінің АТЖ (ФИО), құжат типін, тіркелу датасын және тіркелуші құжат тапсырып отырған ұйым атын шығару.

4. Белгілі бір құжат типімен тіркелген құжаттардың немесе тіркелу датасын белгілі бір мөлшерден артық құжаттар тізімін шығару.

Нұсқа 8

1. Жұмыстан шығу себебі көрсетілген типке сәйкес келетін қызметкерлер тізімін шығару.

2. Тіркелу датасы көрсетілген периодқа сәйкес келетін құжаттар тізімін шығару.

3. Белгілі бір периодта тіркелген құжаткердің АТЖ, жұмыстан шығу себебін, тіркелу уақытын шығару.

4. Құжат коды қажетті диапазонға жататын тіркелген құжаттар тізімін шығару.

Нұсқа 9

1. Көрсетілген типке сәйкес демалыс алған қызметкерлер тізімін шығару.

2. Тіркелу датасы көрсетілген периодқа сәйкес келетін құжаттар тізімін шығару.

3. Көрсетілген периодта тіркелген демалыс типін, тіркелу датасын, құжатты тапсырған қызметкердің АТЖ шығару.

4. Құжат коды көрсетілген диапазонға сәйкес келетін тіркелген құжаттар тізімін шығару.

5. Вывести список сотрудников, бравших отпуск указанного типа.

6. Вывести список документов с датой регистрации в указанный период.

Нұсқа 10

1. Көрсетілген лауазымдағы қызметкерлер тізімін шығару.

2. Құрамында көрсетілген сөз фрагменті бар құжаттар тізімін шығару.

3. Көрсетілген периодта тіркелген жіберушінің АТЖ және қандай ұйымға жібергендігін, тіркелу датасын, құжат типін шығару.

4. Көрсетілген құжат типіне сәйкес келетін немесе құжат коды белгілі мөлшерден кем тіркелген құжаттар тізімін шығару.

Нұсқа 11

1. Көрсетілген лауазымға тағайындалған қызметкерлер тізімін шығару.

2. Тіркелу датасы көрсетілген периодқа сәйкес келетін құжаттар тізімін шығару

3. Көрсетілген периодта тіркелген қызметкердің АТЖ, лауазымын, тіркелу датасын шығару.

4. Құжат коды көрсетілген диапазонда жатқан тіркелген құжаттар тізімін шығару.

Нұсқа 12

1. Көрсетілген типке жататын құралдар тізімін шығару.

2. Белгілі бір клиент уақытша қолданысқа алған құралдар тізімін шығару.

3. Құралдарды уақытша қолданысқа алған клиенттер тізімін және олардың қанша рет бұл қызметке жүгінгендігін шығару, сонымен қатар сол мөлшердің кемуі бойынша сорттау.

4. Адрестері бойынша сортталған клиенттер туралы ақпарат шығару.

Нұсқа 13

1. Көрсетілген типке сәйкес келетін құралдар тізімін шығару.
2. Белгілі бір қызметкер қолданыстан алып тастаған құралдар тізімін шығару.
3. Құрал типтері бойынша топтастырылған, қолданыстан алынған құралдардың санын шығару.
4. Жұмысқа қабылдану датасы белгілі бір мөлшерден артық қызметкерлер тізімін шығару.

Нұсқа 14

1. Жапырақтары көрсетілген типке жататын гүлдердің тізімін шығару.
2. Гүлдердің сатылым соммасы белгілі бір мөлшерден артық жеткізілім кодтарының тізімін шығару (қай жеткізілімдерден көбірек сатылғандығына байланысты).
3. Жеткізушінің коды бойынша жеткізілімдер датасын, соммасын, жеткізушінің және гүлдердің атын шығару.
4. Биіктігі белгілі бір мөлшерден артық немесе гүлдеп тұрған гүлдердің тізімін және сортын шығару.

Нұсқа 15

1. Нөмірлеріне белгілі периодта тұрақтаған клиенттердің тізімін шығару.
2. Әрбір клиенттің нөміріне төлеген жалпы соммасын шығару.
3. Белгілі бір периодта тіркелген клиенттің АТЖ, нөмір типін, тұрақтау датасын шығару.
4. Белгілі бір нөмір типтерінде тіркелген клиенттердің тізімін шығару.

Нұсқа 16

1. Көрсетілген типке сәйкес келетін құралдар тізімін шығару.
2. Белгілі бір клиент уақытша қолданысқа алған құралдар тізімін шығару.
3. Құралдарды уақытша қолданысқа алған клиенттер тізімін және олардың қанша рет бұл қызметке жүгінгендігін шығару, сонымен қатар сол мөлшердің кемуі бойынша сорттау.
4. Адрестері бойынша сортталған клиенттер туралы ақпарат шығару.

Нұсқа 17

1. Сатылып алу бағасы белгілі бір мөлшерден артық немесе гарантия мерзімі көрсетілген мерзімнен артық құндылықтар тізімін шығару.
2. Аттарының құрамында белгілі бір сөз фрагменті бар материалды құндылықтардың табылған жерінің тізімін шығару.
3. Коды белгілі бір диапазонға жататын құндылықтардың бағасының соммасын шығару.

4. Жұмысқа қабылдану уақыты белгілі бір диапазонға жататын, материалды жауапкершілігі бар қызметкерлердің тізімін шығару.

Нұсқа 18

1. Белгілі бір шебер атқарған жөндеу жұмыстарының тізімін шығару.
2. Аттарында белгілі бір сөз фрагменттерін кездесетін жөндеу жұмыстарының кезеңдерін шығару.
3. Жұмыстың коды белгілі бір диапазонға жататын жөндеу жұмыстарының кезеңдерінің бағасының соммасын шығару.
4. Жұмысқа қабылдану уақыты белгілі бір диапазонға жататын шеберлердің тізімін шығару.

Нұсқа 19

1. Белгілі бір қолдану тәлісі бар дәрілердің тізімін шығару.
2. Белгілі бір мөлшерден артық сатылған дәрілердің чек нөмірін шығару.
3. Көрсетілген чек нөмірі бойынша сатылым датасын, кассирдің аты-жөнін, дәрі атын шығару.
4. Қорапшадағы саны белгілі бір мөлшерден артық немесе дәрі коды белгілі бір мөлшерден кем дәрілердің тізімін және олардың өлшену мөлшерін шығару (шт, мл, гр және т.б).

Нұсқа 20

1. Белгілі бір лауазымдағы қызметкерлер тізімін шығару.
2. Құрамында келтірілген сөз фрагменті кездесетін құжаттар тізімін шығару.
3. Құжаттары көрсетілген периодта тіркелген орындаушының АТЖ, тіркелу датасын, құжат типін және орындалу фактісін (орындалды, орындалмады) шығару.
4. Көрсетілген құжат типіне жататын немесе құжат коды белгілі бір диапазонға жататын тіркелген құжаттар тізімін шығару.

Әдебиеттер тізімі

- 1 Интернет-институт информационных технологий.– www.intuit.ru.– Курс «Основы SQL».
- 2 Мамаев Е.В. Microsoft SQL Server 2000.– СПб.: БХВ-Петербург, 2005.– 1280 с.
- 3 Остринская Л.И., Семенова И.И., Дороболук Т.Б. Теория и практика работы с современными базами и банками данных: Учебное пособие. Омск: Изд-во СиБАДИ, 2005.– 250 с.
- 4 Семенова И.И. SQL стандарт в СУБД MS SQL SERVER, ORACLE, VFP И ACCESS: манипулирование данными. – Омск: Изд-во СиБАДИ, 2008.– 57 с.

Мазмұны

Кіріспе	3
1 Зертханалық жұмыс №1. Microsoft SQL Server Management Studio МҚБЖ-нің интерфейсі. MS SQL Server-де мәліметтер қорын құру. MS SQL МҚБЖ құрамына кіретін көмекші бағдарламалар.	4
2 Зертханалық жұмыс №2. Мәліметтермен айла-тәсілдер жасайтын Microsoft SQL Server операторлары.	20
3 Зертханалық жұмыс №3. T-SQL орнатылған тілінің көмегімен бағдарламалауды меңгеру	35
4 Зертханалық жұмыс №4. MICROSOFT SQL SERVER-де сақталынатын процедураны құру.	49
Әдебиеттер тізімі	61

Утепбергенов Ирбулат Туремуратович
Сауанова Клара Тогаевна
Сагындыкова Шолпан Назаровна

МӘЛІМЕТТЕР ҚОРЛАРЫН БАСҚАРУ ЖҮЙЕЛЕРІ

5B070200 - Автоматтандыру және басқару мамандығы студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Редактор Ж. Изтелеуова
Стандарттау бойынша маман Н.Қ. Молдабекова

Басуға _____ қол қойылды
Таралымы 50 дана.
Көлемі 3,8 есептік-баспа-табақ

Пішімі 60x84 1/16
Баспаханалық қағаз №1
Тапсырыс ____. Бағасы 1900 тенге

«Алматы энергетика және байланыс университеті»
коммерциялық емес акционерлік қоғамының
көшірмелі -көбейткіш бюросы
050013, Алматы, Байтұрсынұлы көшесі, 126