

ДӘРІС

Тақырыбы: “SQL реляциялық мәліметтер базасының тілі. Функциялар мен негізгі мүмкіндіктер”

"АЖ деректер базасы" **пәні**

"Ақпараттық жүйелер" **білім беру бағдарламасы**

Авторлары: АЕЖ каф. аға оқытушысы Б.М.Саданова
АЕЖ каф. аға оқытушысы Г.Б.Абилдаева

Дәріс мақсаты:

SQL тіл операторларына жалпы сипаттама беру және SQL-де дерекқорға негізгі сұраныстардың қалай жазылғанын көрсету .



Дәріс жоспары:

1. SQL даму тарихы
2. SQL құрылымы
3. SQL деректердің негізгі түрлері
4. Деректер базасын құру
5. Деректер қорының кестелерін құру (жою)
6. Индекстерді құру (жою)
7. Деректерді іріктеу

SQL даму тарихы

- **SQL (Structured Query Language)** — құрылымдалған сұрау тілі — реляциялық деректер базасымен жұмыс істеу бойынша стандартты сұрау тілі.
- SQL тілі реляциялық алгебрадан кейін пайда болды және оның прототипі 70-ші жылдардың соңында IBM Research-да жасалды.
- Болашақта бұл тіл көптеген коммерциялық ДҚБЖ-де қолданылды және оның кең таралуына байланысты біртіндеп реляциялық ДҚБЖ-де деректерді басқару тілдері үшін "де-факто" стандартына айналды.

SQL құрылымы

- Қазіргі уақытта көптеген реляциялық емес жүйелерде реляциялық деректерге қол жеткізу құралдары бар.
- Стандарттаудың мақсаты-әртүрлі ДҚБЖ арасындағы қосымшалардың *тасымалдануы*.
- Реляциялық алгебрадан айырмашылығы, тек дерекқорға сұрау амалдары ұсынылған, SQL-бұл толық тіл, ол тек сұрау амалдарын ғана емес, сонымен қатар **деректерді анықтау тіліне Data Definition Language (DDL)** сәйкес келетін мәлімдемелерді де қамтиды. Сонымен қатар, тілде дерекқорды басқаруға (басқаруға) арналған операторлар бар.

SQL құрылымы

- SQL интерактивті (сұраныстарды орындау үшін) және кірістірілген (қолданбалы бағдарламаларды құру үшін) ретінде қолданыла алады. Онда бар:
 - мәліметтерді анықтау бойынша ұсыныстар (мәліметтер базасын анықтау, кестелер мен индекстерді анықтау және жою);
 - деректерді таңдау бойынша ұсыныстар;
 - деректерді өзгерту туралы ұсыныстар (деректерді қосу, жою және өзгерту);
 - деректерді басқару ұсыныстары (деректерге қол жеткізу артықшылықтарын беру және жою, транзакцияларды басқару және т.б.).

SQL құрылымы

SQL осы сөйлемдерде орындайды:

- *арифметикалық есептеулер, мәтіндік жолдарды* өңдеу және арифметикалық өрнектер мен мәтіндердің мәндерін *салыстыру*;
- экранға шығару кезінде жолдар мен бағандарды *ретке келтіру*;
- пайдаланушыларға деректер базасында олардың көлемін арттырмай-ақ деректерге өз көзқарасын білдіруге мүмкіндік беретін *көріністер* (виртуалды кестелер) құру;
- сұрау бойынша көрсетілген кесте мазмұнын басқа кестеде есте сақтау (реляциялық *тағайындау* операциясы).
- *деректерді біріктіру*: деректерді топтастыру, Орташа, Сума, Максимум, Минимум, Элементтер саны және т. б. операцияларды қолдану

SQL құрылымы

SQL мәлімдемелерінің 3 тобы бар:

- DDL (Data Definition Language) операторлары - мәліметтер базасының объектілерін анықтау операторлары
- DML операторлары (Data Manipulation Language) - деректерді басқару операторлары
- Деректерді қорғау және басқару операторлары



DDL операторлары (Data Definition Language):

- CREATE DATABASE - деректер қорын құру
- DROP DATABASE - деректер қорын жою
- CREATE TABLE - кестені құру
- ALTER TABLE - кестені өзгерту
- DROP TABLE - кестені жою
- CREATE DOMAIN - доменді құру
- ALTER DOMAIN - доменді өзгерту
- DROP DOMAIN - доменді жою
- CREATE VIEW - ұсыныс құру
- DROP VIEW - ұсынысты жою



DML операторлары (Data Manipulation Language):

- SELECT - кестелерден жолдарды таңдау
- INSERT - кестеге жолдар қосу
- UPDATE - кестедегі жолдарды өзгерту
- DELETE - кестедегі жолдарды өшіру
- COMMIT - енгізілген өзгерістерді тіркеу
- ROLLBACK - енгізілген өзгерістерді кері қайтару



Деректерді қорғау және басқару операторлары:

- GRANT - пайдаланушыға артықшылықтар беру немесе нысандарды айла-шарғы жасау қосымшасы
- REVOKE - пайдаланушы артықшылықтарын немесе қосымшаларды болдырмау

SQL деректердің негізгі түрлері

- INTEGER - бүтін сан;
- SMALLINT - "қысқа бүтін";
- DECIMAL(p,q) - p саны бар ондық сан ($0 < p < 16$) және белгі; q - ондық нүктенің оң жағындағы сандар саны;
- FLOAT - 15 белгісі бар нақты сан;
- CHAR(n) - n символдардан тіркелген ұзындықтың символдық жолы ($0 < n < 256$);
- VARCHAR(n) - n таңбадан аспайтын ұзындықтың символдық жолы ($n > 0$ және әр түрлі ДҚБЖ-да әр түрлі, бірақ 4096 кем емес);

SQL деректердің негізгі түрлері

- DATE - арнайы команда белгілеген форматтағы күн (әдепкі қалпы бойынша mm/dd/yy);
- TIME - арнайы команда белгілеген форматтағы уақыт (әдепкі қалпы бойынша hh.mm.ss);
- DATETIME - күн мен уақыт комбинациясы;
- MONEY - ақша бірлігінің символын анықтайтын форматтағы ақша (\$, руб, ...) және оның орналасуы (суффикс немесе префикс), бөлшек бөлігінің дәлдігі және ақшалай мәндерді көрсету шарты.

SQL ұсыныстарында келесі белгілер қолданылуы мүмкін:

- жұлдызша (*) "барлық" белгілеу үшін - яғни, "анықтауды қанағаттандыратын барлық жағдайлар";
- шаршы жақшалар ([]) – осы жақшада жасалған конструкциялар міндетті емес (яғни, түсірілуі мүмкін);
- фигуралы жақшалар ({}) – бұл жақшаларға жасалған конструкциялар тұтас синтаксистік бірліктер ретінде қарастырылуы тиіс, яғни олар SQL синтаксисінде қолданылатын кәдімгі жақшаларды ауыстыра отырып, синтаксистік конструкцияларды талдау тәртібін анықтауға мүмкіндік береді;

SQL ұсыныстарында келесі белгілер қолданылуы мүмкін:

- көп нүкте (...) – тікелей оған дейінгі синтаксистік бірлік факультативті бір немесе одан көп рет қайталануы мүмкін екенін көрсетеді;
- тікелей сызық (|) – екі немесе одан да көп мүмкіндіктерді таңдау. Мысалы, **ASC|DESC** – ASC немесе DESC терминдерінің бірін таңдауға болатынын көрсетеді; егер таңдау элементтерінің бірі шаршы жақшада болса, ол әдепкі қалпы бойынша таңдалады (**[ASC]|DESC** бұл конструкцияның болмауы ASC таңдауы ретінде қабылданатынын білдіреді));
- үтірлі нүкте (;) – SQL ұсыныстарының аяқталатын элементі;



SQL ұсыныстарында келесі белгілер қолданылуы мүмкін:

- үтір (,) – тізім элементтерін бөлу үшін қолданылады;
- бос орындар () – SQL ұсыныстарының кез келген синтаксистік конструкциялары арасында көрнекілікті арттыру үшін енгізілуі мүмкін;
- бас қалың латын әріптері мен символдары – SQL тіл конструкцияларын жазу үшін қолданылады;
- кіші әріптер – пайдаланушы таңдаған нақты мәндермен ауыстырылуы тиіс конструкцияларды жазу үшін пайдаланылады;

Деректер базасын құру

CREATE DATABASE ұсынысы ДҚ құрады және көрсетілген құрылғыларда және мөлшерде транзакциялар журналы болуы мүмкін.

CREATE DATABASE деректер_қоры_аты ON [DEFAULT] деректер _қоры_құрылғы][= өлшемі][, деректер _қоры_құрылғы][= өлшемі]]...] LOG ON деректер _қоры_құрылғы][= өлшемі][, деректер _қоры_құрылғы][= өлшемі]]...][FOR LOAD]

MyDBroot деректері бар файл және Logdata1 транзакциялар журналының файлы бар MyDB атымен деректер қорын құру мысалы

```
CREATE DATABASE MyDB
ON
(NAME="MyDBroot",
FILENAME="c:\mssql2k\MSSQL\data\mydbroot.mdf",
SIZE=8MB,
MAXSIZE=9MB,
FILEGROWTH=100KB),
LOG ON
(NAME="Logdata1",
FILENAME="e:\log_files\logdata1.ldf",
SIZE=1000MB,
MAXSIZE=1500MB,
FILEGROWTH=100MB)
```

CREATE DATABASE операторының параметрлерін сипаттау:

- **ON** – кілт сөз, командада деректер файлдарының орналасуы, олардың атаулары, көлемі және өсу көлемі көрсетілуі тиіс екенін көрсетеді;
- **NAME** – SQL Server тарапынан осы файлға жүгінетін файлдың логикалық атауы;
- **FILENAME** – файлды кеңейтуді міндетті түрде көрсете отырып, толық жолды көрсететін файлдың физикалық аты;
- **SIZE** – мегабайттағы бастапқы көлемі. Міндетті емес.
- **FILEGROWTH** – файлдың көлемін толтырғаннан кейін өсіру. Өсім ағымдағы көлемнен мегабайларда немесе пайызбен көрсетуге болады;
- **LOG ON** – кілт сөз, командада журнал файлдарының орналасуы, олардың аттары, көлемі және өсу көлемі көрсетілуі тиіс екенін көрсетеді;

Деректер қорының кестелерін құру (жою)

Базалық кестелер **CREATE TABLE** ұсынысымен SQL-да жасалады.

CREATE TABLE *базалық_кесте_аты*

(баған деректер_түрі

[,баған деректер_түрі] ...);

мұнда баған - кесте бағанының логикалық атауы, деректер_түрі ДҚБЖ қолдайтын деректер түрлерінің біріне тиесілі болуы тиіс.

CREAT TABLE - орындалатын ұсыныс. Егер оны терминалдан енгізсе, жүйе алдымен бос болатын кестені құрастырады: ол тек бағандар тақырыбының жолынан тұрады, бірақ әлі де деректермен ешқандай жолдар болмайды.

Деректер қорының кестелерін құру (жою)

Қызметкерлер атымен кестені құру мысалы.

```
CREATE TABLE Қызметкерлер
```

(Код	SMALLINT,
ТАӘ	CHAR (70),
Мекен-жай	CHAR (15),
Телефон	CHAR (10),
Туған күні	DATE,
Қала	CHAR (10),
СТН	CHAR (10));



Бар кестені ***DROP TABLE*** (кестені жою) ұсынысымен
жоюға болады :

DROP TABLE базалық_кесте;

DROP TABLE Қызметкерлер;

Индекстер құру (жою)

- SQL индексін құру үшін **CREATE INDEX** (индексті жасау) ұсынысы бар.
- **Индекс** - бұл берілген кесте бағанының мәні бойынша құрылған жүйелік кесте.
- Онда осы мәндер (кітаптың пәндік көрсеткішіне ұқсас құрылым) кездесетін оның жолдарына сілтеме жасалған кестенің көрсетілген бағанының бірегей мәндерінің тізбесі орналастырылады.

- Мысалы, ДҚ-да кәсіпорын қызметкерлерінің жалақысы туралы мәліметтер сақталатын **ЕСЕПТЕУ** кестесі бар

Қызметкердің ТАӘ	АЙ	Сомма
Иванов	1	10000
Петров	1	12000
Сидоров	1	10000
Иванов	2	10000
Сидоров	2	10000
Иванов	3	10000
Петров	3	12000

- ЕСЕПТЕУ кестесі қызметкерінің аты-жөні баған үшін жасалған **индекс** келесі мәліметтерді қамтиды:

Бағанның мәні	Мұндай мән кездесетін жолдар		
Иванов	1	4	6
Петров	2	7	
Сидоров	3	5	

CREATE INDEX ұсынысының синтаксисі

CREATE [UNIQUE][CLUSTERED|NONCLUSTERED]INDEX
индекс_аты ON
[*деректер_қоры.*]*базалық_кесте*(*баған_аты*[[ASC] |
DESC] [,*баған_аты*[[ASC] | DESC]]...)[WITH][FILLFACTOR
= *x*] [ON *сегмент_аты*]

- CLUSTERED - кластерленген индекс, яғни индексті құрайтын B-ағаштың жапырақтарында деректерге сілтеме емес, деректердің беттерін білдіретін индекс.
- FILLFACTOR - индекстің B-ағашының беттерін толтыруды басқаруға мүмкіндік береді, пайызда, 100% - толық толтыру.
- UNIQUE (бірегей) - индекстелетін базалық кестеде ешқандай екі жол индекстелетін баған үшін бірдей мәнді қабылдауға рұқсат етілмейді

Мысалы, деректер базасында 2 кестелер бар:

1. *Тағам*: БЛ (тағам коды) және Негізі бағандары бар.
2. *Құрамы*: БЛ (тағам коды), ПР (өнім коды), Салмағы бағандары бар.

Тағам кестесінің БЛ және Негізі бағандарының индекстері мына ұсыныстар арқылы жасалады :

```
CREATE UNIQUE INDEX Тағам_БЛ ON Тағам (БЛ);
```

```
CREATE INDEX Тағам_Негізі ON Тағам (Негізі);
```

Құрамы кестесінің бастапқы кілтке (БЛ және ПР бағандары) арналған индекс бұл ұсыныс арқылы жасалады:

```
CREATE UNIQUE INDEX Құрамы_БЛ_ПР ON Құрамы (БЛ, ПР);
```

Деректерді таңдау (SELECT ұсынысы)

SELECT [[ALL] | DISTINCT]{ * | select_элементі [,
select_элементі] ...}

FROM базалық_кесте | ұсыныс [лақап ат]

[, базалық_кесте | ұсыныс[лақап ат]] ...

[WHERE фраза]

[GROUP BY фраза [HAVING фраза]];

мұнда select_элементі - бұл келесі конструкциялардың бірі:

[кесте.]* | [кесте.]баған | SQL_функция | айнымалы
| (өрнек) | жүйелік_айнымалы

- SELECT - көрсетілген бағандардан деректер (таңдау)
- FROM - аталған кестелер(ден)
- WHERE - (мұнда) көрсетілген кестелердегі жолдар жолдарды іріктеу шарттарын қанағаттандыруы тиіс
- GROUP BY - SELECT SQL-SUM (сума), COUNT (Сан), MIN (ең төменгі мән), MAX (ең жоғарғы мән) немесе AVG (орташа мән) фразасын пайдалана отырып, әрбір топ үшін жалғыз біріктірілген мәнді алу үшін көрсетілген бағандар тізімі бойынша (топтастыру)
- ORDER BY - содан кейін деректерді таңдау нәтижелерін реттеу. Бұл ретте, реттеу өсу тәртібімен - ASC (ASCending) немесе DESC (DESCending) кемуі, әдепкі бойынша ASC қабылданады.
- HAVING - (бола тұра) нәтижесінде топтарды іріктеу шартын қанағаттандыратын топтар ғана

- Салыстыру операторларынан басқа (= | <> | < | <= | > | >=) WHERE фразада NOT (жоқ) операторы алдын ала алуы мүмкін BETWEEN (арасында), LIKE (ұқсас), IN (тиесілі), IS NULL (анықталмаған) және EXISTS (бар) шарттары қолданылады.
- Жолдарды іріктеу критерийі логикалық операторлар қосылған бір немесе бірнеше шарттан құралады:
 - AND - шарттарың екеуі де орындалуы тиіс;
 - OR - шарттардың бірі орындалуы тиіс;
 - AND NOT - бірінші шарт қанағаттандырылады және екінші шарт болмауы керек;
 - OR NOT - бірінші шарт қанағаттандырылуы керек немесе екінші шарт қанағаттандырылмауы керек болған жағдайда, AND-дан басымдылық бар (алдымен барлық операциялар орындалады AND және тек осы операциядан кейін ғана OR).

Деректер базасында үш кесте болсын:

■ Товары

Код товара	Наименование	Ед_изм	Категория_товара
1	Товар_1	шт	электроника
2	Товар_2	шт	мебель
3	Товар_3	литр	напитки
4	Товар_4	кг	мясные изделия
5	Товар_5	шт	книги

■ Поставщики

Код поставщика	Наименование	Номер счета	Банк
1	Поставщик_1	1111	Народный
2	Поставщик_2	2222	Альянс
3	Поставщик_3	3333	Народный
4	Поставщик_4	4444	Альянс
5	Поставщик_5	5555	Народный

■ Движение товара

Дата	№ накладной	Код поставщика	Код товара	Количество	Цена
01.01.06	1	1	2	100	10
10.01.06	2	5	3	50	20
20.01.06	3	1	2	100	10
30.01.06	4	2	1	50	15
01.03.06	5	3	4	100	25
10.03.06	6	4	3	200	20
20.03.06	7	1	2	10	12
30.03.06	8	2	1	50	15
01.10.06	9	3	4	10	25
10.10.06	10	5	3	20	20
20.10.06	11	1	2	100	12

Қарапайым іріктеу

Сұрау: тауарлардың атауы мен санаттарын беріңіз
SELECT Наименование, Категория_товара
FROM Товары;

Нәтижесі:

Наименование	Категория_товара
Товар_1	электроника
Товар_2	мебель
Товар_3	напитки
Товар_4	мясные изделия
Товар_5	книги

Жеткізушілер туралы толық ақпарат алу үшін **сұрау** салу:

**SELECT Код поставщика, Наименование, Номер счета,
Банк FROM Поставщики;**

немесе оның қысқа белгісін қолданыңыз:

SELECT *FROM Поставщики;

"Жұлдызша" (*) FROM фразасында көрсетілген кестедегі өрістердің барлық атауларының қысқаша белгісі ретінде қызмет етеді. Сонымен қатар, өрістерді шығару тәртібі кестені құру кезінде осы өрістер анықталған тәртіпке сәйкес келеді.

Сұрау нәтижесі:

Код поставщика	Наименование	Номер счета	Банк
1	Поставщик_1	1111	Народный
2	Поставщик_2	2222	Альянс
3	Поставщик_3	3333	Народный
4	Поставщик_4	4444	Альянс
5	Поставщик_5	5555	Народный

- Тауар жеткізушілеріне қызмет көрсететін банктердің тізімін беру:

SELECT Банк

FROM Поставщики;

Банк
Народный
Альянс
Народный
Альянс
Народный

- Көшірмелерді алып тастау және жолдарды бір уақытта ретке келтіру үшін сұранысты DISTINCT (әр түрлі, әр түрлі) кілт сөзімен толықтыру қажет):

SELECT DISTINCT Банк

FROM Поставщики;

Банк
Альянс
Народный

Есептелген мәндерді таңдау

- SELECT фразасында кесте бағандарының тізімі немесе * таңбасы ғана емес, сонымен қатар өрнектер де болуы мүмкін.
- Жүкқұжат бойынша жеткізілген тауарлардың құнын алу үшін сұрау салудың мысалы:

SELECT № Накладной, (Количество * Цена)

FROM Движение товара;

Нәтижесі:

№ накладной	Количество*Цена
1	1000
2	100
3	1000
4	750
5	2500
6	4000
7	120
8	750
9	250
10	400
11	1200
12	75

Есептелген мәндерді таңдау

- SELECT тіркесі өрнектерді ғана емес, сонымен қатар жеке сандық немесе мәтіндік тұрақтыларды да қамтуы мүмкін.
- Мәтін тұрақтылары апострофтар (') болып табылады.

SELECT № накладной, 'Стоимость =',
(Количество*Цена) FROM Движение товара;

Нәтижесі:

№ накладной		Количество*Цена
1	Стоимость =	1000
2	Стоимость =	100
3	Стоимость =	1000
4	Стоимость =	750
5	Стоимость =	2500
6	Стоимость =	4000
7	Стоимость =	120
8	Стоимость =	750
9	Стоимость =	250
10	Стоимость =	400
11	Стоимость =	1200
12	Стоимость =	75

WHERE фразаны қолдана отырып іріктеу

Кестенің қажетті жолдарын таңдау үшін қай сөйлемде салыстыру операторларын қолдануға болады = (тең), < > (тең емес), < (аз), < = (аз немесе тең), > (көп), > = (көп немесе тең), мысалы, "кем емес" және "артық емес" қатынастарын құра отырып, NOT операторы алдын-ала жасай алады»

- 1 коды бар жазып берілген ұйымдардың жүкқұжаттарының тізбесін алу үшін сұрау салу:

```
SELECT    № накладной, Дата  
FROM      Движение товара  
WHERE     Код поставщика = 1;
```

№ накладной	Дата
1	01.01.06
3	20.01.06
7	20.03.06
11	20.10.06

WHERE фразаны қолдана отырып іріктеу

- AND, OR, AND NOT және OR NOT логикалық операторларымен қосылған бірнеше шарттарды пайдалану мүмкіндігі жолдарды егжей-тегжейлі таңдауға мүмкіндік береді.
- Құны 20 теңгеден кем және саны ≥ 50 тауарларды қамтитын жүкқұжаттар тізбесін алу үшін сұрау салу
SELECT № накладной, Дата. Количество, Цена
FROM Движение товара
WHERE Цена <20 AND Количество ≥ 50 ;

Нәтижесі:

№ накладной	Дата	Количество	Цена
1	01.01.06	100	10
3	20.01.06	100	10
4	30.01.06	50	15
8	30.03.06	50	15
11	20.10.06	100	12

BETWEEN пайдалану

BETWEEN көмегімен ... AND ... бағанның мәні берілген диапозонда болатын жолдарды таңдауға болады.

- Мысалы, тауарлар бағасының мәні 10-нан 15-ке дейінгі диапозонда болатын жүкқұжаттың тізімін беріңіз:

```
SELECT    № накладной, Дата, Количество, Цена  
FROM Движение товара  
WHERE Цена BETWEEN 10 AND 15;
```

№ накладной	Дата	Количество	Цена
1	01.01.06	100	10
3	20.01.06	100	10
4	30.01.06	50	15
7	20.03.06	10	12
8	30.03.06	50	15
11	20.10.06	100	12
12	30.10.06	5	15

IN қолдану

- In операторы берілген мән енгізілуі мүмкін немесе болмауы мүмкін мәндер жиынтығын анықтайды.
- 1 және 2 кодтары бар тауарлар жеткізілген жүкқұжаттар туралы мәліметтер беру:

SELECT *

FROM Движение товара

WHERE Код товара IN (1, 2);

Дата	№ накладной	Код поставщика	Код товара	Количество	Цена
01.01.06	1	1	2	100	10
20.01.06	3	1	2	100	10
30.01.06	4	2	1	50	15
20.03.06	7	1	2	10	12
30.03.06	8	2	1	50	15
20.10.06	11	1	2	100	12
30.10.06	12	2	1	5	15

LIKE пайдалану

- LIKE тек ішкі жолдарды табу үшін қолданылатын CHAR немесе VARCHAR сияқты өрістерге қолданылады.
- LIKE іздейді таңба өрісі оның жолының бір бөлігі Шартқа сәйкес келетіндігін көру. Шарт ретінде ол топтық таңбаларды қолданады-бір нәрсеге сәйкес келетін арнайы таңбалар.
- LIKE-де топтық таңбалардың екі түрі бар:
 - ✓ _ (астын сызу) - кез-келген жалғыз таңбаны ауыстырады,
 - ✓ % (пайыз) – N таңбаларының кез-келген тізбегін ауыстырады (мұндағы N нөл болуы мүмкін),

LIKE пайдалану

- Мәтін түрінің бағанына арналған "бағанның_ атауы LIKE мәтін_константа" синтаксисі "мәтін_константа" берілген үлгіге сәйкес келетін көрсетілген бағанның барлық мәндерін табуға мүмкіндік береді.
- Атауында «оставщик» символдарының тіркесімі бар ұйымдардың тізімін беру

SELECT Наименование

FROM Поставщики

WHERE Наименование LIKE '%оставщик%';

Наименование
Поставщик_1
Поставщик_2
Поставщик_3
Поставщик_4
Поставщик_5

Реттік іріктеу

- Кестелер - бұл реттелмеген мәліметтер жиынтығы, және олардан шыққан мәліметтер белгілі бір кезектілікте көрінбеуі керек.
- SQL шығыс тізімін өзгерту үшін ORDER BY командасын қолданады.
- Көптеген бағандар бір-біріне реттелген және сіз әр баған үшін өсуді (ASC) немесе кемуді (DESC) анықтай аласыз. Әдепкі бойынша өседі.

Реттік іріктеу

- Жеткізілген тауарлар санының кему тәртібімен бойынша жүкқұжаттардың тізбесін беру керек

```
SELECT      *  
FROM Движение товаров  
ORDER      BY Количество DESC;
```

Дата	№ накладной	Код поставщика	Код товара	Количес тво	Цена
10.03.06	6	4	3	200	20
01.01.06	1	1	2	100	10
20.01.06	3	1	2	100	10
01.03.06	5	3	4	100	25
20.10.06	11	1	2	100	12
10.01.06	2	5	3	50	20
30.01.06	4	2	1	50	15
30.03.06	8	2	1	50	15
10.10.06	10	5	3	20	20
20.03.06	7	1	2	10	12
01.10.06	9	3	4	10	25
30.10.06	12	2	1	5	15

Деректерді агрегациялау

Сұраулар жиынтық функцияларды қолдана отырып өрістердің жалпыланған топтық мәнін шығара алады:

COUNT-бағандағы мәндер саны,

SUM-бағандағы мәндердің қосындысы,

AVG-бағандағы орташа мән,

MAX-бағандағы ең үлкен мән,

MIN-бағандағы ең кіші мән.

- SUM және AVG функциялары үшін қарастырылған бағанда сандық мәндер болуы керек.
- COUNT (*) санынан басқа барлық функциялардың алдында DISTINCT кілт сөзі (әр түрлі) болуы мүмкін, бұл функция қолданылғанға дейін қайталанатын мәндерді жою керек екенін көрсетеді.
- COUNT (*) функциясы кестедегі барлық жолдарды ерекшеліксіз санауға арналған (оның көшірмелерін қосқанда).

Деректерді агрегациялау

- Коды=1 бар тауарға жеткізілген тауардың саны және оның орташа бағасы туралы деректер беру:

```
SELECT SUM(Количество),AVG(Цена)
FROM Движение товара
WHERE Код товара=1;
```

SUM(Количество)	AVG(Цена)
105	15

```
SELECT MAX (Количество*Цена)
FROM Движение товара;
```

Количество*Цена
4000

GROUP BY фразасы

- GROUP by ұсынысы басқа өріс тұрғысынан арнайы өрістегі мәндердің ішкі жиынын анықтауға және жиынтық функциясын ішкі жиынға қолдануға мүмкіндік береді. Бұл өрістерді және агрегаттық функцияларды SELECT бірыңғай сөйлемінде біріктіруге мүмкіндік береді.
- Жеткізілген тауарлардың әрқайсысының жалпы массасын есептеу қажет болсын.

SELECT Код товара, SUM(Количество)

FROM Движение товара

GROUP BY Код товара;

Код товара	SUM(Количество)
2	310
3	270
1	105
4	110

GROUP BY фразасы

- GROUP by фразасы order BY дегенді білдірмейді. Нәтижені реттеу үшін сұраныс беру керек

SELECT Код товара, SUM(Количество)

FROM Движение товара

GROUP BY Код товара;

ORDER BY Код товара;

Код товара	SUM(Количество)
1	105
2	310
3	270
4	110

HAVING фразасы

- HAVING сөйлемі жолдар үшін WHERE сөйлемі сияқты топтар үшін бірдей рөл атқарады: топтарды алып тастау үшін қолданылады, сол сияқты WHERE жолдарды алып тастау үшін қолданылады.
- Бұл сөйлем тек GROUP BY сөйлемі болған жағдайда ғана сөйлемге кіреді, ал HAVING өрнегі топ үшін жалғыз мәнге ие болуы керек.
- Екіден астам жүкқұжат бойынша тауар жеткізген жеткізушілердің кодтарының тізімін шығарыңыз:

```
SELECT Код поставщика, COUNT(№ накладной)
FROM Движение товара
GROUP BY Код поставщика
HAVING COUNT(№ накладной)>2;
```

Код поставщика	COUNT(№ накладной)
1	4
2	3

Бірнеше кестені қолданатын сұраулар

- SQL-де бір-бірімен байланысты бірнеше кестелерден алынған мәліметтерді бір уақытта немесе дәйекті түрде өңдеу механизмі бар.
- = 1 кодымен жеткізуші жеткізетін тауарлардың атауын және тауар санатының атауын алыңыз:

```
SELECT      Наименование,          Категория          товара,  
Количество, Цена  
FROM Товары, Движение товара  
WHERE      Товары.Код товара = Движение товара. Код  
товара  
AND      Код поставщика = 1;
```

Бірнеше кестені қолданатын сұраулар

Результат:

Наименование	Категория товара	Количество	Цена
Товар_2	мебель	100	10
Товар_2	мебель	100	10
Товар_2	мебель	10	12
Товар_2	мебель	100	12

- Іріктеу келесі түрде алынды: ДҚБЖ FROM тармағында келтірілген кестелердің декарттық көбейтіндісінің жолдарын дәйекті түрде жасайды, құрылған жолдың деректері WHERE тармағының шарттарына сәйкес келетіндігін тексереді, егер олар сәйкес келсе, онда ол сұрауға жауапқа SELECT сөйлемінде келтірілген өрістерді қосады.

Ішкі сұраулар

- *Кірістірілген ішкі сұрау* дегеніміз жақшаның ішіне алынған және SELECT сөйлемінің WHERE (HAVING) сөйлемі немесе WHERE сөйлемін қолданатын басқа сөйлемдер ішінде орналасқан ішкі сұрау.
- Кірістірілген ішкі сұрау өзінің WHERE (HAVING) сөйлемінде тағы басқа ішкі сұранысты қамтуы мүмкін және т.б.
- *Қарапайым* және *өзара* байланысты ішкі сұраулар бар. Олар IN, EXISTS немесе салыстыру шарттарының бірін (= | <> | < | <= | > | >=) қолдану арқылы WHERE (HAVING) сөйлеміне енгізілген.

Ішкі сұраулар

- Қарапайым ішкі сұраулар *төменнен жоғары* өңделеді. Алдымен ең төменгі ішкі сұрау өңделеді. Оны орындау нәтижесінде алынған мәндер жиынтығы жоғарғы деңгейдегі сұранысты жүзеге асыруда және т.б.
- Ішкі кірістірілген ішкі сұраулары бар сұраулар *кері тәртіпте* өңделеді. Алдымен, негізгі сұраныс жасаған жұмыс парағының бірінші жолы таңдалады, ал ішіне кірістірілген ішкі сұрауда қолданылатын бағандардың мәні таңдалады. Егер бұл мәндер ішкі ішкі сұраудың шарттарына сәйкес келсе, онда таңдалған жол нәтижеге қосылады. Содан кейін екінші қатар таңдалады және солай, нәтижеге кірістірілген ішкі сұранысты қанағаттандыратын барлық жолдар енгізілгенге дейін.

Қарапайым ішкі сұраулар

- Қарапайым кірістірілген ішкі сұраулар кейбір IN предикаттарында тексерілуі керек мәндер жиынтығын ұсыну үшін қолданылады.
- 20 теңге бағамен тауар жеткізіп берген жеткізушілер мен оларға қызмет көрсетілетін банктердің атауын шығарыңыз.

SELECT Наименование, Банк

FROM Поставщики

WHERE Код поставщика IN

(SELECT Код поставщика

FROM Движение товара

WHERE Цена = 20);

Қарапайым ішкі сұраулар

Наименование	Банк
Поставщик_5	Народный
Поставщик_4	Альянс
Поставщик_5	Народный

- Толық сұранысты өңдеген кезде жүйе алдымен ішкі ішкі сұранысты орындайды. Бұл ішкі сұраныста тауарларды 20 теңгеден жеткізетін жеткізушілердің бірқатар жиынтығы, атап айтқанда жиынтық берілген (5,4,5). Сондықтан түпнұсқа сұрау осы қарапайым сұрауға тең:

SELECT Название, банк

FROM Поставщики

WHERE Код поставщика IN (4,5);

Өзара байланысты ішкі сұраулар

- 20 теңге бағамен тауар жеткізіп берген жеткізушілер мен оларға қызмет көрсетілетін банктердің атауын шығарыңыз.

SELECT Наименование, Банк

FROM Поставщики

WHERE 20 IN

(SELECT Код поставщика

FROM Движение товара

WHERE Код поставщика = Движение
товара.Код поставщика

AND Движение товара.Цена = 20);

SQL деректерін өзгерту бойынша ұсыныстар

Деректерді өзгерту ұсыныстармен жүзеге асырылуы мүмкін

- DELETE (жою)
- INSERT (қосу)
- UPDATE (жаңарту)

Олар негізгі кестелерде де, көріністерде де жұмыс істей алады.

DELETE сөйлемі

DELETE сөйлемінің форматы:

DELETE

FROM базалық кесте | көрініс

[WHERE фраза];

көрсетілген кестенің барлық жолдарының мазмұнын (WHERE сөйлемі болмаған кезде) немесе WHERE сөйлемімен ерекшеленетін жолдарды жоюға мүмкіндік береді.

DELETE сөйлемі

«Барлық ет тағамдарын жою» сұранысы

DELETE FROM Блюда

WHERE Основа = 'Мясо';

«Алматыдан жеткізушіге барлық жеткізілімдерді жою» ішкі сұранысы бар жою

DELETE

FROM Поставки

WHERE Код_Поставщика IN

(SELECT Код_Поставщика FROM Поставщики

WHERE Город = 'Алматы');

INSERT сөйлемі

INSERT сөйлемі келесі форматтардың біріне ие:

1. INSERT

INTO {базалық кесте | көрініс} [(баған [, баған] ...)]

VALUES ({константа | айнымалы} [{константа | айнымалы}] ...);

немесе

2. INSERT

INTO {базалық кесте | көрініс} [(баған [, баған] ...)]

ішкі сұрау;

INSERT сөйлемі

1. Бірінші форматта кестеге VALUES (мәндер) сөз тіркесінің тізімінде көрсетілген өріс мәндері бар жол енгізіледі, ал і-ші мән бағандар тізіміндегі і-ші бағанға сәйкес келеді.
2. Екінші форматта алдымен сұраныс орындалады, яғни. SELECT сөйлемі жадында жұмыс кестесін жасайды, содан кейін жұмыс кестесінің жолдары өзгертілген кестеге жүктеледі. Бұл жағдайда жұмыс кестесінің і-ші бағанасы (SELECT тізімінің і-ші элементі) өзгертілген кестенің бағандарының тізіміндегі і-ші бағанға сәйкес келеді.

INSERT сөйлемі

Кестеге жалғыз жазбаны енгізіңіз

Кестеге қосу Шашлык тағамы (Код_Блюда - 34, Блюдо - Шашлык, Основа - Мясо, Выход - 150)

INSERT

INTO Блюда (Код_Блюда, Блюдо, Основа, Выход)

VALUES (34, 'Шашлык', 'Мясо', 150);

34-тағамға жаңа жазба жасалды.

UPDATE сөйлемі

UPDATE сөйлемінде келесі форматтардың бірі бар:

1. UPDATE (базалық кесте | көрініс)

SET баған = мәні [, баған = мәні] ...

[WHERE фраза]

осында мәні- бұл баған | өрнек | тұрақты | айнымалы және бағандарды тек жаңартылатын кестеден қамтуы мүмкін, яғни өзгертіліп отырған кестенің бір бағанының мәні оның екінші бағанының мәнімен немесе кестенің өзгертілген кестесін қоса, оның бірнеше бағанының мәндерін қамтитын өрнекпен ауыстырылуы мүмкін.

Егер WHERE сөйлемі жоқ болса, өзгертілген кестенің барлық жолдарындағы көрсетілген бағандардың мәндері жаңартылады. WHERE сөйлемі жаңартылған жолдардың санын олардың таңдау шарттарын көрсете отырып азайтуға мүмкіндік береді.

UPDATE сөйлемі

2. Екінші формат өзгертілген кесте мәндерін басқа кестелердегі бағандар мәндері бойынша жаңартуға мүмкіндік беретін ұсынысты сипаттайды.

UPDATE {базалық кесте | көрініс}

SET баған = мәні [, баған = мәні] ...

FROM {базалық кесте | көрініс} [псевдоним],

{базалық кесте | көрініс} [псевдоним]

[,{базалық кесте | көрініс} [псевдоним]] ...

[WHERE фраза]

UPDATE сөйлемі

Бірнеше жазбаларды жаңарту мысалы

«Жеткізу кестесіндегі барлық өнімдердің бағасын үш есеге арттырыңыз (кофе бағасынан басқа - Код_Продукта = 17)»

```
UPDATE Поставки
```

```
SET  Цена = Цена * 3
```

```
WHERE  Код_Продукта <> 17;
```

Ішкі сұраумен жаңарту мысалы

«Алматы мен Тараздан жеткізушілер үшін баға мен өнімнің санын нөлге тең етіп белгілеу».

```
UPDATE Поставки
```

```
SET  Цена = 0, К_во = 0
```

```
WHERE  Код_Поставщика IN
```

```
    (SELECT Код_Поставщика FROM Поставщики  
     WHERE  Город IN ('Алматы', 'Тараз'));
```

Қауіпсіздік және қол жетімділікті растау

- Деректер базасы тұрғысынан *қауіпсіздік термині* мәліметтерді рұқсат етілмеген ашылудан, өзгеруден немесе жойылудан қорғауды білдіреді.
- SQL бүкіл кестелерді де, олардың жеке өрістерін де жеке қорғауға мүмкіндік береді.
- Мұның екі мүмкіндігі бар:
 - құпия деректерді рұқсат етілмеген пайдаланушылардан жасыру үшін қолданылатын **көріністер механизмі**;
 - белгілі бір пайдаланушыларға деректерге қол жеткізу үшін артықшылықтар беруге мүмкіндік беретін және оларға артықшылықтардың бір бөлігін басқа пайдаланушыларға беру мүмкіндігін беретін, қажет болған жағдайда кейіннен осы артықшылықтардан бас тартуға мүмкіндік беретін **қол жетімділік рұқсаты** ішкі жүйесі.

Қауіпсіздік және қол жетімділікті растау

- Әдетте, ДҚБЖ-ны орнатқан кезде оған қандай-да бір идентификатор енгізіледі, оны әрі қарай ең артықшылығы бар пайдаланушының - **жүйелік администратордың** идентификаторы ретінде қарастыру қажет.
- Жүйелік әкімші мәліметтер базасын құра алады және оларды пайдалану үшін толық артықшылықтарға ие. Бұл артықшылықтар немесе олардың бір бөлігі басқа пайдаланушыларға берілуі мүмкін.
- Өз кезегінде, жүйе әкімшісінен артықшылықтар алған пайдаланушылар оларды (немесе олардың бір бөлігін) келесіге бере алатын басқа пайдаланушыларға бере алады және т.б.

Қауіпсіздік және қол жетімділікті растау

Артықшылықтар GRANT ұсынысы арқылы беріледі:

GRANT артықшылықтар ON объект TO
пайдаланушылар;

- ✓ мұндағы «артықшылықтар» дегеніміз - үтірмен бөлінген бір немесе бірнеше артықшылықтардың тізімі немесе ALL PRIVILEGES (барлық артықшылықтар) тіркесі;
- ✓ «объект» - объектінің атауы және қажет болған жағдайда (мәліметтер базасы, кесте, көрініс, индекс және т.б.);
- ✓ «пайдаланушылар» дегеніміз - үтірлермен бөлінген бір немесе бірнеше авторизация идентификаторларының тізімі немесе арнайы PUBLIC (public) кілт сөзі.

Қауіпсіздік және қол жетімділікті растау

- Кестелер мен көріністерде SELECT, DELETE, INSERT және UPDATE артықшылықтары бар, олар оқуға (SELECT қолданатын кез-келген әрекетті орындауға), жоюға, қосуға немесе көрсетілген кестеден жолдарды өзгертуге мүмкіндік береді.
- Өзгерістер белгілі бір бағандармен шектелуі мүмкін. Мысалы, сөйлем

GRANT SELECT, UPDATE (Труд) ON Блюда TO cook;

өзін жүйеге аспаз ретінде көрсеткен пайдаланушыға Блюда кестесіндегі ақпаратты пайдалануға мүмкіндік береді, бірақ ол ондағы тек қана Труд бағанының мәндерін өзгерте алады.

Қауіпсіздік және қол жетімділікті растау

- Егер USER_1 пайдаланушысы басқа пайдаланушыға USER_2 қандай да бір артықшылықтар берген болса, ол кейіннен сол артықшылықтардың кейбірін немесе барлығын жоя алады.
- Жойылу REVOKE сөйлемінің көмегімен жүзеге асырылады:

REVOKE привилегии ON объект FROM пользователи;

Мысалы, аспаз пайдаланушысынан Труд бағанының мәндерін өзгерту құқығын алып тастай аласыз:

REVOKE UPDATE (Труд) ON Блюда FROM cook;

Бақылау сұрақтары:

1. Реляциялық ДБ-ның негізгі ұғымдарына анықтама беріңіз. Әр ұғымына мысал ретінде өзіңіздің жобаланған ДБ-ны пайдаланып түсіндіріңіз.
2. «Кортеж-дубликаттардың болмауы» қасиеті неге негізделген?
3. «Кортеждердің реттілігінің болмауы» қасиеті неге негізделген?
4. «Атрибут мәндерінің атомарлылығы» қасиеті неге негізделген?
5. Теориялық-көптік операцияларға қандай операциялар жатады? Мысалдар келтіріңіз.
6. Арнайы реляциялық операцияларға қандай операциялар жатады? Мысалдар келтіріңіз.

Әдебиеттер тізімі :

1. Т. Конноли. Базы данных. Проектирование, реализация и сопровождение. Теория и практика.: Пер.с англ. – М.: Изд.дом «Вильямс», 2012. – 1440 с.
2. К.Дейт. Введение в системы БД. изд.6-е. :Пер.с англ. - М.: Изд.дом "Вильямс", 2014. – 1200с.
3. Д.Ульман. Введение в системы баз данных. – М.: Издательство «Лори», 2015. – 853с.



Назарларыңызға рахмет!