

SQL ТІЛІ: МӘЛІМЕТТЕРМЕН ЖҰМЫС. СҰРАНЫС ҚҰРУ



Дәріс 9

SELECT операторы

Сұраныс құру (ДҚ бір не бірнеше кестелерінен жолдарды таңдау) үшін *SELECT* оператор қолданылады. Сұраныстың нәтижесі- үнемі кесте болады. Сұраныс нәтижесі жаңа кесте құруға қолданылу мүмкін. Сұраныс нәтижесінде пайда болған кесте басқа сұраныста қолданылу мүмкін.

SELECT операторының жалпы формасы:

SELECT бағаналар FROM кестелер

[WHERE шарттар]

[GROUP BY топ [HAVING топтық_шарттар]]

[ORDER BY өріс_аты]

[LIMIT шектеулер];

SELECT операторы

Барлық кесте бағандарын шығару керек болса, select операторынан кейін тізбектеп жазбаларды қоюға қажеті жоқ, тізімді тек * символымен ауыстырса жеткілікті. Мысал:

```
mysql> SELECT * FROM orders;
```

order_ID	o_user_ID	o_book_ID	o_time	o_number
1	3	8	2009-01-04 10:39:38	1
2	6	10	2009-02-10 09:40:29	2
3	1	20	2009-02-18 13:41:05	4
4	4	20	2009-03-10 18:20:00	1
5	3	20	2009-03-17 19:15:36	1

```
5 rows in set (0.02 sec)
```

SELECT операторы

Бар мәнді нәтижеде қайта қайталамау үшін сұраныста *DISTINCT* кілт сөзін қолдану керек, мысалы:

```
mysql> SELECT cat_name, cat_ID FROM catalogs;
+-----+-----+
| cat_name      | cat_ID |
+-----+-----+
| Программирование |      1 |
| Интернет      |      2 |
| Базы данных   |      3 |
| Сети          |      4 |
| Мультимедиа  |      5 |
+-----+-----+
5 rows in set (0.01 sec)
```

Таңдау шарты

Шығарылатын жолдар санын өзгерту жағдай өте жиі кездеседі. Аталған іздеу критерийларын қанағаттандыратын жазба таңдау үшін *WHERE* конструкциясын қолдану қажет :

```
mysql> SELECT user_ID, u_surname FROM users  
-> WHERE u_status='active';
```

user_ID	u_surname
1	Иванов
3	Симонов
4	Кузнецов

```
3 rows in set (0.03 sec)
```

Таңдау шарты

Шығарылатын жолдар санын өзгерту жағдай өте жиі кездеседі. Аталған іздеу критерийларын қанағаттандыратын жазба таңдау үшін *WHERE* конструкциясын қолдану қажет :

```
mysql> SELECT user_ID, u_surname FROM users  
-> WHERE u_status='active';
```

user_ID	u_surname
1	Иванов
3	Симонов
4	Кузнецов

```
3 rows in set (0.03 sec)
```

Таңдау шарты

- Бар мәнді нәтижеде қайта қайталамау үшін сұраныста *DISTINCT* кілт сөзін қолдану керек, мысалы:

```
mysql> SELECT DISTINCT u_status FROM users;
```

```
+-----+
```

```
| u_status |
```

```
+-----+
```

```
| active  |
```

```
| passive |
```

```
| lock    |
```

```
| gold    |
```

```
+-----+
```

```
4 rows in set (0.01 sec)
```

Сұрыптау

ДҚ-да сақталу тәртібі бойынша орналасқан жазбалар - таңдаудың нәтижесі болып табылады. Бағаналар арасынан біреуінің мәндерін сұрыптау үшін *ORDER BY* конструкциясынан кейін сол бағананы көрсету керек. Мысалы:

```
mysql> SELECT * FROM orders ORDER BY o_user_ID;
```

order_ID	o_user_ID	o_book_ID	o_time	o_number
3	1	20	2009-02-18 13:41:05	4
1	3	8	2009-01-04 10:39:38	1
5	3	20	2009-03-17 19:15:36	1
4	4	20	2009-03-10 18:20:00	1
2	6	10	2009-02-10 09:40:29	2

```
5 rows in set (0.00 sec)
```


Сұрыптау

Жазбалардың сұрыптауын бірнеше бағана бойынша жүргізуге болады (оларды *ORDER BY* сөзінен соң үтір қою арқылы көрсету керек). *ORDER BY* конструкциясында келтірілген бағаналар саны шектеусіз.

Үнсіз келісім бойынша сұрыптау тік тәртіп бойынша орындалады (сұрыптау өрісінің жазбалары ең кіші мәнінен ең үлкенге дейін орналасады). Кері тәртіппен сұрыптауы *DESC* кілттік сөзін қолдану арқылы жүзеге асырылады: Тік сұрыптау *ASC* кілттік сөзін қолдану арқылы іске асырылады, бірақ жазбалар үнсіз келісім бойынша тік тәртіпте іріктелетіндіктен, аталған кілттік сөз жазылмайды

```
mysql> SELECT o_time FROM orders ORDER BY o_time DESC;
+-----+
| o_time                |
+-----+
| 2009-03-17 19:15:36   |
| 2009-03-10 18:20:00   |
| 2009-02-18 13:41:05   |
| 2009-02-10 09:40:29   |
| 2009-01-04 10:39:38   |
+-----+
5 rows in set (0.27 sec)
```

Таңдауды шектеу

- Мысал: Сұраныс алғашқы 5 жазшығарады, осы кезде *b_count* өрісі бойынша кері сұрыптау жүргізіледі:

```
mysql> SELECT book_ID, b_count FROM books  
-> ORDER BY b_count DESC  
-> LIMIT 5;
```

book_ID	b_count
28	20
25	20
26	15
29	12
9	12

```
5 rows in set (0.03 sec)
```

Жазбаларды топтау

- *NULL* – ге тең емес көрсетілген бағана мәндері кестедегі жол санын шығаратын *COUNT()* функциясын пайдалануға мысал:

```
mysql> SELECT COUNT(book_ID) FROM books;  
-----  
| COUNT(book_ID) |  
-----  
|               30 |  
-----  
1 row in set (0.16 sec)
```

Жазбаларды топтау

Мысал: *catalogs* кестеден *cat_ID* өрістің максималды мәні бойынша жазбаларды шығарып алу

```
mysql> SELECT * FROM catalogs WHERE cat_ID=MAX(cat_ID);  
ERROR 1111 (HY000): Invalid use of group function
```

- Есептің шешімі *ORDER BY* конструкциясын пайдалануда жатыр

```
mysql> SELECT * FROM catalogs ORDER BY cat_ID DESC LIMIT 1;  
+-----+-----+  
| cat_ID | cat_name |  
+-----+-----+  
|      5 | Мультимедиа |  
+-----+-----+  
1 row in set (0.00 sec)
```

- Таңдауда арнайы есептерді шешуде MySQL-дің ендірілген функциялары қолайлы болады. Көптеген функциялар *SELECT* және *WHERE* түрінде пайдалану тағайындалған. Арнайы топтау функциялары да бар. *SELECT* операторын кестесіз пайдалану мүмкін. *SELECT* операторы калькулятор ретінде

```
mysql> SELECT 2+3;
+-----+
| 2+3 |
+-----+
|    5 |
+-----+
1 row in set (0.00 sec)
```

Операторлар

MySQL-де 3 оператор типі қолданылады

арифметикалық операторлар

логикалық операторлар

салыстыру операторлары



Арифметикалық операторлар

MySQL-де қарапайым арифметикалық операторлар қолданылады: қосу(+), азайту (-), көбейту(*), бөлу(/) және бүтін санды бөлу *DIV*. Нөлге бөлу *NULL* қауіпсіз нәтиже береді.

Салыстыру операторы

Салыстыру операторымен жұмыс жасау кезінде *NULL* мәнін салыстыру нәтижесінде *NULL* болады:

```
mysql> SELECT NULL=NULL;
+-----+
| NULL=NULL |
+-----+
|          |
+-----+
1 row in set (0.02 sec)
```

- Сұранысты түзулегенде пайдаланылады

```
mysql> SELECT NULL IS NULL;
+-----+
| NULL IS NULL |
+-----+
|             1 |
+-----+
1 row in set (0.00 sec)
```


Логикалық операторлар

Оператор	Мысал	Мәні
<i>AND</i>	<i>n AND m</i>	Логикалық ЖӘНЕ: ақиқат <i>AND</i> ақиқат = ақиқат, жалған <i>AND</i> кез- келген = жалған . Все остальные выражения оцениваются как <i>NULL</i>
<i>OR</i>	<i>n OR m</i>	Логикалық НЕМЕСЕ: ақиқат <i>OR</i> кез- келген = ақиқат, <i>NULL OR</i> жалған = <i>NULL</i> , <i>NULL OR NULL</i> = <i>NULL</i> , жалған <i>OR</i> жалған = жалған
<i>NOT</i>	<i>NOT n</i>	Логикалық ЖОҚ: <i>NOT</i> ақиқат = жалған , <i>NOT</i> жалған = ақиқат. <i>NOT NULL</i> = <i>NULL</i>
<i>XOR</i>	<i>n XOR m</i>	Логикалық <i>исключающее НЕМЕСЕ</i> : ақиқат <i>XOR</i> ақиқат = жалған , ақиқат <i>XOR</i> жалған = ақиқат, жалған <i>XOR</i> ақиқат = ақиқат, жалған <i>XOR</i> жалған = жалған , <i>NULL XOR</i> кез- келген = <i>NULL</i> , кез- келген <i>XOR NULL</i> = <i>NULL</i>

Бақылау сұрақтары:

- 1 Шығарылатын жол санын өзгерту командасы
- 2 MySQL ДҚБЖ функциялары?
- 3 Для чего используется оператор *where* оператор не үшін қолданылады?
- 4 Салыстыру операторлары
- 5 Логикалық операторлар
- 6 Арифметикалық операторлар
- 7 Жазбаларды топтастыру командасы