

## Зертханалық жұмыс №4

**Мамандық:** Деректер туралы ғылым

**Пән атауы:** Деректер ғылымына кіріспе

**Оқытушы:** Базарбек Жания Пархатқызы

Мәліметтерді басқару негіздері. Мәліметтерді түрлендіру және әртүрлі деректер түрлерін өңдеу. Having, distinct, Union, EXISTS, INSERT ALL операторларымен танысу және мүмкіндіктерін тексеру.

SQL HAVING операторы жиынтық функцияларды орындау нәтижесінің көрсеткіші болып табылады. SQL тіліндегі жиынтық функция – баған мәндерінің жиынына негізделген жалғыз мәнді қайтаратын функция. Бұл функциялар: SQL COUNT(), SQL MIN(), SQL MAX(), SQL AVG(), SQL SUM().

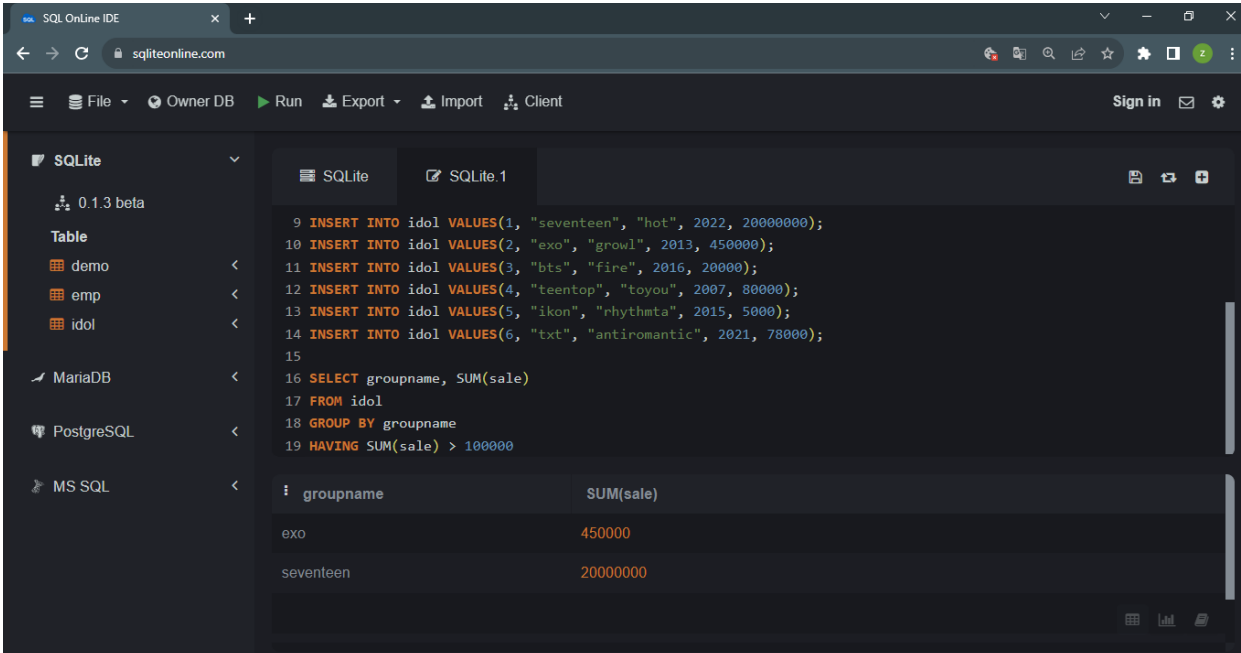
SQL HAVING операторында келесі синтаксис бар:

### **HAVING aggregate\_function(column\_name) operator value**

SQL HAVING операторы SQL WHERE операторына ұқсас, тек кесте бағандарының барлық жиынына емес, SQL GROUP BY операторы арқылы жасалған жиынға қолданылады және әрқашан одан кейін қатаң түрде қолданылады.

Мысалы:

SQL HAVING операторын пайдаланып, альбом сатылымы (sale) саны 100 000-нан асатын әртістердің (groupname) атын көрсетіңіз:



The screenshot shows a web-based SQL IDE interface. The main area displays a series of SQL queries and their results. The queries include several INSERT statements into a table named 'idol' and a SELECT statement using HAVING to filter results based on the sum of sales.

```
9 INSERT INTO idol VALUES(1, "seventeen", "hot", 2022, 2000000);
10 INSERT INTO idol VALUES(2, "exo", "growl", 2013, 450000);
11 INSERT INTO idol VALUES(3, "bts", "fire", 2016, 20000);
12 INSERT INTO idol VALUES(4, "teentop", "toyoul", 2007, 80000);
13 INSERT INTO idol VALUES(5, "ikon", "rhythmta", 2015, 5000);
14 INSERT INTO idol VALUES(6, "txt", "antiromantic", 2021, 78000);
15
16 SELECT groupname, SUM(sale)
17 FROM idol
18 GROUP BY groupname
19 HAVING SUM(sale) > 100000
```

The results table shows the following data:

groupname	SUM(sale)
exo	450000
seventeen	2000000

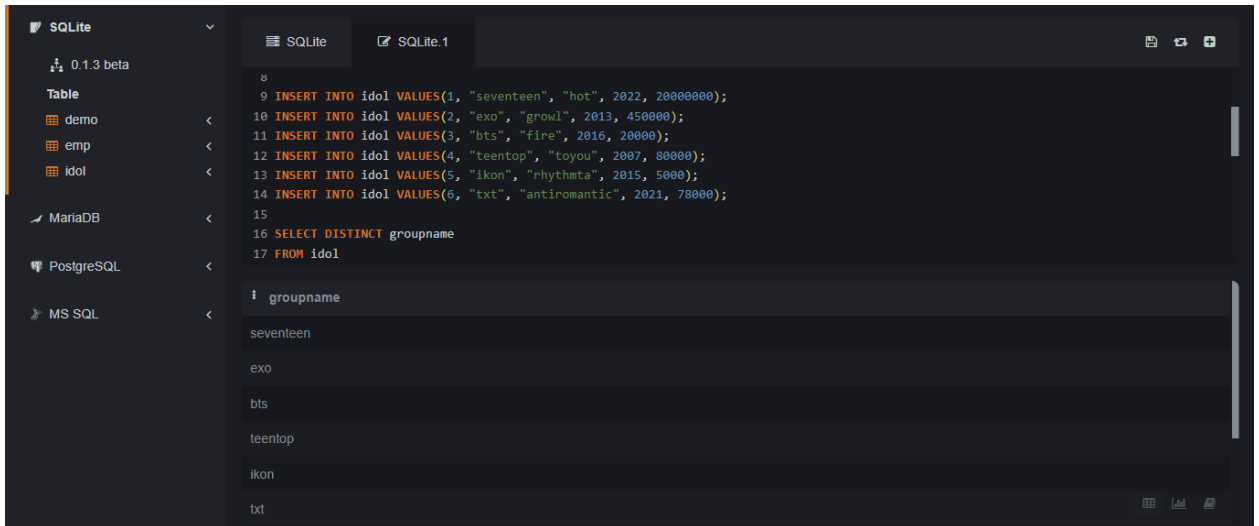
SQL DISTINCT операторы тек бағанның бірегей мәндерімен жұмыс істеу керектігін көрсету үшін қолданылады.

SQL DISTINCT операторы бірегей мәндерді таңдау үшін SQL SELECT операторында кеңінен қолданылады. Сондай-ақ жиынтық функцияларда қолданылады.

SQL DISTINCT операторында келесі синтаксис бар:

### **SELECT DISTINCT column\_name FROM table\_name**

Мысалы:



```
8
9 INSERT INTO idol VALUES(1, "seventeen", "hot", 2022, 2000000);
10 INSERT INTO idol VALUES(2, "exo", "growl", 2013, 450000);
11 INSERT INTO idol VALUES(3, "bts", "fire", 2016, 20000);
12 INSERT INTO idol VALUES(4, "teentop", "toyoy", 2007, 80000);
13 INSERT INTO idol VALUES(5, "ikon", "rhythmta", 2015, 5000);
14 INSERT INTO idol VALUES(6, "txt", "antiromantic", 2021, 78000);
15
16 SELECT DISTINCT groupname
17 FROM idol
```

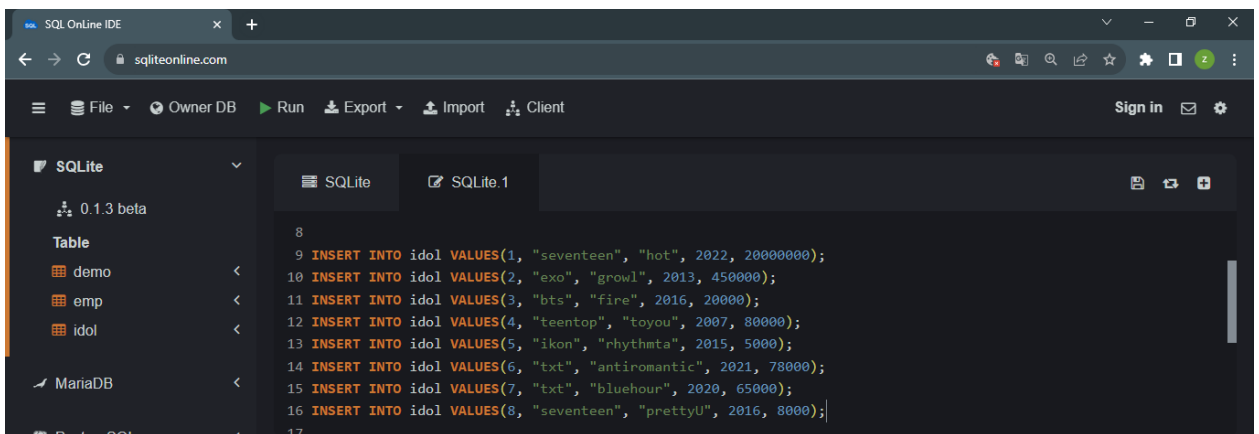
groupname
seventeen
exo
bts
teentop
ikon
txt

SQL UNION операторы екі немесе одан да көп SQL SELECT операторының сұрауларын біріктіру үшін пайдаланылады.

SQL UNION операторында келесі синтаксис бар:

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2
```

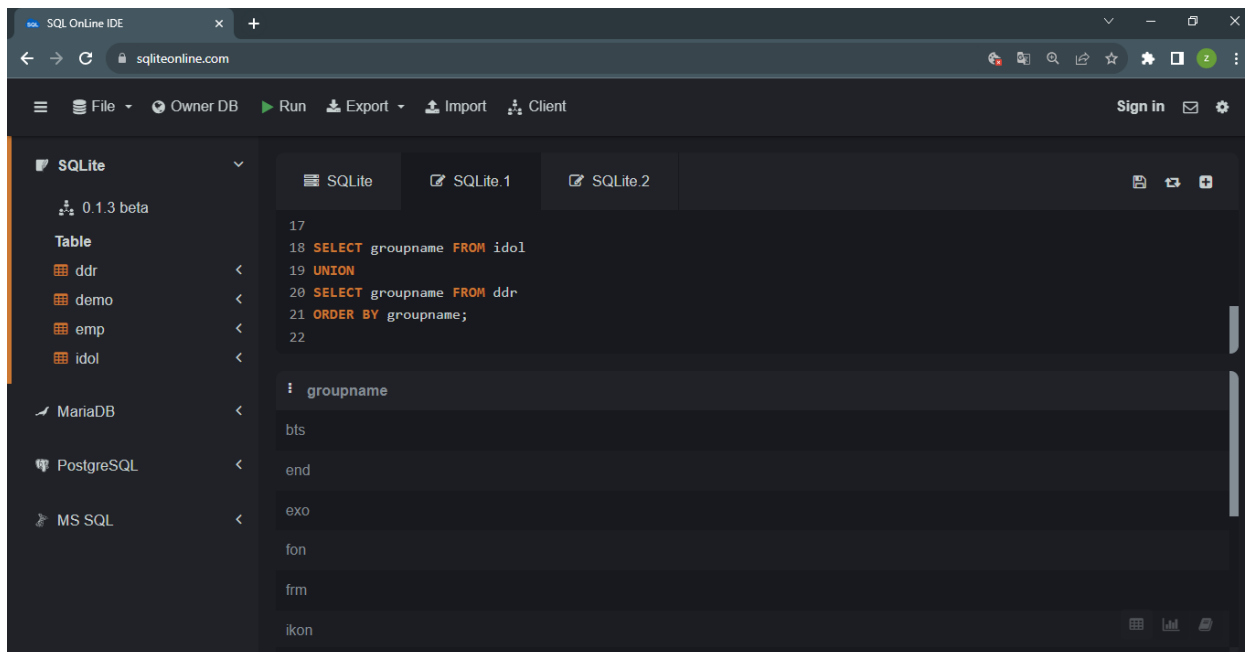
Әрбір SQL SELECT операторының сұрауында бірдей бағандар саны және қайтарылатын деректер түрлері болуы керек екенін ескеру маңызды, әйтпесе нәтиже кестені құру кезінде қате орын алады.



```
8
9 INSERT INTO idol VALUES(1, "seventeen", "hot", 2022, 2000000);
10 INSERT INTO idol VALUES(2, "exo", "growl", 2013, 450000);
11 INSERT INTO idol VALUES(3, "bts", "fire", 2016, 20000);
12 INSERT INTO idol VALUES(4, "teentop", "toyoy", 2007, 80000);
13 INSERT INTO idol VALUES(5, "ikon", "rhythmta", 2015, 5000);
14 INSERT INTO idol VALUES(6, "txt", "antiromantic", 2021, 78000);
15 INSERT INTO idol VALUES(7, "txt", "bluehour", 2020, 65000);
16 INSERT INTO idol VALUES(8, "seventeen", "prettyU", 2016, 8000);
```



```
3 groupname VARCHAR(10),
4 album VARCHAR(10),
5 year NUMBER(4),
6 sale NUMBER
7 );
8 INSERT INTO ddr VALUES(5, "fon", "rhythmta", 2015, 5000);
9 INSERT INTO ddr VALUES(6, "end", "antiromantic", 2021, 78000);
10 INSERT INTO ddr VALUES(7, "me", "bluehour", 2020, 65000);
11 INSERT INTO ddr VALUES(8, "frm", "prettyU", 2016, 8000);
```



Oracle EXISTS шарты ішкі сұраумен бірге пайдаланылады және егер ішкі сұрау кем дегенде бір жолды қайтарса, «қанағаттандырылуы керек» болып саналады. EXISTS SELECT, INSERT, UPDATE немесе DELETE мәлімдемелерінде қолданылуы мүмкін.

Oracle/PLSQL жүйесінде EXISTS шарт синтаксисі:

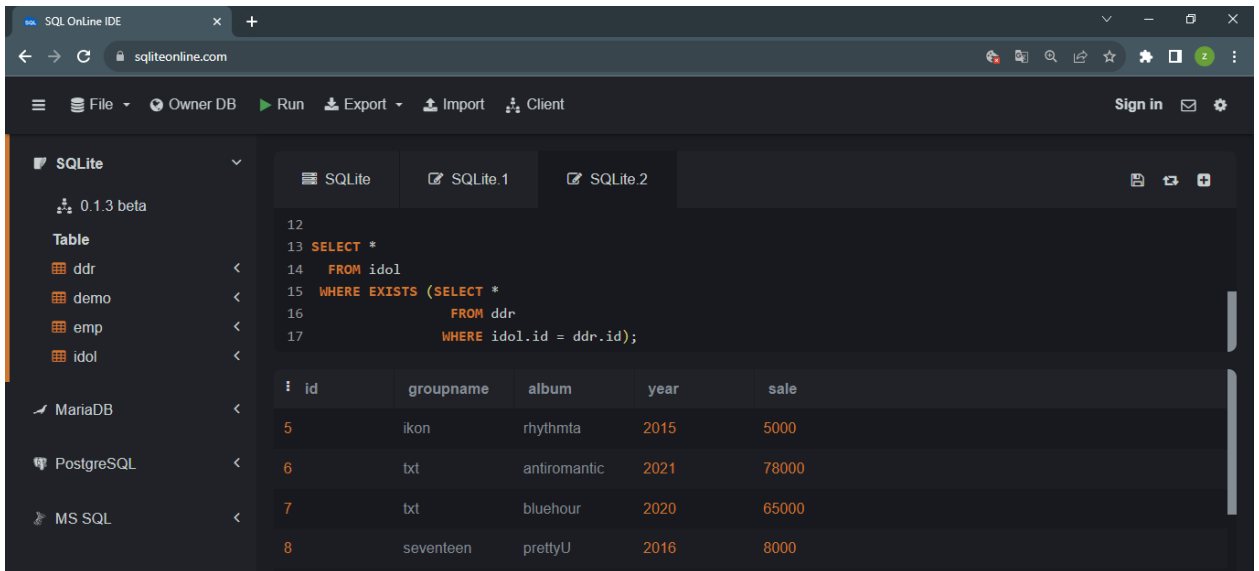
### **WHERE EXISTS ( subquery );**

Subquery Ішкі сұрау - SELECT сөйлемі. Егер ішкі сұрау нәтижелер жиынында кем дегенде бір жазбаны қайтарса, EXISTS шарты TRUE мәніне бағаланады және EXISTS шарты қанағаттандырылады. Ішкі сұрау ешқандай жазбаларды қайтармаса, EXISTS шарты FALSE деп бағаланады және EXISTS шарты орындалмайды.

### **Ескерту**

EXISTS шартын пайдаланатын Oracle SQL сұрауы өте тиімсіз, себебі ішкі сұрау сыртқы сұрау кестесіндегі әрбір жол үшін қайта іске қосылады.

EXISTS шартын пайдаланбайтын сұраулардың көпшілігін жазудың жақсы жолдары бар.



Oracle INSERT ALL операторы бір INSERT операторы арқылы бірнеше жолдарды қосу үшін пайдаланылады. Жолдарды бір кестеге немесе бірнеше кестеге бір ғана SQL пәрменін пайдаланып кірістіруге болады.

Oracle/PLSQL жүйесіндегі INSERT ALL операторының синтаксисі:

## INSERT ALL

**INSERT INTO mytable (column1, column2, column\_n) VALUES (expr1, expr2, expr\_n)**

**INSERT INTO mytable (column1, column2, column\_n) VALUES (expr1, expr2, expr\_n)**

**INSERT INTO mytable (column1, column2, column\_n) VALUES (expr1, expr2, expr\_n)**

**SELECT \* FROM dual;**

The screenshot shows the SQL Worksheet interface. The query editor contains the following SQL code:

```

1 CREATE TABLE suppliers(
2   supplier_id NUMBER,
3   supplier_name VARCHAR(100)
4 )
5
6 INSERT ALL
7   INTO suppliers(supplier_id, supplier_name) VALUES(1, 'INTEL')
8   INTO suppliers(supplier_id, supplier_name) VALUES(2, 'Linux')
9   INTO suppliers(supplier_id, supplier_name) VALUES(3, 'Google')
10 SELECT * FROM dual;
11
12 SELECT * FROM suppliers

```

The results table is displayed below the query:

SUPPLIER_ID	SUPPLIER_NAME
1	INTEL
2	Linux
3	Google

Біз осы уақытқа дейін пайдаланып жүрген SQLite-та кейбір операторлар жұмыс жасамайтындықтан [livesql.oracle.com](https://livesql.oracle.com) пайдаланғаныңыз дұрыс.

### **Тапсырма**

Having, distinct, Union, EXISTS, INSERT ALL операторларын қолдану.