

Зертханалық жұмыс №5

Мамандық: Деректер туралы ғылым

Пән атауы: Деректер ғылымына кіріспе

Оқытушы: Базарбек Жания Пархатқызы

Функционалдық программалаудың элементтері. dplyr көмегімен деректерді өңдеу. TRUNCATE TABLE, UNION ALL, INTERSECT, MINUS, PIVOT, Subqueries операторларымен танысу және мүмкіндіктерін тексеру.

SQL TRUNCATE операторы барлық деректерден кестені тазалау үшін қолданылады. SQL TRUNCATE операторы SQL WHERE операторынсыз пайдаланылатын SQL DELETE операторына ұқсас. Негізгі айырмашылықтар:

SQL TRUNCATE мәлімдемесі оқиғалар журналында жойылған деректерді жазбайды.

SQL DELETE жолы жолды құлыптайды, ал SQL TRUNCATE бүкіл бетті құлыптайды. Нәтижесінде SQL TRUNCATE операторы ешбір мәнді қайтармайды, ал SQL DELETE операторы жойылған жолдардың санын қайтарады.

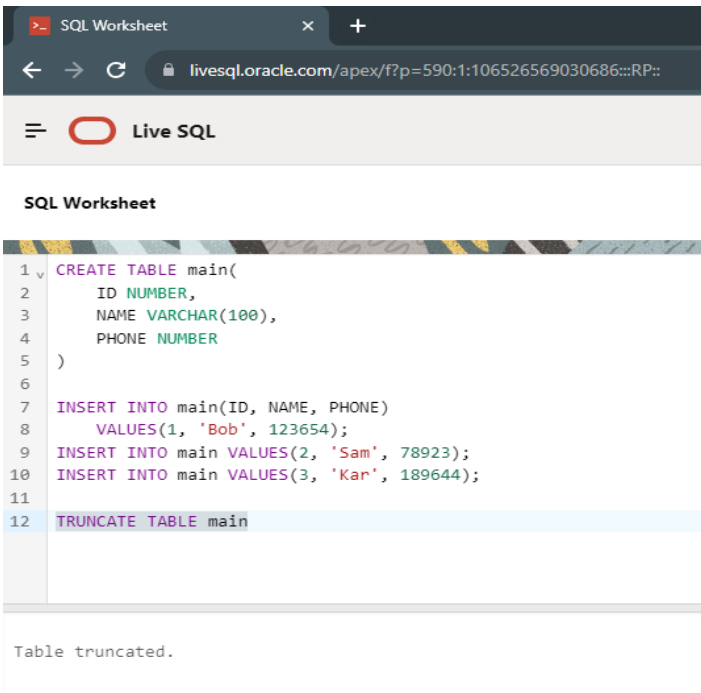
SQL DELETE операторын пайдаланғаннан кейін операцияны кері қайтаруға және жойылған деректерді қалпына келтіруге болады (ROLLBACK пәрмені). SQL TRUNCATE операторын пайдалану кезінде мұны істеу мүмкін емес, бірақ MS SQL Server ДҚБЖ-да операторды транзакцияларда пайдалануға болады.

SQL TRUNCATE операторының орындалуы әрбір ДҚБЖ үшін әртүрлі, сондықтан сіз пайдаланып жатқан жүйенің құжаттамасына жүгіну ұсынылады.

SQL TRUNCATE операторында келесі синтаксис бар:

TRUNCATE TABLE Table_name

Мысалы:



```
SQL Worksheet
livesql.oracle.com/apex/f?p=590:1:106526569030686::RP::
Live SQL

SQL Worksheet

1 CREATE TABLE main(
2   ID NUMBER,
3   NAME VARCHAR(100),
4   PHONE NUMBER
5 )
6
7 INSERT INTO main(ID, NAME, PHONE)
8   VALUES(1, 'Bob', 123654);
9 INSERT INTO main VALUES(2, 'Sam', 78923);
10 INSERT INTO main VALUES(3, 'Kar', 189644);
11
12 TRUNCATE TABLE main

Table truncated.
```

SQL UNION ALL операторы 2 немесе одан да көп SELECT операторларының нәтижелер жиынын біріктіру үшін қолданылады. Ол әртүрлі SELECT мәлімдемелері арасындағы қайталанатын жолдарды жоймайды (барлық жолдар қайтарылады).

UNION ALL ішіндегі әрбір SELECT мәлімдемесінде деректер түрлері бірдей нәтижелер жиындарында бірдей өрістер саны болуы керек.

- UNION операторы қайталанатын жолдарды жояды.
 - UNION ALL қайталанатын жолдарды жоймайды
- SQL тіліндегі UNION ALL операторының синтаксисі.

SELECT expression1, expression2, ... expression_n

FROM tables

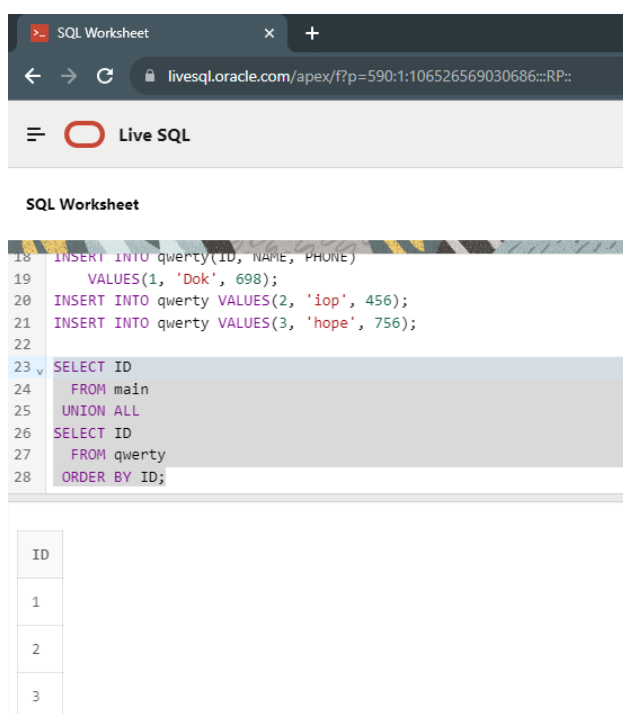
[WHERE conditions]

UNION ALL

SELECT expression1, expression2, ... expression_n

FROM tables

[WHERE conditions];



The screenshot shows a web browser window with the URL `livesql.oracle.com/apex/f?p=590:1:106526569030686::RP::`. The page title is "SQL Worksheet" and it features the "Live SQL" logo. Below the header, the SQL code is as follows:

```
18 INSERT INTO qwerty(ID, NAME, PHONE)
19 VALUES(1, 'Dok', 698);
20 INSERT INTO qwerty VALUES(2, 'iop', 456);
21 INSERT INTO qwerty VALUES(3, 'hope', 756);
22
23 SELECT ID
24 FROM main
25 UNION ALL
26 SELECT ID
27 FROM qwerty
28 ORDER BY ID;
```

The output of the query is displayed in a table with the following data:

ID
1
2
3

Oracle INTERSECT операторы 2 немесе одан да көп SELECT сұрауларының нәтижелерін қайтару үшін пайдаланылады. Дегенмен, ол барлық сұраулар немесе деректер жиындары үшін таңдалған жолдарды қайтарады. Жазба бір сұрауда болса, бірақ басқасында болмаса, ол INTERSECT нәтижелер жинағынан шығарылады.



INTERSECT сұрауы сұр көлеңкеленген аймақтағы жазбаларды қайтарады. Бұл SELECT 1 және SELECT 2 екеуінде де бар жазбалар.

INTERSECT бағдарламасындағы әрбір SELECT мәлімдемесінде деректер түрлері бірдей нәтижелер жиынындағы өрістердің саны бірдей болуы керек.

Oracle/PLSQL жүйесіндегі INTERSECT операторының синтаксисі:

```

SELECT expression1, expression2, ... expression_n
FROM tables
[WHERE conditions]
INTERSECT
SELECT expression1, expression2, ... expression_n
FROM tables
[WHERE conditions];

```

The screenshot shows a browser window with the URL `livesql.oracle.com/apex/f?p=590:1:106526569030686:::RP::`. The page title is 'SQL Worksheet' and it features the 'Live SQL' logo. Below the header, the SQL code is as follows:

```

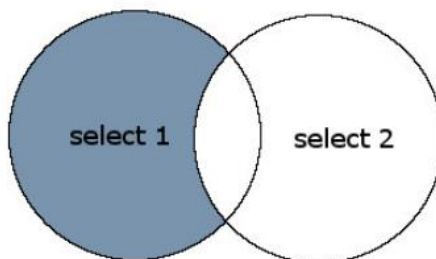
19 INSERT INTO qwerty(ID, NAME, PHONE)
20     VALUES(1, 'Dok', 698);
21 INSERT INTO qwerty VALUES(2, 'iop', 456);
22 INSERT INTO qwerty VALUES(3, 'hope', 756);
23 INSERT INTO qwerty VALUES(4, 'Nae', 1000);
24
25 SELECT NAME
26     FROM main
27 INTERSECT
28 SELECT NAME
29     FROM qwerty;

```

The output of the query is displayed in a table below the code:

NAME
Nae

Oracle MINUS операторы екінші SELECT қайтармаған бірінші SELECT сұрауының барлық жолдарын қайтару үшін пайдаланылады. Әрбір SELECT сұрауы деректер жиынын анықтайды. MINUS операторы бірінші деректер жиынындағы барлық жазбаларды қайтарады, содан кейін нәтижелерден екінші деректер жиынынан барлық жазбаларды жояды.



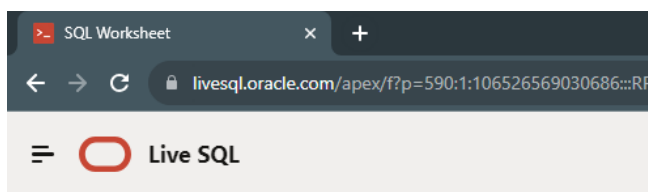
MINUS сұрауы сұр көлеңкелі аймақтағы жазбаларды қайтарады. Бұл SELECT 2-де емес, SELECT 1-де бар жазбалар.

MINUS сұрауындағы әрбір SELECT бірдей деректер түрлерімен нәтижелер жиындарында бірдей өрістерге ие болуы керек.

Oracle/PLSQL жүйесіндегі MINUS операторының синтаксисі:

```
SELECT expression1, expression2, ... expression_n  
FROM tables  
[WHERE conditions]  
MINUS  
SELECT expression1, expression2, ... expression_n  
FROM tables  
[WHERE conditions];
```

Мысалы:



SQL Worksheet

```
1  INSERT INTO main(ID, NAME, PHONE)
2     VALUES(1, 'Bob', 123654);
3  INSERT INTO main VALUES(2, 'Sam', 78923);
4  INSERT INTO main VALUES(3, 'Kar', 189644);
5  INSERT INTO main VALUES(4, 'Nae', 1000);
6  INSERT INTO main VALUES(5, 'Tyu', 2000);
7  INSERT INTO main VALUES(6, 'Kim', 3000);
8
9  INSERT INTO qwerty(ID, NAME, PHONE)
10     VALUES(1, 'Dok', 698);
11  INSERT INTO qwerty VALUES(2, 'iop', 456);
12  INSERT INTO qwerty VALUES(3, 'hope', 756);
13  INSERT INTO qwerty VALUES(4, 'Nae', 1000);
14
```

The screenshot shows a browser window with the URL `livesql.oracle.com/apex/f?p=590:1:10652656`. The page title is "SQL Worksheet". The code editor contains the following SQL statements:

```

10 VALUES(1, 'Dok', 698);
11 INSERT INTO qwerty VALUES(2, 'iop', 456);
12 INSERT INTO qwerty VALUES(3, 'hope', 756);
13 INSERT INTO qwerty VALUES(4, 'Nae', 1000);
14
15 SELECT ID
16 FROM main
17 MINUS
18 SELECT ID
19 FROM qwerty;

```

Below the code editor, a table displays the results of the query:

ID
5
6

Subqueries (подзапросы)

Oracle-да ішкі сұрау - сұрау ішіндегі сұрау. Сіз мәлімдемелеріңізде ішкі сұрауларды жасай аласыз. Бұл ішкі сұраулар WHERE сөйлемінде, FROM сөйлемінде немесе SELECT сөйлемінде болуы мүмкін.

WHERE

Көбінесе ішкі сұрау WHERE сөйлемінде қолданылады. Бұл ішкі сұраулар кірістірілген ішкі сұраулар деп те аталады.

Oracle PIVOT Oracle 11g жүйесінде қолданыла бастаған кестеаралық сұранысты жазуға мүмкіндік береді. Бұл нәтижелерді біріктіріп, жолдарды бағандарға айналдыра алатыныңызды білдіреді.

Oracle/PLSQL жүйесіндегі PIVOT операторының синтаксисі:

```

SELECT * FROM
(
  SELECT column1, column2
  FROM tables
  WHERE conditions
)
PIVOT
(
  aggregate_function(column2)
  FOR column2
  IN ( expr1, expr2, ... expr_n) | subquery
)
ORDER BY expression [ ASC | DESC ];

```

aggregate_function

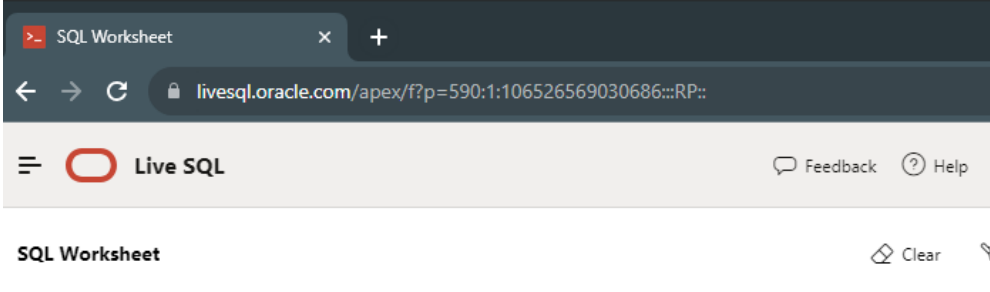
Бұл SUM, COUNT, MIN, MAX немесе AVG сияқты функция болуы мүмкін.

IN (expr1, expr2, ... expr_n)

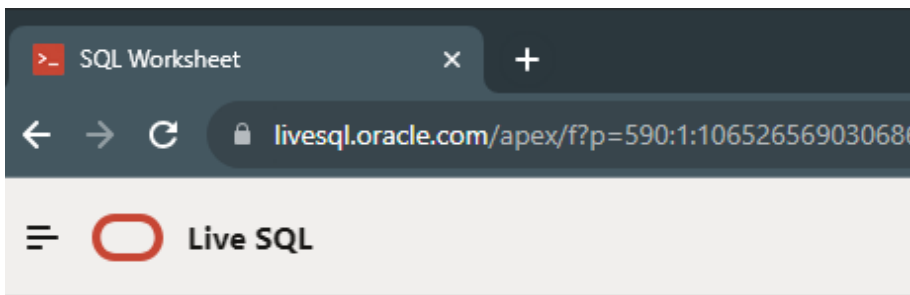
Айқас кестелік сұрау нәтижесі тақырыбындағы 2-бағанды айналдыруға арналған мәндер тізімі.

subquery

Мәндер тізімінің орнына ішкі сұрауды пайдалануға болады. Бұл жағдайда ішкі сұрау нәтижелері айқас кестелік сұрау нәтижесінің тақырыбындағы 2-бағанды айналдыру мәндерін анықтау үшін пайдаланылады.



```
22
23 v CREATE TABLE orders(
24     order_id integer NOT NULL,
25     customer_ref varchar2(50) NOT NULL,
26     order_date date,
27     product_id integer,
28     quantity integer,
29     CONSTRAINT orders_pk PRIMARY KEY (order_id)
30 );
31 INSERT INTO orders VALUES(1, 'aaa', to_date('17-12-1980','dd-mm-yyyy'), 10, 20);
32 INSERT INTO orders VALUES(2, 'bbb', to_date('11-10-1981','dd-mm-yyyy'), 20, 30);
33 INSERT INTO orders VALUES(3, 'ccc', to_date('10-12-1983','dd-mm-yyyy'), 40, 50);
34 INSERT INTO orders VALUES(4, 'rrr', to_date('16-11-1989','dd-mm-yyyy'), 60, 70);
35 INSERT INTO orders VALUES(5, 'yyy', to_date('27-12-1986','dd-mm-yyyy'), 80, 90);
36 INSERT INTO orders VALUES(6, 'ccc', to_date('10-12-1983','dd-mm-yyyy'), 40, 50);
37 INSERT INTO orders VALUES(7, 'rrr', to_date('16-11-1989','dd-mm-yyyy'), 60, 70);
38 INSERT INTO orders VALUES(8, 'yyy', to_date('27-12-1986','dd-mm-yyyy'), 80, 90);
39
```



SQL Worksheet

```
41 v SELECT order_id, customer_ref, product_id
42 FROM orders
43 ORDER BY order_id;
44
45 v SELECT * FROM(
46     SELECT customer_ref, product_id
47     FROM orders)
48 PIVOT(
49     COUNT(product_id)
50     FOR product_id IN (40, 60, 80))
51 ORDER BY customer_ref;
```

CUSTOMER_REF	40	60	80
aaa	0	0	0
bbb	0	0	0
ccc	2	0	0
rrr	0	2	0

Тапсырма

TRUNCATE TABLE, UNION ALL, INTERSECT, MINUS, PIVOT, Subqueries операторларын қолдану.