SET Operators

Version 1.0

codingcircuit23@gmail.com

1 - SET OPERATORS

SET operators are used to combine the data from one or more select queries. SET operators combines the results of two independent queries into a single result. SET Operators are also called as vertical joins because they combine data from two or more SELECT statements based on columns instead of rows.

Syntax:-

```
SELECT column_list FROM table_name
SET_OPERATOR
SELECT column_list FROM table_name
ORDER BY column_list;
```

- ✓ All the SELECT statements in the query should have the same number of columns in the columns_list.
- ✓ Data types of the corresponding columns in the column_lists in all the SELECT statements should be same.
- ✓ The column names to be displayed on the screen, has to be specified in the first query.
- ✓ Data types of the column lists must be compatible
- ✓ Use the ORDER BY clause in the last select statement to sort the results.

Following are the different SET operators used

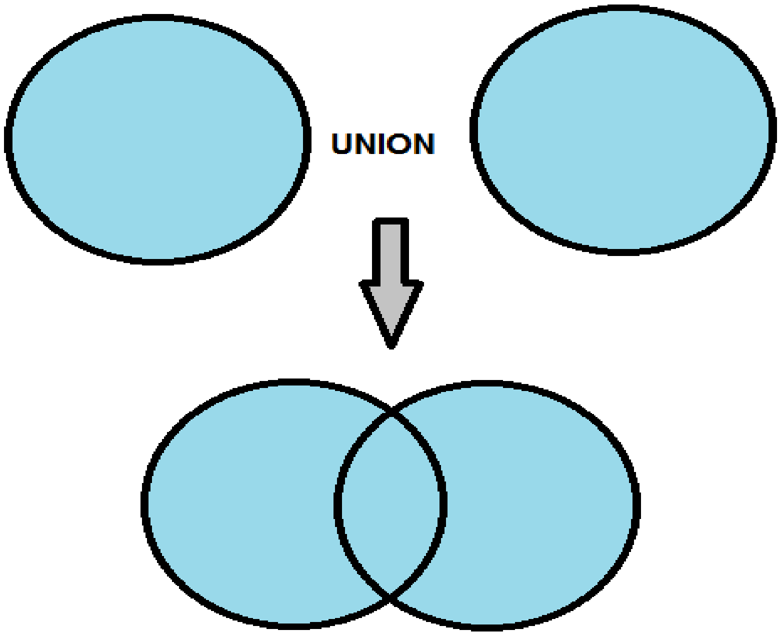
- UNION
- UNION ALL
- INTERSECT
- MINUS

1.1 UNION

UNION combines the result of two SELECT statements into one result set, and then eliminates any duplicates rows from that result.

Syntax:-

```
SELECT column_list FROM table_name
UNION
SELECT column_list FROM table_name
ORDER BY column_list;
```



Example:-

Table1		
Column A	**Column B**	**Column C**
A100	A	100
B200	B	200
D400	D	400

UNION

Table2		
Column D	**Column E**	**Column F**
B200	B	200
C300	C	300
D400	D	400

Select * from Table1
UNION
Select * from Table2

Output

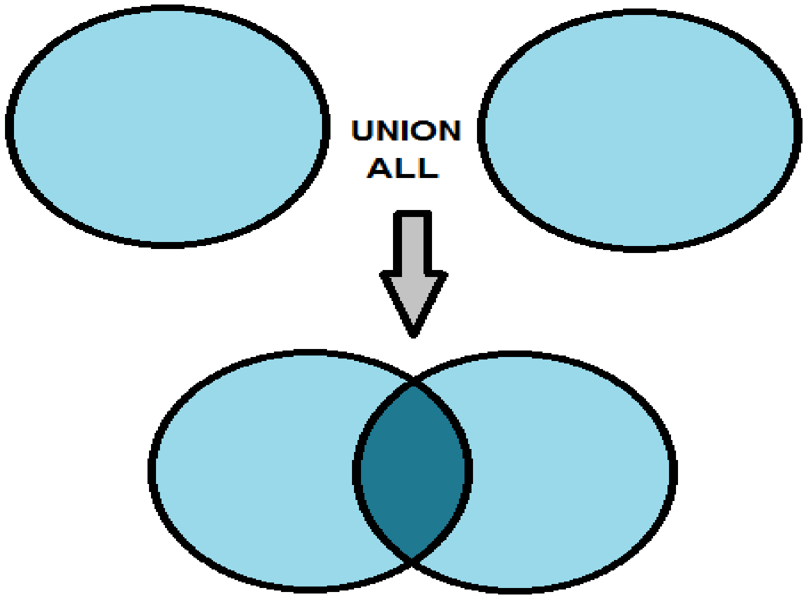
Column A	**Column B**	**Column C**
A100	A	100
B200	B	200
C300	C	300
D400	D	400

1.2 UNION ALL

UNION ALL combines the result of two SELECT statements into one result set. The duplicates are not removed.

Syntax:-

```
SELECT column_list FROM table_name
UNION ALL
SELECT column_list FROM table_name
ORDER BY column_list;
```



Example:-

Table1		
Column A	**Column B**	**Column C**
A100	A	100
B200	B	200
D400	D	400

UNION ALL

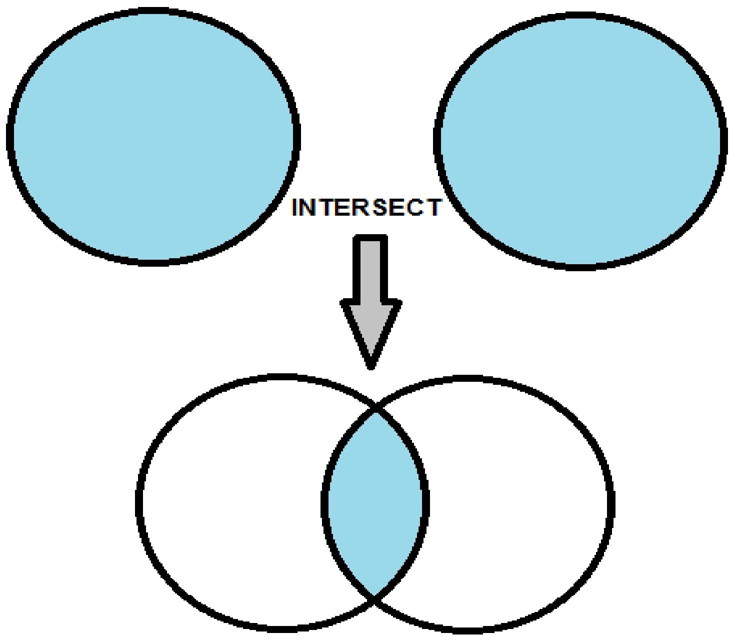
Table2		
Column D	**Column E**	**Column F**
B200	B	200
C300	C	300
D400	D	400

Select * from Table1
UNION ALL
Select * from Table2

Output

Column A	**Column B**	**Column C**
A100	A	100
B200	**B**	**200**
D400	*D*	*400*
B200	**B**	**200**
C300	C	300
D400	*D*	*400*

SELECT column_list FROM table_name
INTERSECT
SELECT column_list FROM table_name
ORDER BY column_list;



Example:-

Table1		
Column A	**Column B**	**Column C**
A100	A	100
B200	B	200
D400	D	400

INTERSECT

Table2		
Column D	**Column E**	**Column F**
B200	B	200
C300	C	300
D400	D	400

Select * from Table1
INTERSECT
Select * from Table2

Output

Column A	**Column B**	**Column C**
B200	B	200
D400	D	400

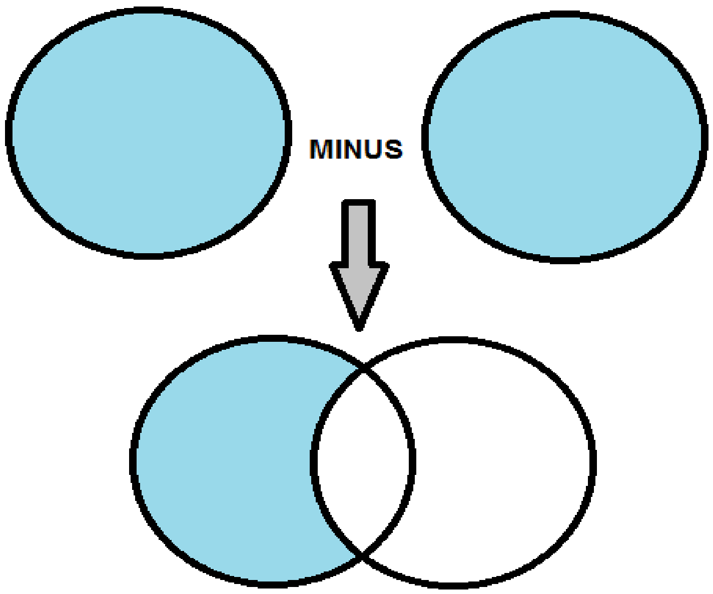
1.4 MINUS

MINUS is the operator, when applied will return the records which are retrieved only by the first SELECT statement and not by the second SELECT statement.

Syntax:-

SELECT column_list FROM table_name
MINUS

SELECT column_list FROM table_name
ORDER BY column_list;



Example:-

Table1		
Column A	**Column B**	**Column C**
A100	A	100
B200	B	200
D400	D	400

Table2		
Column D	**Column E**	**Column F**
B200	B	200
C300	C	300
D400	D	400

Case 1 :

Select * from Table1
MINUS

Select * from Table2

Output

Column A	**Column B**	**Column C**
A100	A	100

Case 1 :

Select * from Table2
MINUS
Select * from Table1

Output

Column A	**Column B**	**Column C**
C300	C	300