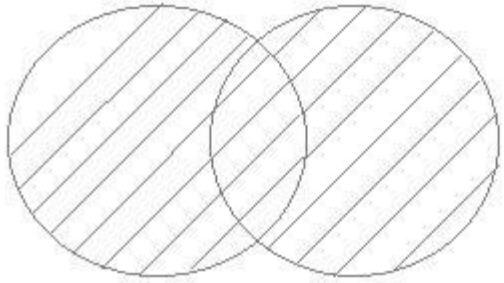
# SET OPERATOR

**Set operators** combine the results of two component queries into a single result. Queries containing **set operators** are called compound queries.

| | |
|---|---|
| UNION | All distinct rows selected by either query |
| UNION ALL | All rows selected by either query, including all duplicates |
| INTERSECT | All distinct rows selected by both queries |
| MINUS | All distinct rows selected by the first query but not the second |

## Union:

UNION is used to combine the results of two or more Select statements. However, it will eliminate duplicate rows from its result set. In case of union, number of columns and datatype must be same in both the tables.



## Example of UNION

| ID | Name |
|---|---|
| 1 | abhi |
| 2 | adam |

First

| ID | Name |
|---|---|
| 2 | adam |
| 3 | Chester |

Second

**Query**

```
select * from First
UNION
select * from second
```

| ID | NAME |
|---|---|
| 1 | abhi |
| 2 | adam |
| 3 | Chester |

Output

UNION CLAUSE

```
SQL> create table client_master1
  2  (client_no varchar2(20),
  3  fname varchar2(30),
  4  city varchar2(20));

Table created.
```

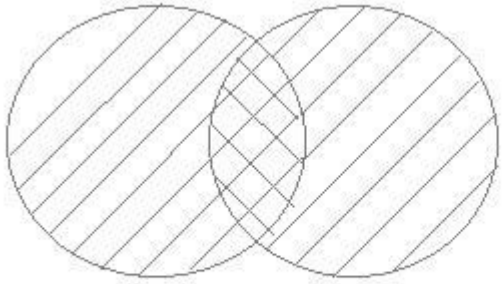
```
SQL> SELECT * FROM client_master1;

CLIENT_NO            FNAME                          CITY
-------------------- ------------------------------ ----------
C0001                ASHOK MEHRA                      MUMBAI
C0002                VISHAL PARIKH                   DELHI
C0003                AJAY MEHETA                     MUMBAI
C0004                ROHIT ROY                      CALCUTTA
C0005                NALINI DEEWAN;                  MUMBAI
C0006                PREM IYER                      DELHI
C0007                RAHUL DESAI                     BARODA
```

- When multiple SELECT queries are joined using UNION operator,oracle displays the combined result from all the compounded select queries,removing all duplicate and in sorted order.
- Same the number of columns must be selected by all participating select statements.
- Column name used in the display are taken from the first query.
- Data types of the column list must be compatible.
- Union and Intersect operators are commutative, i.e. the order of queries is not important, It doesn’t change the final result.

## Union All

This operation is similar to Union. But it also shows the duplicate rows.



## Example of UNION

| ID | Name |
|---|---|
| 1 | abhi |
| 2 | adam |

First

| ID | Name |
|---|---|
| 2 | adam |
| 3 | Chester |

Second

**Query**

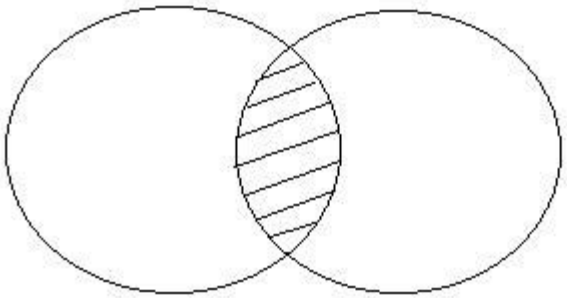
```
select * from First
UNION ALL
select * from second
```

| ID | NAME |
|---|---|
| 1 | abhi |
| 2 | adam |
| 2 | adam |
| 3 | Chester |

Output

## Intersect

Intersect operation is used to combine two SELECT statements, but it only retuns the records which are common from both SELECT statements. In case of **Intersect** the number of columns and datatype must be same. MySQL does not support INTERSECT operator.



## Example of Intersect

| ID | NAME |
|---|---|
| 1 | abhi |
| 2 | adam |

First

| ID | NAME |
|---|---|
| 2 | adam |
| 3 | Chester |

Second

Query

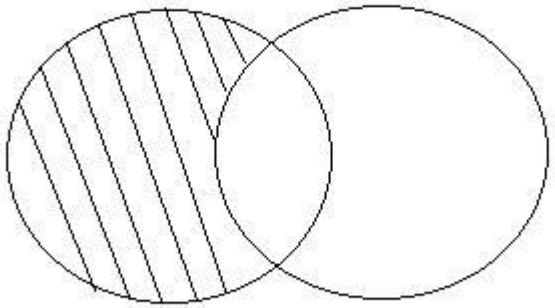
```
select * from First
INTERSECT
select * from second
```

| ID | NAME |
|---|---|
| 2 | adam |

output

## Minus

Minus operation combines result of two Select statements and return only those result which belongs to first set of result.



## Example of Minus

| ID | NAME |
|---|---|
| 1 | abhi |
| 2 | adam |

First

| ID | NAME |
|---|---|
| 2 | adam |
| 3 | Chester |

Second

Query

```
select * from First
Minus
select * from second
```

| ID | NAME |
|---|---|
| 1 | abhi |

output

**MINUS CLAUSE**

SQL> select * from sales_order_details;

| ORDERNO | PRODUCTNO |
|---|---|
| 1 | p1 |
| 2 | p2 |
| 3 | p3 |
| 4 | p4 |
| 5 | P5 |
| 6 | P6 |

SQL> select * from product_master_details;

| PRODUCTNO | DESCRIPTION |
|---|---|
| p1 | monitors |
| p3 | floppydisks |
| p4 | mouse |
| p5 | HDD |

```
SQL> select productno from sales_order_details
  2  minus
  3  select productno from product_master_details;

PRODUCTNO
------------------------------
p2
p6
```