

## **View:**

A view is a database object that has no values. Its contents are based on the base table.

It contains rows and columns similar to the real table. In MySQL, the view is a virtual table created by a query by joining one or more tables. It is operated similarly to the base table but does not contain any data of its own.

The view and table have one main difference that the views are definitions built on top of other tables (or views) . If any changes occur in the underlying table, the same changes are reflected in the view also.

## **Types of View in MySQL:**

- Simple View.
- Complex View.

## **Update salary in myview:**

```
mysql> update myview set salary = 7000 where id = 101;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from myview;
+----+-----+-----+-----+
| id | name  | salary | dno |
+----+-----+-----+-----+
| 101 | Valan | 7000  | 10  |
| 102 | Naveen | 4000  | 10  |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## **Update salary in Base Table (tbl\_employee1) :**

```
mysql> update tbl_employee1 set salary = null where dno = 10;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from myview;
+-----+-----+-----+-----+
| id | name | salary | dno |
+-----+-----+-----+-----+
| 101 | Valan | NULL | 10 |
| 102 | Naveen | NULL | 10 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from tbl_employee1;
+-----+-----+-----+-----+
| id | name | salary | dno |
+-----+-----+-----+-----+
| 101 | Valan | NULL | 10 |
| 102 | Naveen | NULL | 10 |
| 103 | Mahesh | 5000 | 20 |
| 104 | Gaythri | 6000 | 20 |
| 105 | Aravind | 7000 | 20 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## Types of View in MySQL:

- Simple View.
- Complex View.

### Simple View.

Contains only one single base table or is created from only one table.

```
mysql> create view myview as select * from tbl_employee1 where dno = 10;
Query OK, 0 rows affected (0.03 sec)

mysql> select * from myview;
+-----+-----+-----+-----+
| id | name | salary | dno |
+-----+-----+-----+-----+
| 101 | Valan | 3000 | 10 |
| 102 | Naveen | 4000 | 10 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

### Update salary in myview:

```
mysql> update myview set salary = 7000 where id = 101;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from myview;
+-----+-----+-----+-----+
| id | name  | salary | dno |
+-----+-----+-----+-----+
| 101 | Valan | 7000   | 10  |
| 102 | Naveen | 4000   | 10  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Update salary in Base Table (tbl\_employee1) :

```
mysql> update tbl_employee1 set salary = null where dno = 10;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from myview;
+-----+-----+-----+-----+
| id | name  | salary | dno |
+-----+-----+-----+-----+
| 101 | Valan | NULL   | 10  |
| 102 | Naveen | NULL   | 10  |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from tbl_employee1;
+-----+-----+-----+-----+
| id | name    | salary | dno |
+-----+-----+-----+-----+
| 101 | Valan   | NULL   | 10  |
| 102 | Naveen  | NULL   | 10  |
| 103 | Mahesh  | 5000   | 20  |
| 104 | Gaythri | 6000   | 20  |
| 105 | Aravind | 7000   | 20  |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## Joins:

Inner join is the default join , based on the condition the inner join displays the result which it is matched.

### Types of Joins:

- Inner Join.
- Outer Join.

4 syntax is possible to apply joins.

- 1) Table name with where clause.
- 2) Alias name with where clause.
- 3) Table name with join and on clause.
- 4) Alias name with join and on clause.

- Table name with where clause.(syntax)

```
mysql> select tbl_agents.Agent_code,tbl_agents.Agent_name,tbl_customers.Customer_name from tbl_agents,tbl_customers where tbl_agents.Working_area =tbl_customers.Customer_area;
```

- Alias name with where clause.(syntax)

```
mysql> select a.Agent_code,a.Agent_name,c.Customer_name from tbl_agents a,tbl_customers c where a.Working_area =c.Customer_area;
```

- Table name with join and on clause.(syntax)

```
mysql> select tbl_agents.Agent_code,tbl_agents.Agent_name,tbl_customers.Customer_name from tbl_agents join tbl_customers on tbl_agents.Working_area =tbl_customers.Customer_area;
```

- Alias name with join and on clause.(syntax)

```
mysql> select a.Agent_code,a.Agent_name,c.Customer_name from tbl_agents a join tbl_customers c on a.Working_area =c.Customer_area;
```

### Table name with where clause

```
mysql> select tbl_agent.AGENT_CODE,tbl_agent.AGENT_NAME,tbl_customers.CUSTOMER_NAME from tbl_agent,tbl_customers where tbl_agent.WORKING_AREA = tbl_customers.CUSTOMER_AREA;
```

AGENT_CODE	AGENT_NAME	CUSTOMER_NAME
A102	agent2	customer1
A101	agent1	customer1
A102	agent2	customer2
A101	agent1	customer2
A102	agent2	customer3
A101	agent1	customer3
A104	agent4	customer4
A103	agent3	customer4
A104	agent4	customer5
A103	agent3	customer5

10 rows in set (0.00 sec)

Alias name with where clause:

```
mysql> select a.AGENT_CODE, a.AGENT_NAME, c.CUSTOMER_NAME from tbl_agent a, tbl_customers c where a.WORKING_AREA = c.CUSTOMER_AREA;
```

AGENT_CODE	AGENT_NAME	CUSTOMER_NAME
A102	agent2	customer1
A101	agent1	customer1
A102	agent2	customer2
A101	agent1	customer2
A102	agent2	customer3
A101	agent1	customer3
A104	agent4	customer4
A103	agent3	customer4
A104	agent4	customer5
A103	agent3	customer5

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

### Table name with join and on clause:

```
mysql> select tbl_agents.Agent_code,tbl_agents.Agent_name,tbl_customers.Customer_name from tbl_agents join tbl_customers on tbl_agents.Working_area =tbl_customers.Customer_area;
```

Agent_code	Agent_name	Customer_name
A102	agent2	customer1
A101	agent1	customer1
A102	agent2	customer2
A101	agent1	customer2
A102	agent2	customer3
A101	agent1	customer3
A104	agent4	customer4
A103	agent3	customer4
A104	agent4	customer5
A103	agent3	customer5

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

### Alias name with join and on clause:

```
mysql> select a.AGENT_CODE, a.AGENT_NAME, c.CUSTOMER_NAME from tbl_agent a join tbl_customers c on a.WORKING_AREA = c.CUSTOMER_AREA;
```

AGENT_CODE	AGENT_NAME	CUSTOMER_NAME
A102	agent2	customer1
A101	agent1	customer1
A102	agent2	customer2
A101	agent1	customer2
A102	agent2	customer3
A101	agent1	customer3
A104	agent4	customer4
A103	agent3	customer4
A104	agent4	customer5
A103	agent3	customer5

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

### Types of Inner Join:

- Equi Join
- NON-Equi Join

## Equi Join:

```
mysql> select a.AGENT_CODE, a.AGENT_NAME, c.CUSTOMER_NAME from tbl_agent a inner join tbl_customers c on a.WORKING_AREA = c.CUSTOMER_AREA;
```

AGENT_CODE	AGENT_NAME	CUSTOMER_NAME
A102	agent2	customer1
A101	agent1	customer1
A102	agent2	customer2
A101	agent1	customer2
A102	agent2	customer3
A101	agent1	customer3
A104	agent4	customer4
A103	agent3	customer4
A104	agent4	customer5
A103	agent3	customer5

10 rows in set (0.00 sec)

## NON-Equi Join:

```
mysql> select a.AGENT_CODE, a.AGENT_NAME, c.CUSTOMER_NAME from tbl_agent a inner join tbl_customers c on a.WORKING_AREA != c.CUSTOMER_AREA;
```

AGENT_CODE	AGENT_NAME	CUSTOMER_NAME
A104	agent4	customer1
A103	agent3	customer1
A104	agent4	customer2
A103	agent3	customer2
A104	agent4	customer3
A103	agent3	customer3
A102	agent2	customer4
A101	agent1	customer4
A102	agent2	customer5
A101	agent1	customer5

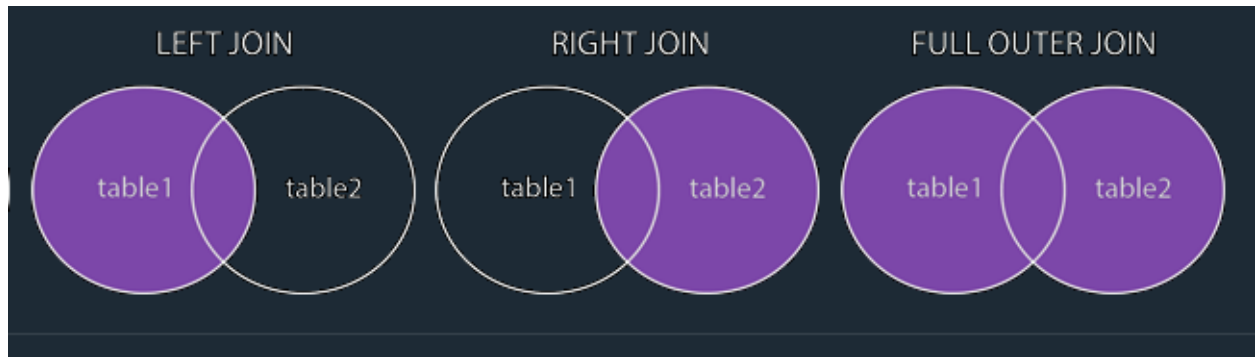
10 rows in set (0.00 sec)

## Outer Join:

### Types of Outer Join:

- Left Outer Join.
- Right Outer Join.
- Full Outer Join. (MySQL does not support Full outer join).
- Cross join.

- **LEFT (OUTER) JOIN** : Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN** : Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN** : Returns all records when there is a match in either left or right table



### Left Outer Join.

```
mysql> select * from tbl_agents a left outer join tbl_customers c on a.Working_area= c. Customer_area;
```

Agent_code	Agent_name	Working_area	Customer_code	Customer_name	Customer_area
A101	agent1	chennai	C103	customer3	chennai
A101	agent1	chennai	C102	customer2	chennai
A101	agent1	chennai	C101	customer1	chennai
A102	agent2	chennai	C103	customer3	chennai
A102	agent2	chennai	C102	customer2	chennai
A102	agent2	chennai	C101	customer1	chennai
A103	agent3	Bangalore	C105	customer5	Bangalore
A103	agent3	Bangalore	C104	customer4	Bangalore
A104	agent4	Bangalore	C105	customer5	Bangalore
A104	agent4	Bangalore	C104	customer4	Bangalore
A105	agent5	Kolkata	NULL	NULL	NULL

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

### Right Outer Join:

```
mysql> select * from tbl_agents a right outer join tbl_customers c on a.Working_area= c. Customer_area;
```

Agent_code	Agent_name	Working_area	Customer_code	Customer_name	Customer_area
A102	agent2	chennai	C101	customer1	chennai
A101	agent1	chennai	C101	customer1	chennai
A102	agent2	chennai	C102	customer2	chennai
A101	agent1	chennai	C102	customer2	chennai
A102	agent2	chennai	C103	customer3	chennai
A101	agent1	chennai	C103	customer3	chennai
A104	agent4	Bangalore	C104	customer4	Bangalore
A103	agent3	Bangalore	C104	customer4	Bangalore
A104	agent4	Bangalore	C105	customer5	Bangalore
A103	agent3	Bangalore	C105	customer5	Bangalore
NULL	NULL	NULL	C106	customer6	Delhi

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

### Full Outer Join. (MySQL does not support Full outer join).

```
mysql> select * from tbl_agents a full outer join tbl_customers c on a.Working_area= c. Customer_area;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'full outer jo
in tbl_customers c on a.Working_area= c. Customer_area' at line 1
mysql>
```

## Cross join:

```
mysql> select * from tbl_agents cross join tbl_customers;
```

Agent_code	Agent_name	Working_area	Customer_code	Customer_name	Customer_area
A105	agent5	Kolkata	C101	customer1	chennai
A104	agent4	Bangalore	C101	customer1	chennai
A103	agent3	Bangalore	C101	customer1	chennai
A102	agent2	chennai	C101	customer1	chennai
A101	agent1	chennai	C101	customer1	chennai
A105	agent5	Kolkata	C102	customer2	chennai
A104	agent4	Bangalore	C102	customer2	chennai
A103	agent3	Bangalore	C102	customer2	chennai
A102	agent2	chennai	C102	customer2	chennai
A101	agent1	chennai	C102	customer2	chennai
A105	agent5	Kolkata	C103	customer3	chennai
A104	agent4	Bangalore	C103	customer3	chennai
A103	agent3	Bangalore	C103	customer3	chennai
A102	agent2	chennai	C103	customer3	chennai
A101	agent1	chennai	C103	customer3	chennai
A105	agent5	Kolkata	C104	customer4	Bangalore
A104	agent4	Bangalore	C104	customer4	Bangalore
A103	agent3	Bangalore	C104	customer4	Bangalore
A102	agent2	chennai	C104	customer4	Bangalore
A101	agent1	chennai	C104	customer4	Bangalore
A105	agent5	Kolkata	C105	customer5	Bangalore
A104	agent4	Bangalore	C105	customer5	Bangalore
A103	agent3	Bangalore	C105	customer5	Bangalore
A102	agent2	chennai	C105	customer5	Bangalore
A101	agent1	chennai	C105	customer5	Bangalore
A105	agent5	Kolkata	C106	customer6	Delhi
A104	agent4	Bangalore	C106	customer6	Delhi
A103	agent3	Bangalore	C106	customer6	Delhi
A102	agent2	chennai	C106	customer6	Delhi
A101	agent1	chennai	C106	customer6	Delhi

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

```
mysql>
```