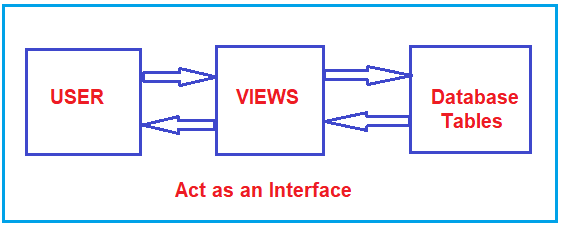
**What is a View in MySQL?**

VIEW is a database object that can be created like a table. In SQL, a VIEW is similar to a virtual table. But unlike tables VIEWS don’t actually store data. VIEWS are complex SELECT statements used as virtual tables for ease of reference and reuse. The VIEWS are useful for storing complex SQL statements as a virtual table and request the VIEW as a single table instead of a complex query.

For security purposes, we can restrict users from accessing underlying tables and instead give access to views or virtual tables with limited columns. Since, every time user request view, the database engine recreates the result set, which always returns up-to-date data rows from views.

So, when we query a view, it actually gets the data from the underlying database tables as shown in the below image.



So, in simple words, we can say that the views in MySQL act as an interface between the actual database table(s) and the user.

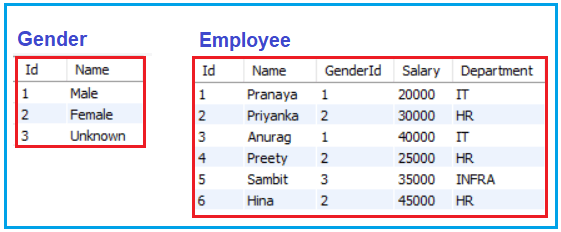
**What are the differences between a table and a view in MySQL?**

When we compared a view with a table, we have the following differences.

1. The table is physical i.e. it is an actual table whereas the view is logical i.e. it is a virtual table.
2. A Table is an independent object whereas a view is a dependent object. That is a view depends on the table(s) from which it is going to fetch the data.
3. The Table stores the actual data of the database whereas View creates a logical subset of data from one or more tables.
4. When a new table is created from an existing table, the new and old tables are independent themselves, that is the changes of one table will not be reflected into the other table whereas if a view is created based on a table, any changes that are performed on the table will reflect into the view and any changes performed on the view reflected in the table also.

**Examples to understand Views in MySQL**

We are going to use the following Gender and Employee table to understand the Views in MySQL.



**Please use the below SQL Script to create and populate the database EmployeeDB, and Gender and Employee table with the required sample data.**

**CREATE** **DATABASE** EmployeeDB;

**USE** EmployeeDB;

-- Create Gender Table

**CREATE** **TABLE** Gender

(

Id **INT** **PRIMARY KEY**,

Name **VARCHAR**(50)

);

-- Populate the Gender Table with test data

**INSERT** **INTO** Gender **VALUES**(1, 'Male');

**INSERT** **INTO** Gender **VALUES**(2, 'Female');

**INSERT** **INTO** Gender **VALUES**(3, 'Unknown');

-- Create Employee Table

**CREATE** **TABLE** Employee

(

Id **INT** **PRIMARY KEY**,

Name **VARCHAR**(50),

GenderId **VARCHAR**(50),

Salary **INT**,

Department **VARCHAR**(50)

);

-- Populate the Employee Table with test data

**INSERT** **INTO** Employee **VALUES**(1, 'Pranaya', 1, 20000, 'IT');

**INSERT** **INTO** Employee **VALUES**(2, 'Priyanka', 2, 30000, 'HR');

**INSERT** **INTO** Employee **VALUES**(3, 'Anurag', 1, 40000, 'IT');

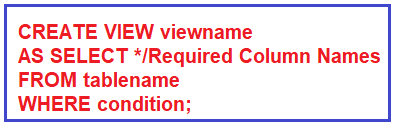
**INSERT** **INTO** Employee **VALUES**(4, 'Preety', 2, 25000, 'HR');

**INSERT** **INTO** Employee **VALUES**(5, 'Sambit', 3, 35000, 'INFRA');

**INSERT** **INTO** Employee **VALUES**(6, 'Hina', 2, 45000, 'HR');

**How to Create a View in MySQL?**

The Syntax to Create a View in MySQL is shown below. As you can see in the below image, the MySQL Views will be created by using a select statement.



**How many types of views are there in MySQL?**

There are two types of views in MySQL, they are as follows.

1. **Simple view or Updatable views**
2. **Complex view or non-updatable views.**

**What is a simple view or an Updatable view in MySQL?**

The view that is created based on the columns of a single table is known as a simple view in MySQL. In the Simple View, we can perform all the DML operations, so it is also called an updatable view. Let us understand Simple Views in MySQL with Examples.

**Simple View with All Columns:**

Now we will create a view that will retrieve all the column data from the Employee table.

**CREATE VIEW vwEmployee AS SELECT \* FROM Employee;**

Once you execute the above statement, the view **vwEmployee**is created inside the Views folder as shown in the below image.



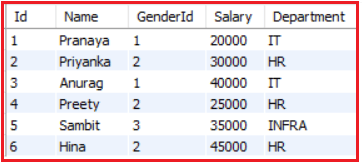
As the above view is created based on a single table i.e. Employee table, so it is called a Simple view and updatable view. As we already discussed we can perform all the DML Operations on Simple view. Let us prove this.

**Retrieve Operation on MySQL View:**

To retrieve the data from a view, we need to use the SELECT statement, in the same way, we used to select the data from a table. So, the following SQL query will retrieve all the data from the Employee table as the view created on the Employee table.

**SELECT \* FROM vwEmployee;**

When you execute the above SQL Statement, you will get the following output. Here, we are querying the view which internally fetches the data from the underlying base table i.e. Employee.



**Insert Operation on MySQL View:**

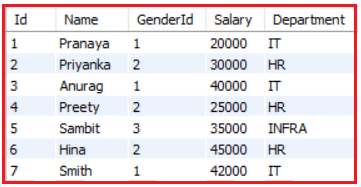
It is possible in MySQL to perform INSERT Operation on Simple View. In this case, the record actually inserted into the underlying base table. Let us try to insert one record using our vwEmployee view by executing the below SQL INSERT Statement.

**INSERT INTO vwEmployee VALUES(7, ‘Smith’, 1, 42000, ‘IT’);**

When you execute the above INSERT statement, one record is inserted into the Employee table. To check the same, execute the below SELECT query.

**SELECT \* FROM Employee;**

Once you execute the above SQL Statement, you will get the following output and notice the new record with id 7 is there in the Employee table.



**Update Operation on MySQL Views:**

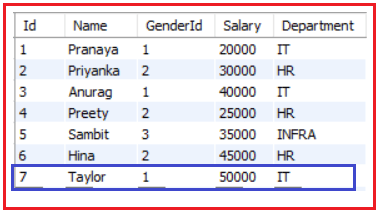
It is also possible in MySQL Simple Views to Perform UPDATE DML Operation. Let’s try to update the Name and Salary of the employee whose id is 7 using the vwEmployee view by executing the below UPDATE Statement.

**UPDATE vwEmployee SET Name = ‘Taylor’, Salary = 50000 WHERE Id = 7;**

Once you execute the above UPDATE SQL Statement, the Employee Name and Salary is Updated whose Id is 7. To verify whether the data is updated successfully or not make a select query against the Employee table by executing the below SELECT Statement.

**SELECT \* FROM Employee;**

Once you execute the above SELECT Statement, you will get the following output and notice the Name and Salary of the Employee who’s with id 7 is updated in the Employee table as expected as shown in the below image.



**Delete Operation on MySQL View:**

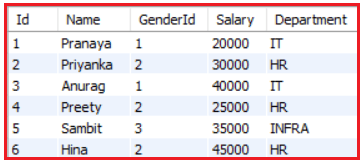
It is also possible in MySQL Simple view to perform the DML DELETE Operation. Let’s try to Delete the Employee whose ID is 7 using the vwEmployee view by executing the below DELETE Statement.

**DELETE FROM vwEmployee where ID = 7;**

Once you execute the above DELETE Statement, then check the Employee table and you will see that the Employee whose ID is 7 is deleted as expected from the Employee table. To verify whether the same, issue a select query against the Employee table by executing the below SELECT Statement.

**SELECT \* FROM Employee;**

Once you execute the above SELECT Statement, you will get the following output and notice that the Employee who’s with id 7 is no longer exists in the Employee table.



So, this proofs that we can perform ALL the DML operations on a Simple View in MySQL which is also called an Updatable View.

**Simple View with Required Columns in MySQL:**

The view that we created in our example, retrieves all the columns from the Employee table. Now we will see how to create a view with specific columns from the Employee table. We want all the columns except the Salary column. To do so, please execute the following SQL Statement.

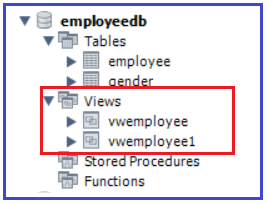
**CREATE** **VIEW** vwEmployee1

**AS**

**SELECT** Id, Name, GenderId, Department

**FROM** Employee;

Once you execute the above CREATE VIEW statement, the view **vwEmployee1**is created inside the Views folder as shown in the below image.



In this vwEmployee1 view, we can also perform the SELECT, INSERT, UPDATE, and DELETE operations as the above view is also a simple or you can say updatable view.

**How to DROP a View in MySQL?**

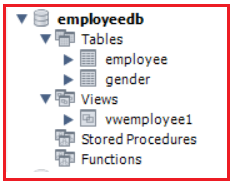
To drop a view in MySQL, we need to use the following syntax.

**DROP VIEW viewname;**

As we have created two views so far, let us delete the **vwEmployee** view by executing the below statements.

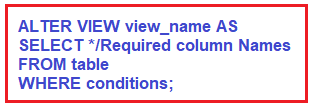
**DROP VIEW vwEmployee;**

Once you execute the above statement, the **vwEmployee** view should be deleted from the Views folder as shown in the below image.



**How to Modify a view in MySQL?**

Once the view is created, later if you want to update the view, then you can also do the same in MySQL. To modify or update the already created view without dropping it we can use ALTER VIEW statement. Following is the syntax.



For example, from our **vwEmployee1**view, we also want to remove the GenderId column. Then we can do the same by executing the below ALTER View Statement.

**ALTER** **VIEW** vwEmployee1

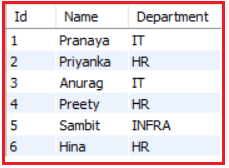
**AS**

**SELECT** Id, Name, Department

**FROM** Employee;

Now, issue a select query against the view and you will see that only three columns’ data are returned from the Employee table.

**SELECT \* FROM vwEmployee1;** will give you the following output.



**What is a complex view in MySQL?**

When we create a view based on more than 1 table by using MySQL JOIN, then it is known as a complex view and on a complex view, we may or may not perform DML operations. So, a complex view is also called a non-updatable view.