Rename View

There are various SQL statements that perform different operations on database objects, such as creating, updating, deleting and also renaming a database object. And since a view is also a database object, all these operations can also be performed on a view, you can create a view, update a view, delete a view and also rename a view.

There is no direct query to rename a view in SQL. In MySQL we can rename a view using the **RENAME TABLE** statement and in MS SQL Server we can rename a view using the **sp_rename procedure**.

In many cases, deleting the existing view and then re-creating it with a new name is rather recommended.

Renaming a View in MySQL

The **RENAME TABLE** statement in MySQL database is used to rename views. You just have to make sure that the new name of the view does not overlap with the name of any existing views.

Syntax

Following is the basic syntax to rename a view in MySQL

```
RENAME TABLE old_view_name TO new_view_name;
```

Here, we must ensure that old view name is present in the database and that new view name does not already exist. Otherwise, it will issue a warning. Also before executing this statement, it is important to make sure that the table is not locked and there are no active transactions.

Example

In this example, let us first create a table with the name CUSTOMERS which contains the personal details of customers including their name, age, address and salary etc. As shown below:

```
CREATE TABLE CUSTOMERS (
   ID INT NOT NULL,
   NAME VARCHAR (20) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR (25),
   SALARY DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

Now insert values into this table using the INSERT statement as follows:

```
INSERT INTO CUSTOMERS
VALUES
   (1, 'Ramesh', 32, 'Ahmedabad', 2000.00),
   (2, 'Khilan', 25, 'Delhi', 1500.00),
   (3, 'Kaushik', 23, 'Kota', 2000.00),
   (4, 'Chaitali', 25, 'Mumbai', 6500.00),
   (5, 'Hardik', 27, 'Bhopal', 8500.00),
   (6, 'Komal', 22, 'Hyderabad', 4500.00),
   (7, 'Muffy', 24, 'Indore', 10000.00);
```

The table will be created as follows:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	Hyderabad	4500.00
7	Muffy	24	Indore	10000.00

Following query creates a view based on the above created table:

```
CREATE VIEW CUSTOMERS_VIEW
AS
SELECT *
FROM CUSTOMERS
WHERE AGE > 25;
```

You can verify the contents of a view using the select query as shown below:

```
SELECT *
FROM CUSTOMERS_VIEW;
```

The view is displayed as follows:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
5	Hardik	27	Bhopal	8500.00

Now we know that a view with the name **CUSTOMERS_VIEW** exists in our database. So, we are directly going to rename this view to **VIEW_CUSTOMERS**, using the following

query

```
RENAME TABLE CUSTOMERS_VIEW TO VIEW_CUSTOMERS;
```

Output

The result obtained is as shown below

```
Query OK, 0 rows affected (0.08 sec)
```

Verification

We can verify whether the view is renamed or not by retrieving its contents using its new name in the SELECT statement. Following is the query to display the records in the VIEW_CUSTOMERS view:

```
SELECT * FROM VIEW_CUSTOMERS;
```

The view displayed is as follows:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
5	Hardik	27	Bhopal	8500.00

Renaming a View in SQL Server

There isn't a query in SQL Server that can rename a view directly. But, it does give you access to a stored procedure called sp_rename that can rename a view. You have to make sure there are no active transactions being performed on the view using its old name before renaming it.

The **sp_rename** is a system stored procedure (set of pre-built subroutines that perform tasks within the database) in SQL that can be used to rename various database objects including tables, columns, indexes, and constraints.

Syntax

Following is the basic syntax to rename a view in SQL:

```
EXEC sp_rename 'old_view_name', 'new_view_name';
```

Here, we must ensure that old view name is present in the database and that new view name does not already exist. Otherwise, it will issue a warning. Before executing this statement, it is important to make sure that the table is not locked and there are no active transactions.

Example

In this example, let us first try to create a table with the name **CUSTOMERS** which contains the personal details of customers including their name, age, address and salary etc. As shown below:

```
CREATE TABLE CUSTOMERS (
   ID INT NOT NULL,
   NAME VARCHAR (20) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR (25),
   SALARY DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

Now insert values into this table using the INSERT statement as follows:

```
INSERT INTO CUSTOMERS
VALUES
   (1, 'Ramesh', 32, 'Ahmedabad', 2000.00),
   (2, 'Khilan', 25, 'Delhi', 1500.00),
   (3, 'Kaushik', 23, 'Kota', 2000.00),
   (4, 'Chaitali', 25, 'Mumbai', 6500.00);
```

Following query creates a view based on the above created table:

```
CREATE VIEW CUSTOMERS_VIEW AS
SELECT *
FROM CUSTOMERS
WHERE SALARY > 2000;
```

You can verify the contents of a view using the select query as shown below:

```
SELECT *
FROM CUSTOMERS_VIEW;
```

The view will be created as:

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500.00

Now, we know that we have an existing view **CUSTOMERS_VIEW** in our database. So, we are going to rename this view to **VIEW_CUSTOMERS**, using the following query:

```
EXEC sp_rename CUSTOMERS_VIEW, VIEW_CUSTOMERS;
```

Verification

We can verify whether the view is renamed or not by retrieving its contents using its new name in the SELECT statement. Following is the query to display the records in the VIEW_CUSTOMERS view:

```
SELECT * FROM VIEW_CUSTOMERS;
```

The view displayed is as follows:

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500.00

We have renamed the view to VIEW_CUSTOMERS; if the user tries to get the details by using the old view name, it will throw an error showing that the view does not exist.

Rules to be followed while Renaming Views

When renaming views in SQL, there are some rules and best practices that should be followed to ensure that the renaming process goes smoothly and does not cause any unintended consequences or issues.

Here are some general rules to keep in mind when renaming views in SQL:

- **Avoid renaming system views** − System views are views that contain all the information about the database management system. Renaming these views can cause issues with the functioning of the database system, so it is generally not recommended to rename system views.
- **Update all references to the view** − After renaming a view, any stored procedures, triggers, or other database objects that reference the view will need to be updated to use the new name of the view. Failure to update these references can result in errors or issues with the functioning of the database system.
- **Test thoroughly** − Before renaming a view in a production environment, it is important to test the renaming process thoroughly in a development or testing environment to ensure that all references to the view have been updated correctly and that the database system continues to function as expected.
- **Use a consistent naming convention** − It is a good practice to use a consistent naming convention for views and other database objects to make it easier to understand and maintain the database system. If you need to rename a view, consider following the same naming convention that you have used for other views in the database.
- **Backup the database** − Before renaming a view, it is recommended to create a backup of the database to ensure that you have a restore point; in case anything goes wrong during the renaming process.