SQL - Rename Table

SQL provides two ways to rename an MySQL table. You can use either SQL RENAME TABLE or ALTER TABLE statement to change a table name in MySQL RDBMS.

The SQL RENAME TABLE Statement

You can change a MySQL table name using SQL **RENAME TABLE** statement.

Syntax

Following is the syntax of the SQL RENAME TABLE Statement −

```
RENAME TABLE table_name TO new_table_name;
```

Where, **table_name** is the current name of an existing table and **new_table_name** is the new name of the table.

Example

Let us 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,
```

```
   PRIMARY KEY (ID)
);
```

Now, let us insert few records into this table using the INSERT statement as follows −

```
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (2, 'Khilan', 25, 'Delhi', 1500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (3, 'kaushik', 23, 'Kota', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (4, 'Chaitali', 25, 'Mumbai', 6500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (5, 'Hardik', 27, 'Bhopal', 8500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (6, 'Komal', 22, 'MP', 4500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (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

Following SQL Query changes the name of the **CUSTOMERS** table to **BUYERS** −

```
RENAME TABLE CUSTOMERS to BUYERS;
```

Verification

Once you change the name of a table, you can start using the new table name in your SQL queries.

```
SELECT * FROM BUYERS;
```

If table name got changed successfully, then it should list down all the records which were available in CUSTOMERS table.

The SQL ALTER TABLE Statement

The **ALTER TABLE** statement can be used to change or modify the structure of an existing table i.e. using this statement you can add/delete columns, create/destroy indexes, change the datatypes of the existing columns, rename the columns and, we can even rename the table.

Syntax

Following is the syntax of the SQL **ALTER TABLE** statement to rename an existing table −

```
ALTER TABLE table_name RENAME [TO|AS] new_table_name
```

Example

Following SQL **ALTER TABLE** statement will change the table name from **BUYERS** to **CUSTOMERS**.

```
ALTER TABLE BUYERS RENAME TO CUSTOMERS;
```

Once you change the name of the table to CUSTOMERS, you can start using this name in your SQL queries.

```
SELECT * FROM CUSTOMERS;
```

This will produce the following result:

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

Renaming a Table in SQL Server

There isn't a query in SQL Server that can rename a table directly. However, it does give you access to a stored procedure called **sp_rename** that enables you to rename a table.

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 table in SQL Server -

```
EXEC sp_rename 'old_table_name', 'new_table_name'
```

warning. Second important point is to make sure that the table is not locked and there is no active transaction involving this table.

Example

Assume we already have the CUSTOMERS table in our database. Now, we are going to rename this table from **CUSTOMERS** to **WORKERS**, using the following query -

```
EXEC sp_rename 'CUSTOMERS', 'WORKERS';
```

Output

The result obtained is as shown below -

```
Completion time: 2023-08-15T19:21:49.1144163+05:30
```

Verification

We can verify whether the changes are reflected by retrieving its contents using the SELECT statement as follows -

```
SELECT * FROM WORKERS;
```

This will list down all the records available in WORKERS table 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