SQL Table

Table is a collection of data, organized in terms of rows and columns. In DBMS term, table is known as relation and each row is a tuple.

Let's see an example of an student table:

Reg_No	Name	Branch
20184165	Saurav Chaudhary	CSE
20184063	Rajan Kr Jaiswal	CSE
20184092	Rohit Kumar	CSE
20184192	Lokesh Raj Singhi	CSE

In the above table, "Student" is the table name, "Reg_No", "Name" and "Branch" are the column names. The combination of data of multiple columns forms a row.

SQL TABLE Variable

The **SQL Table variable** is used to create, modify, rename, copy and delete tables.

The SQL CREATE TABLE Statement

The CREATE TABLE statement is used to create a new table in a database.

Syntax

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

Here the column parameters specify the names of the column table and datatype decide the type of data to be stored in that column.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

> For complete detail about datatype you can refer to Introduction section.

SQL CREATE TABLE Example

```
CREATE TABLE Students(
    Reg_No int,
    Name varchar(255),
    Branch varchar(255)
);
```

Here Reg_No column is of type integer , Name & Branch is of type varchar.

The empty "Students" table will now look like this:

Reg_No	Name	Branch

Create Table Using Another Table

We can create a copy of an existing table can also be created using CREATE TABLE.

The new table gets the same column definitions. All columns or specific columns can be selected.

If you create a new table using an existing table, the new table will be filled with the existing values from the old table.

Syntax

CREATE TABLE *new_table_name* AS
SELECT *column1, column2,...*
FROM *existing_table_name*
WHERE ....;

The following SQL creates a new table called "TestTable" (which is a copy of the "Student" table):

```
CREATE TABLE TestTable AS
SELECT Reg_No, Name
FROM students;
```

SQL DROP TABLE Statement

The DROP TABLE statement is used to drop an existing table in a database.

Syntax

```
DROP TABLE table_name;
```

> **Note:** This is very important to know that once a table is deleted all the information available in the table is lost forever, so we have to be very careful when using this command.

SQL DROP TABLE Example

```
DROP TABLE TestTable;
```

SQL TRUNCATE TABLE

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

Syntax

```
TRUNCATE TABLE table_name;
```

SQL DELETE TABLE

The DELETE statement is used to delete rows from a table. If you want to remove a specific row from a table you should use WHERE condition nor it will simply delete all the rows from the table.

```
DELETE FROM table_name;
```

To delete specific row from table

```
DELETE FROM table_name [WHERE condition];
```

Difference between DELETE and TRUNCATE statements

There is a slight difference b/w delete and truncate statement. The **DELETE statement** only deletes the rows from the table based on the condition defined by WHERE clause or delete all the rows from the table when condition is not specified.

But it does not free the space containing by the table.

The **TRUNCATE statement:** it is used to delete all the rows from the table **and free the containing space.**

Difference b/w DROP and TRUNCATE statements

When you use the drop statement it deletes the table's row together with the table's definition so all the relationships of that table with other tables will no longer be valid.

When you drop a table:

- Table structure will be dropped
- Relationship will be dropped
- Integrity constraints will be dropped
- Access privileges will also be dropped

On the other hand when we **TRUNCATE** a table, the table structure remains the same, so you will not face any of the above problems.

SQL RENAME TABLE Statement

The RENAME TABLE statement is used to rename the table.

```
ALTER TABLE table_name
RENAME To new_table_name;
```

Or

```
RENAME old_table _name To new_table_name;
```

SQL ALTER TABLE Statement

The ALTER TABLE statement is used to add, modify or delete columns in an existing table. It is also used to rename a table.

You can also use SQL ALTER TABLE command to add and drop various constraints on an existing table. ALTER TABLE - ADD Column

ALTER TABLE - ADD COLUMN

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
ADD column_name datatype;
```

The following SQL adds an "Email" column to the "Students" table:

```
ALTER TABLE Students
Add Email varchar(255);
```

ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

The following SQL deletes the "Email" column from the "Students" table:

Example

```
ALTER TABLE Students
DROP COLUMN Email;
```

ALTER TABLE - ALTER/MODIFY COLUMN

To change the data type of a column in a table, use the following syntax:

SQL Server

```
ALTER TABLE table_name
ALTER COLUMN column_name datatype;
```

Oracle Live SQL

```
ALTER TABLE table_name
MODIFY column_name datatype;
```

ALTER TABLE - RENAME Column

The syntax of alter table rename column is given below:

```
ALTER TABLE table_name
RENAME COLUMN old_name to new_name;
```