

SQL Table

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

Note: A table has a specified number of columns, but can have any number of rows.

Table is the simple form of data storage. A table is also considered as a convenient representation of relations.

Let's see an example of an employee table:

Employee		
EMP_NAME	ADDRESS	SALARY
Ankit	Lucknow	15000
Raman	Allahabad	18000
Mike	New York	20000

In the above table, "Employee" is the table name, "EMP_NAME", "ADDRESS" and "SALARY" are the column names. The combination of data of multiple columns forms a row e.g. "Ankit", "Lucknow" and 15000 are the data of one row.

SQL TABLE Variable

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

It was introduced with SQL server 2000 to be an alternative of temporary tables.

It is a variable where we temporary store records and results. This is same like temp table but in the case of temp table we need to explicitly drop it.

Table variables are used to store a set of records. So declaration syntax generally looks like CREATE TABLE syntax.

1. **create table** "tablename"
2. ("column1" "data type",
3. "column2" "data type",
4. ...
5. "columnN" "data type");

When a transaction rolled back the data associated with table variable is not rolled back.

A table variable generally uses lesser resources than a temporary variable.

Table variable cannot be used as an input or an output parameter.

Topics of SQL TABLE Statement

SQL TABLE Variable

What TABLE variable can do?

SQL CREATE TABLE

How to create a table using SQL query>

SQL DROP TABLE

How to drop a table?

SQL DELETE TABLE

How to delete all the records of a table?

SQL RENAME TABLE

How to rename a table?

SQL TRUNCATE TABLE

How to truncate a table?

SQL COPY TABLE

How to copy a table?

SQL TEMP TABLE

What is temporary table? What are the advantage of temporary table?

SQL ALTER TABLE

How to add, modify, rename and drop column.

SQL CREATE TABLE

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

If you want to create a table, you should name the table and define its column and each column's data type.

Let's see the simple syntax to create the table.

1. **create table** "tablename"
2. ("column1" "data type",
3. "column2" "data type",
4. "column3" "data type",
5. ...
6. "columnN" "data type");

The data type of the columns may vary from one database to another. For example, NUMBER is supported in Oracle database for integer value whereas INT is supported in MySQL.

Let us take an example to create a STUDENTS table with ID as primary key and NOT NULL are the constraint showing that these fields cannot be NULL while creating records in the table.

1. SQL> **CREATE TABLE** STUDENTS (
2. ID **INT** NOT NULL,
3. **NAME VARCHAR** (20) NOT NULL,
4. AGE **INT** NOT NULL,
5. ADDRESS **CHAR** (25),
6. **PRIMARY KEY** (ID)
7.);

You can verify it, if you have created the table successfully by looking at the message displayed by the SQL Server, else you can use DESC command as follows:

```
SQL> DESC STUDENTS;
```

FIELD	TYPE	NULL	KEY	DEFAULT	EXTRA
ID	Int(11)	NO	PRI		
NAME	Varchar(20)	NO			
AGE	Int(11)	NO			
ADDRESS	Varchar(25)	YES		NULL	

4 rows in set (0.00 sec)

Now you have the STUDENTS table available in your database and you can use to store required information related to students.

SQL CREATE TABLE Example in MySQL

Let's see the command to create a table in MySQL database.

1. **CREATE TABLE** Employee
2. (
3. EmployeeID **int**,
4. FirstName **varchar**(255),
5. LastName **varchar**(255),
6. Email **varchar**(255),
7. AddressLine **varchar**(255),
8. City **varchar**(255)
9.);

Create a Table using another table

We can create a copy of an existing table using the create table command. The new table gets the same column signature as the old table. We can select all columns or some specific columns.

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

The basic syntax for creating a table with the other table is:

1. **CREATE TABLE** table_name **AS**
2. **SELECT** column1, column2,...
3. **FROM** old_table_name **WHERE** ;
4. The following SQL creates a copy **of** the employee **table**.
5. **CREATE TABLE** EmployeeCopy **AS**
6. **SELECT** EmployeeID, FirstName, Email
7. **FROM** Employee;

SQL Primary Key with CREATE TABLE Statement

The following query creates a PRIMARY KEY on the "D" column when the "Employee" table is created.

MySQL

1. **CREATE TABLE** Employee(
2. EmployeeID NOT NULL,
3. FirstName **varchar**(255) NOT NULL,
4. LastName **varchar**(255),
5. City **varchar**(255),
6. **PRIMARY KEY** (EmployeeID)
7.);

SQL DROP TABLE

A SQL DROP TABLE statement is used to delete a table definition and all data from a table.

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.

Let's see the syntax to drop the table from the database.

1. **DROP TABLE** "table_name";

Let us take an example:

First we verify STUDENTS table and then we would delete it from the database.

1. SQL> **DESC** STUDENTS;

FIELD	TYPE	NULL	KEY	DEFAULT	EXTRA
ID	Int(11)	NO	PRI		
NAME	Varchar(20)	NO			
AGE	Int(11)	NO			
ADDRESS	Varchar(25)	YES		NULL	

1. 4 rows in set (0.00 sec)

This shows that STUDENTS table is available in the database, so we can drop it as follows:

1. SQL>**DROP TABLE** STUDENTS;

Now, use the following command to check whether table exists or not.

1. SQL> **DESC** STUDENTS;

1. Query OK, 0 rows affected (0.01 sec)

As you can see, table is dropped so it doesn't display it.

SQL DROP TABLE Example in MySQL

Let's see the command to drop a table from the MySQL database.

1. **DROP TABLE** table_name;

SQL DROP TABLE Example in Oracle

Let's see the command to drop a table from Oracle database. It is same as MySQL.

1. **DROP TABLE** table_name;

SQL DROP TABLE Example in Microsoft SQLServer

Let's see the command to drop a table from SQLServer database. It is same as MySQL.

1. **DROP 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.

1. **DELETE FROM** table_name [**WHERE** condition];

But if you do not specify the WHERE condition it will remove all the rows from the table.

1. **DELETE FROM** table_name;

There are some more terms similar to DELETE statement like as DROP statement and TRUNCATE statement but they are not exactly same there are some differences between them.

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**.

Let's see an "employee" table.

Emp_id	Name	Address	Salary
1	Aryan	Allahabad	22000
2	Shurabhi	Varanasi	13000
3	Pappu	Delhi	24000

Execute the following query to truncate the table:

1. **TRUNCATE TABLE** employee;
-

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 TRUNCATE TABLE

A truncate SQL statement is used to remove all rows (complete data) from a table. It is similar to the DELETE statement with no WHERE clause.

TRUNCATE TABLE Vs DELETE TABLE

Truncate table is faster and uses lesser resources than DELETE TABLE command.

TRUNCATE TABLE Vs DROP TABLE

Drop table command can also be used to delete complete table but it deletes table structure too. TRUNCATE TABLE doesn't delete the structure of the table.

Let's see the syntax to truncate the table from the database.

1. **TRUNCATE TABLE** table_name;

For example, you can write following command to truncate the data of employee table

1. **TRUNCATE TABLE** Employee;

Note: The rollback process is not possible after truncate table statement. Once you truncate a table you cannot use a flashback table statement to retrieve the content of the table.

SQL COPY TABLE

If you want to copy the data of one SQL table into another SQL table in the same SQL server, then it is possible by using the SELECT INTO statement in SQL.

The SELECT INTO statement in Structured Query Language copies the content from one existing table into the new table. SQL creates the new table by using the structure of the existing table.

Syntax of SELECT INTO statement in SQL

1. SELECT * INTO New_table_name FROM old_table_name;

Examples of SELECT INTO statement in SQL

In this article, we have taken the following three different SQL examples which will help you how to copy the content of one table into another table in SQL:

Example 1: In this example, we have a table called **Cars** with three columns:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose you want to copy the content of the above Car table into the new table **Car_Details**. For this, you have to type the following query in SQL:

1. SELECT * INTO Car_Details FROM Cars;

- Let's check the **Car_Details** table is created successfully or not in the database:

1. SELECT * FROM Car_Details;

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000

Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Car_Details

Example 2: In this example, we have a table called **Employee** with four columns:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

- Suppose you want to copy the record of the above Employee table into the new table **Coding_Employees**. For this, you have to type the following query in SQL:

1. SELECT * INTO Coding_Employees FROM Employee;

- Let's check the **Coding_Employees** table is created successfully or not in the database:

1. SELECT * FROM Coding_Employees;

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi

203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Coding_Employees

Example 3: In this example, we have a table called **Student** with four columns:

RollNo	Name	Marks	Age
1001	Bhanu	88	17
1002	Raman	82	16
1003	Sumit	80	16
1004	Shobhit	95	15
1005	Akash	85	16

Table: Student

- Suppose you want to copy the record of the above Student table into the new table **Class_12_Students**. For this, you have to type the following query in SQL:

1. SELECT * INTO Class_12_Students FROM Student;

- Let's check the table is **Class_12_Students** table created successfully or not in the database:

1. SELECT * FROM Class_12_Students;

RollNo	Name	Marks	Age
1001	Bhanu	88	17

1002	Raman	82	16
1003	Sumit	80	16
1004	Shobhit	95	15
1005	Akash	85	16

Table: Class_12_Students

Example 4: In this example, we have a table called **Cars** with three columns:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose you want to copy **Car_Color** and **Car_Name** columns of the above Cars table into the new table **Car_Color**. For this, you have to type the following query in SQL:

1. SELECT Car_Name, Car_Color INTO Car_Color FROM Cars;

- Let's check the **Car_Color** table is created successfully or not in the database:

1. SELECT * FROM Car_Color;

Car Name	Car Color
Hyundai Creta	White
Hyundai Venue	White
Hyundai i20	Red
Kia Sonet	White
Kia Seltos	Black
Swift Dezire	Red

Table: Car_Color

Syntax of SELECT INTO statement with WHERE clause in SQL

1. `SELECT * INTO New_table_name FROM old_table_name WHERE [condition] ;`

Examples of SELECT INTO statement with WHERE clause in SQL

Here, we have taken the following three different SQL examples, which will help you how to copy the content of one table into another table with a specific condition in SQL:

Example 1: In this example, we have a table called **Cars** with three columns:

Car Name	Car Color	Car Cost
Hyundai Creta	Black	10,85,000
Hyundai Venue	Black	9,50,000
Hyundai i20	Red	9,00,000

Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose we want to copy only the record of those cars whose color is black. For this, we have to type the following query in SQL:

1. SELECT * INTO Black_Car_Details FROM Cars WHERE Car_Color = 'Black';

- Let's check the **Black_Car_Details** table is created successfully or not in the database:

1. SELECT * FROM Black_Car_Details;

Car Name	Car Color	Car Cost
Hyundai Creta	Black	10,85,000
Hyundai Venue	Black	9,50,000
Kia Seltos	Black	8,00,000

Table: Black_Car_Details

Example 2: In this example, we have a table called **Employee** with four columns:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	45000	Goa
202	Ankit	45000	Delhi
203	Bheem	38000	Goa

204	Ram	49000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose we want to copy only the record of those employees whose Salary is more than 40,000. For this, we have to type the following query in SQL:

1. `SELECT * INTO Emp_Salary_40000 FROM Cars WHERE Emp_Salary > 40000;`

- Let's check the **Emp_Salary_40000** table created successfully or not in the database:

1. `SELECT * FROM Emp_Salary_40000;`

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	45000	Goa
202	Ankit	45000	Delhi
204	Ram	49000	Goa

Table: Emp_Salary_40000

SQL ALTER TABLE

The ALTER TABLE statement in Structured Query Language allows you to add, modify, and delete columns of an existing table. This statement also allows database users to add and remove various SQL constraints on the existing tables.

Any user can also change the name of the table using this statement.

ALTER TABLE ADD Column statement in SQL

In many situations, you may require to add the columns in the existing table. Instead of creating a whole table or database again you can easily add single and multiple columns using the ADD keyword.

Syntax of ALTER TABLE ADD Column statement in SQL

1. ALTER TABLE table_name ADD column_name column-definition;

The above syntax only allows you to add a single column to the existing table. If you want to add more than one column to the table in a single SQL statement, then use the following syntax:

1. ALTER TABLE table_name
2. ADD (column_Name1 column-definition,
3. column_Name2 column-definition,
4.
5. column_NameN column-definition);

Examples of ALTER TABLE ADD Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to add the single and multiple columns in the existing table using ALTER TABLE statement:

Example 1: Let's take an example of a table named **Cars**:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000

Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose, you want to add the new column Car_Model in the above table. For this, you have to type the following query in the SQL:

1. ALTER TABLE Cars ADD Car_Model Varchar(20);

This statement will add the Car_Model column to the Cars table.

Example 2: Let's take an example of a table named **Employee:**

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose, you want to add two columns, **Emp_ContactNo.** and **Emp_EmailID**, in the above Employee table. For this, you have to type the following query in the SQL:

1. ALTER TABLE Employee ADD (Emp_ContactNo. Number(13), Emp_EmailID var char(50) ;

This statement will add Emp_ContactNo. and Emp_EmailID columns to the Employee table.

ALTER TABLE MODIFY Column statement in SQL

The MODIFY keyword is used for changing the column definition of the existing table.

Syntax of ALTER TABLE MODIFY Column statement in SQL

1. ALTER TABLE table_name MODIFY column_name column-definition;

This syntax only allows you to modify a single column of the existing table. If you want to modify more than one column of the table in a single SQL statement, then use the following syntax:

1. ALTER TABLE table_name
2. MODIFY (column_Name1 column-definition,
3. column_Name2 column-definition,
4.
5. column_NameN column-definition);

Examples of ALTER TABLE MODIFY Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to modify single and multiple columns of the existing table using ALTER TABLE statement:

Example 1: Let's take an example of a table named **Cars**:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000

Swift Dezire	Red	7,95,000
--------------	-----	----------

Table: Cars

- Suppose, you want to modify the datatype of the **Car_Color** column of the above table. For this, you have to type the following query in the SQL:

1. ALTER TABLE Cars ADD Car_Color Varchar(50);

Example 2: Let's take an example of a table named **Employee:**

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose, you want to modify the datatypes of two columns **Emp_ContactNo.** and **Emp_EmailID** of the above Employee table. For this, you have to type the following query in the SQL:

1. ALTER TABLE Employee ADD (Emp_ContactNo. Int, Emp_EmailID varchar(80) ;

ALTER TABLE DROP Column statement in SQL

In many situations, you may require to delete the columns from the existing table. Instead of deleting the whole table or database you can use DROP keyword for deleting the columns.

Syntax of ALTER TABLE DROP Column statement in SQL

1. ALTER TABLE table_name DROP Column column_name ;

Examples of ALTER TABLE DROP Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to delete a column from the existing table using ALTER TABLE statement:

Example 1: Let's take an example of a table named **Cars**:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose, you want to delete the Car_Color column from the above table. For this, you have to type the following query in the SQL:

1. ALTER TABLE Cars DROP COLUMN Car_Color ;

- Let's check using the following statement that the Car_Color column is deleted from the table or not:

1. SELECT * FROM Cars;

Car Name	Car Cost
----------	----------

Hyundai Creta	10,85,000
Hyundai Venue	9,50,000
Hyundai i20	9,00,000
Kia Sonet	10,00,000
Kia Seltos	8,00,000
Swift Dezire	7,95,000

Table: Cars

Example 2: Let's take an example of a table named **Employee**:

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose, you want to delete the **Emp_Salary and Emp_City** column from the above Employee table. For this, you have to type the following two different queries in the SQL:
1. ALTER TABLE Cars DROP COLUMN Emp_Salary ;
 2. ALTER TABLE Cars DROP COLUMN Emp_City ;

ALTER TABLE RENAME Column statement in SQL

The RENAME keyword is used for changing the name of columns or fields of the existing table.

Syntax of ALTER TABLE RENAME Column statement in SQL

1. ALTER TABLE table_name RENAME COLUMN old_name to new_name;

Examples of ALTER TABLE RENAME Column statement in SQL

Here, we have taken the following two different SQL examples, which will help you how to change the name of a column of the existing table using ALTER TABLE statement:

Example 1: Let's take an example of a table named **Cars**:

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

- Suppose, you want to change the name of the **Car_Color** column of the above Cars table. For this, you have to type the following query in the SQL:

1. ALTER TABLE Cars RENAME COLUMN Car_Color to Colors;

This statement will change the name of a column of the Cars table. To see the changes, you have to type the following query:

1. `SELECT * FROM Cars;`

Car Name	Car Color	Car Cost
Hyundai Creta	White	10,85,000
Hyundai Venue	White	9,50,000
Hyundai i20	Red	9,00,000
Kia Sonet	White	10,00,000
Kia Seltos	Black	8,00,000
Swift Dezire	Red	7,95,000

Table: Cars

Example 2: Let's take an example of a table named **Employee:**

Emp_Id	Emp_Name	Emp_Salary	Emp_City
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee

- Suppose, you want to change the name of **the Emp_City** column of the above Employee table. For this, you have to type the following query in the SQL:

1. ALTER TABLE Employee RENAME COLUMN Emp_City to Emp_Address;

This statement will change the name of a column of the Employee table. To see the changes, you have to type the following query:

1. SELECT * FROM Employee;

Emp_Id	Emp_Name	Emp_Salary	Emp_Address
201	Abhay	25000	Goa
202	Ankit	45000	Delhi
203	Bheem	30000	Goa
204	Ram	29000	Goa
205	Sumit	40000	Delhi

Table: Employee