**DATA DEFINITION LANGUAGE**

The various commands in DDL are :-

1. **Create**
2. **Alter**
3. **Rename**
4. **Drop**
5. **Truncate**

**CREATE:**

1. It creates the table.
2. Before we study the **Create** command, let us first study the some of the basic **data types** we use in SQL.

**Syntax:**

SYNTAX:

CREATE TABLE table\_name(

column1 datatype(size) constraints,

column2 datatype(size) constraints,

……

.....

columnN datatype(size) constraints);

**EX:**

STUDENT-USN(PK),SNAME,SEMESTER,CNAME

LIBRARY-LID(PK),BNAME,BPRICE,AUTHOR

PROJECT-PID(PK),PNAME,PLOC

**Rename:**

1. It renames a table.

**Syntax:**

Rename table\_name to new\_table\_name

**EX:**

SQL> rename student to stu;

Table renamed.

**Syntax rename the column**

Alter table table\_Name

rename column Column\_Name to New\_Column\_Name

**TRUNCATE:**

1. It removes all the data permanently, but the structure of the table remains as it is.

**Syntax:**

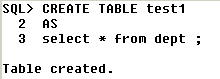
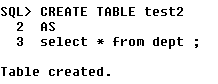
**TRUNCATE TABLE <TABLE\_NAME> ;**

**DROP:**

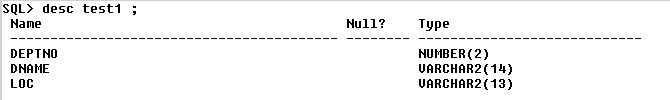
It removes both data and the structure of the table permanently from the database.

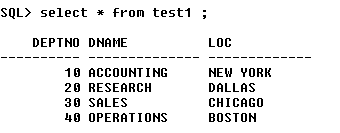
**Ex – SQL > DROP TABLE test ;**

Let us understand the difference between **drop & truncate** using the below shown example,

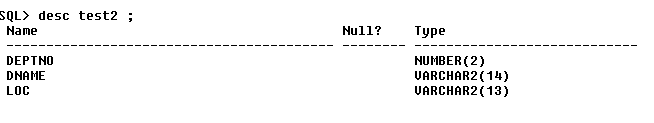
 

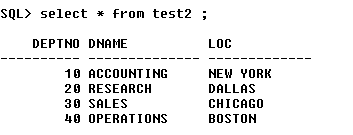
Let us create 2 tables Test1 and Test2 as shown above.





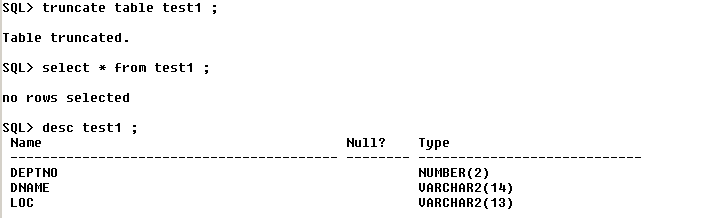
The above shows the description of the table test1.





The above gives the description of the table Test2.

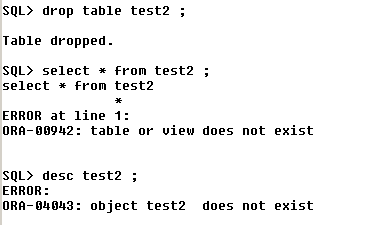
Now, let us use the **Truncate query on Test1** and **Drop query on Test2** and see the difference.



The above 3 queries show that – 1st query has the table test1 truncated.

2nd query – it shows **no rows selected** – thus only the records from the table has been removed. 3rd query – it shows that the structure of the table is still present. Only the records will be removed.

Thus, this **explains the truncate query.**



Thus from the above queries we can explain how **drop** works. 1st query – it drops the table. Thus – the entire structure and records of the table are dropped.

2nd and 3rd query – since, there is no table – **select & desc** query for **test2** will throw an error.

Thus, this **explains the drop query.**

Hence, we have seen the difference between **drop & truncate** query.

**10g Recycle Bin**

**TABLE**

**DROP**

**RECYCLE**

**BIN**

**What to do?**

**PERMANENTLY REMOVE**

**RESTORE IT**

**FLASHBACK**

**PURGE**

* The functionality of Recycle Bin was introduced in Oracle 10G version only.
* Thus even though the table has been dropped, we can still restore it using **flashback command** or we can permanently remove it using the **purge** command.
* This concept of Recycle bin was not there in the earlier versions of Oracle.

**Recovering Drop Tables (Undo Drop Table):**

To recover this dropped table a user can type the command

SQL> Flashback table table\_name to before drop;

You can also restore the dropped table by giving it a different name like this

SQL> Flashback table emp to before drop rename to emp2;

**Permanently deleting table after drop:**

Purge table tableName

**Permanently Dropping Tables**If you want to permanently drop tables without putting it into Recycle Bin drop tables with purge command like this

SQL> drop table tablename purge;

This will drop the table permanently and it cannot be restored.

**Syntax to drop column**

ALTER TABLE table\_name DROP COLUMN column\_name;

ALTER TABLE table\_name DROP (column\_name1, column\_name2);

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.

Topics of SQL TABLE Statement

[**SQL TABLE Variable**](https://www.javatpoint.com/sql-table-variable)

What TABLE variable can do?

[**SQL CREATE TABLE**](https://www.javatpoint.com/sql-create-table)

How to create a table using SQL query>

[**SQL DROP TABLE**](https://www.javatpoint.com/sql-drop-table)

How to drop a table?

[**SQL DELETE TABLE**](https://www.javatpoint.com/sql-delete-table)

How to delete all the records of a table?

[**SQL RENAME TABLE**](https://www.javatpoint.com/sql-rename-table)

How to rename a table?

[**SQL TRUNCATE TABLE**](https://www.javatpoint.com/sql-truncate-table)

How to truncate a table?

[**SQL COPY TABLE**](https://www.javatpoint.com/sql-copy-table)

How to copy a table?

[**SQL TEMP TABLE**](https://www.javatpoint.com/sql-temp-table)

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

[**SQL ALTER TABLE**](https://www.javatpoint.com/sql-alter-table)

How to add, modify, rename and drop column.

# 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.

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. );

SQL CREATE TABLE Example in Oracle

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

1. **CREATE** **TABLE** Employee
2. (
3. EmployeeID number(10),
4. FirstName varchar2(255),
5. LastName varchar2(255),
6. Email varchar2(255),
7. AddressLine varchar2(255),
8. City varchar2(255)
9. );

SQL CREATE TABLE Example in Microsoft SQLServer

Let's see the command to create a table in SQLServer database. It is same as MySQL and Oracle.

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. );

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;
2. 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 RENAME TABLE

**SQL RENAME TABLE** syntax is used to change the name of a table. Sometimes, we choose non-meaningful name for the table. So it is required to be changed.

Let's see the syntax to rename a table from the database.

1. **ALTER** **TABLE** table\_name
2. RENAME **TO** new\_table\_name;

Optionally, you can write following command to rename the table.

1. RENAME old\_table \_name **To** new\_table\_name;

Let us take an example of a table named "STUDENTS", now due to some reason we want to change it into table name "ARTISTS".

Table1: students

|  |  |  |
| --- | --- | --- |
| **Name** | **Age** | **City** |
| Amrita gill | 25 | Amritsar |
| Amrender sirohi | 22 | Ghaziabad |
| Divya khosla | 20 | Delhi |

You should use any one of the following syntax to RENAME the table name:

1. **ALTER** **TABLE** STUDENTS
2. RENAME **TO** ARTISTS;

Or

1. RENAME STUDENTS **TO** ARTISTS;
2. **After** that the **table** "students" will be changed **into** **table** **name** "artists"

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 a SQL table into another table in the same SQL server database, it is possible by using the select statement.

The syntax of copying table from one to another is given below:

1. **SELECT** \* **INTO** <destination\_table> **FROM** <source\_table>

For example, you can write following command to copy the records of hr\_employee table into employee table.

1. **SELECT** \* **INTO** admin\_employee **FROM** hr\_employee;

**Note: SELECT INTO is totally different from INSERT INTO statement.**

1. **Create table stu as select \* from emp;**

For example, you can write following command to copy the records of emp table into stu table.

1. **Create table stu as select \* from emp where ename=’yyyy’;**

For example, you can write following command to copy the structure of emp table into stu table.

1. **Insert into stu select \* from emp;**

For example, you can write following command to copy the result to existing table.

SQL TEMP TABLE

The concept of temporary table is introduced by SQL server. It helps developers in many ways:

**Temporary tables** can be created at run-time and can do all kinds of operations that a normal table can do. These temporary tables are created inside tempdb database.

There are two types of temp tables based on the behavior and scope.

1. Local Temp Variable
2. Global Temp Variable

Local Temp Variable

Local temp tables are only available at current connection time. It is automatically deleted when user disconnects from instances. It is started with hash (#) sign.

1. **CREATE** **TABLE** #**local** **temp** **table** (
2. User id **int**,
3. Username **varchar** (50),
4. User address **varchar** (150)
5. )

Global Temp Variable

Global temp tables name starts with double hash (##). Once this table is created, it is like a permanent table. It is always ready for all users and not deleted until the total connection is withdrawn.

1. **CREATE** **TABLE** ##new **global** **temp** **table** (
2. User id **int**,
3. User **name** **varchar** (50),
4. User address **varchar** (150)
5. )

# SQL ALTER TABLE

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.

### SQL ALTER TABLE Add Column

If you want to add columns in SQL table, the SQL alter table syntax is given below:

1. **ALTER** **TABLE** table\_name **ADD** column\_name **column**-definition;

If you want to add multiple columns in table, the SQL table will be

1. **ALTER** **TABLE** table\_name
2. **ADD** (column\_1 **column**-definition,
3. column\_2 **column**-definition,
4. .....
5. column\_n **column**-definition);

### SQL ALTER TABLE Modify Column

If you want to modify an existing column in SQL table, syntax is given below:

1. **ALTER** **TABLE** table\_name **MODIFY** column\_name column\_type;

If you want to modify multiple columns in table, the SQL table will be

1. **ALTER** **TABLE** table\_name
2. **MODIFY** (column\_1 column\_type,
3. column\_2 column\_type,
4. .....
5. column\_n column\_type);

### SQL ALTER TABLE DROP Column

The syntax of alter table drop column is given below:

1. **ALTER** **TABLE** table\_name **DROP** **COLUMN** column\_name;

### SQL ALTER TABLE RENAME Column

The syntax of alter table rename column is given below:

1. **ALTER** **TABLE** table\_name
2. RENAME **COLUMN** old\_name **to** new\_name;