**SQL | DDL, DML, DCL and TCL Commands**

**Structured Query Language(SQL)** as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert etc. to carry out the required tasks.

These SQL commands are mainly categorized into four categories as discussed below:

1. **DDL(Data Definition Language) :**DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in database.

**Examples of DDL commands:**

* + [**CREATE**](https://www.geeksforgeeks.org/sql-create/) – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
  + [**DROP**](https://www.geeksforgeeks.org/sql-drop-truncate/) – is used to delete objects from the database.
  + [**ALTER**](https://www.geeksforgeeks.org/sql-alter-add-drop-modify/)-is used to alter the structure of the database.
  + [**TRUNCATE**](https://www.geeksforgeeks.org/sql-drop-truncate/)–is used to remove all records from a table, including all spaces allocated for the records are removed.
  + [**COMMENT**](https://www.geeksforgeeks.org/sql-comments/) –is used to add comments to the data dictionary.
  + [**RENAME**](https://www.geeksforgeeks.org/sql-alter-rename/)–is used to rename an object existing in the database.

1. **DML(Data Manipulation Language) :**The SQL commands that deals with the manipulation of data present in database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

**Examples of DML:**

* + [**SELECT**](https://www.geeksforgeeks.org/sql-select-clause/) – is used to retrieve data from the a database.
  + [**INSERT**](https://www.geeksforgeeks.org/sql-insert-statement/) – is used to insert data into a table.
  + [**UPDATE**](https://www.geeksforgeeks.org/sql-update-statement/) – is used to update existing data within a table.
  + [**DELETE**](https://www.geeksforgeeks.org/sql-delete-statement/) – is used to delete records from a database table.

1. **DCL(Data Control Language) :**DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.

**Examples of DCL commands:**

* + **GRANT**-gives user’s access privileges to database.
  + **REVOKE**-withdraw user’s access privileges given by using the GRANT command.

1. **TCL(transaction Control Language) :**TCL commands deals with the [transaction within the database](https://www.geeksforgeeks.org/sql-transactions/).

**Examples of TCL commands:**

* + **COMMIT**– commits a Transaction.
  + [**ROLLBACK**](https://www.geeksforgeeks.org/sql-transactions/)– rollbacks a transaction in case of any error occurs.
  + **SAVEPOINT**–sets a savepoint within a transaction.
  + **SET TRANSACTION**–specify characteristics for the transaction.

**1 . DDL(Data Definition Language):**

1.1 **CREATE**

There are two CREATE statements available in SQL:

1. CREATE DATABASE 2. CREATE TABLE

1. The **CREATE DATABASE** statement is used to create a new database in SQL.

**Syntax**: CREATE DATABASE database\_name; where **database\_name**: name of the database.

Ex : CREATE DATABASE my\_database;

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

**Syntax**:

CREATE TABLE table\_name ( column1 data\_type(size), column2 data\_type(size),

column3 data\_type(size), ....);

**table\_name**: name of the table.

**column1** name of the first column.

**data\_type**: Type of data we want to store in the particular column.

Ex: CREATE TABLE Students ( ROLL\_NO int(3), NAME varchar(20),SUBJECT varchar(20));

**1.2 DROP**

It is used to delete a whole database or just a table.The DROP statement destroys the objects like an existing database, table, index, or view.

**Syntax:**

**DROP object object\_name**

**Examples:**

**DROP TABLE table\_name;**

**table\_name**: Name of the table to be deleted.

**DROP DATABASE database\_name;**

**database\_name**: Name of the database to be deleted.

**1.3 TRUNCATE**

Truncate statement is a Data Definition Language (DDL) operation that is used to mark the extents of a table for deallocation (empty for reuse). The result of this operation quickly removes all data from a table.

The TRUNCATE TABLE mytable statement is logically (though not physically) equivalent to the DELETE FROM mytable statement (without a WHERE clause).  
**Syntax:**

**TRUNCATE TABLE table\_name; where table\_name**: Name of the table to be truncated.

**DATABASE name - student\_data**

**DROP vs TRUNCATE**

* Truncate is normally ultra-fast and its ideal for deleting data from a temporary table.
* Truncate preserves the structure of the table for future use, unlike drop table where the table is deleted with its full structure.
* Table or Database deletion using DROP statement **cannot** be rolled back, so it must be used wisely.
* To delete the whole database

DROP DATABASE student\_data;

After running the above query whole database will be deleted.

* To truncate Student\_details table from student\_data database.

TRUNCATE TABLE Student\_details;

After running the above query Student\_details table will be truncated, i.e, the data will be deleted but the structure will remain in the memory for further operations.

* 1. **ALTER (ADD, DROP, MODIFY)**

ALTER TABLE is used to add, delete/drop or modify columns in the existing table. It is also used to add and drop various constraints on the existing table.

**ALTER TABLE – ADD**

ADD is used to add columns into the existing table. Sometimes we may require to add additional information, in that case we do not require to create the whole database again, **ADD** comes to our rescue.

**Syntax:**

**ALTER TABLE table\_name ADD (Columnname\_1 datatype, Columnname\_2 datatype,**

**… Columnname\_n datatype);**

**ALTER TABLE – DROP**

DROP COLUMN is used to drop column in a table. Deleting the unwanted columns from the table.

**Syntax:**

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

**ALTER TABLE-MODIFY**

It is used to modify the existing columns in a table. Multiple columns can also be modified at once.  
*\*Syntax may vary slightly in different databases.*

**Syntax(Oracle,MySQL,MariaDB):**

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

**Syntax(SQL Server):**

**ALTER TABLE table\_name ALTER COLUMN column\_name column\_type;**

**Queries: Sample Table:Student**

| **ROLL\_NO** | **NAME** |
| --- | --- |
| 1 | Ram |
| 2 | Abhi |
| 3 | Rahul |
| 4 | Tanu |

**QUERY:**

* To ADD 2 columns AGE and COURSE to table Student.

**ALTER TABLE Student ADD (AGE number(3),COURSE varchar(40));**

**OUTPUT:**

| **ROLL\_NO** | **NAME** | **AGE** | **COURSE** |
| --- | --- | --- | --- |
| 1 | Ram |  |  |
| 2 | Abhi |  |  |
| 3 | Rahul |  |  |
| 4 | Tanu |  |  |

* MODIFY column COURSE in table Student

**ALTER TABLE Student MODIFY COURSE varchar(20);**

After running the above query maximum size of Course Column is reduced to 20 from 40.

* DROP column COURSE in table Student.

**ALTER TABLE Student DROP COLUMN COURSE;**

**OUTPUT:**

| **ROLL\_NO** | **NAME** | **AGE** |
| --- | --- | --- |
| 1 | Ram |  |
| 2 | Abhi |  |
| 3 | Rahul |  |
| 4 | Tanu |  |

1.5 ALTER (RENAME)

Sometimes we may want to rename our table to give it a more relevant name. For this purpose we can use **ALTER TABLE** to rename the name of table.  
*\*Syntax may vary in different databases.*

**Syntax(Oracle,MySQL,MariaDB):**

**ALTER TABLE table\_name RENAME TO new\_table\_name;**

Columns can be also be given new name with the use of **ALTER TABLE**.  
**Syntax(Oracle):**

**ALTER TABLE table\_name RENAME COLUMN old\_name TO new\_name;**

**Syntax(MySQL,MariaDB):**

**ALTER TABLE table\_name CHANGE COLUMN old\_name TO new\_name;**

**Sample Table:**Student

| **ROLL\_NO** | **NAME** | **AGE** |
| --- | --- | --- |
| 1 | Ram | 20 |
| 2 | Abhi | 21 |
| 3 | Rahul | 22 |
| 4 | Tanu | 19 |

**QUERY:**

* Change the name of column NAME to FIRST\_NAME in table Student.

**ALTER TABLE Student RENAME COLUMN NAME TO FIRST\_NAME;**

**OUTPUT:**

| **ROLL\_NO** | **FIRST\_NAME** | **AGE** |
| --- | --- | --- |
| 1 | Ram | 20 |
| 2 | Abhi | 21 |
| 3 | Rahul | 22 |
| 4 | Tanu | 19 |

* Change the name of the table Student to Student\_Details

**OUTPUT:**Student\_Details

| **ROLL\_NO** | **FIRST\_NAME** | **AGE** |
| --- | --- | --- |
| 1 | Ram | 20 |
| 2 | Abhi | 21 |
| 3 | Rahul | 22 |
| 4 | Tanu | 19 |

**1.6 Comments**

Comments can be written in the following three formats:

1. Single line comments.
2. Multi line comments
3. In line comments

* **Single line comments:** Comments starting and ending in a single line are considered as single line comments.  
  Line starting with ‘–‘ is a comment and will not be executed.  
  Syntax:
* --single line comment
* --another comment
* SELECT \* FROM Customers;
* **Multi line comments:**Comments starting in one line and ending in different line are considered as multi line comments. Line starting with ‘/\*’ is considered as starting point of comment and are terminated when ‘\*/’ is encountered.  
  Syntax:

/\* multi line commentanother comment \*/

SELECT \* FROM Customers;

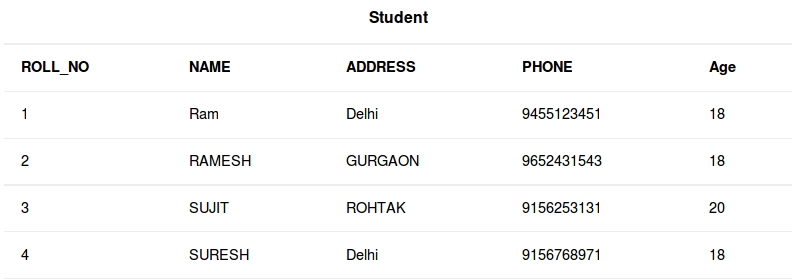
* **In line comments:**In line comments are an extension of multi line comments, comments can be stated in between the statements and are enclosed in between ‘/\*’ and ‘\*/’.  
  Syntax:

SELECT \* FROM /\* Customers; \*/

**2. DML(Data Manipulation Language) :**

# 2.1 SELECT Query

Select is the most commonly used statement in SQL. The SELECT Statement in SQL is used to retrieve or fetch data from a database. We can fetch either the entire table or according to some specified rules. The data returned is stored in a result table. This result table is also called result-set.With the SELECT clause of a SELECT command statement, we specify the columns that we want to be displayed in the query result and, optionally, which column headings we prefer to see above the result table.The select clause is the first clause and is one of the last clauses of the select statement that the database server evaluates. The reason for this is that before we can determine what to include in the final result set, we need to know all of the possible columns that could be included in the final result set.

**SampleTable:**  
[](http://cdncontribute.geeksforgeeks.org/wp-content/uploads/table.jpg)  
**Basic Syntax:**

**SELECT column1,column2 FROM table\_name ;**

**column1 , column2**: names of the fields of the table

**table\_name:** from where we want to fetch

This query will return all the rows in the table with fields column1 , column2.

To fetch the entire table or all the fields in the table:

SELECT \* FROM table\_name;

* Query to fetch the fields ROLL\_NO, NAME, AGE from the table Student:

SELECT ROLL\_NO, NAME, AGE FROM Student;

Output:

| **ROLL\_NO** | **NAME** | **Age** |
| --- | --- | --- |
| 1 | Ram | 18 |
| 2 | RAMESH | 18 |
| 3 | SUJIT | 20 |
| 4 | SURESH | 18 |

* To fetch all the fields from the table Student:

SELECT \* FROM Student;

Output:

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9652431543 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |

2.2 INSERT INTO Statement

The INSERT INTO statement of SQL is used to insert a new row in a table. There are two ways of using INSERT INTO statement for inserting rows:

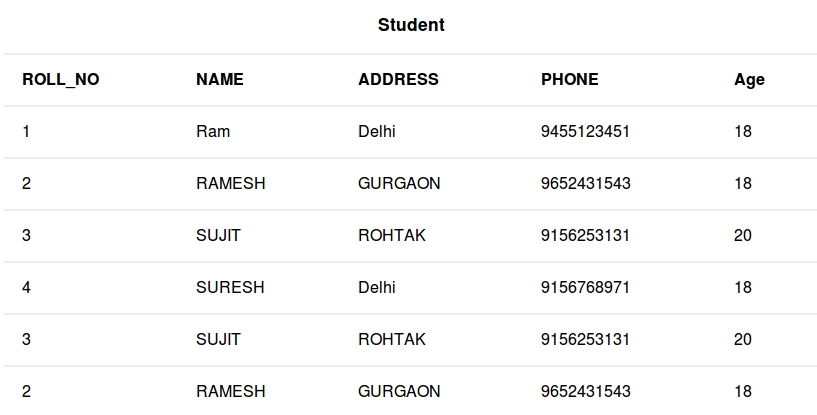
**1.Only values:** First method is to specify only the value of data to be inserted without the column names.  
**Syntax:**

**INSERT INTO table\_name VALUES (value1, value2, value3,...);**

**table\_name**: name of the table.

**value1, value2,..** : value of first column, second column,... for the new record

**2.Column names and values both:** In the second method we will specify both the columns which we want to fill and their corresponding values as shown below:  
Syntax:

**INSERT INTO table\_name (column1, column2, column3,..) VALUES ( value1, value2, value3,..);table\_name**: name of the table.**column1**: name of first column, second column ...**value1, value2, value3** : [](http://cdncontribute.geeksforgeeks.org/wp-content/uploads/table11.jpg)

value of first column, second column,... for the new record.

**Queries:Method 1 (Inserting only values) :**

INSERT INTO Student VALUES ('5','HARSH','WEST BENGAL','8759770477','19');

**Output:** The table **Student** will now look like:

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 5 | HARSH | WEST BENGAL | 8759770477 | 19 |

**Method 2 (Inserting values in only specified columns):**

INSERT INTO Student (ROLL\_NO, NAME, Age) VALUES ('5','PRATIK','19');

**Output:** The table **Student** will now look like:

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 5 | PRATIK | null | null | 19 |

Notice that the columns for which the values are not provided are filled by null. Which is the default values for those columns.

**Using SELECT** **in INSERT INTO Statement**

We can use the SELECT statement with INSERT INTO statement to copy rows from one table and insert them into another table.The use of this statement is similar to that of INSERT INTO statement. The difference is that the SELECT statement is used here to select data from a different table. The different ways of using INSERT INTO SELECT statement are shown below:

* **Inserting all columns of a table:** We can copy all the data of a table and insert into in a different table.  
  **Syntax:**
* **INSERT INTO first\_table SELECT \* FROM second\_table;**
* **first\_table**: name of first table.
* **second\_table**: name of second table.

We have used the SELECT statement to copy the data from one table and INSERT INTO statement to insert in a different table.

* **Inserting specific columns of a table:** We can copy only those columns of a table which we want to insert into in a different table.  
  **Syntax:**
* **INSERT INTO first\_table(names\_of\_columns1) SELECT names\_of\_columns2 FROM second\_table;**
* **first\_table**: name of first table.
* **second\_table**: name of second table.
* **names of columns1**: name of columns separated by comma(,) for table 1.
* **names of columns2**: name of columns separated by comma(,) for table 2.

We have used the SELECT statement to copy the data of the selected columns only from the second table and INSERT INTO statement to insert in first table.

* **Copying specific rows from a table**: We can copy specific rows from a table to insert into another table by using WHERE clause with the SELECT statement. We have to provide appropriate condition in the WHERE clause to select specific rows.  
  **Syntax:**
* **INSERT INTO table1 SELECT \* FROM table2 WHERE condition;**
* **first\_table**: name of first table.
* **second\_table**: name of second table.
* **condition:** condition to select specific rows.

Table2: LateralStudent

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 7 | SOUVIK | DUMDUM | 9876543210 | 18 |
| 8 | NIRAJ | NOIDA | 9786543210 | 19 |
| 9 | SOMESH | ROHTAK | 9687543210 | 20 |

**Queries:**

* + **Method 1(Inserting all rows and columns):**
  + INSERT INTO Student SELECT \* FROM LateralStudent;

**Output:**  
This query will insert all the data of the table LateralStudent in the table Student. The table Student will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 7 | SOUVIK | DUMDUM | 9876543210 | 18 |
| 8 | NIRAJ | NOIDA | 9786543210 | 19 |
| 9 | SOMESH | ROHTAK | 9687543210 | 20 |

* **Method 2(Inserting specific columns):**
* INSERT INTO Student(ROLL\_NO,NAME,Age) SELECT ROLL\_NO, NAME, Age FROM LateralStudent;

**Output:**  
This query will insert the data in the columns ROLL\_NO, NAME and Age of the table LateralStudent in the table Student and the remaining columns in the Student table will be filled by *null* which is the default value of the remaining columns. The table Student will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 7 | SOUVIK | null | null | 18 |
| 8 | NIRAJ | null | null | 19 |
| 9 | SOMESH | null | null | 20 |

* **Select specific rows to insert**:
* INSERT INTO Student SELECT \* FROM LateralStudent WHERE Age = 18;

**Output:**  
This query will select only the first row from table LateralStudent to insert into the table Student. The table Student will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 7 | SOUVIK | DUMDUM | 9876543210 | 18 |

**2.3 UPDATE Statement**

The UPDATE statement in SQL is used to update the data of an existing table in database. We can update single columns as well as multiple columns using UPDATE statement as per our requirement.

**Basic Syntax**

**UPDATE table\_name SET column1 = value1, column2 = value2,...**

**WHERE condition;**

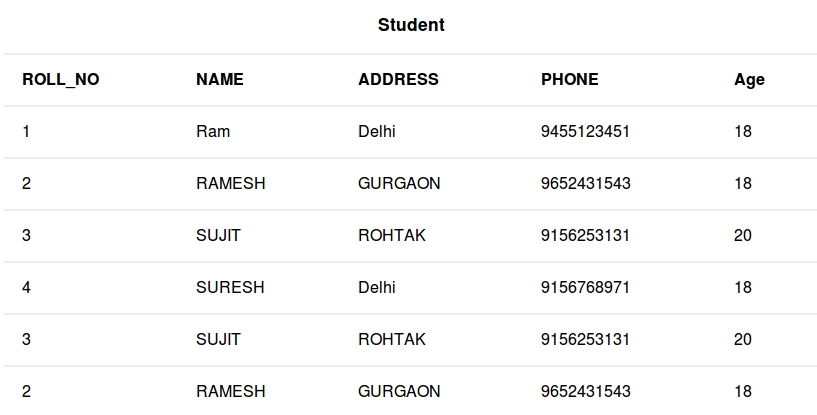
**table\_name:** name of the table

**column1**: name of first , second, third column....

**value1**: new value for first, second, third column....

**condition**: condition to select the rows for which the

values of columns needs to be updated.

**NOTE:** In the above query the**SET**statement is used to set new values to the particular column and the **WHERE** clause is used to select the rows for which the columns are needed to be updated. If we have not used the WHERE clause then the columns in **all** the rows will be updated. So the WHERE Clause is used to choose the particular rows.  
[](http://cdncontribute.geeksforgeeks.org/wp-content/uploads/table11.jpg)

**Example Queries**

* + **Updating single column**: Update the column NAME and set the value to ‘PRATIK’ in all the rows where Age is 20.
  + UPDATE Student SET NAME = 'PRATIK' WHERE Age = 20;

**Output:**  
This query will update two rows(third row and fifth row) and the table **Student** will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | PRATIK | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | PRATIK | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |

* + **Updating multiple columns:** Update the columns NAME to ‘PRATIK’ and ADDRESS to ‘SIKKIM’ where ROLL\_NO is 1.
  + UPDATE Student SET NAME = 'PRATIK', ADDRESS = 'SIKKIM' WHERE ROLL\_NO = 1;

**Output**:  
The above query will update two columns in the first row and the table **Student** will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | PRATIK | SIKKIM | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | PRATIK | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | PRATIK | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |

**Note:** For updating multiple columns we have used comma(,) to separate the names and values of two columns.

* **Omitting WHERE clause:** If we omit the WHERE clause from the update query then all of the rows will get updated.
* UPDATE Student SET NAME = 'PRATIK';

**Output:**  
The table **Student** will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | PRATIK | Delhi | 9455123451 | 18 |
| 2 | PRATIK | GURGAON | 9562431543 | 18 |
| 3 | PRATIK | ROHTAK | 9156253131 | 20 |
| 4 | PRATIK | Delhi | 9156768971 | 18 |
| 3 | PRATIK | ROHTAK | 9156253131 | 20 |
| 2 | PRATIK | GURGAON | 9562431543 | 18 |

**2.4 DELETE Statement**

The DELETE Statement in SQL is used to delete existing records from a table. We can delete a single record or multiple records depending on the condition we specify in the WHERE clause.

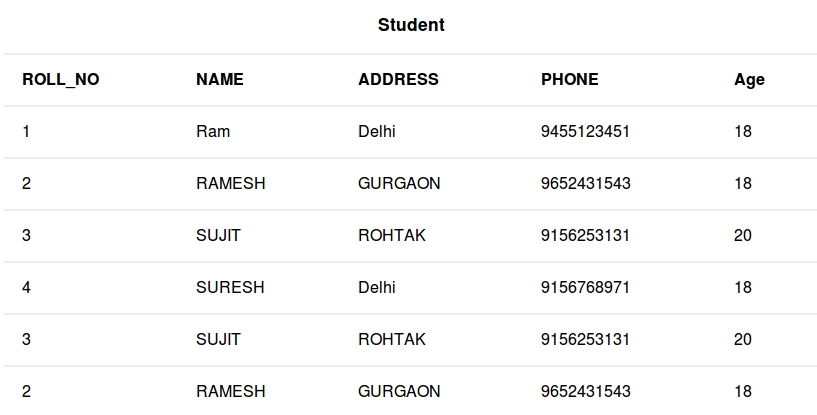
**Basic Syntax:**

DELETE FROM table\_name WHERE some\_condition;

**table\_name**: name of the table

**some\_condition**: condition to choose particular record.

**Note:** We can delete single as well as multiple records depending on the condition we provide in WHERE clause. If we omit the WHERE clause then all of the records will be deleted and the table will be empty.

Sample Table:  
[](http://cdncontribute.geeksforgeeks.org/wp-content/uploads/table11.jpg)

**Example Queries:**

* **Deleting single record**: Delete the rows where NAME = ‘Ram’. This will delete only the first row.
* DELETE FROM Student WHERE NAME = 'Ram';

**Output:**The above query will delete only the first row and the table **Student** will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 3 | SUJIT | ROHTAK | 9156253131 | 20 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |

* **Deleting multiple records**: Delete the rows from the table Student where Age is 20. This will delete 2 rows(third row and fifth row).
* DELETE FROM Student WHERE Age = 20;

**Output:**  
The above query will delete two rows(third row and fifth row) and the table **Student** will now look like,

| **ROLL\_NO** | **NAME** | **ADDRESS** | **PHONE** | **Age** |
| --- | --- | --- | --- | --- |
| 1 | Ram | Delhi | 9455123451 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |
| 4 | SURESH | Delhi | 9156768971 | 18 |
| 2 | RAMESH | GURGAON | 9562431543 | 18 |

* **Delete all of the records:** There are two queries to do this as shown below,
* query1: "DELETE FROM Student";
* query2: "DELETE \* FROM Student";

**Output:**  
All of the records in the table will be deleted, there are no records left to display. The table**Student** will become empty!