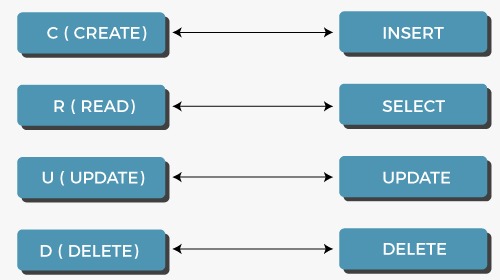
# CRUD Operations in SQL

As we know, CRUD operations act as the foundation of any computer programming language or technology. So before taking a deeper dive into any programming language or technology, one must be proficient in working on its CRUD operations. This same rule applies to databases as well.

Let us start with the understanding of CRUD operations in SQL with the help of examples. We will be writing all the queries in the supporting examples using the MySQL database.



## 1. Create:

In CRUD operations, 'C' is an acronym for **create,** which means to add or insert data into the SQL table. So, firstly we will create a table using CREATE command and then we will use the INSERT INTO command to insert rows in the created table.

**Syntax for table creation:**

Backward Skip 10sPlay VideoForward Skip 10s

1. **CREATE** **TABLE** Table\_Name (ColumnName1 Datatype, ColumnName2 Datatype,..., ColumnNameN Datatype);

where,

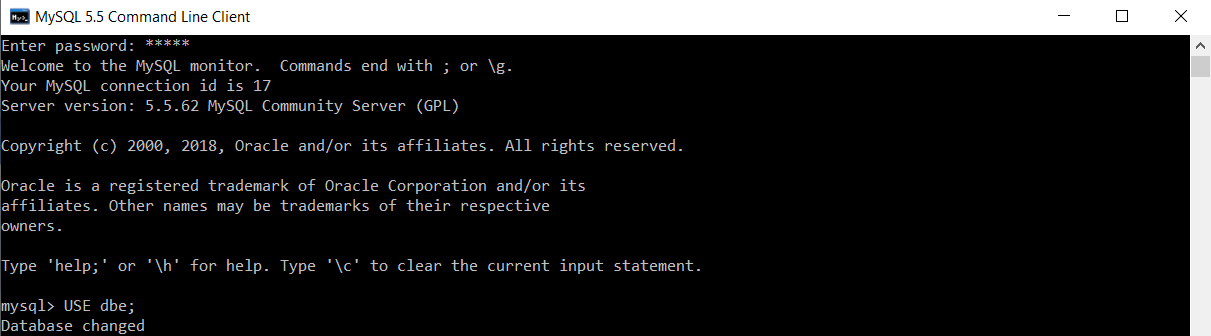
* Table\_Name is the name that we want to assign to the table.
* Column\_Name is the attributes under which we want to store data of the table.
* Datatype is assigned to each column. Datatype decides the type of data that will be stored in the respective column.

**Syntax for insertion of data in table:**

1. **INSERT** **INTO** Table\_Name (ColumnName1,...., ColumnNameN) **VALUES** (Value 1,....,Value N),....., (Value 1,....,Value N);

Prior to the creation of a table in SQL, we need to create a database or select an existing database. Since we already had a database, we will select the database with the USE command.

1. mysql> USE dbe;



Now, we will write a query to create a table named **employee** in the database named **dbe.**

1. mysql> **CREATE** **TABLE** employee(ID **INT** **PRIMARY** **KEY**, First\_Name **VARCHAR**(20), Last\_Name **VARCHAR**(20), Salary **INT**, Email\_Id **VARCHAR**(40));

IMG_258

To ensure that the table is created as per the column names, data types and sizes which we have assigned during table creation, we will execute the following query:

1. mysql> **DESC** employee;

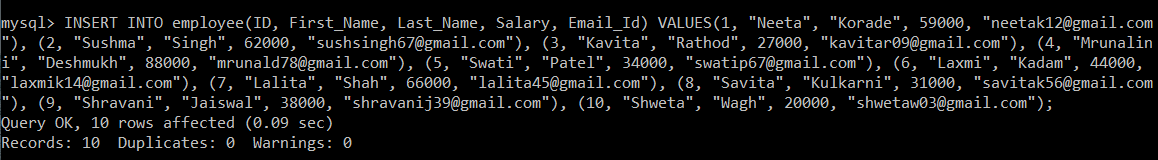
You will get the following output:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| ID | int(11) | NO | PRI | NULL |  |
| First\_Name | varchar(20) | YES |  | NULL |  |
| Last\_Name | varchar(20) | YES |  | NULL |  |
| Salary | int(11) | YES |  | NULL |  |
| Email\_Id | varchar(40) | YES |  | NULL |  |

The above results verify that the table is created successfully as per the requirements.

We will execute the following query to insert multiple records in the employee table:

1. **INSERT** **INTO** employee(ID, First\_Name, Last\_Name, Salary, Email\_Id) **VALUES**(1, "Neeta", "Korade", 59000, "neetak12@gmail.com"), (2, "Sushma", "Singh", 62000, "sushsingh67@gmail.com"), (3, "Kavita", "Rathod", 27000, "kavitar09@gmail.com"), (4, "Mrunalini", "Deshmukh", 88000, "mrunald78@gmail.com"), (5, "Swati", "Patel", 34000, "swatip67@gmail.com"), (6, "Laxmi", "Kadam", 44000, "laxmik14@gmail.com"), (7, "Lalita", "Shah", 66000, "lalita45@gmail.com"), (8, "Savita", "Kulkarni", 31000, "savitak56@gmail.com"), (9, "Shravani", "Jaiswal", 38000, "shravanij39@gmail.com"), (10, "Shweta", "Wagh", 20000, "shwetaw03@gmail.com");



## 2. Read:

In CRUD operations, 'R' is an acronym for **read,** which means **retrieving or fetching the data from the SQL table.** So, we will use the SELECT command to fetch the inserted records from the SQL table. We can retrieve all the records from a table using an asterisk (\*) in a SELECT query. There is also an option of retrieving only those records which satisfy a particular condition by using the WHERE clause in a SELECT query.

**Syntax to fetch all the records:**

1. **SELECT** \***FROM** TableName;

**Syntax to fetch records according to the condition:**

1. **SELECT** \***FROM** TableName **WHERE** CONDITION;

**Example 1:**

Write a query to fetch all the records stored in the employee table.

**Query:**

1. mysql> **SELECT** \***FROM** employee;

Here, an asterisk is used in a SELECT query. This means all the column values for every record will be retrieved.

You will get the following output after executing the above query:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **First\_Name** | **Last\_Name** | **Salary** | **Email\_Id** |
| 1 | Neeta | Korade | 59000 | neetak12@gmail.com |
| 2 | Sushma | Singh | 62000 | sushsingh67@gmail.com |
| 3 | Kavita | Rathod | 27000 | kavitar09@gmail.com |
| 4 | Mrunalini | Deshmukh | 88000 | mrunald78@gmail.com |
| 5 | Swati | Patel | 34000 | swatip67@gmail.com |
| 6 | Laxmi | Kadam | 44000 | laxmik14@gmail.com |
| 7 | Lalita | Shah | 66000 | lalita45@gmail.com |
| 8 | Savita | Kulkarni | 31000 | savitak56@gmail.com |
| 9 | Shravani | Jaiswal | 38000 | shravanij39@gmail.com |
| 10 | Shweta | Wagh | 20000 | shwetaw03@gmail.com |

All the records are successfully retrieved from the employee table.

**Example 2:**

Write a query to fetch only those records from the employee table whose salary is above 35000.

**Query:**

1. mysql> **SELECT** \***FROM** employee **WHERE** Salary > 35000;

Here, an asterisk is used in a SELECT query. This means all the column values for every record will be retrieved. We have applied the WHERE clause on Salary, which means the records will be filtered based on salary.

You will get the output as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **First\_Name** | **Last\_Name** | **Salary** | **Email\_Id** |
| 1 | Neeta | Korade | 59000 | neetak12@gmail.com |
| 2 | Sushma | Rathod | 62000 | sushsingh67@gmail.com |
| 4 | Mrunalini | Deshmukh | 88000 | mrunald78@gmail.com |
| 6 | Laxmi | Kadam | 44000 | laxmik14@gmail.com |
| 7 | Lalita | Shah | 66000 | lalita45@gmail.com |
| 9 | Shravani | Jaiswal | 38000 | shravanij39@gmail.com |

There are six records in the employee table whose salary is above 35000.

## 3. Update:

In CRUD operations, 'U' is an acronym for the **update,** which **means making updates to the records present in the SQL tables.** So, we will use the UPDATE command to make changes in the data present in tables.

**Syntax:**

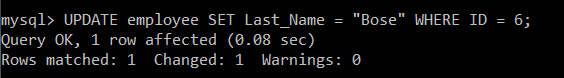
1. **UPDATE** Table\_Name **SET** ColumnName = Value **WHERE** CONDITION;

**Example 1:**

Write a query to update an employee's last name as 'Bose', whose employee id is 6.

**Query:**

1. mysql> **UPDATE** employee **SET** Last\_Name = "Bose" **WHERE** ID = 6;



Here in the SELECT query, we have used the SET keyword to update an employee's last name as 'Bose'. We want to update an employee's last name only for the employee with id 6, so we have specified this condition using the WHERE clause.

To ensure that an employee's last name with employee id 6 is updated successfully, we will execute the SELECT query.

1. mysql> **SELECT** \***FROM** employee;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **First\_Name** | **Last\_Name** | **Salary** | **Email\_Id** |
| 1 | Neeta | Korade | 59000 | neetak12@gmail.com |
| 2 | Sushma | Singh | 62000 | sushsingh67@gmail.com |
| 3 | Kavita | Rathod | 27000 | kavitar09@gmail.com |
| 4 | Mrunalini | Deshmukh | 88000 | mrunald78@gmail.com |
| 5 | Swati | Patel | 34000 | swatip67@gmail.com |
| 6 | Laxmi | Bose | 44000 | laxmik14@gmail.com |
| 7 | Lalita | Shah | 66000 | lalita45@gmail.com |
| 8 | Savita | Kulkarni | 31000 | savitak56@gmail.com |
| 9 | Shravani | Jaiswal | 38000 | shravanij39@gmail.com |
| 10 | Shweta | Wagh | 20000 | shwetaw03@gmail.com |

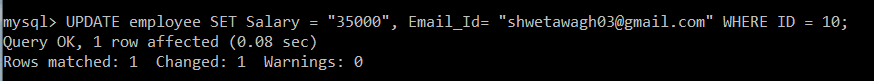
The results above verify that an employee's last name with employee id 6 is now changed to 'Bose'.

**Example 2:**

Write a query to update the salary and email id of an employee as '35000' and 'shwetawagh03@gmail.com', respectively, whose employee id is 10.

**Query:**

1. mysql> **UPDATE** employee **SET** Salary = "35000", Email\_Id= "shwetawagh03@gmail.com" **WHERE** ID = 10;



Here in the UPDATE query, we have used the SET keyword to update an employee's salary as '35000' and the email id as 'shwetawagh03@gmail.com'. We want to update the salary and email id of an employee only for the employee with id 10, so we have specified this condition using the WHERE clause.

To ensure that the salary and email id of an employee with employee id 10 is updated successfully, we will execute the SELECT query.

1. mysql> **SELECT** \***FROM** employee;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **First\_Name** | **Last\_Name** | **Salary** | **Email\_Id** |
| 1 | Neeta | Korade | 59000 | neetak12@gmail.com |
| 2 | Sushma | Singh | 62000 | sushsingh67@gmail.com |
| 3 | Kavita | Rathod | 27000 | kavitar09@gmail.com |
| 4 | Mrunalini | Deshmukh | 88000 | mrunald78@gmail.com |
| 5 | Swati | Patel | 34000 | swatip67@gmail.com |
| 6 | Laxmi | Bose | 44000 | laxmik14@gmail.com |
| 7 | Lalita | Shah | 66000 | lalita45@gmail.com |
| 8 | Savita | Kulkarni | 31000 | savitak56@gmail.com |
| 9 | Shravani | Jaiswal | 38000 | shravanij39@gmail.com |
| 10 | Shweta | Wagh | 35000 | shwetaw03@gmail.com |

The results above verify that the salary and email id of an employee with employee id 10 is now changed to '35000' and 'shwetawagh03@gmail.com', respectively.

## 4. Delete:

In CRUD operations, 'D' is an acronym for **delete,** which means **removing or deleting the records from the SQL tables.** We can delete all the rows from the SQL tables using the DELETE query. There is also an option to remove only the specific records that satisfy a particular condition by using the WHERE clause in a DELETE query.

**Syntax to delete all the records:**

1. **DELETE** **FROM** TableName;

**Syntax to delete records according to the condition:**

1. **DELETE** **FROM** TableName **WHERE** CONDITION;

**Example 1:**

Write a query to delete the employee record from the employee table whose salary is above 34000.

**Query:**

1. mysql> **DELETE** **FROM** employee **WHERE** Salary > 34000;

IMG_262

Here we have applied the DELETE query on the employee table. We want to delete only the employee record whose salary is 34000, so we have specified this condition using the WHERE clause.

We will execute the SELECT query to ensure that the employee record with salary as 34000 is deleted successfully.

1. mysql> **SELECT** \***FROM** employee;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **First\_Name** | **Last\_Name** | **Salary** | **Email\_Id** |
| 1 | Neeta | Korade | 59000 | neetak12@gmail.com |
| 2 | Sushma | Singh | 62000 | sushsingh67@gmail.com |
| 3 | Kavita | Rathod | 27000 | kavitar09@gmail.com |
| 4 | Mrunalini | Deshmukh | 88000 | mrunald78@gmail.com |
| 6 | Laxmi | Bose | 44000 | laxmik14@gmail.com |
| 7 | Lalita | Shah | 66000 | lalita45@gmail.com |
| 8 | Savita | Kulkarni | 31000 | savitak56@gmail.com |
| 9 | Shravani | Jaiswal | 38000 | shravanij39@gmail.com |
| 10 | Shweta | Wagh | 35000 | shwetaw03@gmail.com |

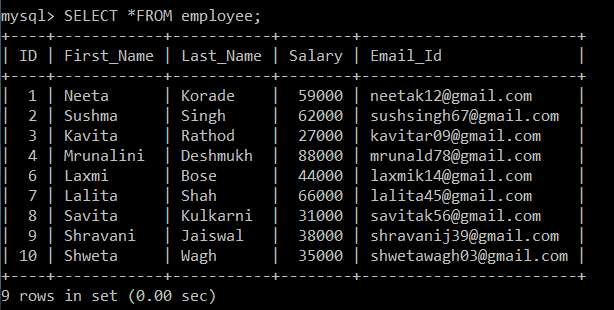
The results above verify that the employee with a salary of 34000 no longer exists in the employee table.

**Example 2:**

Write a query to delete all the records from the employee table.

First, let us see the employee table, which is available currently.

1. mysql> **SELECT** \***FROM** employee;



To remove all the records from the employee table, we will execute the DELETE query on the employee table.

1. mysql> **DELETE** **FROM** employee;

IMG_264

We will execute the SELECT query to ensure that all the records are deleted successfully from the employee table.

1. mysql> **SELECT** \***FROM** employee;

IMG_265

The results above verify that the employee table does not contain any record now.