MySQL - Insert Query

After creating a table in a MySQL database with the CREATE TABLE statement, we will only have an empty table that only has its structure defined. To populate it with data, we need to add records manually using separate queries.

The MySQL INSERT Statement

To insert data into a MySQL table, we would need to use the MySQL INSERT statement. We can insert data into the MySQL table by using the 'mysql>' prompt or by using any client program such as PHP, Java etc.

Since the structure of a table is already defined, the MySQL INSERT statement will only accept the data which is according to the structure of the table. Data inserted into a table must have same data types, satisfy the constraints (if any), etc. If the inserted data does not satisfy these conditions, the INSERT INTO statement displays an error.

Syntax

Following is the syntax of the MySQL INSERT statement −

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)

VALUES (value1, value2, value3,...valueN);
```

To insert string values, it is required to keep all the values into double or single quotes. For example "value".

Inserting Data from the Command Prompt

To insert data from the command prompt, we will use SQL **INSERT INTO** statement to insert data into an MySQL table.

Example

First of all, let us create a table named CUSTOMERS using the following query

```
CREATE TABLE CUSTOMERS (
   ID INT NOT NULL,
   NAME VARCHAR(20) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR (25),
   SALARY DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

```
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY) VALUES
(1, 'Ramesh', 32, 'Ahmedabad', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY) VALUES
(2, 'Khilan', 25, 'Delhi', 1500.00 ),
(3, 'Kaushik', 23, 'Kota', 2000.00 ),
(4, 'Chaitali', 25, 'Mumbai', 6500.00 ),
(5, 'Hardik', 27, 'Bhopal', 8500.00 );
```

Inserting records into a database is also possible even if you do not specify the column name if the comma separated values in the query match the attributes of corresponding columns as shown below –

```
INSERT INTO CUSTOMERS VALUES
(6, 'Komal', 22, 'Hyderabad', 4500.00 ),
(7, 'Muffy', 24, 'Indore', 10000.00 );

SELECT * FROM CUSTOMERS;
```

Inserting Data Into a Table Using Another Table

Sometimes, we just need to copy the data from one existing table in a database to another table in the same database. And there are various ways to do so −

- Using INSERT... SELECT
- Using INSERT... TABLE

INSERT... SELECT Statement

We can populate the data into a table through the select statement over another table; provided the other table has a set of fields, which are required to populate the first table.

Here is the syntax –

```
INSERT INTO table_name1 [(column1, column2, ... columnN)]
SELECT column1, column2, ...columnN
FROM table_name2
[WHERE condition];
```

Example

In the following query, we are creating another table **CUSTOMERS_Copy** with the same structure as **CUSTOMERS** table –

```
CREATE TABLE CUSTOMERS_Copy (
   ID INT AUTO_INCREMENT,
   NAME VARCHAR(20) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR (25),
   SALARY DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

Now, let us use the INSERT...INTO statement to insert the records into the CUSTOMERS_Copy table from CUSTOMERS table.

```
INSERT INTO CUSTOMERS_Copy SELECT * from CUSTOMERS;
```

```
SELECT * FROM CUSTOMERS_Copy;
```

INSERT...TABLE Statement

On the other hand, instead of selecting specific columns, we can insert the contents of one table into another using
the **INSERT...TABLE** statement.

Following is the syntax to do so −

```
INSERT INTO table1 TABLE table2;
```

Example

In this example, let us use the same CUSTOMERS table we have created in the previous example and copy its contents to another table CUSTOMERS_dummy.

For that, first of all, we will create the table CUSTOMERS_dummy with the same structure as CUSTOMERS table −

```
CREATE TABLE CUSTOMERS_dummy (
   ID INT AUTO_INCREMENT,
   NAME VARCHAR(20) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR (25),
   SALARY DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

Using the CUSTOMERS table, we will insert all its values into CUSTOMERS_dummy table −

```
INSERT INTO CUSTOMERS_dummy TABLE CUSTOMERS;
```

```
SELECT * FROM CUSTOMERS_dummy;
```

INSERT ... SET

You can insert a record by setting values to selected columns using the INSERT...SET statement. Following is the syntax of this statement −

```
INSERT INTO table_name SET column_name1 = value1,
column_name2=value2,.....;
```

Where, table_name is the name of the table into which you need to insert the record and column_name1 = value1, column_name2 = value2 ..... are the selected column names and the respective values.

If you are inserting values into a table using the INSERT ... SET statement and if you provide values for only a certain columns the values in the remaining will be NULL.

Example

Following query inserts a record into the CUSTOMERS table using the INSERT...SET statement. Here, we are passing values only to the ID, NAME and, AGE columns (remaining values will be NULL) –

```
INSERT INTO CUSTOMERS
SET ID = 8, NAME = 'Sarmista', AGE = 35;

SELECT * FROM CUSTOMERS WHERE ID=8;
```

Inserting Data Using a Client Program

```
sql = "INSERT INTO tutorials_tbl VALUES (1, 'Learn PHP', 'John Paul',
'2023-3-28')"
cursorObj.execute(sql)
```

```
import mysql.connector
import datetime
#establishing the connection
connection = mysql.connector.connect(
   host='localhost',
   user='root',
   password='password',
   database='tut'
)
table_name = 'tutorials_tbl'
new_tutorial_data = [
   (2, 'Learn MySQL', 'Abdul S', '2023-03-28'),
   (3, 'JAVA Tutorial', 'Sanjay', '2007-05-06'),
   (4, 'Python Tutorial', 'Sasha Lee', '2016-09-04'),
   (5, 'Hadoop Tutorial', 'Chris Welsh', '2023-03-28'),
   (6, 'R Tutorial', 'Vaishnav', '2011-11-04')
]
#Creating a cursor object
cursorObj = connection.cursor()
cursorObj.execute("truncate table tutorials_tbl")
sql = "INSERT INTO tutorials_tbl VALUES (1, 'Learn PHP', 'John Paul', '2023-3-28')"
cursorObj.execute(sql)
insert_query = f'INSERT INTO {table_name} (tutorial_id, tutorial_title, tutorial_author,
submission_date) VALUES (%s, %s, %s, %s)'
cursorObj.executemany(insert_query, new_tutorial_data)
connection.commit()
print("Row inserted successfully.")
```

```
cursorObj.close()

connection.close()
```