Insert Into Select

The MySQL Insert Into Select Statement

In MySQL, the **INSERT INTO... SELECT** statement is used to add/insert one or more rows from an existing table to target table.

This statement is a combination of two different statements: INSERT INTO and SELECT.

- The MySQL INSERT INTO statement is a commonly used command in database management and it requires only the name of the table and the values to be inserted into a table. However, it is important to ensure that the data being inserted matches the structure and data types of the table columns.
- The SELECT statement is used to fetch data from an existing database table.

When the above mentioned statements are used together, the SELECT statement first fetches the data from an existing table and the **INSERT INTO** statement inserts the retrieved data into another table (if they have same table structures).

Syntax

Following is the syntax for using insert into select statement –

```
INSERT INTO table2 (column1, column2, column3, ...)
SELECT column1, column2, column3, ...
FROM table1
WHERE condition;
```

Following are some important points that we have to consider before we execute the below queries −

- In the database where we are going to insert data, a table must already exist.
- Both the source and target tables must match its structure.

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 ),
(2, 'Khilan', 25, 'Delhi', 1500.00 ),
(3, 'Kaushik', 23, 'Kota', 2000.00 ),
(4, 'Chaitali', 25, 'Mumbai', 6500.00 ),
(5, 'Hardik', 27, 'Bhopal', 8500.00 ),
(6, 'Komal', 22, 'Hyderabad', 4500.00 ),
(7, 'Muffy', 24, 'Indore', 10000.00 );
```

```
Select * From CUSTOMERS;
```

Inserting Required Data from one Table to Another Table

We may come across some instances where we only want to add small number of records to another table. This can be achieved by using a WHERE clause to select all the number of rows that the query returned.

Example

Before that, let us create a another table named **CUSTOMERS_copy** with similar structure of previously created 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)
);
```

In the following query, we are trying to fetch the records from the CUSTOMERS table and insert them into the CUSTOMERS_copy table.

```
INSERT INTO CUSTOMERS_copy (ID, NAME, AGE, ADDRESS, SALARY)
SELECT ID, NAME, AGE, ADDRESS, SALARY FROM CUSTOMERS
WHERE AGE >= 25;
```

```
SELECT * FROM CUSTOMERS_copy;
```

Inserting the rows with LIMIT

Using the MySQL LIMIT clause, we can specify the number of rows from the query that should be added to the target table.

Example

Before proceeding further, let us first truncate all rows in the **CUSTOMERS_copy** table using the following query –

```
TRUNCATE TABLE CUSTOMERS_copy;
```

Now, we are going to insert the top 3 records from CUSTOMERS table sorted by their AGE using the LIMIT clause –

```
INSERT INTO CUSTOMERS_copy (ID, NAME, AGE, ADDRESS, SALARY)
SELECT ID, NAME, AGE, ADDRESS, SALARY FROM CUSTOMERS
ORDER BY AGE LIMIT 3;
```

```
SELECT * FROM CUSTOMERS_copy;
```

Inserting All Columns from one Table to Another Table

We can also insert every column from one to another table. To do so, following is the syntax –

```
INSERT INTO table2
SELECT * FROM table1
```

Before inserting all the records, first truncate all rows in the CUSTOMERS_copy table by using the statement –

```
TRUNCATE TABLE CUSTOMERS_copy;
```

In the following query, we are trying to add all the columns from the CUSTOMERS table to the CUSTOMERS_copy table –

```
INSERT INTO CUSTOMERS_copy SELECT * FROM CUSTOMERS;
```

```
SELECT * FROM CUSTOMERS_copy;
```

```
sql = "INSERT INTO new_tutorials_tbl
(tutorial_id, tutorial_title, tutorial_author, submission_date)
SELECT tutorial_id, tutorial_title, tutorial_author, submission_date FROM tutorials_tbl"
insert_into_select_query = sql
cursorObj.execute(insert_into_select_query)
```

```
import mysql.connector
#establishing the connection
connection = mysql.connector.connect(
   host='localhost',
   user='root',
   password='password',
   database='tut'
)
cursorObj = connection.cursor()
insert_into_select_query = "INSERT INTO new_tutorials_tbl (tutorial_id, tutorial_title, tutorial_author,
submission_date) SELECT tutorial_id, tutorial_title, tutorial_author, submission_date FROM
tutorials_tbl"
cursorObj.execute(insert_into_select_query)
connection.commit()
print("Data inserted into new table successfully.")
cursorObj.close()
connection.close()
```