

# INSERT INTO to Add Row

by Sophia Tutorial



## WHAT'S COVERED

This tutorial explores using INSERT statements to add a single row into an existing table in two parts:

1. Using the INSERT INTO Statement
2. Examples

## 1. Using the INSERT INTO Statement

We have used the INSERT statement a few times in prior tutorials, but we will get a chance to explore the details around the statement in this tutorial. The basic syntax of the INSERT statement looks like this:

```
INSERT INTO <tablename> (<column1>, <column2>, ...)  
VALUES (<value1>, <value2>, ...);
```

To break this down, we have to identify the table that we want to insert into to replace <tablename> after the keywords INSERT INTO. Then we must list the column names separated by commas. Next, we have a list of the values in parenthesis after the VALUES keyword in the same order that we have the columns. It is important that we use quotes around dates, characters, and strings in the value list. Note that we must include all columns in a table that has a PRIMARY KEY or NOT NULL constraint in place. The only exceptions are if the column has a DEFAULT value or if the column has a sequence attached to it, which we will address in the next tutorial.

## 2. Examples

For example, to insert into the artist table, we can run the following statement:

```
INSERT INTO artist (artist_id, name)  
VALUES (1000, 'Bob Dylan');
```

### Query Results

Query ran successfully. 0 rows to display.

We can change the order of the columns if we also change the order of the values to match:

```
INSERT INTO artist (name, artist_id)
VALUES ('Michael Jackson', 1001);
```

### Query Results

Query ran successfully. 0 rows to display.

If we accidentally swapped the values and columns, we may get an error:

```
INSERT INTO artist (name, artist_id)
VALUES (1001, 'Michael Jackson');
```

### Query Results

Query failed because of: error: invalid input syntax for integer: "Michael Jackson"

However, this is unreliable error-checking. It catches that we tried to insert a string into an integer column. There are many other instances where we may be inserting wrong data into the same data type, like swapping two integer value columns. It is important that we are double-checking the order of the column list and the value list to match.

We do have to be careful of the existing constraints on a table, such as foreign keys, as this can prevent data from being inserted. For example, if we tried to insert the following statement:

```
INSERT INTO album (artist_id, album_id, title)
VALUES (999, 2000, 'Latest Hits');
```

The artist\_id in the album table references the artist table. If we looked at the artist table for the artist\_id equal to 999, we will find that this doesn't exist:

```
SELECT *
FROM artist
WHERE artist_id = 999;
```

As such, trying to insert it into the table would result in an error:

### Query Results

Query failed because of: error: insert or update on table "album" violates foreign key constraint "album\_artist\_id\_fkey"

If we wanted to add the album Latest Hits to the album table using the artist\_id 1000 that we inserted for Bob Dylan, the INSERT statement would look like this:

```
INSERT INTO album (artist_id, album_id, title)
VALUES (1000, 1001, 'Latest Hits');
```

## Query Results

Query ran successfully. 0 rows to display.

However, if we accidentally swapped the values:

```
INSERT INTO album (artist_id, album_id, title)
VALUES (1001, 1000, 'Latest Hits');
```

## Query Results

Query ran successfully. 0 rows to display.

No error would be displayed in this case, as we also have added an artist with the artist\_id value of 1001. This is why it is important to be careful regarding what the values represent, as these logical errors will not be caught by the database.

---

## Video Transcription

[MUSIC PLAYING] The insert statement is a statement in which we can actually utilize to be able to insert or add data to a table. The framework of the statement looks like this. It has insert into-- those are the keywords. Then we have the table name. In his case here, it's genre.

Then in parentheses, what we're going to include are the columns in which we're going to insert into. So in this case here, we're going to be choosing genre ID and the name. Then afterwards, we're going to have the key word values. And in the same order that we have the columns, we're going to pass in the specific data. So in this case here, we're going to pass in genre ID of 1,000 and then a name of funk. Going to go ahead and run that, and that's going to add in that data component.

It is important to note that every time that you insert data into the table, it will have to have a verification against all the different constraints that we have on the table. So in this case here, being that genre ID is the primary key, if we try to insert it again, we should get an error in this case, being that we're passing 1,000 again for that, and it does violate the constraint.



TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.



SUMMARY

The INSERT INTO statement allows us to insert data into an existing table.

Source: Authored by Vincent Tran