In this lesson we are going to look at INSERT INTO and that is a SQL command that lets us insert data into a table.

We have two empty tables now called *users* and *videos.* Let’s insert some data into our *user’s* table. If you have tried to insert data into our videos table. Remember that it is a FOREIGN KEY which is why it won’t let us do that.

Let’s INSERT one row into our *user’s* table.

INSERT INTO users(id, name)

VALUES (1, ‘Mary’);

After we execute our query. The data gets inserted into the table.

Now, we can use our SELECT command to view our table.

SELECT \* FROM users;

|  |  |
| --- | --- |
| id  integer | name  character varying (100) |
| 1 | Mary |

Now we will insert data into this table again, but this time without specifying the columns.

INSERT INTO users

VALUES (‘2’, ‘Rolf’);

SELECT \* FROM *users;*

|  |  |
| --- | --- |
| id  integer | name  character varying (100) |
| 1 | Mary |
| 2 | Rolf |

Because ‘id’ and ‘name’ are the only two columns on the table, we can omit the columns. If we are not inserting data on all the columns of the table then we need to specify the column names on our query.

Now let’s insert some data in our videos’ table.

|  |  |  |
| --- | --- | --- |
| id integer | user\_id | name  character varying (255) |

INSERT INTO videos

VALUES (1, 2, ‘Test video’);

SELECT \* FROM videos;

|  |  |  |
| --- | --- | --- |
| id  integer | user\_id  integer | name  character varying (255) |
| 1 | 2 | Test video |

Now as we have the FOREIGN KEY and PRIMARY KEY between the ids and the user ids we can perform an INNER JOIN between the ids in *users’* table and the user\_ids in videos table.

SELECT \* FROM videos INNER JOIN users ON users.id = videos.user\_id;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id  Integer | user\_id  integer | name  character varying (255) | id  integer | name  character varying (100) |
| 1 | 2 | Test video | 2 | rolfsmith |