# Creating the table

In this section of the tutorial, we will create the new table Employee. We have to mention the database name while establishing the connection object.

We can create the new table by using the CREATE TABLE statement of SQL. In our database PythonDB, the table Employee will have the four columns, i.e., name, id, salary, and department\_id initially.

The following query is used to create the new table Employee.

>  create table Employee (name varchar(20) **not** null, id int primary key, salary float **not** null, Dept\_Id int **not** null)

### Example

**import** mysql.connector

*#Create the connection object*

myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "google",database = "PythonDB")

*#creating the cursor object*

cur = myconn.cursor()

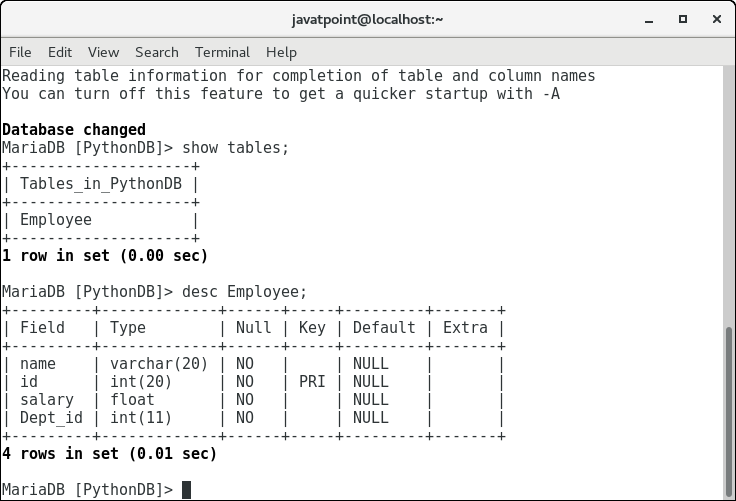
**try**:

*#Creating a table with name Employee having four columns i.e., name, id, salary, and department id*

    dbs = cur.execute("create table Employee(name varchar(20) not null, id int(20) not null primary key, salary float not null, Dept\_id int not null)")

**except**:

    myconn.rollback()

myconn.close()  

Now, we may check that the table Employee is present in the database.

## Alter Table

Sometimes, we may forget to create some columns, or we may need to update the table schema. The alter statement used to alter the table schema if required. Here, we will add the column branch\_name to the table Employee. The following SQL query is used for this purpose.

alter table Employee add branch\_name varchar(20) **not** null

Consider the following example.

### Example

**import** mysql.connector

*#Create the connection object*

myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "google",database = "PythonDB")

*#creating the cursor object*

cur = myconn.cursor()

**try**:

*#adding a column branch name to the table Employee*

    cur.execute("alter table Employee add branch\_name varchar(20) not null")

**except**:

    myconn.rollback()

myconn.close()

