

10 Database Programming

It is important for any application to store Data on Database Server so that it can be accessed easily whenever required.

10.1 API

API stands for Application Programming Interface. It is a set of Rules (Protocols), routine and Tools required for the communication between two applications. In other words API is an intermediary that allows two applications to communicate with each other.

Need for Database API

- If you design a python application that require storing data, retrieving data, updating data and delete data (CRUD) whenever required, we need a Database.
- In order to have communication between Python applications and Database Server we need Database Interface.
- Thus Database Interface is a middleman between the Python Application and Database Server.
- Most of the Python Database Interface follows (adhere) Python **DB-API V2** Standards.



10.2 Python Database Interface

Python Database API supports a wide range of Database available. Some of the name of the Databases and their Database Interfaces are given below.

Database Server	Database Interface
Informix	IfxPy
Postgresql	Psycopg2
Oracle	Cx_Oracle
MySQL	pymysql
Microsoft Access	pyodbc
Sybase	sqlanydb

10.2.1 pymysql

What is the Role of pip?

Pip is a package manager used to install pymysql database Interface required for database programming.

Installing pymysql

- 1) Check whether pymysql is already installed on your system or not.

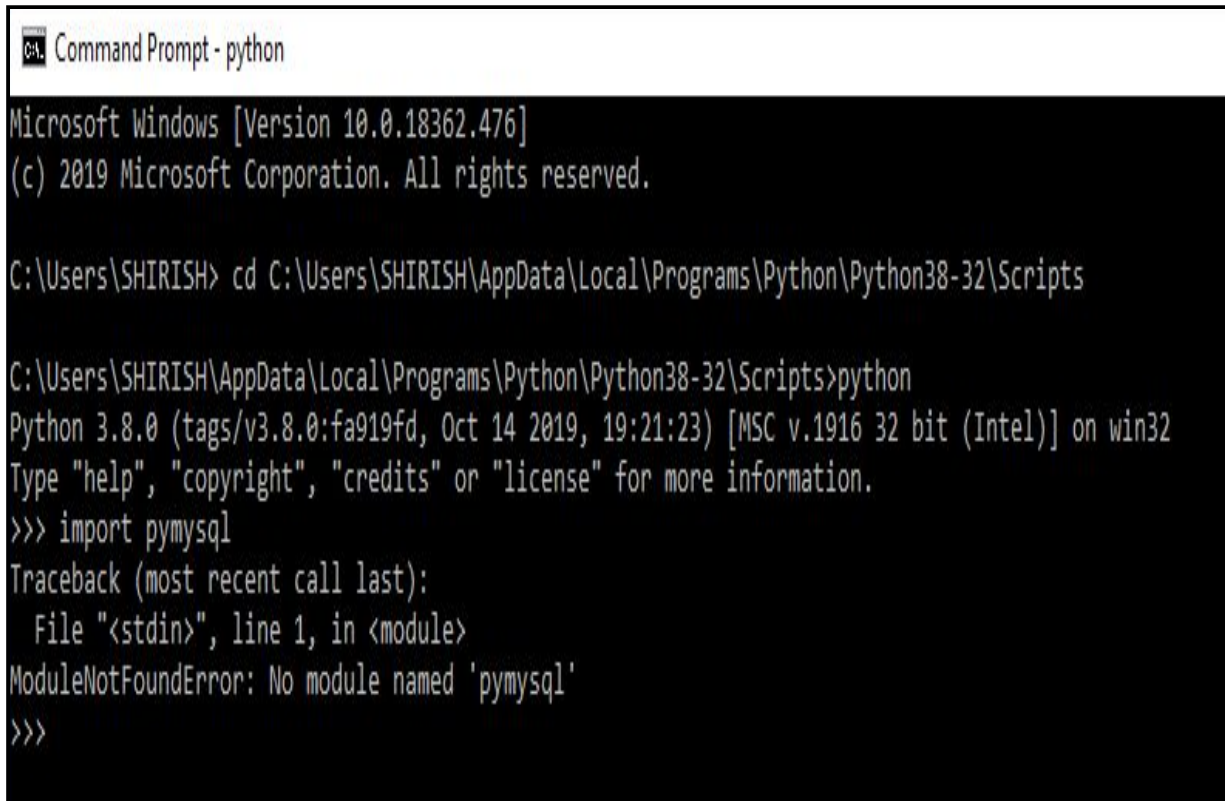
cmd-prompt

- a) Open cmd-prompt
- b) Change drive to

C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts

- c) After changing, type: python, it will give you the prompt (>>>)

d) >>> import pymysql



```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

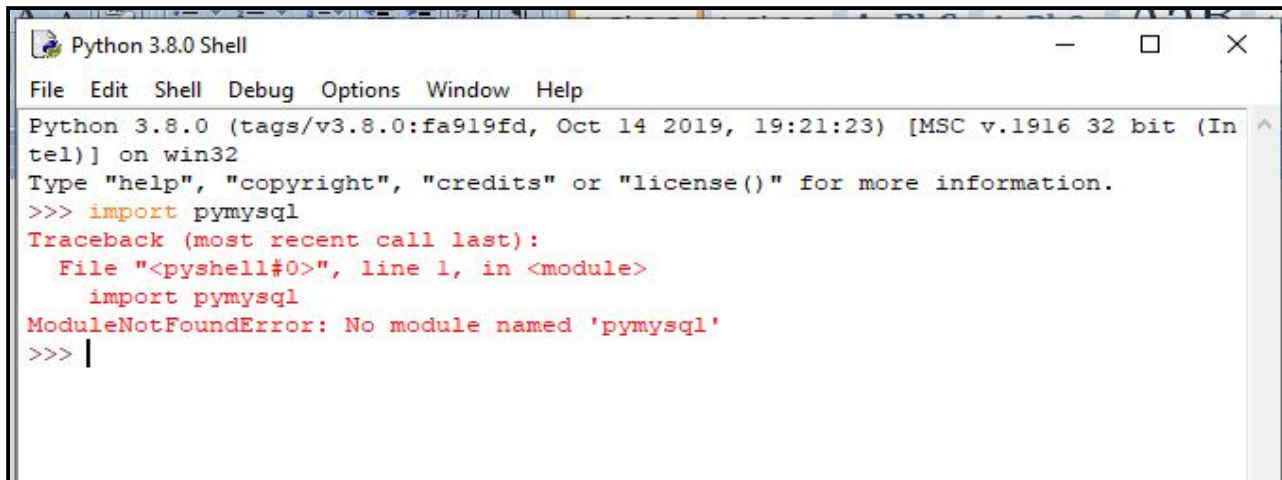
C:\Users\SHIRISH> cd C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts

C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts>python
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pymysql
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'pymysql'
>>>
```

e) If pymysql is not installed , you will get an error : **No Module named 'pymysql'**

10.2.2 IDLE:

- a) Open IDLE
- b) At prompt type: >>> import pymysql



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pymysql
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    import pymysql
ModuleNotFoundError: No module named 'pymysql'
>>> |
```

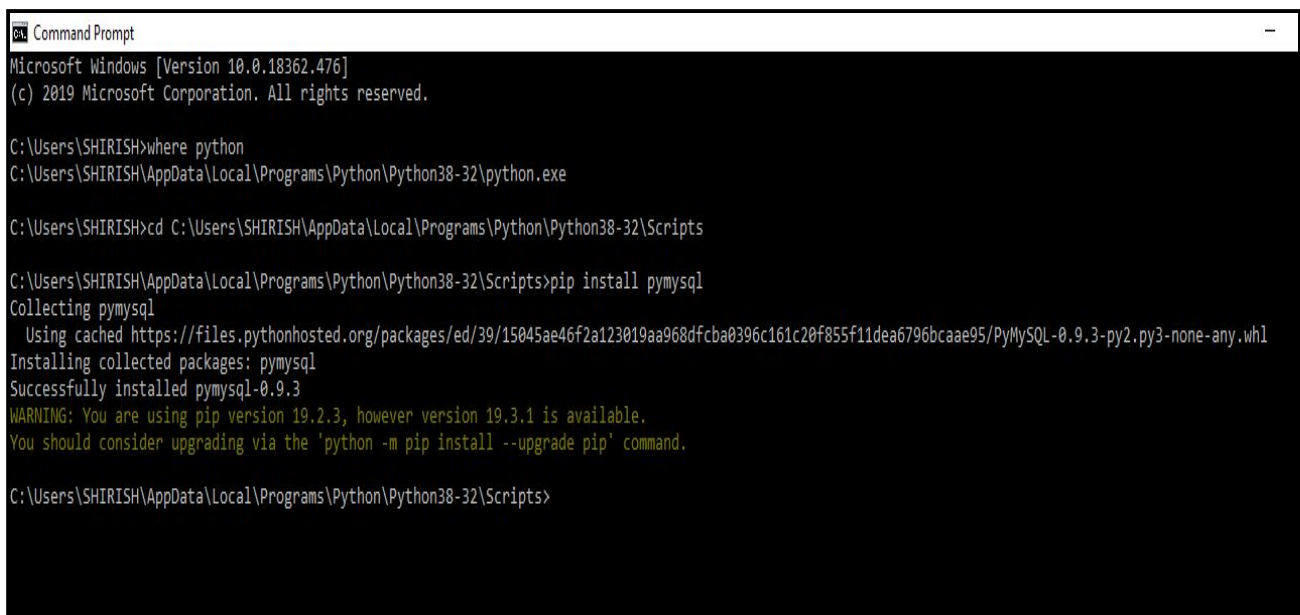
c) If pymysql is not installed , you will get an error : No Module named 'pymysql'

2) Installing pymysql database interface.

- a) Open cmd-prompt
- b) Change drive to

C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts>

c) Type **pip install pymysql**, it will start installing pymysql.



```
Command Prompt
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

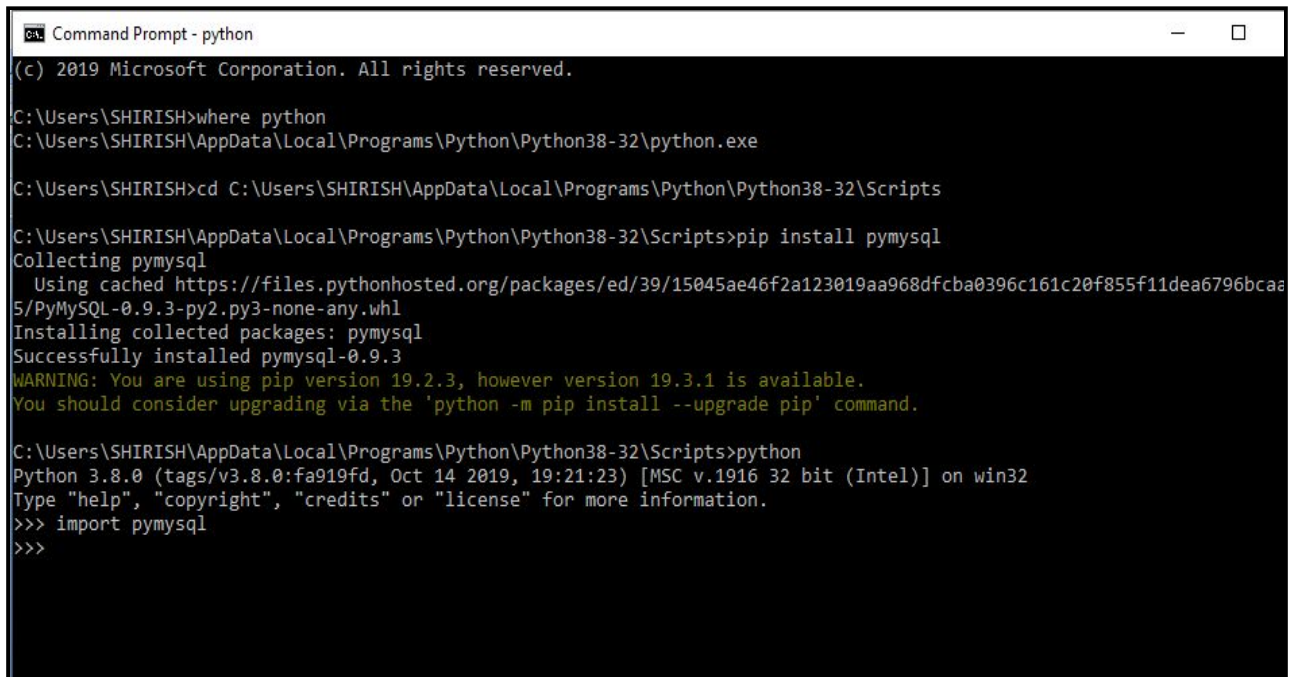
C:\Users\SHIRISH>where python
C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\python.exe

C:\Users\SHIRISH>cd C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts

C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts>pip install pymysql
Collecting pymysql
  Using cached https://files.pythonhosted.org/packages/ed/39/15045ae46f2a123019aa968dfcba0396c161c20f855f11dea6796bcaae95/PyMySQL-0.9.3-py2.py3-none-any.whl
Installing collected packages: pymysql
Successfully installed pymysql-0.9.3
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts>
```

- d) Type: python on cmd prompt, it will start prompt.
- e) At prompt type: >>> import pymysql.
- f) If you get an prompt again i.e >>>, then pymysql is successfully installed.



```
Command Prompt - python
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\SHIRISH>where python
C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\python.exe

C:\Users\SHIRISH>cd C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts

C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts>pip install pymysql
Collecting pymysql
  Using cached https://files.pythonhosted.org/packages/ed/39/15045ae46f2a123019aa968dfcba0396c161c20f855f11dea6796bcaae5/PyMySQL-0.9.3-py2.py3-none-any.whl
Installing collected packages: pymysql
Successfully installed pymysql-0.9.3
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

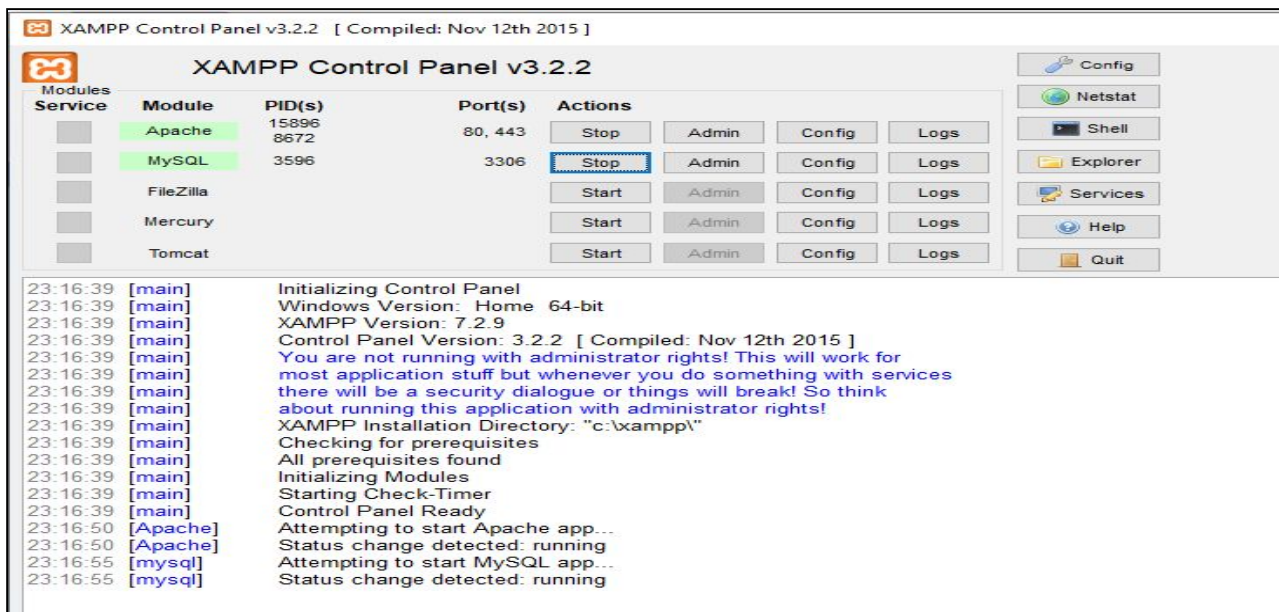
C:\Users\SHIRISH\AppData\Local\Programs\Python\Python38-32\Scripts>python
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pymysql
>>>
```

10.2.3 Python and Mysql

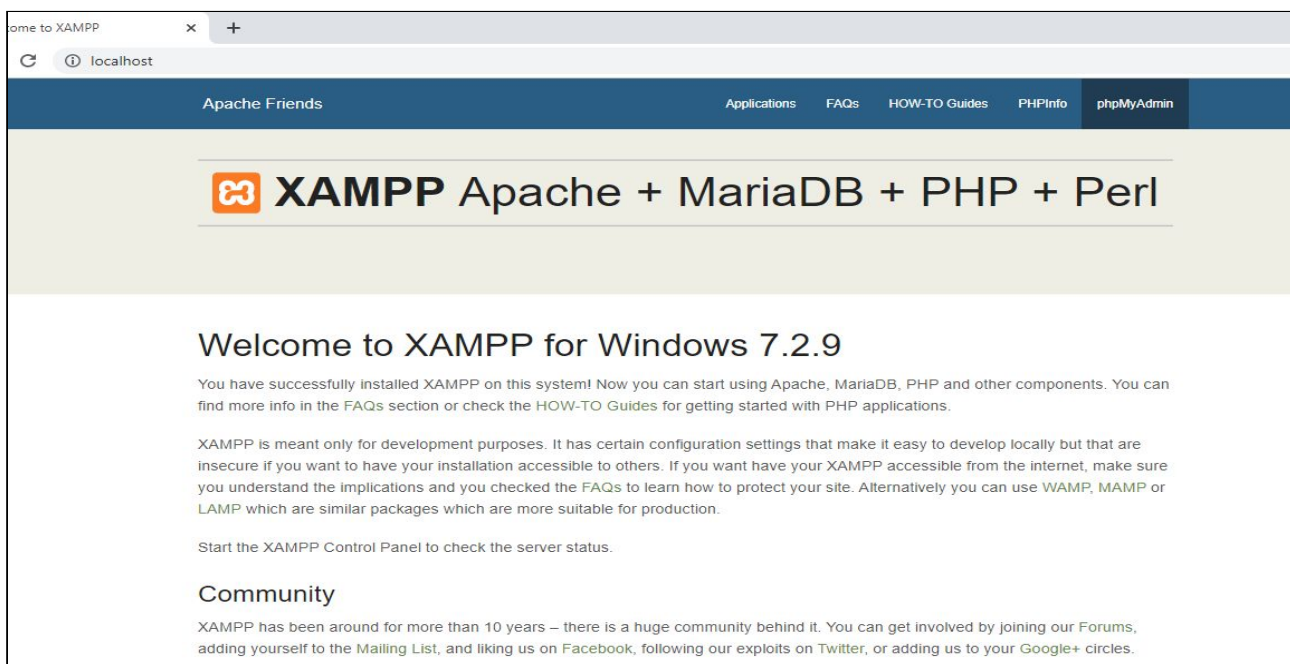
For Database Programming we need Server and Database. For this we install Xampp which is bundled with Apache Server and MariaDB (Mysql) database.

10.3 Creating Database

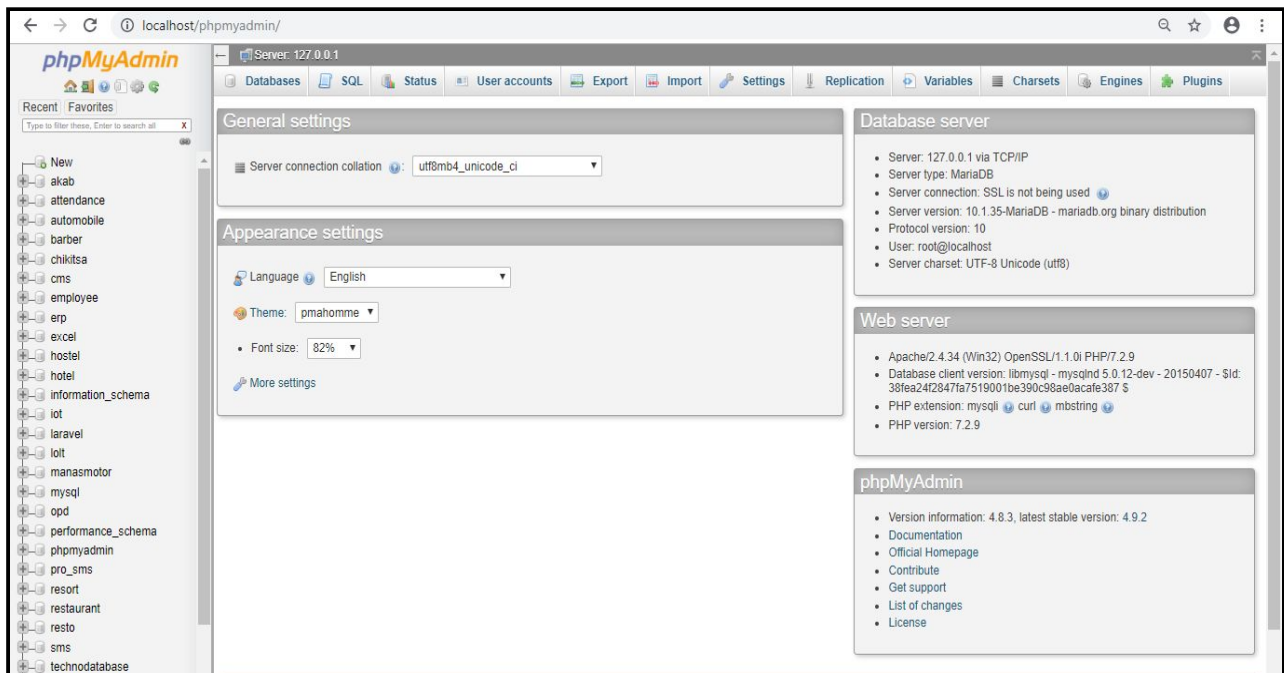
- 1) Start Xampp.
- 2) Start Apache and Mysql Service.



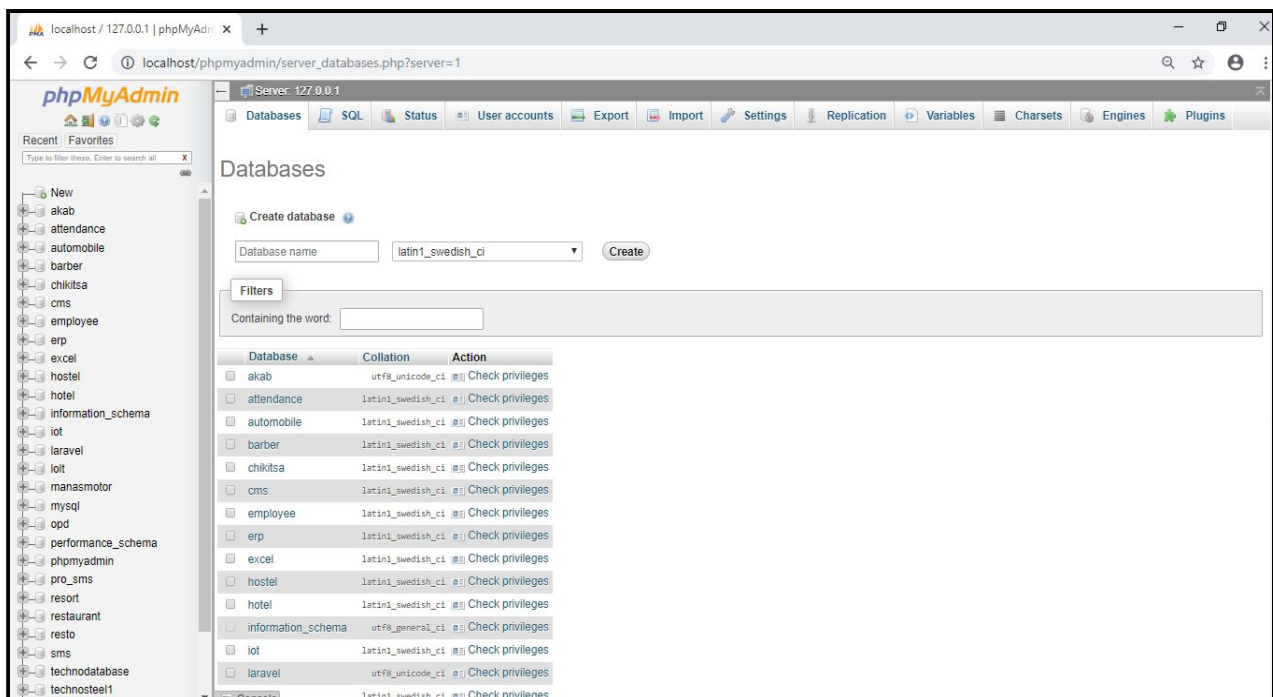
3) Open Browser and type localhost or localhost/dashboard in URL. You get the Xampp dashboard as shown.



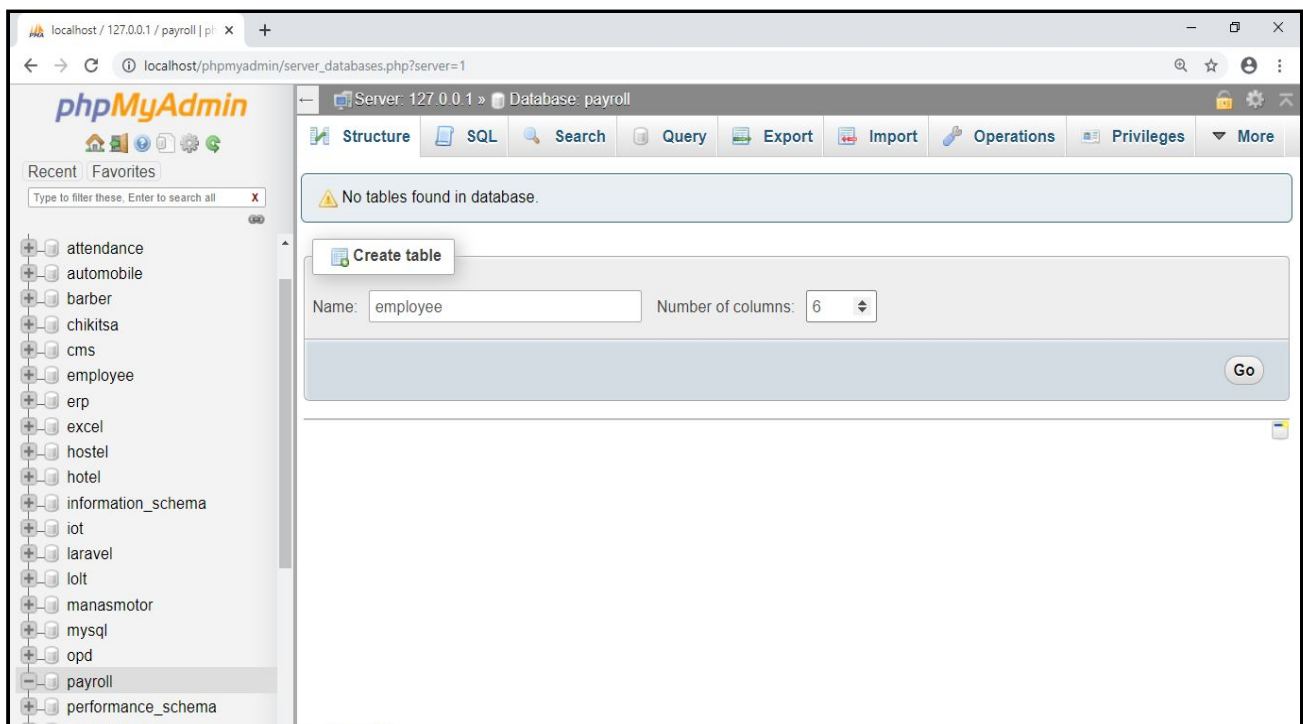
4) Click on **phpMyadmin** from Menubar. You will get phpMyadmin Dashboard.



5) On the left hand Side Click on New to Create Database.



- 6) Enter database name: payroll and click on the create Button. It will create a database payroll and will open a new window to create Table.
- 7) You can see the name of the Database **payroll** on the left-hand side.
- 8) Create Table with Table name: employee, Number of Column: 6 and click on GO to create a table.



- 9) After clicking Go, it will open a new window to enter the name of the columns, their data-type and their Length.

localhost / 127.0.0.1 / payroll / e/ x +

localhost/phpmyadmin/server_databases.php?server=1

phpMyAdmin

Recent Favorites

Type to filter these, Enter to search all

Server: 127.0.0.1 » Database: payroll » Table: employee

Table name: employee Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
id	INT	11	None			<input type="checkbox"/>	PRIMARY
Pick from Central Columns							
fname	VARCHAR	255	None			<input type="checkbox"/>	---
Pick from Central Columns							
lname	VARCHAR	255	None			<input type="checkbox"/>	---
Pick from Central Columns							
age	INT	11	None			<input type="checkbox"/>	---
Pick from Central Columns							
sex	VARCHAR	255	None			<input type="checkbox"/>	---
Pick from Central Columns							
income	FLOAT	11	None			<input type="checkbox"/>	---
Pick from Central Columns							

Table comments: Collation: Storage Engine: InnoDB

PARTITION definition:

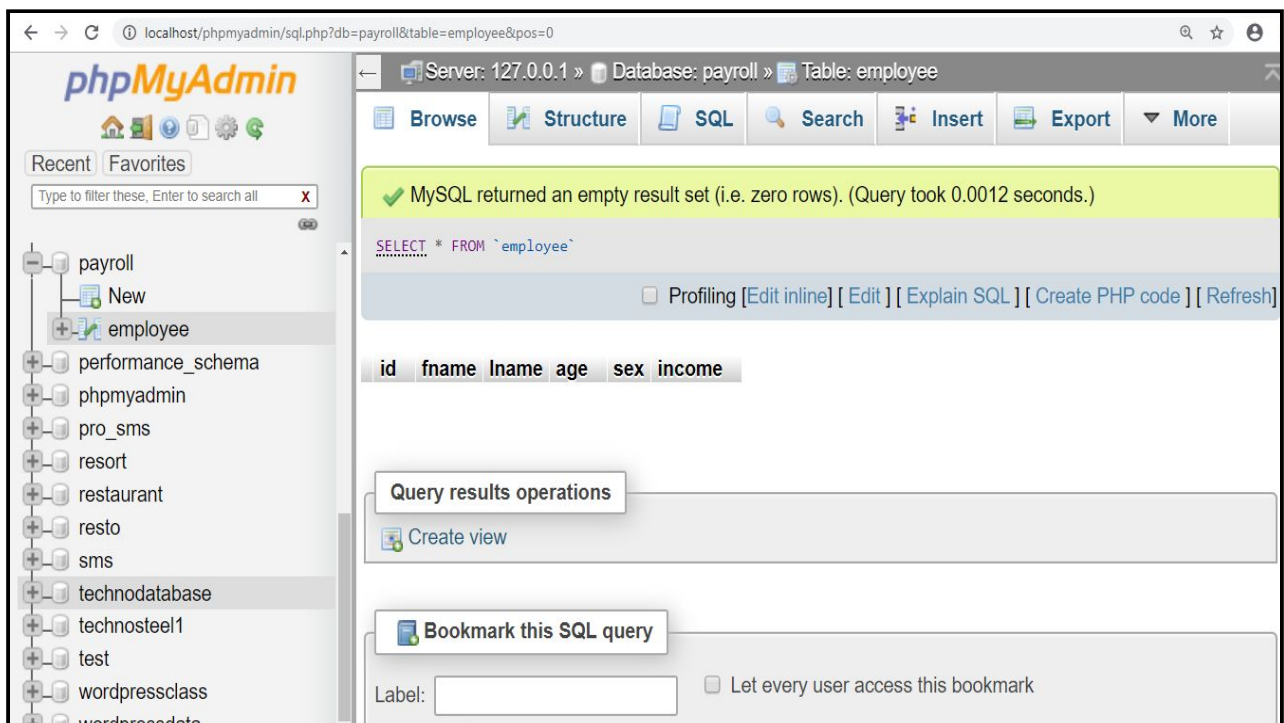
Partition by: (Expression or column list)

Partitions:

Console Preview SQL

Name	Type	Length	Auto-Increment
id	int	11	yes
fname	varchar	255	---
lname	varchar	255	---
age	int	11	---
sex	varchar	255	---
income	float	11	---

10) After clicking on SAVE (At right Bottom Corner), your employee database is created. Click on the employee database to view its column.



Creating employee.py file.

- **Inserting Record into Database**

1) Import pymysql module in the file.

- **import pymysql**

2) Use connect () method to connect to the database.

For Xampp

```
servername="localhost"
```

```
username="root"
```

```
password=""
```

```
dbname="payroll"
```

```
db=pymysql.connect (servername, username,password,dbname)
```

3) Creating Cursor

```
cursor = db.cursor()
```

4) Write a query to Insert the Record into Database.

```
sql="INSERT INTO employee (fname,lname,age,sex,income) VALUES  
( 'HARRY' , 'ROX' , '30' , 'Male' , '30000' )"
```

5) Execute the Query

```
cursor. execute(sql)
```

6) Commit changes in the database.

```
db.commit()
```

7) close the connection.

```
db.close()
```

- **Commit() and rollback in Database**

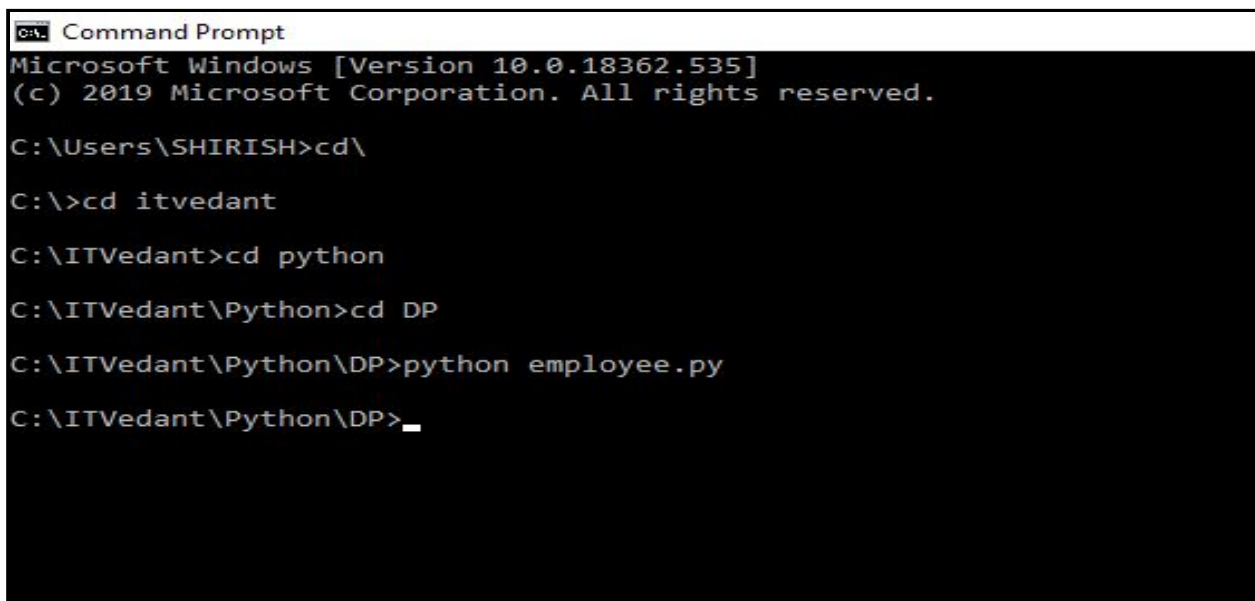
Whenever a transaction is successful and completed, the COMMIT statement helps the database to make changes permanently.

On the other hand if a failure occurs while executing ROLLBACK statement helps to undo the changes and return to the original unchanged state.

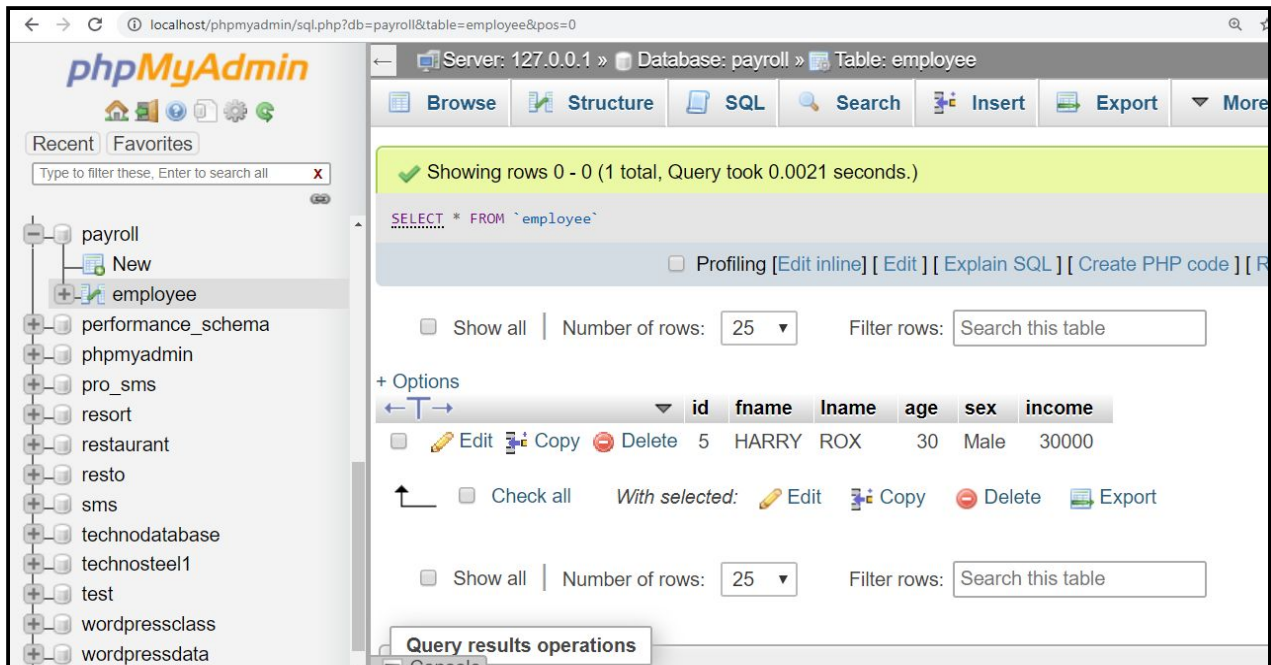
1. COMMIT commits the current transaction, making its changes permanent.
2. ROLLBACK rolls back the current transaction, canceling its changes.

- **Executing employee.py file**

- 1) Open command prompt or Terminal.
- 2) Change the drive to the folder where we had saved the employee.py.
- 3) Type: python employee.py

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the following text: "Microsoft Windows [Version 10.0.18362.535] (c) 2019 Microsoft Corporation. All rights reserved." followed by a series of directory change commands: "C:\Users\SHIRISH>cd\" (with a blank line), "C:\>cd itvedant", "C:\ITVedant>cd python", "C:\ITVedant\Python>cd DP", and "C:\ITVedant\Python\DP>python employee.py". The final line shows the prompt "C:\ITVedant\Python\DP>_" with a cursor.

- 4) After executing the command, the record gets inserted in the database.



- **Reading Record from the Database**

1) Import pymysql module in the file.

- **import pymysql**

2) Use connect () method to connect to the database.

For Xampp

```
servername="localhost"
```

```
username="root"
```

```
password=""
```

```
dbname="payroll"
```

```
db=pymysql.connect (servername, username,password,dbname)
```

3) Creating Cursor

```
cursor = db.cursor()
```

4) Write query to Insert the Record into Database.

```
sql="SELECT * FROM employee"
```

5) Execute the Query

```
cursor. execute(sql)
```

6) Fetch all the records from the Database

```
result=cursor.fetchall()
```

7) Printing records stored in the **result**

```
for x in result:  
    print(x)
```

8) Commit changes in the database.

```
db.commit()
```

9) close the connection.

```
db.close()
```

Command Prompt

Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\SHIRISH>cd\
C:\>cd itvedant
C:\ITVedant>cd python
C:\ITVedant\Python>cd DP
C:\ITVedant\Python\DP>python employee.py
C:\ITVedant\Python\DP>python employee.py
(5, 'HARRY', 'ROX', 30, 'Male', 30000.0)
C:\ITVedant\Python\DP>

localhost/phpmyadmin/sql.php?db=payroll...

Server: 127.0.0.1 » Database: payroll » Table: employee

BrowseStructureSQLSearchInsert

Showing rows 0 - 0 (1 total, Query took 0.0021 seconds.)

SELECT * FROM `employee`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25

+ Options

	id	fname	lname	age	sex	income
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	HARRY	ROX	30	Male	30000

☐ Check all With selected: ☐ Edit ☐ Copy ☐ Delete

Export

- **Updating Record from the Database**

In update we will be changing First name HARRY to MAC

1) Import pymysql module in the file.

```
import pymysql
```

2) Use connect () method to connect to the database.

For Xampp

```
servername="localhost"
```

```
username="root"
```

```
password=""
```

```
dbname="payroll"
```

```
db=pymysql.connect (servername, username,password,dbname)
```

3) Creating Cursor

```
cursor = db.cursor()
```

4) Write query to Insert the Record into Database.

```
sql="UPDATE  employee SET fname='MAC' WHERE  id=5"
```

5) Execute the Query

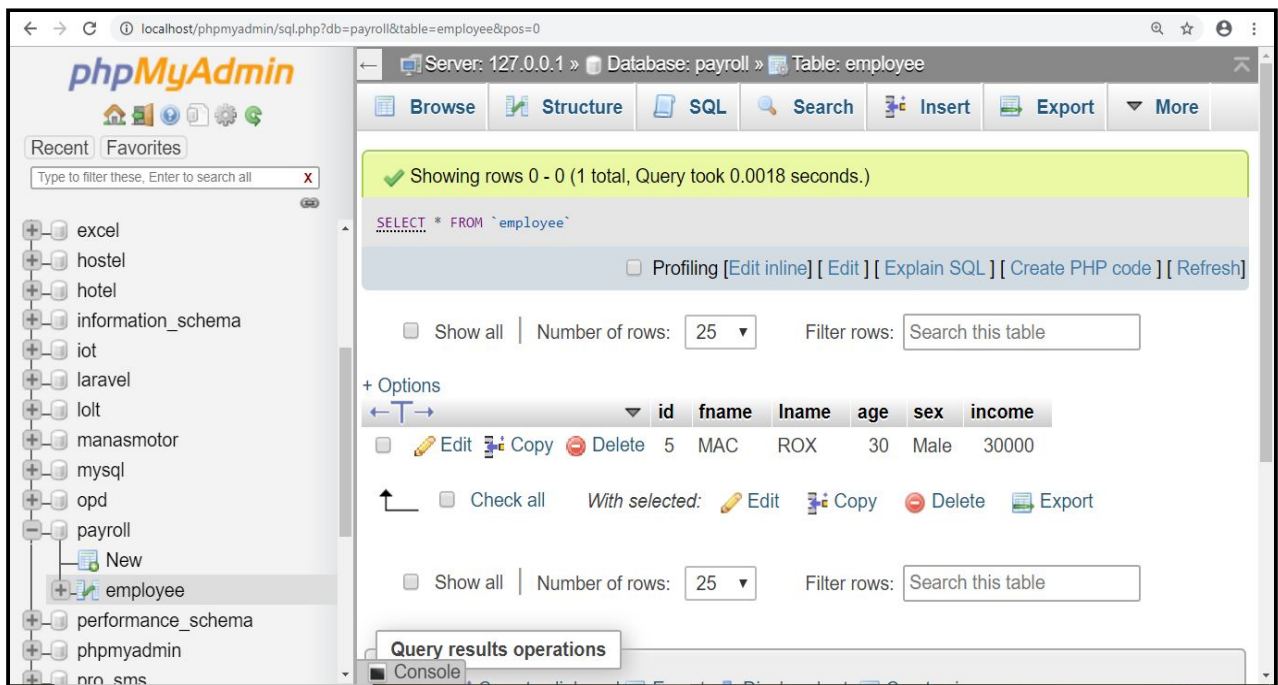
```
cursor. execute(sql)
```

6) Commit changes in the database.

```
db.commit ()
```

7) close the connection.

```
db.close ()
```



In the Database fname is changed from HARRY to MAC.

- **Deleting Record from the Database**

1) Import pymysql module in the file.

```
import pymysql
```

2) Use connect () method to connect to the database.

For Xampp

```
servername="localhost"
```

```
username="root"
```

```
password=""
```

```
dbname="payroll"
```

```
db=pymysql.connect (servername, username,password,dbname)
```

3) Creating Cursor

```
cursor = db.cursor()
```

4) Write a query to Insert the Record into Database.

```
sql="DELETE FROM employee WHERE id=5"
```

5) Execute the Query

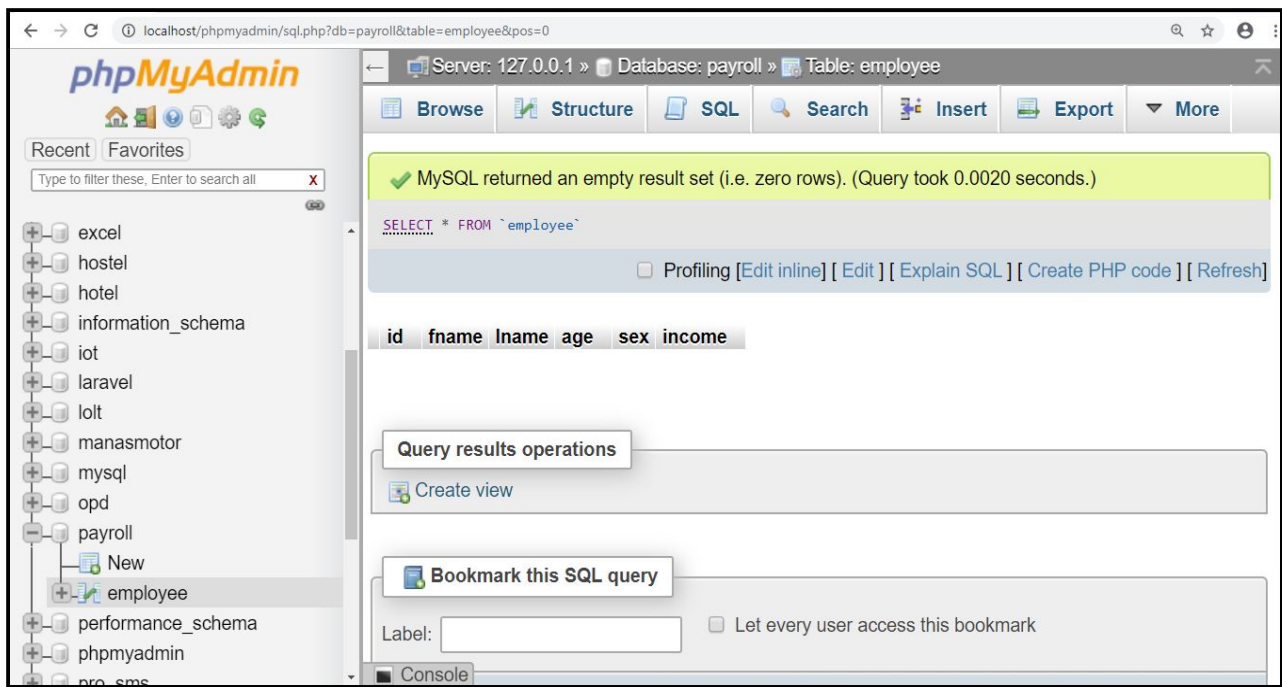
```
cursor.execute (sql)
```

6) Commit changes in the database.

```
db.commit ()
```

7) close the connection.

```
db.close ()
```



Thus you can see, after executing the Query record with id=5 is deleted.

10.4 Transaction

- 1) Performing operations like Inserting record in the database, Updating record, deleting record and selecting records from the database refer to Transaction.
- 2) This transaction is an operation on the database that makes certain changes to the database.
- 3) For any transaction to be consistent it must follow **ACID** properties.
- 4) **A**-Atomicity , **C**-Consistent , **I**-Isolated and **D**-Durability

10.5 Atomicity

- 1) By Atomicity it means Transaction on the database either get completed or does not get completed. There is no partial completion of the database Transaction.

10.6 Consistent

- 1) Consistency means the transaction does not break the rules defined by the program.
- 2) If supposed while filling form user does not enter one of the field in the form. When you press the submit button and transaction occurs, the form is rejected

as it is not submitted according to the rules defined by the program.(i.e. user must fill all the fields)

10.7 Isolation

- 1) Consider a situation where different users submit the forms while logging into an application. Thus multiple transactions occur on the database at the same time. But these transactions do not interfere with each other and are isolated from each other.

10.8 Durability

- 1) Durability is something that lasts a long.
- 2) Once a transaction is committed it gets saved in the database permanently.
- 3) Even if the system failure occurs, the transaction in the database remains as it is, thus it is durable.

10.9 Cursor

- 1) A cursor is the memory space allocated to store the result of the CRUD (Create, Read, Update and Delete) operations performed on the Database.
- 2) Cursor object/instance of that memory location is created and it is used to fetch the result of the operation performed on the database table.
- 3) By using cursor object and fetchone () or fetchall () method we can fetch the result stored in the cursor.