Introduction to Database Programming in Python

Interacting with a database is an important feature in many programming languages including python. In comparision to storing data in flat files, its much easier to store, retrive and modify data in a database. We are going to learn the following concepts and programming skills.

- Creating a Database connection
- Creating a Database
- Create a Table
- Inserting into the table
- Retrieving data from Table
- Updating Records in a table
- Deleting Data in a table

Before you can start working with MySQL database, you need to start the database server. I am using WAMP server for this tutorial. You also need to install the latest **mysql-connetor** for this purpose. use **pip install mysql-connector** in the command window to download and install it.

Connecting to the database server

In [23]:
```
import mysql.connector
con = mysql.connector.connect(host="localhost", user="root", passwd="")
mycursor = con.cursor()
con.close()
```

Creating a Database

In [3]:
```
import mysql.connector
con = mysql.connector.connect(host="localhost", user="root", passwd="")
mycursor = con.cursor()
mycursor.execute("DROP DATABASE IF EXISTS student")
mycursor.execute("CREATE DATABASE student")
mycursor.execute("USE student")
```

Creating the Table

In [11]:
```
mycursor.execute("DROP TABLE IF EXISTS studentinfo")
mycursor.execute("CREATE TABLE studentinfo (name VARCHAR(30), age INT(3), gender CHAR(1))")
```

Inserting data into the table

In [12]:
```
sql = """INSERT INTO studentinfo(name, age, gender)
        VALUES('Ashok',17,'M')"""
mycursor.execute(sql)
con.commit()
```

Inserting multiple rows simultaniously

Here we are going to use the **executemany()** function that accept two parameters as shpown below.

In [15]:
```
sql = """INSERT INTO studentinfo(name, age, gender)
VALUES(%s, %s, %s)"""
rows = [('Amit', 18,'M'),('Sudha', 17, 'F')]
mycursor.executemany(sql, rows)
con.commit()
con.close()
```

Reading from Database Table

- **fetchone()** − It fetches the next row of a query result set. A result set is an object that is returned when a cursor object is used to query a table.
- **fetchall()** − It fetches all the rows in a result set. If some rows have already been extracted from the result set, then it retrieves the remaining rows from the result set.

In [17]:
```
import mysql.connector
con = mysql.connector.connect(host="localhost", user="root", passwd="", database="student")
mycursor = con.cursor()

sql = "SELECT * FROM studentinfo"

mycursor.execute(sql)

result = mycursor.fetchall()

for row in result:
    name = row[0]
    age = row[1]
    gender = row[2]
    print("Name=%s, Age=%d, Gender=%c" % (name,age,gender))
con.close()
```

```
Name=Ashok, Age=17, Gender=M
Name=Amit, Age=18, Gender=M
Name=Sudha, Age=17, Gender=F
Name=Amit, Age=18, Gender=M
Name=Sudha, Age=17, Gender=F
```

Updating records in a Table

In [1]:
```
import mysql.connector
con = mysql.connector.connect(host="localhost", user="root", passwd="", database="student")
mycursor = con.cursor()

sql = "UPDATE studentinfo SET age=age-3 WHERE age='%d'" % (21)
mycursor.execute(sql)

sql = "SELECT * FROM studentinfo"

mycursor.execute(sql)

result = mycursor.fetchall()

for row in result:
    name = row[0]
    age = row[1]
    gender = row[2]
    print("Name=%s, Age=%d, Gender=%c" % (name,age,gender))
con.close()
```

```
Name=Ashok, Age=17, Gender=M
Name=Amit, Age=18, Gender=M
Name=Sudha, Age=17, Gender=F
Name=Amit, Age=18, Gender=M
Name=Sudha, Age=17, Gender=F
```

Deleting Records from a Table

In [ ]:

```
import mysql.connector
con = mysql.connector.connect(host="localhost", user="root", passwd="", database="student")
mycursor = con.cursor()

sql = "DELETE FROM studentinfo WHERE name='%s'" % ('Ashok')
mycursor.execute(sql)

sql = "SELECT * FROM studentinfo"

mycursor.execute(sql)

result = mycursor.fetchall()

for row in result:
    name = row[0]
    age = row[1]
    gender = row[2]
    print("Name=%s, Age=%d, Gender=%c" % (name,age,gender))
con.close()
```