Tutorial - 3

Introduction to Databases

MySQLdb package

MySQLdb is an interface for connecting to a MySQL database server from Python.

Installation

For Ubuntu, use the following command -

```
$ sudo apt-get install python-pip python-dev libmysqlclient-dev
```

For Fedora, use the following command -

```
$ sudo dnf install python python-devel mysql-devel redhat-rpm-config gcc
```

For Python command prompt, use the following command -

```
pip install MySQL-python
```

Connecting to Mysql

```
#!/usr/bin/python

import MySQLdb

# Open database connection
db = MySQLdb.connect("localhost","user","password","Database" )
```

Perform a Query

To perform a query, you first need a cursor, and then you can execute queries on it:

```
c=db.cursor()
max_year= 2013
c.execute("""SELECT Fname,Lname FROM STUDENT
        WHERE YoJ < %s""", (max_year,))
```

Read the Output

Once the query is executed, you can use following functions to get the output of the query

- **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.
- **rowcount:** This is a read-only attribute and returns the number of rows that were affected by an execute() method.

For example, results = cursor.fetchall()

INSERT Operation

```
# Prepare SQL query to INSERT a record into the database.
sql = """INSERT INTO COURSE(Cno, CName,Level, NumberOfCredits)
      VALUES (110, 'Databases', 2, 4)"""

try:
  # Execute the SQL command
  cursor.execute(sql)
  # Commit your changes in the database
  db.commit()
except:
  # Rollback in case there is any error
  db.rollback()

# disconnect from server
db.close()
```

Commit and Rollback Operation

db.commit()

It gives a green signal to database to finalize the changes, and after this operation, no change can be reverted back

db.rollback()

If you are not satisfied with one or more of the changes and you want to revert back those changes completely, then use **rollback()** method.

Update Operation

```
# Prepare SQL query to UPDATE required records
sql = "UPDATE STUDENT SET CGPA = CGPA + 1
                    WHERE GENDER='M';”
try:
   # Execute the SQL command
   cursor.execute(sql)
   # Commit your changes in the database
   db.commit()
except:
   # Rollback in case there is any error
   db.rollback()
```

DELETE Operation

```
# Prepare SQL query to DELETE required records
sql = "DELETE FROM COURSE WHERE NumberOfCredits <  '%d'" % (4)
try:
   # Execute the SQL command
   cursor.execute(sql)
   # Commit your changes in the database
   db.commit()
except:
   # Rollback in case there is any error
   db.rollback()
```

Other Ways

- For Python
  - MySQL connector: https://dev.mysql.com/doc/connector-python/en/
  - To avoid writing SQL manually and manipulate your tables as they were Python objects, you can check out SQLAlchemy : http://www.sqlalchemy.org/

  For Java

    https://alvinalexander.com/java/java-mysql-select-query-example

    https://www.javatpoint.com/java-jdbc

- Java API to connect and execute queries with a database (Not just MySQL)

For various other platforms: www.mysqltutorial.org