PYTHON CRUD WITH MYSQL

base training: nov 3 2021

1. Open command prompt (use admin console if fail on install)

```
$ pip3 install virtualenv

$ pip3 install pipenv
```

2. Setup the virtual environment for python in a sandbox project

```
$ virtualenv –p python sandbox

$ cd sandbox
```

3. Open vscode from your folder (if it does not – use vscode from start menu and open folder)

```
$ code .
```

4. In visual studio code hit Ctrl + Shift + P to open the commands menu on top
5. Type “Python: Select Interpreter” (it does lookahead search and will give options)
6. Select the “Sandbox environment” python interpreter version
7. You will see it in the status bar at the bottom left of vscode
8. Open the terminal within vscode and observe that the environment is used on terminal prompt
9. The terminal should show (sandbox) before the prompt. Exit the vscode terminal and reopen it if it is not displaying.

10. Install the mysql connector in the terminal that has the virtual environment

```
$ pip install mysql-connector-python
```

11. Create the database connection by creating a new file called database.py

```
import mysql.connector as connector
config = {
  'user': 'root',
  'password': 'GM3n3z3s',
  'host': 'localhost'
}
db = connector.connect(**config)
cursor = db.cursor()
```

12. Initialize the setup required with a new database by creating setup.py file

```
import mysql.connector as connector
from database import cursor
DB_NAME = 'sandbox'
def create_database():
  cursor.execute("CREATE DATABASE IF NOT EXISTS {} DEFAULT
CHARACTER SET 'utf8'".format(DB_NAME))
  print("Database {} created!".format(DB_NAME))
create_database()
```

13. Use the Run and Debug in vscode to run the setup.py script
14. Terminal will give a response that the sandbox is created!
15. Append setup.py to have table creation functionality
16. After DB_NAME in code

```
TABLES = {}
TABLES['users'] = (
    "CREATE TABLE `users`  ("
    " `id` int(11) NOT NULL AUTO_INCREMENT,"
    " `username` varchar(255) NOT NULL,"
    " `password` varchar(255) NOT NULL,"
    " `created` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,"
    "PRIMARY KEY (`id`)"
    ") ENGINE=InnoDB"
)
```

17. After create_database()

```
def create_tables():
  cursor.execute("USE {}".format(DB_NAME))
  for table_name in TABLES:
    table_description = TABLES[table_name]
    try:
      print("Creating table ({})".format(table_name))
      cursor.execute(table_description)
    except connector.Error as err:
      if err.errno == errorcode.ER_TABLE_EXISTS_ERROR:
        print("Table ({}) already exists".format(table_name))
      else:
        print(err.msg)
```

18. at the end

```
create_tables()
```

19. Change database.py to have the database name

```
import mysql.connector as connector
config = {
  'user': 'root',
  'password': 'password123',
  'host': 'localhost',
  'database': ‘sandbox’
}
db = connector.connect(**config)
cursor = db.cursor()
```

20. Initialize CRUD with main.py

```
from database import cursor, db


def create_user(username, password):
  sql = ("INSERT INTO users(username, password) VALUES (%s,
%s)")
  cursor.execute(sql, (username, password,))
  db.commit()
  user_id = cursor.lastrowid
  print("Added user {}".format(user_id))


def get_users():
  sql = ("SELECT * FROM users ORDER BY created DESC")
  cursor.execute(sql)
  results = cursor.fetchall()


  for row in results:
    print(row)
```

```
def get_user(id):
  sql = ("SELECT * FROM users WHERE id = %s")
  cursor.execute(sql, (id,))
  result = cursor.fetchone()

  print(result)

def update_user_password(id, new_password):
  sql = ("UPDATE users SET password = %s WHERE id = %s")
  cursor.execute(sql, (new_password, id,))
  db.commit()

  print("Updated password for user with id {}".format(id))

def delete_user(id):
  sql = ("DELETE FROM users WHERE id = %s")
  cursor.execute(sql, (id,))
  db.commit()

  print("Deleted user with id {}".format(id))

# create_user('Christopher', 'pAsSwOrD')
# create_user('John', 'doe123')
# create_user('Jane', 'hello123')
# get_users()
# get_user(1)
# update_user_password(1, 'newP@55')
```

```
delete_user(1)

get_users()
```

reference:

https://www.codiculum.com/tutorial/2020/06/crud-operations-with-python-and-mysql/#section3