

Submission Details

Name: Kelly Joseph Calvadores
 Course and Section: CPE32S3
 Date of Submission: June 07, 2024
 Instructor: Engr. Roman M. Richard

✓ Working with Python and SQLite

✓ Part 1: Python and SQL

✓ Step 1: Create a SQL connection to our SQLite database

```
import sqlite3
import os

DefaultPath = os.path.join("sqlite.db")


def db_connect(db_path = DefaultPath):
    conn = sqlite3.connect(db_path)
    return conn
```

✓ Step 2: Create a table on the SQLite database

```
con1 = db_connect()
cur1 = con1.cursor()
CustomersSQL = """CREATE TABLE customers (id INTEGER PRIMARY KEY, first_name TEXT NOT NULL, last_name TEXT NOT NULL)"""
cur1.execute(CustomersSQL)
ProductsSQL = """CREATE TABLE products (id INTEGER PRIMARY KEY, name TEXT NOT NULL, price real NOT NULL)"""
cur1.execute(ProductsSQL)
```

 [Show hidden output](#)

```
cur1.execute("SELECT name FROM sqlite_master WHERE type = 'table'")
print(cur1.fetchall())
```

 `[('customers',), ('products',), ('orders',), ('lineitems',)]`

```
cur1.execute("SELECT sql FROM sqlite_master WHERE type = 'table' AND name = 'customers'")
print(cur1.fetchone()[0])
```


 `CREATE TABLE customers (id INTEGER PRIMARY KEY, first_name TEXT NOT NULL, last_name TEXT NOT NULL)`

```
OrdersSQL = """
CREATE TABLE orders(
    id INTEGER PRIMARY KEY,
    date TEXT NOT NULL,
    customer_id INTEGER,
    FOREIGN KEY (customer_id) REFERENCES customers (id))"""
```

```
cur1.execute(OrdersSQL)
```

 `<sqlite3.Cursor at 0x7e6fa72f58c0>`

```
LineItemsSQL = """
CREATE TABLE lineitems(
    id INTEGER PRIMARY KEY,
    quantity integer NOT NULL,
    total REAL NOT NULL,
    product_id INTEGER,
    order_id INTEGER,
    FOREIGN KEY (product_id) REFERENCES products (id),
    FOREIGN KEY (order_id) REFERENCES orders (id))"""
cur1.execute(LineItemsSQL)
```

 `<sqlite3.Cursor at 0x7e6fa72f58c0>`

✓ Step 3: Loading the Data

```

con2 = db_connect()
cur2 = con2.cursor()

ProductSQL = "INSERT INTO products (name, price) VALUES (?, ?)"
cur2.execute(ProductSQL, ('Introduction to Combinatorics', 7.99))
cur2.execute(ProductSQL, ('A Guide to Writing Short Stories', 17.99))
cur2.execute(ProductSQL, ('Data Structures and Algorithms', 11.99))
cur2.execute(ProductSQL, ('Advanced Set Theory', 16.99))
con2.commit()

cur2.execute("SELECT id, name, price FROM products")
Formatted_Result = [f"{id:<5}{name:<35}{price:<5}" for id, name, price in cur2.fetchall()]
id, product, price = "ID", "Product", "Price"
print('\n'.join([f"{id:<5}{product:<35}{price:<5}" + Formatted_Result]))

```

ID	Product	Price
1	Introduction to Combinatorics	7.99
2	A Guide to Writing Short Stories	17.99
3	Data Structures and Algorithms	11.99
4	Advanced Set Theory	16.99

```

CustomerSQL = "INSERT INTO customers (first_name, last_name) VALUES (?, ?)"
cur2.execute(CustomerSQL, ('Alan', 'Turning'))
customer_id = cur2.lastrowid
print(customer_id)
con2.commit()

```

1

Task 1: Insert 3 more records on the customers table

Insert the following records:

1. Donald Knuth
2. Edgar Codd
3. Martin Forest

```

def AddCustomer(Firstname, Lastname):
    cur2.execute(CustomerSQL, (Firstname, Lastname))
    con2.commit()
    return

```

```

AddCustomer('Donald', 'Knuth')
AddCustomer('Edgar', 'Codd')
AddCustomer('Martin', 'Forest')

```

```

cur2.execute("SELECT id, first_name, last_name FROM customers")
FormattedResult = [f"{id:<5}{first_name:<15}{last_name:<5}" for id, first_name, last_name in cur2.fetchall()]
id, first_name, last_name = "ID", "First Name", "Last Name"
print('\n'.join([f"{id:<5}{first_name:<15}{last_name:<5}" + FormattedResult]))

```

ID	First Name	Last Name
1	Alan	Turning
2	Donald	Knuth
3	Edgar	Codd
4	Martin	Forest

```

OrderSQL = "INSERT INTO orders (date, customer_id) VALUES(?, ?)"
date = "1944-02-22"
cur2.execute(OrderSQL, (date, customer_id))
order_id = cur2.lastrowid
print(order_id)
con2.commit()

```

5

Task 2: Insert 3 more records on the orders table

Insert the following records:

1. for Donald Knuth, date is 7/3/1967
2. Edgar Codd, date is 1/12/1969
3. Martin Forest, date is 1/15/2021

```
def AddOrder(Date):
    OrderSQL = "INSERT INTO orders (date, customer_id) VALUES(?, ?)"
    cur2.execute(OrderSQL, (Date, customer_id))
    con2.rollback()
    return
```

```
AddOrder("1967-03-07")
AddOrder("1969-12-01")
AddOrder("2021-15-01")
```

```
cur2.execute("SELECT id, date, customer_id FROM orders")
FormattedResult = [f"{id:<5}{date:<15}{customer_id:<5}" for id, date, customer_id in cur2.fetchall()]
id, Date, CustomerId = "ID", "Date", "Customer Id"
print('\n'.join([f"{id:<5}{Date:<15}{CustomerId:<5}" + FormattedResult]))
```

ID	Date	Customer Id
1	1944-02-22	1
2	1967-03-07	1
3	1969-12-01	1
4	2021-15-01	1

```
cur2.execute("DELETE FROM orders WHERE id = 5")
con2.commit()
```

```
li_sql = """INSERT INTO lineitems
...         (order_id, product_id, quantity, total)
...         VALUES (?, ?, ?, ?)"""
product_id = 1
cur2.execute(li_sql, (order_id, 1, 1, 7.99))
con2.commit()
```

Task 3: Insert 3 more records on the lineitems

Insert the following records:

1. for Donald Knuth, insert (order_id, 2, 2, 17.99)
2. Edgar Codd, insert (order_id, 3, 3, 11.99)
3. Martin Forest, insert (order_id, 4, 4, 10.99)

```
def Addline(ProductId,Quantity, Total):
    cur2.execute(li_sql, (order_id, ProductId, Quantity, Total))
    con2.commit()
    return
```

```
Addline(2, 2, 17.99)
Addline(3, 3, 11.99)
Addline(4, 4, 10.99)
```

```
cur2.execute("SELECT id, order_id, product_id, quantity, total FROM lineitems")
FormattedResult = [f"{id:<5}{order_id:<15}{product_id:<15}{quantity:<15}{total:<5}" for id, order_id, product_id, quantity, total in cur2.fetchall()]
id, order_id, product_id, quantity, total = "ID", "Order Id", "Product ID", "Quantity", "Total"
print('\n'.join([f"{id:<5}{order_id:<15}{product_id:<15}{quantity:<15}{total:<5}" + FormattedResult]))
```

ID	Order Id	Product ID	Quantity	Total
1	4	1	1	7.99
2	5	2	2	17.99
3	5	3	3	11.99
4	5	4	4	10.99

```
cur2.execute("SELECT * FROM customers")
results = cur2.fetchall()
for row in results:
    print(row)
```

1	'Alan'	'Turning'
2	'Donald'	'Knuth'
3	'Edgar'	'Codd'
4	'Martin'	'Forest'

```
cur2.execute("SELECT id, first_name, last_name FROM customers WHERE id = 2")
result = cur2.fetchone()
print(result)
```

2	'Donald'	'Knuth'
---	----------	---------

```

con2.row_factory = sqlite3.Row
cur = con2.cursor()
cur.execute("SELECT id, first_name, last_name FROM customers WHERE id = 1")
result = cur.fetchone()
id, first_name, last_name = result['id'], result['first_name'], result['last_name']
print(f"Customer: {first_name} {last_name}'s id is {id}")

```

↗ Customer: Alan Turning's id is 1

▼ Supplementary Activity

1. Create a database and call it user.db

```

DefaultPath2 = os.path.join("user.db")

def db_connect2(db_path = DefaultPath):
    conn = sqlite3.connect(db_path)
    return conn

conS = db_connect()
curS = con1.cursor()

```

2. Create a table named "users" and insert the following: (id int, name TEXT, email TEXT)

```

UsersSQL = """CREATE TABLE users(
    ID INTEGER PRIMARY KEY,
    name TEXT NOT NULL, email TEXT NOT NULL)"""

curS.execute(UsersSQL)

```

↗ <sqlite3.Cursor at 0x7e6fa708e740>

3. Insert the following data:

```

(1, 'Jonathan', 'jvtaylor@gmail.com'),
(2, 'John', 'jonathan@gmail.com'),
(3, 'cpeEncoders', 'encoders@gmail.com')

```

```

def AddUsers(Name, Email):
    user = "INSERT INTO users (name, email) VALUES (?, ?)"
    curS.execute(user, (Name, Email))
    conS.commit()
    return

```

```

AddUsers("Jonathan", "jvtaylor@gmail.com")
AddUsers("John", "jonathan@gmail.com")
AddUsers("cpeEncoders", "encoders@gmail.com")

```

4. Select all data from users

```

curS.execute("SELECT id, name, email FROM users")
FormattedResultS = [f"{id:<5}{name:<15}{email:<5}" for id, name, email in curS.fetchall()]
id, Name, Email = "ID", "Name", "Email"
print('\n'.join([f"{id:<5}{Name:<15}{Email:<5}" + FormattedResultS]))

```

↗

ID	Name	Email
1	Jonathan	jvtaylor@gmail.com
2	John	jonathan@gmail.com
3	cpeEncoders	encoders@gmail.com

5. Select id = 3 from users.

```

curS.execute("SELECT * FROM users WHERE id = 3")
FormattedResultS = [f"{id:<5}{name:<15}{email:<5}" for id, name, email in curS.fetchall()]
id, Name, Email = "ID", "Name", "Email"
print('\n'.join([f"{id:<5}{Name:<15}{Email:<5}" + FormattedResultS]))

```

↗

ID	Name	Email
3	cpeEncoders	encoders@gmail.com

6. Update user id = 3 name and set it to "James."

```
curS.execute("UPDATE users SET name = 'James' WHERE id = 3")
conS.commit()
```

```
curS.execute("SELECT * FROM users WHERE id = 3")
FormattedResultS = [f"{id:<5}{name:<15}{email:<5}" for id, name, email in curS.fetchall()]
id, Name, Email = "ID", "Name", "Email"
print('\n'.join([f"{id:<5}{Name:<15}{Email:<5}" ] + FormattedResultS))
```

ID	Name	Email
3	James	encoders@gmail.com

7. Insert the following data: (4, 'Cynthia', cynthia@gmail.com)

```
AddUsers("Cynthia", "cynthia@gmail.com")
```

```
curS.execute("SELECT id, name, email FROM users")
FormattedResultS = [f"{id:<5}{name:<15}{email:<5}" for id, name, email in curS.fetchall()]
id, Name, Email = "ID", "Name", "Email"
print('\n'.join([f"{id:<5}{Name:<15}{Email:<5}" ] + FormattedResultS))
```

ID	Name	Email
1	Jonathan	jvtaylor@gmail.com
2	John	jonathan@gmail.com
3	James	encoders@gmail.com
4	Cynthia	cynthia@gmail.com

8. Delete id = 4 from users.

```
curS.execute("DELETE FROM users WHERE id = 4")
conS.commit()
```

9. Display all contents in a formatted way

```
curS.execute("SELECT id, name, email FROM users")
FormattedResultS = [f"{id:<5}{name:<15}{email:<5}" for id, name, email in curS.fetchall()]
id, Name, Email = "ID", "Name", "Email"
print('\n'.join([f"{id:<5}{Name:<15}{Email:<5}" ] + FormattedResultS))
```

ID	Name	Email
1	Jonathan	jvtaylor@gmail.com
2	John	jonathan@gmail.com
3	James	encoders@gmail.com

✓ Conclusions/Observations:

- In this activity, I able to perform SQL using python. This time I able to finish this activity and able to understand most of the procedures. I have learned how to manipulate sql using python.