



Experiment No. 13
Program to demonstrate CRUD (create, read, update and delete) operations on database (SQLite/ MySQL) using python
Name : Vaidehi D. Gadag
Branch/Div.: Comps-1 (C47)
Date of Performance: 10/04/2024
Date of Submission: 17/04/2024



Experiment No. 13

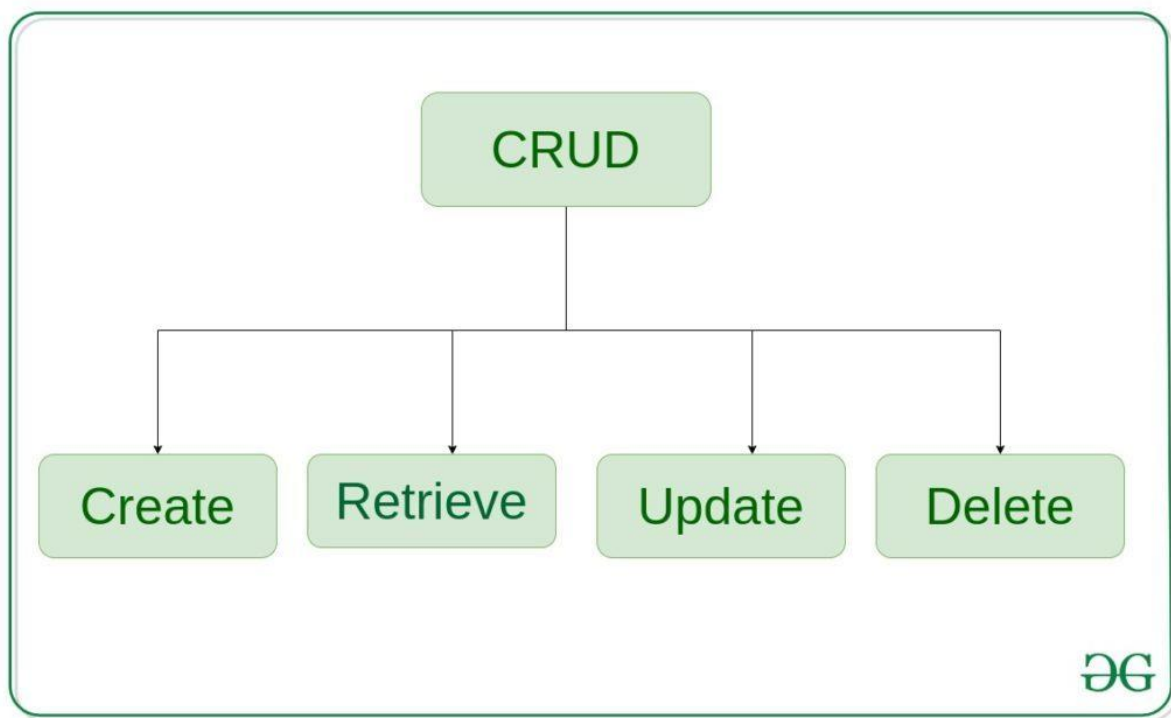
Title: Program to demonstrate CRUD (create, read, update and delete) operations on database (SQLite/ MySQL) using python

Aim: To study and implement CRUD (create, read, update and delete) operations on database (SQLite/ MySQL) using python

Objective: To introduce database connectivity with python

Theory:

In general CRUD means performing Create, Retrieve, Update and Delete operations on a table in a database. Let's discuss what actually CRUD means,



Create – create or add new entries in a table in the database.

Retrieve – read, retrieve, search, or view existing entries as a list(List View) or retrieve a particular entry in detail (Detail View)

Update – update or edit existing entries in a table in the database

Delete – delete, deactivate, or remove existing entries in a table in the database



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Program:

```
import mysql.connector
```

```
# Function to create a new record
```

```
def create_record(conn, values):
```

```
    cursor = conn.cursor()
```

```
    cursor.execute("INSERT INTO records (name, age) VALUES (%s, %s)", values)
```

```
    conn.commit()
```

```
    print("Record created successfully")
```

```
# Function to read all records
```

```
def read_records(conn):
```

```
    cursor = conn.cursor()
```

```
    cursor.execute("SELECT * FROM records")
```

```
    rows = cursor.fetchall()
```

```
    print("ID\tName\tAge")
```

```
    for row in rows:
```

```
        print("{}\t{}\t{}".format(row[0], row[1], row[2]))
```

```
# Function to update a record
```

```
def update_record(conn, record_id, values):
```

```
    cursor = conn.cursor()
```

```
    cursor.execute("UPDATE records SET name=%s, age=%s WHERE id=%s", (*values, record_id))
```



```
conn.commit()
```

```
print("Record updated successfully")
```

```
# Function to delete a record
```

```
def delete_record(conn, record_id):
```

```
    cursor = conn.cursor()
```

```
    cursor.execute("DELETE FROM records WHERE id=%s", (record_id,))
```

```
    conn.commit()
```

```
    print("Record deleted successfully")
```

```
# Main function
```

```
def main():
```

```
    conn = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        password="om@21",
```

```
        database="exp_13"
```

```
    )
```

```
    cursor = conn.cursor()
```

```
# Create table if not exists
```

```
cursor.execute("CREATE TABLE IF NOT EXISTS records
```

```
(id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), age INT)")
```



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
while True:
```

```
print("\n1. Create Record\n2. Read Records\n3. Update Record\n4. Delete Record\n5. Exit")
```

```
choice = input("Enter your choice: ")
```

```
if choice == '1':
```

```
name = input("Enter name: ")
```

```
age = int(input("Enter age: "))
```

```
create_record(conn, (name, age))
```

```
elif choice == '2':
```

```
read_records(conn)
```

```
elif choice == '3':
```

```
record_id = int(input("Enter record ID to update: "))
```

```
name = input("Enter new name: ")
```

```
age = int(input("Enter new age: "))
```

```
update_record(conn, record_id, (name, age))
```

```
elif choice == '4':
```

```
record_id = int(input("Enter record ID to delete: "))
```

```
delete_record(conn, record_id)
```

```
elif choice == '5':
```



```
break
```

```
else:
```

```
print("Invalid choice")
```

```
conn.close()
```

```
if __name__ == "__main__":
```

```
main()
```

Output:

1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit

Enter your choice: 1

Enter name: try_1

Enter age: 19

Record created successfully

1. Create Record
2. Read Records
3. Update Record
4. Delete Record



5. Exit

Enter your choice: 1

Enter name: try_2

Enter age: 20

Record created successfully

1. Create Record

2. Read Records

3. Update Record

4. Delete Record

5. Exit

Enter your choice: 2

ID	Name	Age
----	------	-----

1	try_1	19
---	-------	----

2	try_2	20
---	-------	----

1. Create Record

2. Read Records

3. Update Record

4. Delete Record

5. Exit

Enter your choice: 3

Enter record ID to update: 1

Enter new name: update_1

Enter new age: 25

Record updated successfully

1. Create Record

2. Read Records



3. Update Record

4. Delete Record

5. Exit

Enter your choice: 2

ID	Name	Age
----	------	-----

1	update_1	25
---	----------	----

2	try_2	20
---	-------	----

1. Create Record

2. Read Records

3. Update Record

4. Delete Record

5. Exit

Enter your choice: 4

Enter record ID to delete: 1

Record deleted successfully

1. Create Record

2. Read Records

3. Update Record

4. Delete Record

5. Exit

Enter your choice: 2

ID	Name	Age
----	------	-----

2	try_2	20
---	-------	----



```
mysql> create database exp_13;
Query OK, 1 row affected (0.01 sec)

mysql> use exp_13;
Database changed
mysql> CREATE TABLE records (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     name VARCHAR(255),
->     age INT
-> );
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_exp_13 |
+-----+
| records          |
+-----+
1 row in set (0.00 sec)

mysql> select*from records;
Empty set (0.00 sec)
```

```
1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 1
Enter name: try_1
Enter age: 19
Record created successfully
```

```
1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 1
Enter name: try_2
Enter age: 20
Record created successfully
```

```
1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 2
ID      Name      Age
1       try_1     19
2       try_2     20
```

```
1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice:
```

```
mysql> select*from records;
+----+-----+-----+
| id | name  | age  |
+----+-----+-----+
| 1  | try_1 | 19   |
| 2  | try_2 | 20   |
+----+-----+-----+
2 rows in set (0.00 sec)
```



1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit

Enter your choice: 3

Enter record ID to update: 1

Enter new name: update_1

Enter new age: 25

Record updated successfully

1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit

Enter your choice: 2

ID	Name	Age
1	update_1	25
2	try_2	20

```
mysql> select*from records;
```

id	name	age
1	update_1	25
2	try_2	20

```
2 rows in set (0.00 sec)
```

1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit

Enter your choice: 4

Enter record ID to delete: 1

Record deleted successfully

1. Create Record
2. Read Records
3. Update Record
4. Delete Record
5. Exit

Enter your choice: 2

ID	Name	Age
2	try_2	20

```
mysql> select*from records;
```

id	name	age
2	try_2	20

```
1 row in set (0.00 sec)
```



Conclusion:

In conclusion, the experiment successfully demonstrated CRUD operations (Create, Read, Update, Delete) on a database using Python, specifically with MySQL. The program showcased the seamless connectivity with the database, allowing for the creation, retrieval, updating, and deletion of records. Through simple user prompts, users could perform various operations such as adding new records, viewing existing records, updating records, and deleting records. This experiment underscores the importance of database connectivity in Python applications, enabling efficient data management and manipulation. The program's output validated the functionality of each operation, ensuring the integrity and effectiveness of the CRUD operations. Overall, this experiment provides a practical insight into database operations in Python, facilitating effective data handling and management tasks.