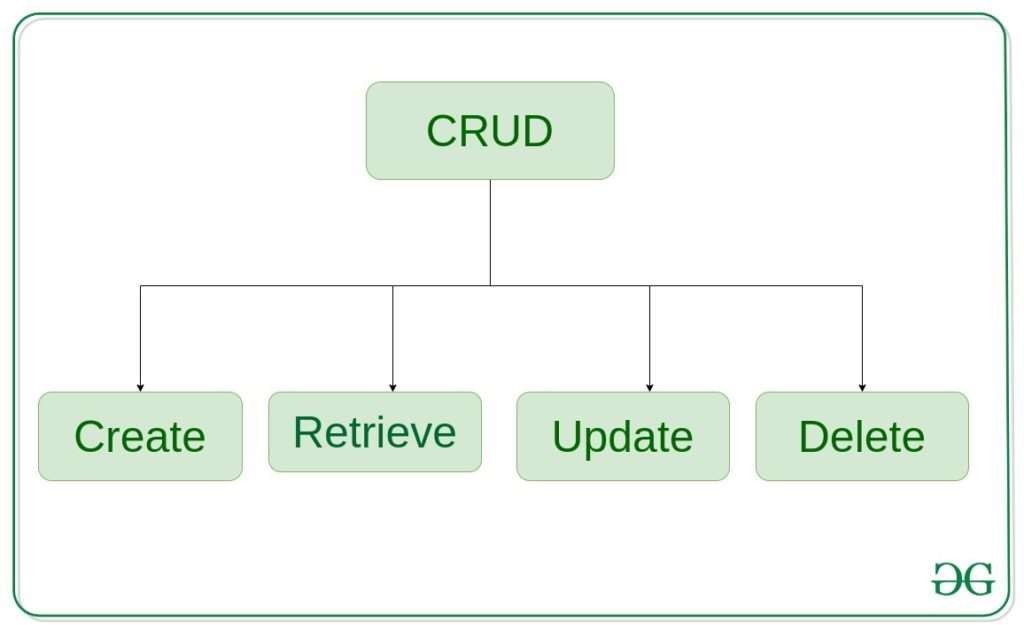
|  |
| --- |
| Experiment No. 13 |
| Program to demonstrate CRUD operations on database (SQLite/MySQL using Python) |
| Date of Performace: 8/04/2024 |
| Date of Submission: 15/04/2024 |



**Experiment No. 13**







**CODE:**

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)''')

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

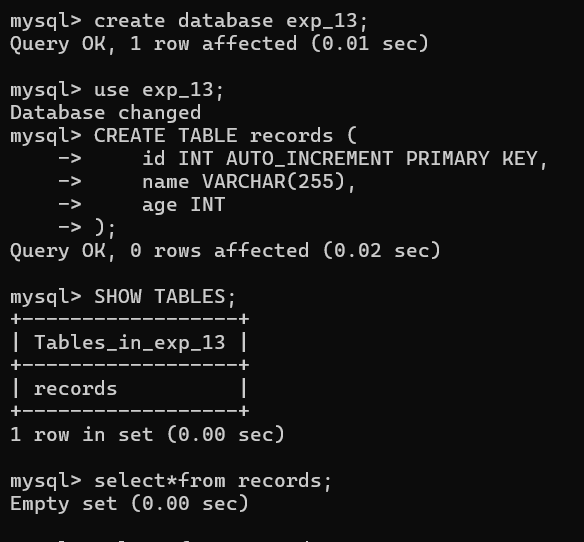
4. Delete Record

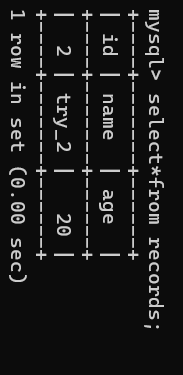
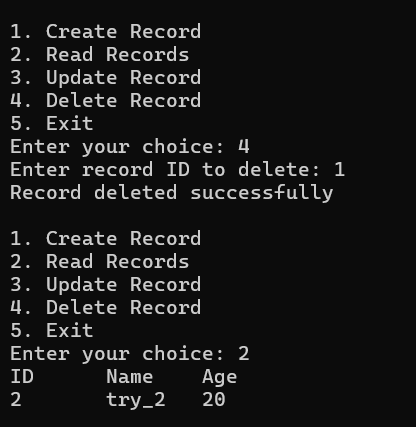
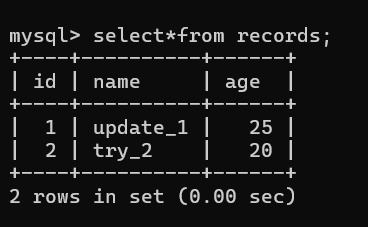
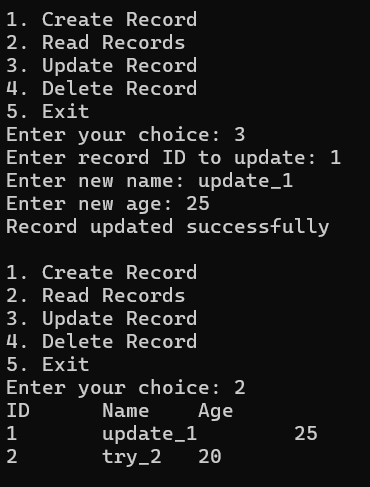
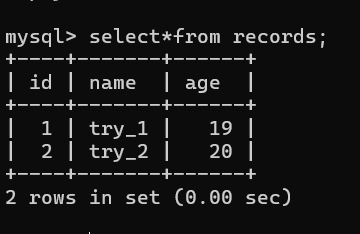
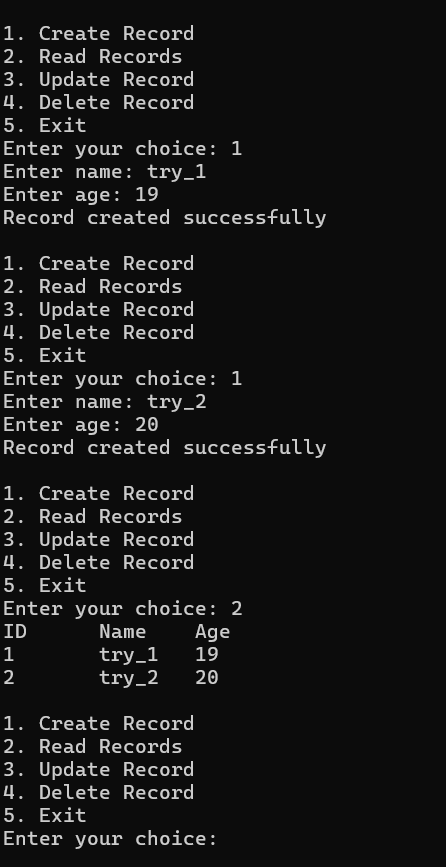
5. Exit

Enter your choice: 2

ID Name Age

2 try\_2 20





**Conclusion:**

In the realm of SQL, CRUD operations serve as the cornerstone for database interaction, facilitating seamless management of relational databases. The Create operation (INSERT) empowers users to append new records, while the Read operation (SELECT) enables efficient data retrieval. With the Update operation (UPDATE), existing records can be modified, while the Delete operation (DELETE) facilitates the removal of unwanted data. Together, these operations empower users to wield full control over their data, ensuring effective management and manipulation of database records.