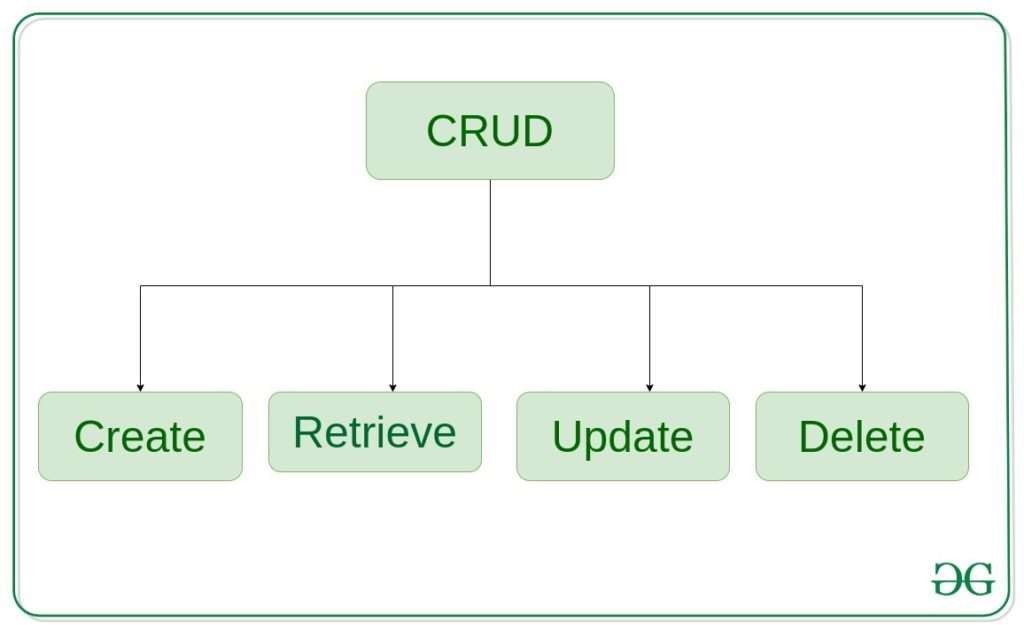
|  |
| --- |
| Experiment No. 13 |
| Program to demonstrate CRUD operations on database (SQLite/MySQL using Python) |
| Date of Performace: 10/04/2024 |
| Date of Submission: 17/04/2024 |



**Experiment No. 13**







Code:  
import mysql.connector

# Establish connection to MySQL server

mydb = mysql.connector.connect(

host="localhost",

user="root",

password="ZXCV0987@1234",

database="python"

)

# Create a cursor object to execute SQL queries

mycursor = mydb.cursor()

# Create table (if it doesn't exist)

mycursor.execute("CREATE TABLE IF NOT EXISTS employees (id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(255), salary DOUBLE)")

# Create operation (insert a new record)

def create\_employee(name, salary):

sql = "INSERT INTO employees (name, salary) VALUES (%s, %s)"

val = (name, salary)

mycursor.execute(sql, val)

mydb.commit()

print("Record inserted successfully")

# Read operation (fetch all records)

def read\_employees():

mycursor.execute("SELECT \* FROM employees")

result = mycursor.fetchall()

for row in result:

print(row)

# Update operation (update a record)

def update\_employee(id, salary):

sql = "UPDATE employees SET salary = %s WHERE id = %s"

val = (salary, id)

mycursor.execute(sql, val)

mydb.commit()

print("Record updated successfully")

# Delete operation (delete a record)

def delete\_employee(id):

sql = "DELETE FROM employees WHERE id = %s"

val = (id,)

mycursor.execute(sql, val)

mydb.commit()

print("Record deleted successfully")

# Example usage

create\_employee("John", 50000)

read\_employees()

update\_employee(1, 55000)

delete\_employee(1)

Output:

