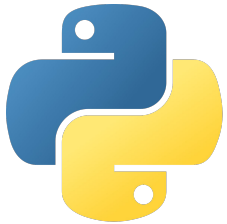

Database in Python Using SQLITE

What is Database?

- Database is a organized collection of data stored and accessed electronically.
- It provides a structured way to store, manage, and retrieve data efficiently.
- **Types**
  - Relational Databases (RDBMS)
    - They organize data into tables with rows and columns, and relationships between tables are established through keys.
    - Example: SQLite, MySQL, Oracle Database, PostgreSQL, etc
  - NoSQL Databases
    - Example: MongoDB, Cassandra, Couchbase, Redis.
  - Object-Oriented Databases (OODBMS)
  - Hierarchical Databases and many more...

Introduction to SQLite

- SQLite is a lightweight, serverless, and self-contained database engine that is included by default in Python's standard library.
- It provides a simple and efficient way to store and retrieve data without requiring a separate process.
- SQLite are suitable for small to medium-sized applications or when you don't need a full-fledged database server.
- To use SQLite in Python, you need to import the **sqlite3** module, which provides the necessary functions and classes for interacting with SQLite databases.

Introduction to SQLite (Topics)

1. Connect To Database
2. Create a Table
3. Insert Operation
4. Select Operation
5. Update Operation
6. Delete Operation

1. Connect To Database

```
import sqlite3
db_name = "users.db"
conn = sqlite3.connect(db_name)
print("Opened database successfully")
```

- First we need to create a new database and open a database connection to allow sqlite3 to work with it.
- **sqlite3.connect** establishes a connection to an SQLite database named **users.db**.
- If the file doesn't exist, it will be created.

What is SQL?

- stands for Structured Query Language.
- SQL is a programming language used for managing relational databases.
- Relational databases organize data into tables with rows and columns, and relationships between tables are established through keys.
- We use SQL language for **CRUD** operations.
  - C = Create
  - R = Read
  - U = Update
  - D = Delete

SQL Data Types

Data Type	Description	Example
INT	Integer value representing a whole number	10, -5, 0
VARCHAR(50)	Variable-length character string with a maximum length of 50 characters	'Python', 'is awesome'
TEXT	does not require specifying a maximum length limit. suitable to store large amounts of text.	'Python is awesome language'
FLOAT	Floating-point number with decimal precision	3.14, -0.5
DATE	Date value in the format 'YYYY-MM-DD'	'2023-06-24', '1990-01-01'
BOOLEAN	Boolean value representing true or false	true, false

2. Create a Table

CREATE TABLE <table_name>(<column_names>)

```
conn.execute(""" CREATE TABLE employee
          (
             ID              INT              PRIMARY KEY     NOT NULL,
             NAME            VARCHAR(1000)                    NOT NULL,
             AGE             INT                              NOT NULL,
             ADDRESS         TEXT,
             SALARY          FLOAT
          )
              """)
```

3. Insert Operations

*conn.execute("**INSERT INTO table_name (column1, column2) VALUES (value1, value2)**")*

```
conn.execute("""INSERT INTO
          employee (ID,NAME,AGE,ADDRESS,SALARY) VALUES (1,'ram', 32, 'kathmandu', 20000.00 )
        """)

conn.execute("""INSERT INTO
         employee (ID,NAME,AGE,ADDRESS) VALUES (2, 'shyam', 20, 'lalitpur')
       """)

conn.execute("""INSERT INTO
        employee (ID,NAME,AGE, SALARY) VALUES (3, 'sita', 15, 3000)
      """)
```

```
conn.commit()
```

```
conn.execute("""INSERT INTO
         employee (ID,NAME,AGE) VALUES (4, 'gita', 40)
       """)
```

4. Select (Read) Operations

*cursor.execute("**SELECT * FROM table_name**")*

```
cursor = conn.execute("SELECT id, name, address, salary from employee")
for row in cursor:
   print("ID = ", row[0])
   print("NAME = ", row[1])
   print("ADDRESS = ", row[2])
   print("SALARY = ", row[3], "\n")
```

```
cursor = conn.execute('select * from employee).fetchall()
print(cursor)
```

5. Update Operations

*cursor.execute(**"UPDATE table_name SET column1 = value1 WHERE condition"**)*

```
conn.execute("UPDATE employee set SALARY = 25000.00 where ID = 4")
```

```
conn.execute("UPDATE employee set address = budhanilkantha where ID = 3")
```

```
conn.commit()
```

5. Delete Operations

```
conn.execute("DELETE From emplyee where name = 'shyam'")

conn.commit()

# close connection to database
conn.close()
```