COMM7360 Big Data Management & Analysis
Laboratory 2 - SQLite & Python

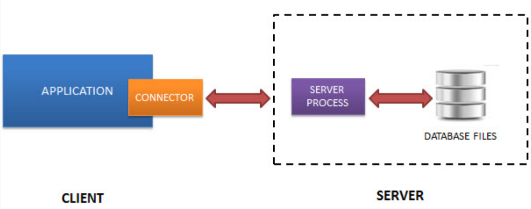
Xiaoyi FU

What is SQLite

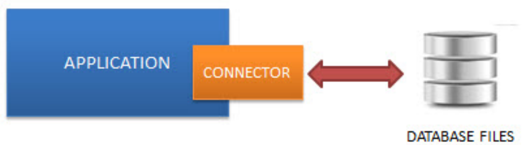
- SQLite is a software library that provides a relational database management system.
- The lite in SQLite means light weight in terms of setup, database administration, and required resource.

What is SQLite

- **Relational Database Management System (RDBMS)**

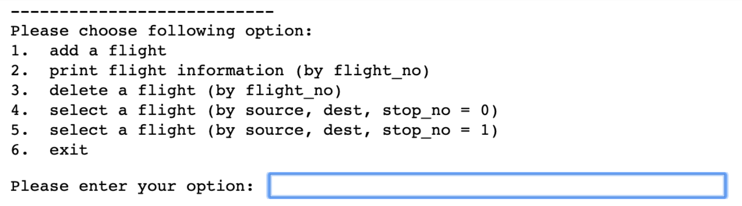


- **SQLite**
  - SQLite database is integrated with the application that accesses the database.
  - The applications interact with the SQLite database read and write directly from the database files stored on disk.



What you can do

- **SQLite Python**
    - How to connect
    - How to update (insert/delete)
    - How to query

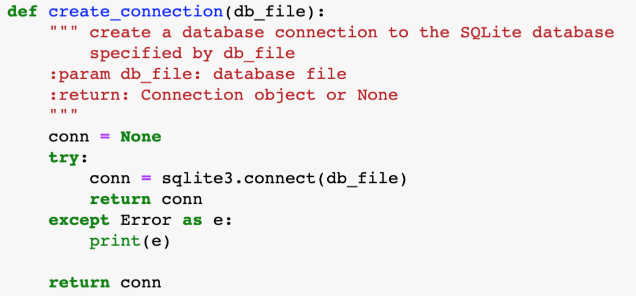


```
---------------------------
Please choose following option:
1.  add a flight
2.  print flight information (by flight_no)
3.  delete a flight (by flight_no)
4.  select a flight (by source, dest, stop_no = 0)
5.  select a flight (by source, dest, stop_no = 1)
6.  exit

Please enter your option:
```

SQLite Connection

- Create a Connection object that represents the database using the connect() function of the sqlite3 module

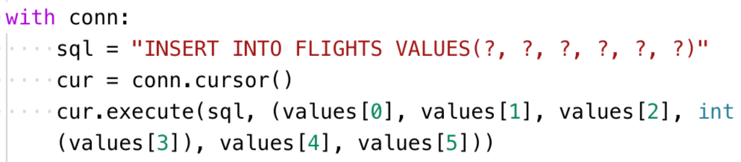


```python
def create_connection(db_file):
    """ create a database connection to the SQLite database
        specified by db_file
    :param db_file: database file
    :return: Connection object or None
    """
    conn = None
    try:
        conn = sqlite3.connect(db_file)
        return conn
    except Error as e:
        print(e)

    return conn
```

SQLite Update

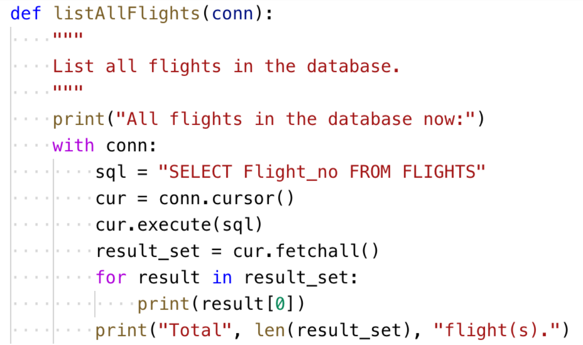
1. Connect to the SQLite database by creating a Connection object.
2. Create a Cursor object by calling the cursor method of the Connection object.
3. Execute an INSERT statement. If you want to pass arguments to the INSERT statement, use the question mark (?) as the placeholder for each argument.



```
with conn:
    sql = "INSERT INTO FLIGHTS VALUES(?, ?, ?, ?, ?, ?)"
    cur = conn.cursor()
    cur.execute(sql, (values[0], values[1], values[2], int
    (values[3]), values[4], values[5]))
```

SQLite Query

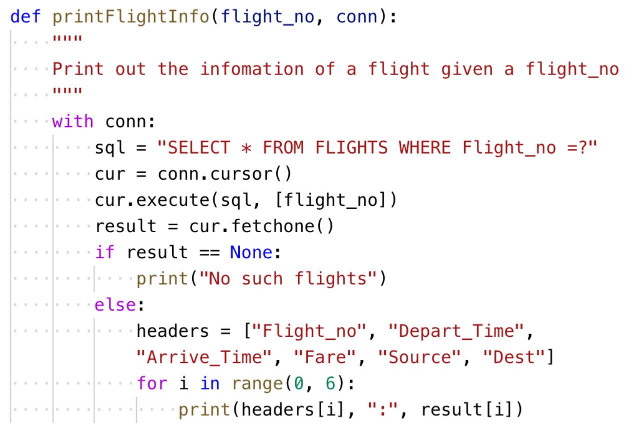
1. Create a Cursor object using the cursor method of the Connection object.
2. Execute a SELECT statement.
3. Call the **fetchall()** method of the cursor object to fetch the data.
4. Finally, loop the cursor and process each row individually.



```python
def listAllFlights(conn):
    """
    List all flights in the database.
    """
    print("All flights in the database now:")
    with conn:
        sql = "SELECT Flight_no FROM FLIGHTS"
        cur = conn.cursor()
        cur.execute(sql)
        result_set = cur.fetchall()
        for result in result_set:
            print(result[0])
        print("Total", len(result_set), "flight(s).")
```

Print Flight Info

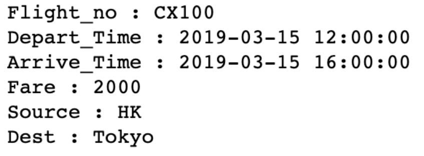
- A useful function is provided to print flight info



```python
def printFlightInfo(flight_no, conn):
    """
    Print out the infomation of a flight given a flight_no
    """
    with conn:
        sql = "SELECT * FROM FLIGHTS WHERE Flight_no =?"
        cur = conn.cursor()
        cur.execute(sql, [flight_no])
        result = cur.fetchone()
        if result == None:
            print("No such flights")
        else:
            headers = ["Flight_no", "Depart_Time",
            "Arrive_Time", "Fare", "Source", "Dest"]
            for i in range(0, 6):
                print(headers[i], ":", result[i])
```

Print Flight Info

- For example, to print the info for CX100:
  - fli_no = ”CX100”
    printFlightInfo(fli_no, conn)



```
Flight_no : CX100
Depart_Time : 2019-03-15 12:00:00
Arrive_Time : 2019-03-15 16:00:00
Fare : 2000
Source : HK
Dest : Tokyo
```