Documentation: SQLite Student Database Management

Overview

This script creates and manages an **SQLite database (student.db)** that stores student records, including their **name, section, year of joining (YOJ), and company**. The script performs the following tasks:

1. **Creates the database and table** (if they don't already exist).
2. **Inserts multiple student records** into the table.
3. **Retrieves and displays** all inserted records.

Features

- **Automatic Database Creation**: If student.db doesn't exist, it will be created.
- **Table Management**: Ensures that the Student table is created only if it doesn't already exist.
- **Bulk Data Insertion**: Uses executemany() for efficient insertion of multiple records.
- **Data Retrieval**: Fetches and prints all records after insertion.
- **Ensures Data Integrity**: Commits changes before closing the database connection.

How It Works

1. Database Connection

- Establishes a connection to **SQLite** using sqlite3.connect("student.db").
- If the file **does not exist**, SQLite automatically creates it.

2. Creating the Student Table

- The script defines a **CREATE TABLE** SQL statement with an **IF NOT EXISTS** condition to ensure the table is only created if it doesn’t already exist.
- The table schema includes:
  - **NAME (TEXT)** – Student’s name.
  - **SECTION (TEXT)** – The student’s academic section (e.g., Data Science, Full Stack, AI & ML).
  - **YOJ (INT)** – Year of joining.
  - **COMPANY (TEXT)** – The company where the student is placed or working.

3. Inserting Multiple Records

- A list of tuples contains **student details**, and executemany() is used to insert all records efficiently.
- This avoids multiple individual insert queries, making the process faster.

4. Retrieving and Displaying Data

- Uses SELECT * FROM Student to fetch all records.
- The records are printed row by row to confirm successful insertion.

5. Committing and Closing the Connection

- The commit() function ensures that all changes are saved in the database.
- The connection is **closed properly** to prevent memory leaks and database locking issues.

How to Run the Script

1. Ensure **Python** is installed on your system.
2. Save the script as student_db.py.
   ```
   python student_db.py
   ```
3. After execution, check the student.db file in the same directory.

To view records manually, you can open student.db using **DB Browser for SQLite** or run:

```
sqlite3 student.db
SELECT * FROM Student;
```

Future Improvements

- Add **user input functionality** to insert new records dynamically.
- Implement **search and filter options** for specific students.
- Develop a **Streamlit UI** to make it interactive.
- Connect the database to an **AI-powered chatbot** for query-based interaction.

Code for this

```
import sqlite3

# Connecting to SQLite database (creates "student.db" if it doesn't exist)

connection = sqlite3.connect("student.db")

# Create a cursor object to execute SQL commands

cursor = connection.cursor()

# Table creation (includes IF NOT EXISTS to prevent errors if it already exists)

table_info = """

CREATE TABLE IF NOT EXISTS Student (

  NAME TEXT,
  SECTION TEXT,
  YOJ INT,
  COMPANY TEXT
);
"""
cursor.execute(table_info)
# Inserting multiple records using executemany()
students_data = [
  ("Surya Vishal", "Data Science", 2024, "Tekworks"),
  ("Nitish", "Data Science", 2024, "Symphonize"),
  ("Ram Charan", "Data Science", 2025, "Symphonize"),
  ("Pradeep", "Full Stack", 2023, "Bitlabs"),
  ("Vineetha", "Full Stack", 2023, "Bitlabs"),
  ("Anjali Sharma", "Data Science", 2025, "TCS"),
  ("Rohit Mehta", "Cyber Security", 2024, "Infosys"),
  ("Sneha Reddy", "Full Stack", 2024, "Capgemini"),
  ("Abhishek Yadav", "AI & ML", 2023, "Google"),
  ("Sandeep Kumar", "Cyber Security", 2025, "Microsoft"),
  ("Neha Verma", "AI & ML", 2024, "Amazon"),

  ("Arjun Kapoor", "Cloud Computing", 2023, "IBM"),

  ("Priyanka Mishra", "Cloud Computing", 2024, "Accenture"),

  ("Ravi Teja", "Full Stack", 2025, "Deloitte"),

  ("Swetha Nair", "Cyber Security", 2023, "Cisco"),

  ("Manoj Singh", "Data Science", 2025, "HCL"),

  ("Aditi Rao", "Cloud Computing", 2024, "Oracle"),

  ("Varun Dhawan", "AI & ML", 2025, "Facebook"),

  ("Megha Agarwal", "Cyber Security", 2023, "Wipro"),

  ("Sahil Khan", "Full Stack", 2024, "Zoho"),

  ("Rajat Sharma", "Data Science", 2025, "TCS"),

  ("Alok Verma", "Cyber Security", 2023, "Microsoft"),

  ("Kriti Sen", "Full Stack", 2024, "Capgemini"),

  ("Anil Kumar", "AI & ML", 2025, "Google"),

  ("Simran Kaur", "Cloud Computing", 2023, "IBM"),

  ("Yash Gupta", "Data Science", 2024, "Tekworks"),

  ("Pooja Sharma", "Full Stack", 2023, "Bitlabs"),

  ("Aman Yadav", "AI & ML", 2025, "Amazon"),
  ("Vikram Singh", "Cloud Computing", 2024, "Accenture"),
  ("Neetu Kapoor", "Cyber Security", 2023, "Cisco"),
  ("Rishabh Pant", "Full Stack", 2025, "Deloitte"),
  ("Kavita Mehta", "AI & ML", 2024, "Facebook"),
  ("Sourabh Jain", "Data Science", 2023, "Wipro"),
  ("Rajesh Khanna", "Full Stack", 2024, "Zoho"),
  ("Naina Sharma", "Cyber Security", 2025, "Microsoft"),
  ("Tushar Raj", "AI & ML", 2023, "Google"),
  ("Avneet Kaur", "Cloud Computing", 2024, "Oracle"),
  ("Ishaan Awasthi", "Data Science", 2025, "HCL"),
  ("Meera Chopra", "Full Stack", 2024, "Bitlabs"),
  ("Krishna Das", "AI & ML", 2023, "Amazon"),
  ("Puneet Singh", "Cloud Computing", 2025, "Accenture"),
  ("Trisha Krishnan", "Cyber Security", 2024, "Cisco"),
  ("Dinesh Reddy", "Data Science", 2023, "TCS"),
  ("Sonia Sharma", "Full Stack", 2024, "Capgemini"),
  ("Vishal Anand", "AI & ML", 2025, "Google"),
  ("Zoya Khan", "Cyber Security", 2023, "Microsoft"),
  ("Karthik Raju", "Cloud Computing", 2024, "Oracle"),
  ("Ravi Kumar", "Data Science", 2025, "Symphonize"),
  ("Anusha Reddy", "Full Stack", 2024, "Zoho"),
  ("Harshita Mehta", "Cyber Security", 2023, "Wipro"),
  ("Akash Sharma", "Data Science", 2024, "Tekworks"),
  ("Shreya Kapoor", "AI & ML", 2025, "Facebook"),
  ("Gaurav Singh", "Cloud Computing", 2023, "IBM"),
  ("Isha Nair", "Full Stack", 2024, "Bitlabs"),
  ("Vikash Reddy", "Cyber Security", 2025, "Microsoft"),
  ("Nikhil Verma", "AI & ML", 2023, "Amazon"),
  ("Aisha Khurana", "Cloud Computing", 2024, "Accenture"),
  ("Karan Johar", "Data Science", 2025, "TCS"),
  ("Deepika Sen", "Full Stack", 2023, "Capgemini"),
  ("Pratham Mehta", "Cyber Security", 2024, "Infosys"),
  ("Ramesh Gupta", "AI & ML", 2025, "Google"),
  ("Saniya Roy", "Cloud Computing", 2023, "IBM"),
  ("Tarun Das", "Full Stack", 2024, "Zoho"),
  ("Madhavi Rao", "Data Science", 2025, "HCL"),
      ("Aarav Sharma", "Cyber Security", 2023, "Wipro"),
      ("Sonali Desai", "AI & ML", 2024, "Amazon"),
      ("Vivek Anand", "Cloud Computing", 2025, "Accenture"),
      ("Ruchi Patel", "Data Science", 2024, "Symphonize")
  ]
  # Insert records in bulk
  cursor.executemany("INSERT INTO Student (NAME, SECTION, YOJ, COMPANY) VALUES (?, ?, ?, ?)", students_data)
  # Display all the records
  print("The Inserted records are:")
  data = cursor.execute("SELECT * FROM Student")
  for row in data:
      print(row)
  # Commit changes and close connection
  connection.commit()
  connection.close()
```