

ASSIGNMENT – 16.3

NAME : BHAWWANA SHRE

HALL TICKET NO : 2403A52311

BATCH NUMBER : 01

COURSE CODE : 24CS002PC215

PROGRAM NAME : B.TECH

YEAR/SEM : 2ND AND 3RD

TASK 1 :

Ask AI to design a schema for a Library Management System (Tables: Books, Members, Loans).

SQL Code

```
CREATE TABLE Members (
    member_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    join_date DATE
);

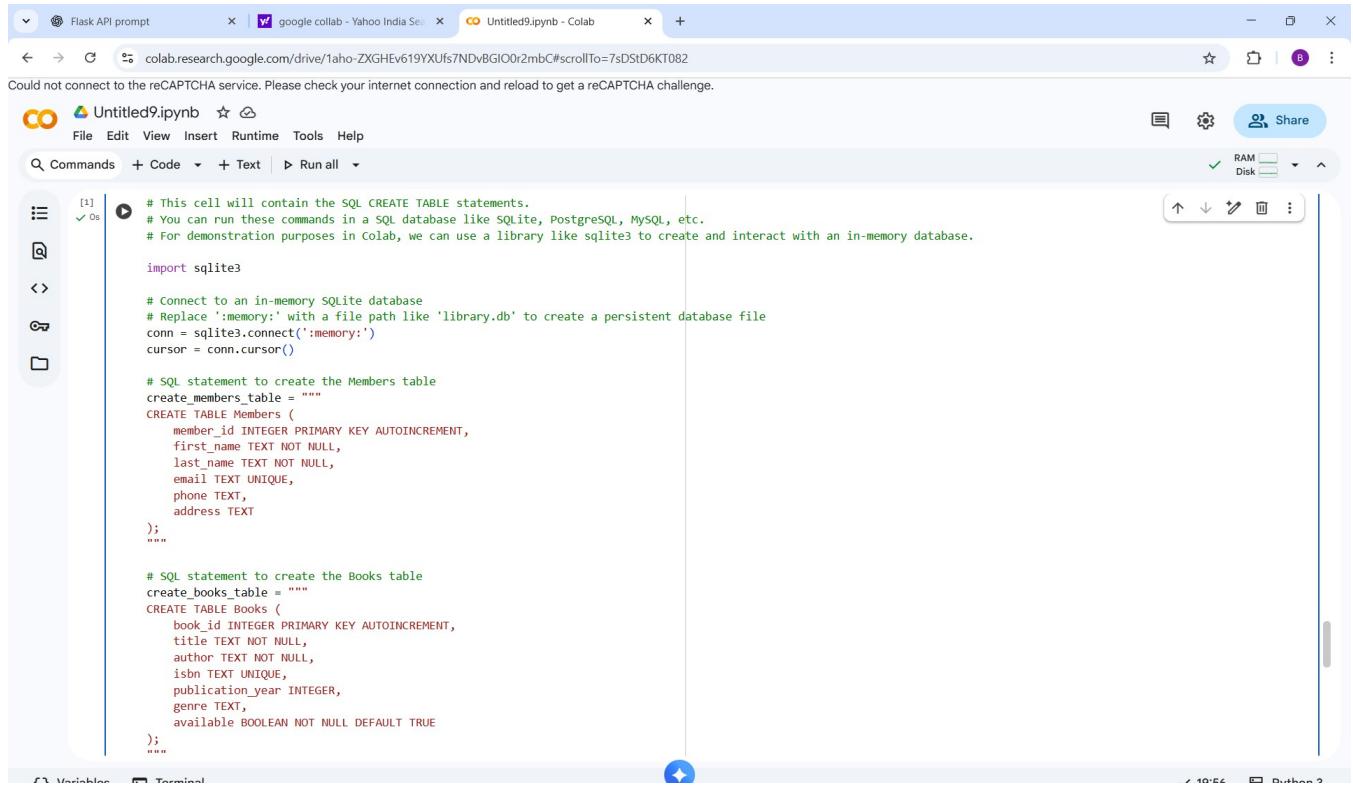
CREATE TABLE Books (
    book_id INT PRIMARY KEY,
    title VARCHAR(200),
    author VARCHAR(100),
    available BOOLEAN
);

CREATE TABLE Loans (
    loan_id INT PRIMARY KEY,
    member_id INT,
    book_id INT,
    loan_date DATE,
    return_date DATE,
    FOREIGN KEY (member_id) REFERENCES Members(member_id),
    FOREIGN KEY (book_id) REFERENCES Books(book_id)
);
```

PROMPT :

“Design a Database Schema for a Library Management System with tables: Members, Books, and Loans — including primary keys, foreign keys, and necessary attributes. Provide SQL CREATE TABLE statements.”

CODE :



The screenshot shows a Google Colab notebook titled "Untitled9.ipynb". The code cell contains the following SQL statements:

```
# This cell will contain the SQL CREATE TABLE statements.
# You can run these commands in a SQL database like SQLite, PostgreSQL, MySQL, etc.
# For demonstration purposes in Colab, we can use a library like sqlite3 to create and interact with an in-memory database.

import sqlite3

# Connect to an in-memory SQLite database
# Replace ':memory:' with a file path like 'library.db' to create a persistent database file
conn = sqlite3.connect(':memory:')
cursor = conn.cursor()

# SQL statement to create the Members table
create_members_table = """
CREATE TABLE Members (
    member_id INTEGER PRIMARY KEY AUTOINCREMENT,
    first_name TEXT NOT NULL,
    last_name TEXT NOT NULL,
    email TEXT UNIQUE,
    phone TEXT,
    address TEXT
);
"""

# SQL statement to create the Books table
create_books_table = """
CREATE TABLE Books (
    book_id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    author TEXT NOT NULL,
    isbn TEXT UNIQUE,
    publication_year INTEGER,
    genre TEXT,
    available BOOLEAN NOT NULL DEFAULT TRUE
);
"""
```

```
create_loans_table = """
CREATE TABLE Loans (
    loan_id INTEGER PRIMARY KEY AUTOINCREMENT,
    book_id INTEGER,
    member_id INTEGER,
    loan_date DATE NOT NULL,
    return_date DATE,
    FOREIGN KEY (book_id) REFERENCES Books(book_id),
    FOREIGN KEY (member_id) REFERENCES Members(member_id)
);

# Execute the CREATE TABLE statements
try:
    cursor.execute(create_members_table)
    cursor.execute(create_books_table)
    cursor.execute(create_loans_table)
    print("Tables created successfully.")

    # You can optionally add some sample data and query it here
    # For example:
    # cursor.execute("INSERT INTO Members (first_name, last_name, email) VALUES (?, ?, ?)", ('Alice', 'Smith', 'alice.smith@example.com'))
    # conn.commit()
    # cursor.execute("SELECT * FROM Members")
    # print("Members:", cursor.fetchall())

except sqlite3.Error as e:
    print(f"An error occurred: {e}")

finally:
    # Close the connection
    conn.close()
... Tables created successfully.
```

OUTPUT :

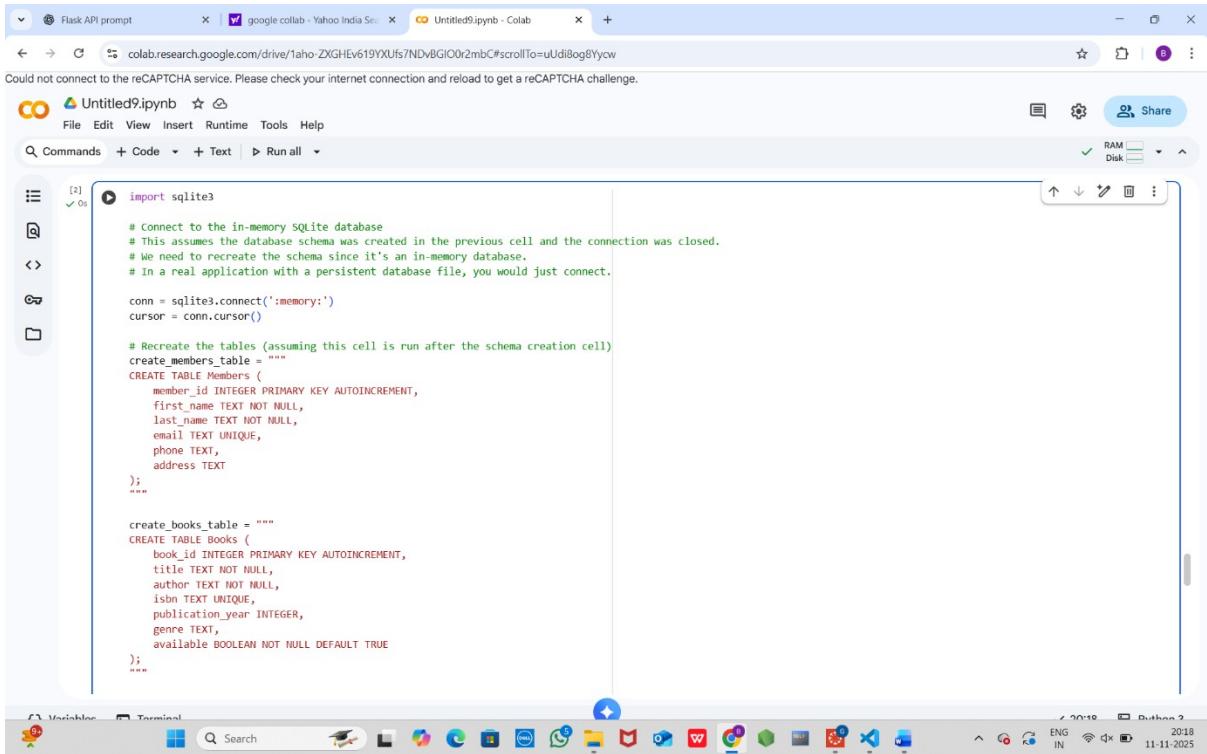
Tables created successfully.

TASK 2 :

*Ask AI to generate **INSERT INTO** queries for the schema above (3 sample records per table).*

PROMPT : Ask AI to generate **INSERT INTO queries for the schema above (3 sample records per table).**

CODE :



The screenshot shows a Google Colab interface with a notebook titled "Untitled9.ipynb". The code cell contains Python code using the sqlite3 module to recreate two tables: Members and Books. The Members table has columns for member_id (primary key, autoincrement), first_name, last_name, email (unique), phone, and address. The Books table has columns for book_id (primary key, autoincrement), title, author, isbn (unique), publication_year, genre, and available (boolean, default true). The code is run in a cell labeled [2].

```
import sqlite3

# Connect to the in-memory SQLite database
# This assumes the database schema was created in the previous cell and the connection was closed.
# We need to recreate the schema since it's an in-memory database.
# In a real application with a persistent database file, you would just connect.

conn = sqlite3.connect(':memory:')
cursor = conn.cursor()

# Recreate the tables (assuming this cell is run after the schema creation cell)
create_members_table = """
CREATE TABLE Members (
    member_id INTEGER PRIMARY KEY AUTOINCREMENT,
    first_name TEXT NOT NULL,
    last_name TEXT NOT NULL,
    email TEXT UNIQUE,
    phone TEXT,
    address TEXT
);
"""

create_books_table = """
CREATE TABLE Books (
    book_id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    author TEXT NOT NULL,
    isbn TEXT UNIQUE,
    publication_year INTEGER,
    genre TEXT,
    available BOOLEAN NOT NULL DEFAULT TRUE
);
"""

Variables Terminal
```

OUTPUT :

Tables recreated for inserting data.

Sample data inserted successfully.

--- Loans ---

[(1, 1, 1, '2023-10-26', None), (2, 2, 2, '2023-10-20', None), (3, 3, 1, '2023-10-25', None)]

TASK 3 :

Use AI to generate a query to list all books borrowed by a specific member

PROMPT:

“Generate an SQL query to list all books borrowed by a specific member using the Library Management System schema.”

CODE :

The screenshot shows a Google Colab notebook titled "Untitled9.ipynb". The code cell contains Python code for creating an in-memory SQLite database and tables for members and books. The code includes comments explaining the steps: connecting to an in-memory database, recreating tables, and defining columns with constraints like PRIMARY KEY, AUTOINCREMENT, NOT NULL, and UNIQUE.

```
[1]: import sqlite3

# Connect to the in-memory SQLite database
# In a real application with a persistent database file, you would just connect.
# Since this is an in-memory database and the previous connection was closed,
# we need to recreate the schema and data first.

conn = sqlite3.connect(':memory:')
cursor = conn.cursor()

# Recreate the tables and insert sample data (assuming the previous cells were run)
create_members_table = """
CREATE TABLE Members (
    member_id INTEGER PRIMARY KEY AUTOINCREMENT,
    first_name TEXT NOT NULL,
    last_name TEXT NOT NULL,
    email TEXT UNIQUE,
    phone TEXT,
    address TEXT
);
"""

create_books_table = """
CREATE TABLE Books (
    book_id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    author TEXT NOT NULL,
    isbn TEXT UNIQUE,
    publication_year INTEGER,
    genre TEXT,
    available BOOLEAN NOT NULL DEFAULT TRUE
);
"""

Variables Terminal
```

OUTPUT :

Database schema and data recreated for querying.

--- Books borrowed by Member ID 1 ---

*('The Hitchhikers Guide to the Galaxy',
'Douglas Adams', '2023-10-26', None)*

('1984', 'George Orwell', '2023-10-25', None)