

(20 points)

Overview

Objectives: To develop the ability to create and manipulate tables using SQLite3 and write simple queries using SQL.

Submission File: [hw1.yaml](#)

What to Submit: Submit the .yaml file you downloaded above, with your answers to Gradescope.

Setup: Use the SQLite command to create a database named `music.db`. If the database already exists, `sqlite3` will open it. If you want to specify a different directory for the database, use the full path like so: `sqlite3 /YourPath/music.db`.

Information available for public disclosure

See the `.yaml` formatting and SQL code [formatting guide](#) on the class website for explicit instructions on submitting proper yaml files.

Warning

This is the only homework with **unlimited submission attempts** to the autograder. Your highest graded submission will be automatically selected.

Creating the Database and Table (5 points)

- Table Creation (1 points):** Write a SQL statement to create a table named `Artists` with the following schema:
 - `ArtistId: Integer` (Primary Key)
 - `Name: String`
- Data Insertion (1 point):** Insert the following artist tuples into the `Artists` table:

```
(1, 'John Lennon')
(2, 'Roger Waters')
(3, 'Bob Marley')
```

```
(4, 'Eric Clapton')
(5, 'B.B. King')
(6, 'Buddy Guy')
(7, 'Jimi Hendrix')
```

3. **Queries (3 points):** Write SQL queries to:

- 3.1 Return all tuples from the `Artists` table.
- 3.2 Find tuples whose artist names start with "B".
- 3.3 Retrieve the `Name` attribute for the artist with `ArtistId` equal to 2.

Understanding SQLite Behavior (3 points)

Using the SQLite documentation and your own understanding, answer the following questions related to data insertion in the `Artists` table:

4. **Insertion Behavior (1 point):**

- Query: Insert the tuple `(-1, 'Michael Jackson')` into the `Artists` table.
- Select the best answer and add only the letter into the `.yaml` file:

- A. SQLite allows negative integers for primary keys by default.
- B. SQLite throws an error because primary keys cannot be negative.
- C. SQLite automatically converts the negative integer to a positive one.
- D. The data is inserted, but 'Michael Jackson' becomes the primary key.

5. **Duplicate Insertion (1 point):**

- Query: Insert the tuple `(2, 'Roger Waters')` again into the `Artists` table.
- Select the best answer and add only the letter into the `.yaml` file:

- A. SQLite inserts the tuple because it allows duplicate primary keys.
- B. SQLite throws a unique constraint violation error because the primary key already exists.
- C. SQLite updates the existing record with the new data.
- D. The insertion is ignored without any error.

6. **NULL Insertion (1 point):**

- Query: Insert the tuple `(NULL, 'Roger Waters')` into the `Artists` table.
- Select the best answer and add only the letter into the `.yaml` file:

- A. SQLite treats NULL as a unique value and inserts the tuple.
- B. SQLite generates a new unique primary key for the tuple and inserts it.
- C. SQLite throws an error because primary keys cannot be NULL.
- D. The insertion is ignored without any error.

Querying and Manipulating Data (12 points)

Warning

Heads up! Questions depend on each other, so solve them sequentially. I.e. do not submit questions 8, without finishing question 7.

7. **Table Creation (3 points):** Create a table named `Albums` with the following schema:

- `AlbumId: Integer` (*Primary Key*)
- `Title: Varchar` (*no need to specify the number of bits this time*)
- `Released: Date` (*Format: 'YYYY-MM-DD'*)
- `ArtistId: Integer` (*Foreign Key to Artists table*)

8. **Data Insertion (1 point):** Insert the following 10 albums into the `Albums` table:

```
(1, 'Imagine', '1971-09-09', 1),
(2, 'The Wall', '1979-11-30', 2),
(3, 'Legend', '1984-05-08', 3),
(4, '461 Ocean Boulevard', NULL, 4),
(5, 'Live at the Regal', '1965-11-21', 5),
(6, 'Damn Right, Ive Got the Blues', '1991-08-27', 6),
(7, 'Axis: Bold as Love', NULL, 7),
(8, 'Plastic Ono Band', '1970-12-11', 1),
(9, 'Three Little Birds', '1980-03-05', 3),
(10, 'Blues Breakers with Eric Clapton', '1966-07-22', 4);
```

9. **Output Formats (2 points):** Using SQLite's formatting commands (e.g. `mode`) for both subquestions (*otherwise it may be graded incorrectly.*), write SQL queries to display all albums in:

- 9.1. Comma-separated form.
- 9.2. Column form, with each column width set to 15 characters.

10. **Artists and Albums (6 points):**

- 10.1. List artist names and their album titles in alphabetical order by artist names.
- 10.2. Retrieve unique artist names who have released at least one album on or after the year 1980. (Refer to [Date Operations](#))
- 10.3. Retrieve the Artist name and titles of all albums without a release date.

Bonus Question (1 point)

11. Provide a SQL query that lists the top 5 artists along with the number of albums they have released in descending order. If the album counts are the same, the artist names should be listed in ascending order.

Date Operations

Feel free to try these SQLite date operations to understand their usage:

- `SELECT date('2011-03-28');`
- `SELECT date('now');`
- `SELECT date('now', '-5 year');`
- `SELECT date('now', '-5 year', '+24 hour');`