SQLite Exercise

- Use the Chinook database

-- Task 1

-- Imagine we would like to throw a promotional Music Festival in the city we made the most money.

--Write a query that returns the 1 city that has the highest sum of invoice totals.

--Return both the city name and the sum of all invoice totals. So, which city has the best customers?

```
SELECT BillingCity,
   SUM(Total) AS [Total Spend]
FROM invoices
GROUP BY BillingCity
ORDER BY [Total Spend] DESC
LIMIT 1;
```

-- Task 2

-- Consider that we know our customers love rock music, and we can decide which musicians to invite to play at the concert.

-- Let's invite the artists who have written the most rock music in our dataset.

-- Write a query that returns the Artist name and total track count of the top 10 rock bands.

```
SELECT ar.name AS Artist,
   COUNT(t.name) AS [Total Tracks]
FROM artists AS ar
JOIN albums AS al ON ar.artistid = al.artistid
JOIN tracks AS t ON al.albumid = t.albumid
JOIN genres AS g ON t.genreid = g.genreid
WHERE g.name = 'Rock'
```

```
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
```

-- Task 3

-- Build a query that returns the person who has spent the most money

---

```
-- Solution 1 (using two tables)
SELECT c.FirstName || " " || c.LastName AS Name,
printf("%.2f", SUM(i.Total)) AS Total
FROM customers AS c
JOIN invoices AS i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName
ORDER BY Total DESC
LIMIT 1;
```

-- Task 4

-- Use your query to return the email, first name, last name, and Genre of all Rock Music listeners.

-- Return your list ordered alphabetically by email address starting with A.

```
SELECT c.Email,
c.FirstName,
c.Lastname,
g.name As Genre
FROM customers AS c
JOIN invoices AS i ON c.customerid = i.customerid
JOIN invoice_items AS l ON i.invoiceid = l.invoiceid
```

```
JOIN tracks AS t ON l.trackid = t.trackid
JOIN genres AS g ON t.genreid = g.genreid
WHERE g.Name = "Rock"
ORDER BY c.Email;
```

-- Task 5

-- First, find which artist has earned the most according to the InvoiceLines?

-- Second, use this artist to find which customer spent the most on this artist.

```
SELECT ar.Name,
SUM(l.UnitPrice * l.Quantity) AS Total
FROM artists AS ar
JOIN albums AS al ON ar.artistid = al.artistid
JOIN tracks AS t ON al.albumid = t.albumid
JOIN invoice_items AS l ON t.trackid = l.trackid
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

-- who spent the most

```
SELECT c.FirstName || " " || c.LastName AS Name,
ar.Name AS [Artist Name],
SUM(i.Total) AS Total
FROM customers AS c
JOIN invoices AS i ON c.CustomerId = i.CustomerId
JOIN invoice_items AS l ON i.invoiceid = l.invoiceid
JOIN tracks AS t ON t.trackid = l.trackid
JOIN albums AS al ON t.albumid = al.albumid
```

```
JOIN artists AS ar ON al.artistid = ar.artistid
WHERE ar.Name = (
   SELECT ar1.Name
   FROM artists AS ar1
   JOIN albums AS al1 ON ar1.artistid = al1.artistid
   JOIN tracks AS t1 ON al1.albumid = t1.albumid
   JOIN invoice_items AS l1 ON t1.trackid = l1.trackid
   GROUP BY 1
   ORDER BY SUM(l1.UnitPrice * l1.Quantity) DESC
   LIMIT 1
)
GROUP BY 1,2
ORDER BY Total DESC;
```

-- Task 6

-- Count how many songs base on genre does customer 12 bought

-- and How much did customer 13 spent across genres?

-- How much did each customers spent per genre?

```
-- customer 12
SELECT c.FirstName,
g.Name AS Genre,
COUNT(t.Name) AS Songs
FROM customers AS c
JOIN invoices AS i ON c.customerid = i.customerid
JOIN invoice_items AS l ON i.invoiceid = l.invoiceid
JOIN tracks AS t ON l.trackid = t.trackid
JOIN genres AS g ON t.genreid = g.genreid
```

```
WHERE c.customerid = 12
GROUP BY 1,2
ORDER BY 1;

-- customer 13
SELECT c.FirstName,
g.Name AS Genre,
COUNT(t.Name) AS Songs,
SUM(l.UnitPrice * l.Quantity) AS Spent
FROM customers AS c
JOIN invoices AS i ON c.customerid = i.customerid
JOIN invoice_items AS l ON i.invoiceid = l.invoiceid
JOIN tracks AS t ON l.trackid = t.trackid
JOIN genres AS g ON t.genreid = g.genreid
WHERE c.customerid = 13
GROUP BY 1,2
ORDER BY 1;

-- each customer
SELECT c.FirstName,
g.Name AS Genre,
COUNT(t.Name) AS Songs,
SUM(l.UnitPrice * l.Quantity) AS Spent
FROM customers AS c
JOIN invoices AS i ON c.customerid = i.customerid
JOIN invoice_items AS l ON i.invoiceid = l.invoiceid
JOIN tracks AS t ON l.trackid = t.trackid
JOIN genres AS g ON t.genreid = g.genreid
GROUP BY 1,2
```

```
ORDER BY 1, 3 DESC;
```

-- Task 7

-- Write a SELECT statement that will return the Artistid column from the artists table (use table alias)

-- and the albumid column from the albums table (use table alias) using a LEFT join.

-- Specify the Artistid column common to both tables as a predicate in the ON clause of the join

-- Execute the written statement.

```
SELECT t.artistid, m.albumid FROM artists AS t
LEFT JOIN albums AS m
   ON t.artistid = m.artistid;
```

-- Task 8

-- Write a SELECT statement that joins the albums table (use table alias "A")

-- with the tracks table (use table alias "T")

-- Use the albumid column common to both tables as a predicate in the ON clause of the join (INEERE JOIN)

-- Return the trackid, name and title columns from the tracks table

-- Execute the written statement.

```
SELECT T.trackid, T.name, A.Title FROM albums AS A
JOIN tracks AS T
   ON A.albumid = T.albumid;
```

-- Task 10

-- Write a SELECT statement that joins(SELF JOIN by using INNER JOIN) the employees

-- employees reports to manager

-- using two columns: EmployeeId and ReportsTo

-- Use ORDER BY clause for manager column

-- Execute the written statement.

```
SELECT M.firstname || ' ' || M.lastname AS Employee, E.firstname || ' ' || E.lastname AS Manager
FROM employees AS E
JOIN employees AS M
   ON E.employeeid = M.reportsto
   ORDER BY Manager;
```

-- Task 11

-- Write a Select statement using Like Operator to search the words which start with "Wild" from the tracks table.

-- Write a Select statement using Glob Operator to search the words which end with "Man" from the tracks table.

```
SELECT * FROM tracks
WHERE name LIKE 'Wild%';

SELECT * FROM tracks
WHERE name GLOB '*Man';
```

-- Task 12

-- Use the albums and tracks table:

-- Write a querry and a subquerry to return all the tracks in the album that have the title 'Let There Be Rock'. Make sure your resul should have three columns of 'trackid', 'album id' and 'name'.

```
SELECT trackid, albumid, name FROM tracks
WHERE albumid = (
SELECT albumid FROM albums
WHERE title = 'Let There Be Rock'
);
```

-- Task 13

-- Write a query to return all customers whose sales representative is in Canada.

-- Make sure use the IN operator

-- Use employees and customers tables

```
SELECT * FROM customers
WHERE SupportRepId IN (
SELECT EmployeeId FROM employees
WHERE Country = 'Canada'
);
```