

DBI Lab 006: JOINS, UPDATE, DELETE, & more

COMP1048: Databases and Interfaces

Matthew Pike & Yuan Yao

Lab Overview

This week's lab is a continuation of the previous lab. We will be using an extended version of the database from the previous lab, but we will be performing more advanced queries. For this lab, you will be given a pre-populated database. You will be required to write SQL queries to answer the questions below.

Lab Setup

1. Download the database file from [Moodle](#). This is a SQLite database file (Note the `.db` file extension).
2. Open the database file using the SQLite Command Line Interface (CLI).

Mac and Linux Users

You can do this by opening a terminal window and typing `sqlite3` followed by the path to the database file. For example, if the database file is in your Downloads folder, you would type `sqlite3 ~/Downloads/DBI_Lab006.db`.

Windows Users

Ensure that the `DBI_Lab006.db` file and the `sqlite.exe` are in the same directory. Double click on the `sqlite.exe` file to open the SQLite CLI. Then type `.open DBI_Lab006.db` to open the database file.

3. In order to include table headers in the output, type `.headers on` and press enter. This will ensure that the column names are displayed in the output. We can also change the output format to be more readable by typing `.mode column` and pressing enter.

Lab Questions

Before you start, you should familiarize yourself with the database schema.

You can do this by typing `.schema` and pressing enter. This will display the schema of the database.

1. Write a query to display the title and year of all movies. You should also list the names of the leading actor that starred in each movie. All movies should be listed even if they do not have a leading actor. The output should be ordered by **year of release first in ascending order**, then **by movie title in ascending order**. Your output should look like this:

Year	Movie Title	Leading Actor
...

2. Write a query that displays the combined sales number for each region (for all movies). The output should be ordered by region name in ascending order. Your output should look like this:

Region	Total Sales
...	...

3. Write a query that displays the **sales number and region for all films**. If a film has sales data from many regions, each region should be displayed on a separate line. **All films should be listed at least once**, even if they have no sales data. The output should be ordered by year of release first in ascending order, then by movie title in ascending order. Your output should look like this:

Year	Movie Title	Region	Sales
...

4. Write a query that displays the combined sales number **(summed over all regions) for each movie**. The output should be **ordered by sales in descending order**. All films should be listed at least once, even if they have no sales data. Your output should look like this:

Year	Movie Title	Sales Number
...

5. Write a query that displays the combined sales numbers for all films (across all regions) that an actor has starred in. **All actors should be listed at least once, even if they have not starred in any films**. The output should be ordered by sales in descending order. Your output should look like this:

Actor Name	Sales Number
...	...

6. Write a query that updates the sales number of all movies that were released in Europe to be 10% higher than the original. Updated sales numbers should be **rounded to the nearest integer**. Your query should produce no output.
7. Write a query that reports the **income for each movie**. The income is the total sales number (summation over all regions) multiplied by the movie price. The output should be ordered by year of release first in ascending order, then by movie title in ascending order. Your output should look like this

Year	Movie Title	Income
...

8. Write a query that identifies the most popular actor in each region. The actor's popularity in one region is measured by the **average sales of all movies in that region that the actor has starred in**. The output should be ordered by region name in ascending order. Your output should look like this:

Region	Actor Name	Average Sales
...

Hint:

In SQL, **GROUP BY** can be used to group rows that have the same value in more than one column. For example, the following query will group all rows that have the same value in the **year** and **title** columns.

```
SELECT year, title
FROM movies
GROUP BY year, title
```

9. Write a query that deletes all movies that have no sales data. Your query should produce no output.

Submission

Please submit a PDF document containing your SQL solutions to the above tasks. **You do not need to include the output of your queries.**

Submitting this assignment will contribute 1% to your overall Module grade.

Your submission should demonstrate reasonable effort and fulfil the specified requirements set out in this lab sheet to receive the available marks. There is no granularity to the marking, the marking is on a pass-or-fail basis.

The submission point is available on Moodle.

Submission Deadline - Monday, 14 November 2022 @ 15:00