

Assignment 7: Indexes and Transactions

The purposes of this assignment is to explore a few topics related to database **efficiency** and **reliability**: indexes and transactions!

In the starter files, you'll find a copy of the **movies.db** database; import this into a new Codespace. Create a file called **a7.sql** in which to write your queries.

In **a7.sql**, add the following:

.mode box

This will activate box-formatted result tables when the script is run.

Label each task clearly with its number, using comments, within the **a7.sql** file.

Task 1: Creating and Timing the Query

Your first task is the following:

- Write a query that returns all movies in which the actor **Burgess Meredith** has starred, with columns **movie_title** and **actor_name**. Figure out how to have SQLite explain the process by which its query optimizer will execute the query.
- Execute this query, and output the execution duration. (You can use **.timer on** directly in the script to start the timer, and **.timer off** to stop it after execution of the query.) Make a note of the **real** time.

Here's what the result set should look like:

movie_title	actor_name
A Fan's Notes	Burgess Meredith
The Man	Burgess Meredith
Hay que matar a B.	Burgess Meredith
Golden Needles	Burgess Meredith
92 in the Shade	Burgess Meredith
The Day of the Locust	Burgess Meredith
Burnt Offerings	Burgess Meredith
The Great Bank Hoax	Burgess Meredith

Task 2: Speeding Up the Query

As you can see, this isn't exactly a fast query to run. If we were to implement a movie info website using this database, the speed may be unacceptable, especially with latency from other processes like the HTTP request/response cycle.

Figure out a way to create one or more indexes to speed up this query. To demonstrate your results, set up your script in the following way:

- First, drop the index/indexes;
- Generate the query plan explanation;
- Run the timed query **WITHOUT** the indexes.
- Create the indexes;
- Once again, generate the query plan explanation and run the timed query **WITH** the indexes. Make a note of the difference in run times.

Write a short explanation of **WHY** your chosen indexes improved the speed of the query; the explanation should reference this case specifically. (You'll later paste this explanation in the Comment area of the Brightspace Hand In form.)

Task 3: Effect of Indexes on Delete

Write a query to delete all rows from the **stars** table with ID > 10,000 (this will be most of the rows). Run this query—and time it—with and without the indexes in place. However, to avoid permanently losing data, figure out a way to run and time the queries without permanently committing the results! (Hint: this is where **transactions** come in.)

To make it easier for the marker, label these events using the following queries:

SELECT 'now deleting stars without indexes' AS status;
SELECT 'now deleting stars WITH indexes' AS status;

Compare the **real** duration of the queries with and without the indexes. Write an explanation of **WHY** you're seeing this result.

Grading

- Task 1:
 - [1 mark] query to fetch results;
 - [1 mark] software-generated explanation of query plan and timer reports;
- Task 2:
 - [1 mark] indexes set up; timing with indexes in place;
 - [1 mark] your written explanation of why the indexes improve query speed for this case;
- Task 3:
 - [1 mark] delete query implemented;
 - [1 mark] your written explanation of the difference in timing with and without indexes.

Total: **6**

Hand In

Place the **movies.db** and **a7.sql** files in a folder called **a7-firstname-lastname**, remove any unnecessary files (like these instructions), zip the folder, and navigate to the hand in area of the appropriate Brightspace folder. Copy your written explanations for tasks 2 and 3 and paste them into the comment area. Submit your work.