In-Class Assignment 18

Comparing Query Performance

Follow the instructions below and answer the questions that follow. Add your solutions to the SQL script called **In-Class Assignment 18.sql** and submit to Canvas by the deadline listed above. Save your file frequently to avoid losing work!

Instructions:

Write queries based on the **sakila** schema in your MySQL script to answer the questions below. Include the answer to your question in a /*block comment*/ in the script underneath the query that answers it.

Note that execution times will be slower on the first run than on subsequent runs. Therefore, for each question, **run each query once before examining performance with EXPLAIN ANALYZE.**

Queries:

1. Is retrieving an indexed variable faster than retrieving a non-indexed variable?

   In the **customers** table, last name is indexed and first name is not. Write queries selecting each one and compare the performance. Note that there are an equal number of first names and last names recorded in the customers table, so differences in processing time is not due to varying amounts of data in each column.

2. Is sorting by an indexed variable faster than sorting by a non-indexed variable?

   In the **customers** table, last name is indexed and first name is not. Write queries selecting first name and last name from the table, and compare the performance when sorting descending by first name vs. by last name.

3. Does an inner join execute faster than an outer join?

   Write a query that displays **category id** and **category name,** only for categories that match records in the **film_category** table. Formulate one query using an inner join and one using a left join and compare performance. Be sure that both versions of the query produce the same result set to ensure you're evaluating comparable queries.

4. Are searches with leading wildcards in search strings really that slow?

   Refer to question 1: how long did it take to retrieve 599 unindexed first names?
   Now write a query to display only movie titles (an indexed field) that end in 'DINOSAUR'.
   How many records are returned and how long does the query take to execute? What does this comparison tell you about the speed of leading wildcard characters in search strings?

5. How does adding an unnecessary clause affect performance?

   Write a simple query to select all fields in the **rental** table. Then, write another query that selects all variables and filters by rental ID > 0 (all values of rental ID are greater than 0). Compare the performance of each query.

6. Think of two queries (different than the ones above) that will produce the same result set and compare their performance. They don't have to follow the categories shown in the "Tips" slide from lectures. Report your findings. If you find anything unexpected, explain why you think it might have occurred.