RMIT UNIVERSITY

School of Computing Technologies

ISYS1101/ 1102 Database Applications

Week 4: Tute/Lab – Query Optimisation

Semester 2 2022

1 Objective

There are two parts in this week's Tute/lab session.

The objective of the first part is to learn more about qery evaluation and optimisation:

- Use EXPLAIN PLAN to see the workings behind a query;
- Try out a few queries with and without indexes and see the performance differences;
- Explore the behaviour of AND and OR conditions;
- Explore the impact of join order;
- Explore the performance of nested joins and nested queries.

In the second half, you will demonstrate your database design for the Assignment 1 (first milestone) to the marker (your tutor). The objective of this milestone is to offer you initial feedback on your database design. The marker will provide you feedback on "fit-for-the-purpose" of your database design and point you to major flaws (if any). If there are major flaws, the marker will give you one last chance to rectify them and present to them for remarking.

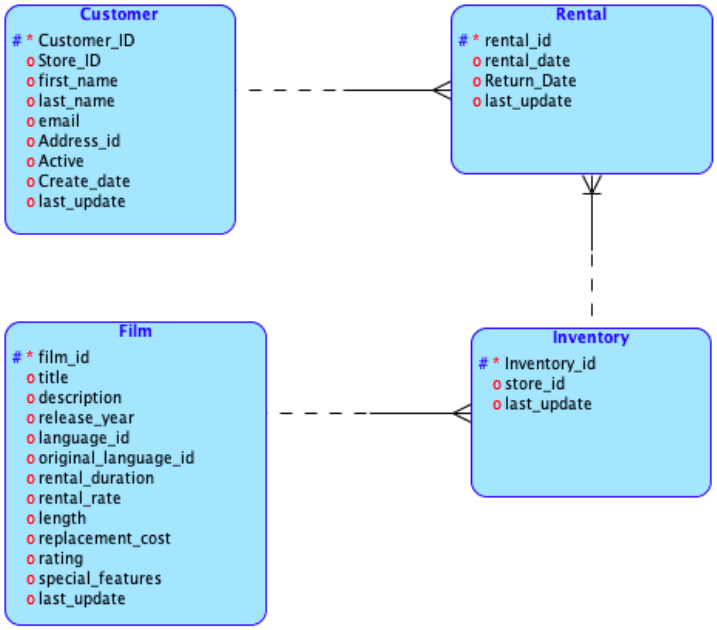
2 Preparation Tasks

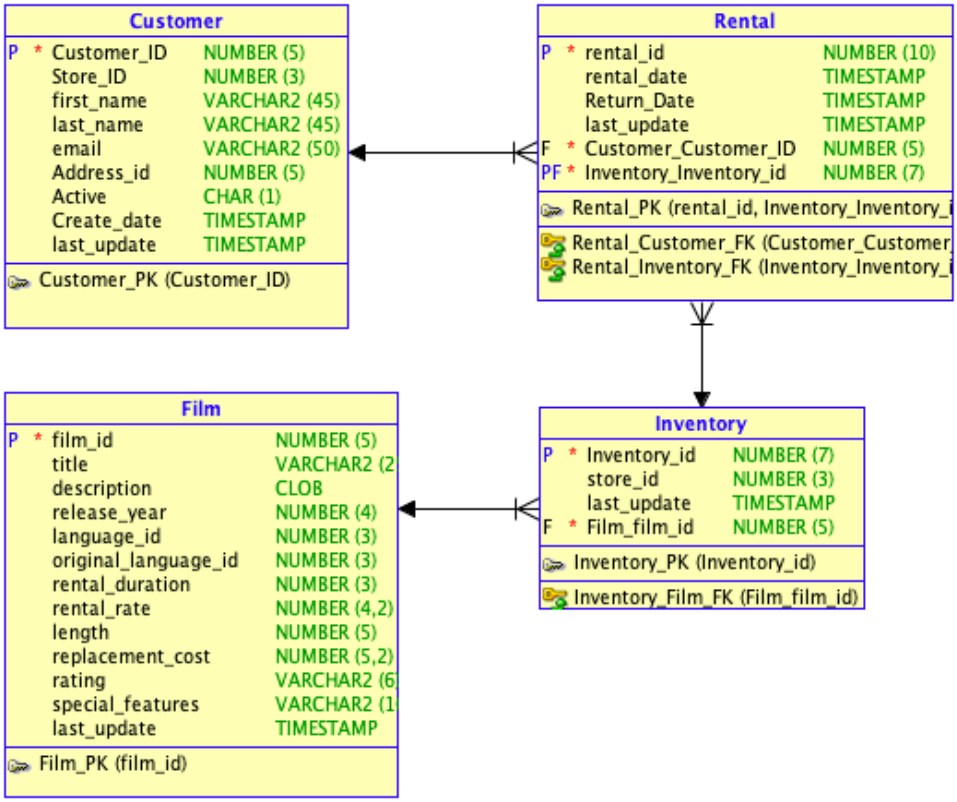
From now on, you cannot use sample databases. You must have your own database schema. Contact ITS if you still did not receive an email on connection settings.

Last week we downloaded and built the rentals table using `rental-DDL.sql` and `rental-populate-bulk-ALL.sql`. This table is a part of a movie rentals database (extracted from MySQL sample database – Sakila movie rental database). For the activities in this week's tute/ lab session, you are required to download DDL and populate scripts for the following tables in this database.

- Customer
- Film
- Inventory

Download `<tablename>-DDL.sql` and `<tablename>-populate-bulk-ALL.sql` scripts from Canvas (Modules → Course Resources → Movie Rentals Database). Run them on SQL Developer to build these four tables.
The underlying data model (conceptual and physical) that these tables are built on is as follows:





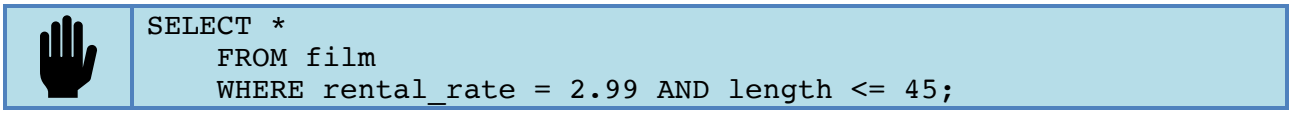
Please note that the tables do not have any constraints (no primary keys or foreign keys) built on to them. That's intentional, as we explore the impact of indexes and add them later.

Your Lab assistant will discuss with you the performance of simple INSERTs and bulk INSERTs, especially you upload thousands of rows at once.

3 Tutorial Questions on Query Evaluation

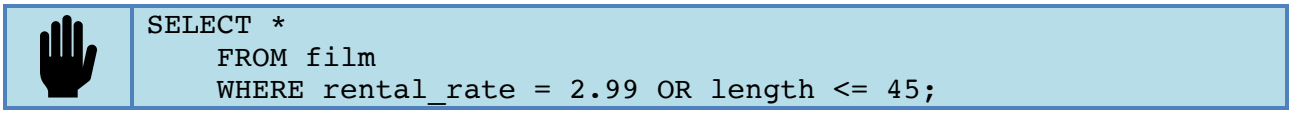
3.1 Compare AND and OR conditions

Activity 1: Run the following query:



```
SELECT *
    FROM film
    WHERE rental_rate = 2.99 AND length <= 45;
```

Obtain the query execution plan for the query.
Replace AND with OR.



```
SELECT *
    FROM film
    WHERE rental_rate = 2.99 OR length <= 45;
```

Obtain the query execution plan for the query.

Compare the performance between these two queries.

How can you explain the similarities and differences?

Activity 2: Add an index on the `length` attribute, and re-run both of the queries.

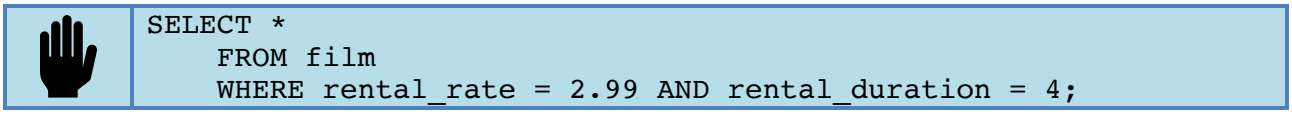
Obtain new query execution plans for the both queries.

Compare the performance between these two queries.

How can you explain the similarities and differences?

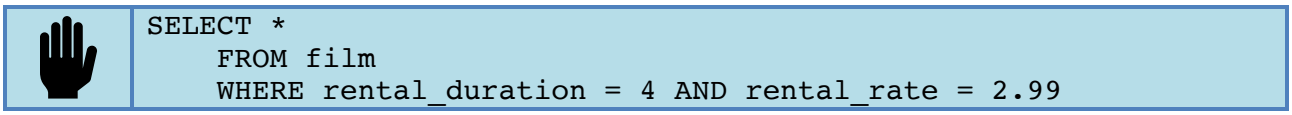
3.2 Changing the order of conditions

Activity 3: Run the following two queries:



```
SELECT *
    FROM film
    WHERE rental_rate = 2.99 AND rental_duration = 4;
```

and



```
SELECT *
    FROM film
    WHERE rental_duration = 4 AND rental_rate = 2.99
```

Obtain the query execution plans for both queries.

Compare the performance between these two queries.

How can you explain the similarities and differences?

3.3 Changing the order of joins

SQL can only join two input sources (such tables or views) at a time. So, if your query involves with joining three or more input sources, they are done with nested joins. This means that two input sources are joined first, then a third input is joined in to that joined result, and so on. So, if we have three tables --- a, b and c, there are different ways of joining them: join a and b then c; join a and c and then b; join b and c and then a.

When a query is optimised the join order is an important factor.

Activity 4:

Write a query using following methods to find first name and last name of customers who rented out the film "Metropolis Coma".

1. Using a single SQL statement with all required tables joined together.
2. Re-write the query using same tables joined in a different order.

Hint: In this sample data set, there are 26 customers who borrowed this movie. Both queries must produce the same result. (The ordering of rows can be different.)

Obtain the query execution plans for both queries.

Compare the performance between these two queries.

How can you explain the similarities and differences?

Activity 5:
Repeat the above two queries by adding different indexes on join attributes (such as customer_id, rental_id, inventory_id, etc) and filter condition attribute (film title).

Obtain the query execution plans for new queries.

Compare the performance among new queries and also compare between with index and without indexes.

How can you explain the similarities and differences?

Your Lab assistant will explain to you an issue with using functions like lower() on attributes that are indexed.

3.4 Query performance with nested queries

Activity 6:

Re-write the query again, this time with no joins, but with a number of nested queries, using IN clause, in the following format:
SELECT something

```
    FROM source1
    WHERE some_attrib IN
        (SELECT another_attrib
             FROM source2
             WHERE some_condition);
```

Obtain the query execution plans for the new query.

Compare the performance between nested JOINs and nested queries.

Was this new query executed in the same way as you entered?

Has the query optimiser made any changes?

4 Assignment 1: Milestone 1 – Feedback on database design

As part of the assignment 1, you are required to demonstrate your Entity-Relationship model, physical database design on SQL Developer, and the database schema you built for your Computerised Voting System back-end.

You can achieve this milestone in Week 4 or 5 labs.

It is highly recommended that you do the demo and receive the marker's feedback prior to start working on programming work on the front-end. If your database design does not meet the specifications (business requirements outlined on assignment specs), the tutor will give you some informal feedback and you will gent one more (only one) chance to do the design and get it marked by the tutor.