

# INFS2200/7903 PROJECT ASSIGNMENT 2

Semester Two 2022

Total Marks:	70 marks
Due Date:	4:00PM 28-October-2022
What to Submit:	SQL script file + short report
Where to Submit:	Electronic submission via Blackboard

---

The goal of the project assignments is to gain practical experience in applying several database management concepts and techniques using the Oracle DBMS. In particular, this assignment mainly focuses on improving database efficiency with views, indexing, and query planning.

Your main task is to first populate your database with appropriate data, then design, implement, and test the appropriate queries to perform the tasks explained in the next sections.

You must work on this project **individually**. Academic integrity policies apply. Please refer to *3.60.04 Student Integrity and Misconduct* of the University Policy for more information.

**Roadmap:** Section 1 describes the database schema for the assignment and provides instructions on downloading the script file needed to create and populate the database. Section 2 describes the tasks to be completed for this assignment. Finally, Section 3 explains the submission guidelines and marking scheme.

Enjoy the project!

## SECTION 1. THE MOVIES DATABASE

**The Database:** The MOVIES database (Figure 1) captures the information regarding movies and the actors in these movies. The database includes six tables: **film**, **actor**, **category**, **language**, **film\_actor**, and **film\_category**. **Film** keeps track of film details. **Actor** stores information about all actors in the movie industry. **Category** stores the information about the different types of film categories. **Language** stores the different languages in which these movies are released. **Film\_actor** and **film\_category** keep track of which actors have acted in which films, and which films are classified under which categories, respectively.

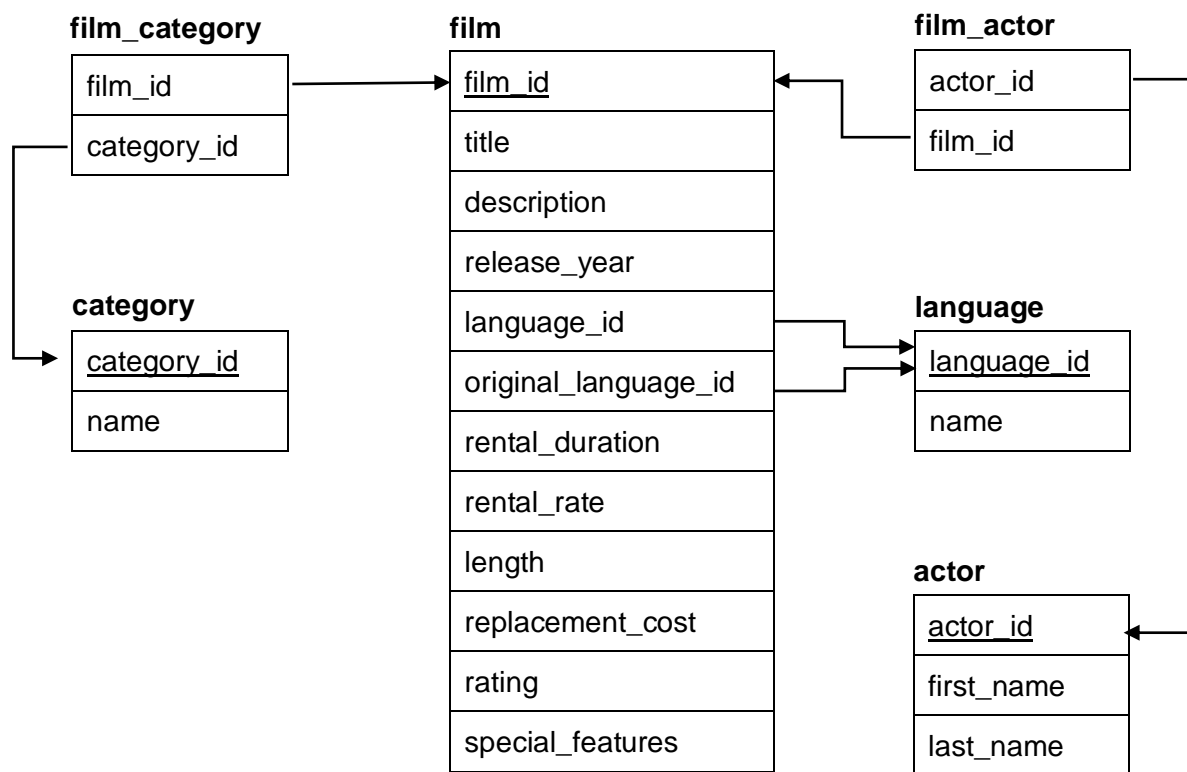


Figure 1 Database schema

**The Script File:** Please go to Blackboard and download the supplementary script file for this project assignment "MoviesDB.sql".

## SECTION 2. ASSIGNMENT TASKS

**Create and Populate Database:** You need to execute the script file “MoviesDB.sql” to create and populate your database before working on the following tasks. Wait until you see the message “DONE! All data has been inserted.” It should only take one or two minutes. The script will also drop related tables.

### Task 1 – Views

1. Write a SQL statement to find all the short (i.e., length < 50) English Comedy films. Here, ‘English’ is the language (not the original language) of the film and ‘Comedy’ is the category of the film. Your query should display the titles of the films.
2. Write a SQL statement to find all actors who have acted in the films you obtained in Task 1.1. Your query should display the ids, first names and last names of the actors. (**Note:** Each actor should only appear once in the query result, even if they may have acted in multiple films)
3. Write a SQL statement to create a (virtual) view called V\_HR\_MU\_2010\_ACTORS that lists the ids, first names and last names of all the actors who have acted in a high-rate (i.e., rental\_rate > 4) Music film released in the year 2010. Here, ‘Music’ is the film category. (**Note:** Each actor should only appear once in the view, even if they may have acted in multiple films)
4. Write a SQL statement to create a materialized view MV\_HR\_MU\_2010\_ACTORS that lists the same information as in Task 1.3.
5. Execute the following two SQL statements and report their query execution time and query execution plan.  
**Question:** Did the materialized view improve the query efficiency? Explain your answer. (**Hint:** You should look at both the elapsed time and the cost in the query execution plan)

```
SELECT * FROM V_HR_MU_2010_ACTORS;
SELECT * FROM MV_HR_MU_2010_ACTORS;
```

### Task 2 – Indexes

1. Write a SQL statement to find the first 10 films (in ascending alphabetical order of the film titles) that take place in a ‘Boat’, i.e., the word ‘Boat’ appears in the film description. Your query should display the film titles. (**Note:** You should avoid using LIKE in the SQL statement and instead use string manipulation functions)

2. In order to potentially speed up the query in Task 2.1, a function-based index could be created on the film table. Write a SQL statement to create an index IDX\_BOAT that best fits the task.
3. Report the execution time and execution plan of the SQL query you wrote in Task 2.1 before and after creating the index in Task 2.2.  
**Question:** Did the index improve the query efficiency? Explain your answer. (*Hint: You should look at both the elapsed time and the cost in the query execution plan*)
4. Write a SQL statement to count the number of films for which there are at least 40 other films with the same release\_year, rating, and special\_features values.
5. In order to potentially speed up the query in Task 2.4, bitmap index can be created on the film table. Write the SQL statements to create bitmap indexes BIDX\_YEAR, BIDX\_RATE, and BIDX\_FEATURE that best fit the task.
6. Report the execution time and execution plan of the SQL query you wrote in Task 2.4 before and after creating the indexes in Task 2.5.  
**Question:** Did the indexes improve query efficiency? Explain your answer. (*Hint: You should look at both the elapsed time and the cost in the query execution plan*)

### Task 3 – Execution Plan

1. A B+ tree index PK\_FILMID has been generated automatically for the primary key column film\_id of the film table. Write the SQL statements to answer the following **Questions:**
  - What is the height of the B+ tree index?
  - What is the number of leaf blocks in the B+ tree index?
  - What is the number of block access needed for a full table scan of the film table?

**Hint:** You may find the following documents from Oracle helpful for Task 3.1:

- [https://docs.oracle.com/cd/B28359\\_01/server.111/b28320/statviews\\_5119.htm#REFRN29025](https://docs.oracle.com/cd/B28359_01/server.111/b28320/statviews_5119.htm#REFRN29025)
- [https://docs.oracle.com/cd/B19306\\_01/server.102/b14237/statviews\\_4473.htm#REFRN26286](https://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_4473.htm#REFRN26286)
- [https://docs.oracle.com/cd/B19306\\_01/server.102/b14237/statviews\\_2105.htm#REFRN20286](https://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_2105.htm#REFRN20286)

2. The following SQL statement lists all the films with a film\_id larger than 100:

```
SELECT * FROM FILM WHERE FILM_ID > 100;
```

Report the rule-based execution plan chosen by the Oracle optimizer for executing this query.

**Question:** Explain the query processing steps taking place in this plan.

3. Report the cost-based execution plan chosen by the Oracle optimizer for executing the query in Task 3.2.

**Question:** Explain the query processing steps taking place in this plan. In your opinion, what are the main differences between the plans you obtained in Task 3.2 and Task 3.3, based on the statistics from Task 3.1 and your calculation? (*Hint: You need to estimate the number of block accesses with and without index. You can assume the film table is sorted by film\_id*)

4. The following SQL statement lists all the films with a film\_id larger than 19,990:

```
SELECT * FROM FILM WHERE FILM_ID > 19990;
```

Report the cost-based execution plan chosen by the Oracle optimizer for executing this query.

**Question:** Explain the query processing steps taking place in this plan. In your opinion, what are the main differences between the plans you obtained in Task 3.3 and Task 3.4, based on the statistics from Task 3.1 and your calculation?

5. The following SQL statement lists all information for the film with a film\_id of 100:

```
SELECT * FROM FILM WHERE FILM_ID = 100;
```

Report the cost-based execution plan chosen by the Oracle optimizer for executing this query.

**Question:** Explain the query processing steps taking place in this plan. In your opinion, what are the main differences between the plans you obtained in Task 3.3 and Task 3.5, based on the statistics from Task 3.1 and your calculation?

### SECTION 3. Deliverables & Marking Scheme

The project is due by **4:00PM, 28 October 2022**. Late submissions will be penalized unless you are approved for an extension (refer to Section 5.3 of the ECP).

You are required to turn in two files (use StudentID to name your files):

1. StudentID.pdf: (replacing StudentID) – *Submit on Blackboard via the Turnitin link “Report Submission”*  
A report that answers all the questions in Section 2 including all the necessary SQL statements and the screenshots of their outputs.
2. StudentID.sql: (replacing StudentID) – *Submit on Blackboard via the upload link “SQL Script Submission”*  
A plain-text script file that includes all your SQL statements.

Your **report** file should include the following content:

- Answers to all the **Questions** in Section 2.
- If you are asked to write SQL statements, you need to include those statements in your report.
- After you execute a SQL statement, if Oracle produces any output (e.g. query result, query execution time, query plan, etc), you should also include a screenshot of the output as well. (**Note:** Please be sensible when including query output. Any output close to the size of one page can be shown by just including the first 10 lines and the last 10 lines. A report that includes multiple pages of a query output will lose presentation marks. You may find some helpful instructions for formatting query output in Practical 1 or the following Oracle documentation)  
[https://docs.oracle.com/cd/A57673\\_01/DOC/server/doc/SP33/ch4.htm](https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SP33/ch4.htm)

Your **script** file is in plain text format. You must make sure that your script file can be executed on the ITEE lab computers by the “@” command. The same SQL statements in your script file should also be copied and pasted into your report file (as explained above). Even though the script file does not introduce any new information compared to the report, it is intended to help the marking tutors to quickly check the correctness of your SQL statements before checking the details in your report file.

Your final mark will be halved if you only submit the script file or the report (not both).

Enjoy the project! Good luck!

### Marking Scheme:

Tasks	Marks	Marking Criteria
1.1	5	<ul style="list-style-type: none"><li>Write only one SQL and generate the correct result</li></ul>
1.2	5	<ul style="list-style-type: none"><li>Write only one SQL and generate the correct result</li></ul>
1.3	6	<ul style="list-style-type: none"><li>View is created with the correct name and semantics. The correctness of the view will be tested by: <code>SELECT * FROM V_HR_MU_2010_ACTORS;</code></li></ul>
1.4	4	<ul style="list-style-type: none"><li>Materialized view is created with the correct name and semantics. The correctness of the view will be tested by: <code>SELECT * FROM MV_HR_MU_2010_ACTORS;</code></li></ul>
1.5	4	<ul style="list-style-type: none"><li>Query execution time and query plans are reported for both view and materialized view</li><li>Answer the Question correctly</li></ul>
2.1	4	<ul style="list-style-type: none"><li>Write only one SQL and generate the correct result (avoid using the LIKE keyword)</li></ul>
2.2	2	<ul style="list-style-type: none"><li>Function-based index is created with the correct name and semantics</li></ul>
2.3	4	<ul style="list-style-type: none"><li>Query execution time and query plans are reported for both before and after index</li><li>Answer the Question correctly</li></ul>
2.4	4	<ul style="list-style-type: none"><li>Write only one SQL and generate the correct result</li></ul>
2.5	3	<ul style="list-style-type: none"><li>Bitmap indexes are created with the correct name and semantics</li></ul>
2.6	4	<ul style="list-style-type: none"><li>Query execution time and query plans are reported for both before and after index</li><li>Answer the Question correctly</li></ul>
3.1	5	<ul style="list-style-type: none"><li>Write the correct SQLs to generate index statistics and table statistics respectively</li><li>Answer the Questions correctly</li></ul>
3.2	3	<ul style="list-style-type: none"><li>Rule-based execution plan is generated</li><li>Answer the Question correctly</li></ul>
3.3	5	<ul style="list-style-type: none"><li>Cost-based execution plan is generated</li><li>Answer the Question correctly</li></ul>
3.4	5	<ul style="list-style-type: none"><li>Cost-based execution plan is generated</li><li>Answer the Question correctly</li></ul>
3.5	5	<ul style="list-style-type: none"><li>Cost-based execution plan is generated</li><li>Answer the Question correctly</li></ul>
Presentation	2	<ul style="list-style-type: none"><li>No query result exceeds one page in the screenshot</li><li>No tuple is broken into multiple lines in the screenshot</li></ul>