

COMP3010E Database Systems Homework 3 (100 pts)

Due: Nov. 14, 2025 11:59PM

- Please submit a **single PDF file** with your answers and email it to zhouxun2023@hit.edu.cn
- Put "[COMP3010E] HW3 <name>" in your email subject, replace <name> with your name.

This homework will continue to use the movie database from HW2. For detailed information about the database, please refer to HW2.

Some Notes: (1) in this dataset, "actors" include both male and female personnel. In the following questions, we simply use "Actors" to refer to all the actors and actresses in the data. Also "movie" may contain some TV shows as well. (2) YR represents the year in which a movie is released (4 digits). (3) ORD represents the order of an actor in a movie's cast. ORD = 1 means this actor is the lead actor. (4) All the names (actors, directors) are in the following format: 'Last, First'. There is a whitespace before the first name.

Part 1: Write PL/SQL code to answer the following 2 questions. Include screenshots of DBMS output (not query results) of your code.

1. (15pts) Write a **stored procedure** called SHOW_DIRECTOR_SUMMARY that takes the director's name as an input parameter and displays the number of movies directed by the director, the average score, and the highest score. For director X, you should let the DBMS display a line like this: "Director X has directed N movies, with an average score of Y and a highest score of Z." If the director has no directed movies, output "No movies directed by this director were found."

For example, if the director you choose is 'Nolan, Christopher', then your output should look like the following (**You must choose a different director in your code**).

Director Nolan, Christopher has directed 8 movies, with an average score of 8.325 and a highest score of 9.

2. (15pts) Create a **trigger** called TRG_SCORE_VALIDATION that performs a validity check on the score field when inserting or updating data in the MOVIE table. First, check if the score field is NULL; if it is, raise an exception with the message "Movie score cannot be null!" Additionally, if the rating is less than 0 or greater than 10, raise an exception with the message "Score must be between 0 and 10!" Lastly, if the validity check passes, output "Operation successful."

Test your trigger with three different SQL commands, which must generate **three different outputs**. Please copy and paste screenshots of the outputs into your submission file.

Part 2: Write an **expression in relational algebra to express each of the following 6 queries. Here is the field information for the three tables in the movie database for your reference.**

Movie (MovieID, Title, Yr, Score, Votes, Director)
 Actor (ActorID, Name)
 Casting (MovieID, ActorID, Ord)

3. (10pts) Find the titles (Title) and years (Yr) of all movies released in 2000 or later.

4. (10pts) Find the names of all actors who appeared in movies directed by "Steven Spielberg".

5. (10pts) Find the IDs of movies that are directed by "Christopher Nolan" and star "Leonardo DiCaprio" as the lead actor.

6. (10pts) Find the IDs of all actors who have never appeared in a movie directed by "Christopher Nolan."

7. (10pts) Find the IDs of all other actors who have collaborated with the actor whose ID is 94888 in the same movie.

8. (10pts) Find the IDs of all actors who appeared in each movie directed by "Christopher Nolan."

Part 3: Draw a **tree diagram to represent the corresponding query execution plan of the following relational algebra expression.**

9. (10pts)

$$\pi_{Name} \left(\sigma_{Score > 8.0 \wedge Votes > 30000} \left(Movie \bowtie \left(Actor \bowtie \pi_{MovieID, ActorID}(Casting) \right) \right) \right)$$