DUBLIN INSTITUTE OF TECHNOLOGY

DT228 BSc. (Honours) Degree in Computer Science

Year 3

WINTER EXAMINATIONS 2015-16

DATABASES 2 [CMPU3010]

PATRICIA O'BYRNE
DR. DEIRDRE LILLIS
PAUL COLLINS

TUESDAY 5TH JANUARY 1.00 P.M. – 3.00 P.M.

TWO HOURS

INSTRUCTIONS TO CANDIDATES

READ THE CASE STUDY ON PAGE 2 BEFORE ATTEMPTING QUESTION 1.

THERE IS A SYNTAX GUIDE ON THE FINAL PAGE OF THIS PAPER.

ANSWER **QUESTION 1 (40%)** AND **TWO OTHERS (30%** EACH).

Actor Database

A group of movie production companies use a database to keep track of the actors they use in their movies. Thw database is independent of any one production company. Movies that are pre- and post-production are stored in it. Auditions that are planned to fill acting roles are put into it. Actors can see upcoming auditions and decide whether or not to audition. Once a role has been filled, the actor's stage name is stored with the role in the database. Sometimes an actor will play more than one role in the same movie. Members are movie-goers who review and rate the movies after release. When ten reviews have been recorded for a movie, the 'movie_rating' attribute shows the average review rating. This is recalculated every time a new review is added.

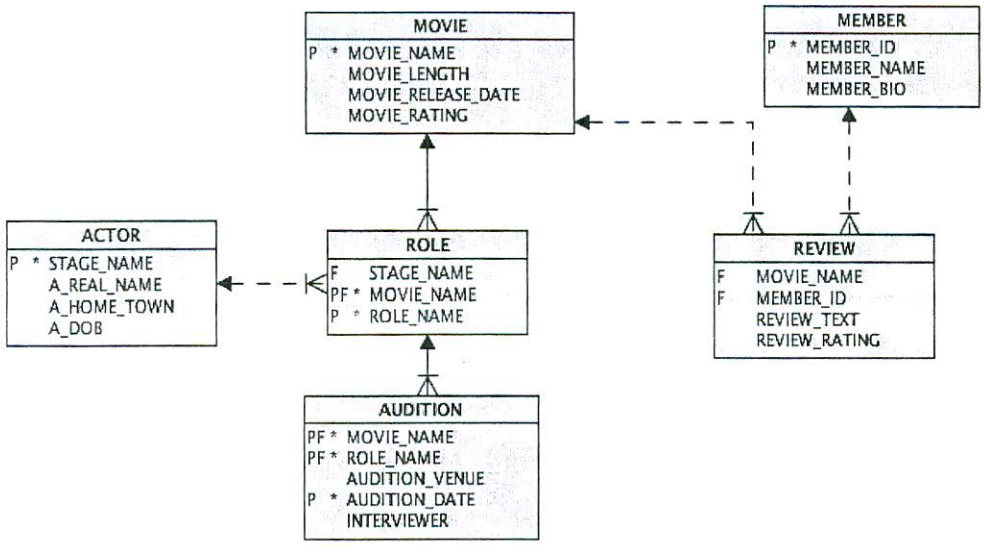


Figure 1 Entity Relationship Diagram – Actor System

1. The diagram in Figure 1 shows the Entity Relationship Diagram, or conceptual schema for the Actor system. Please note: If two movies have the same name, the later movie is entered into the system with the release year as part of its 'Movie_name' (e.g. 'Robocop' was released in 1987, 'Annie' was released in 1982 and 'Robocop (2014)' and 'Annie (2014) were both released in 2014). The system facilitates the following four different types of user:

(i) **The production company** agent who books actors for movies. The production company agent can see all of the stored data about the actor and their previous roles. The agent can add new movies (just the name initially) and add, amend and delete auditions for the roles in the movie. When a role has been filled, the agent can add the stage_name of the actor who will play the role. If an actor is not yet recorded in the database, the production company agent can add the actor (just the stage_name) as soon as the actor has been allocated a role. When the film is ready for release, the agent can add the movie length (in minutes) and release date.

(ii) **The actor.** The actor can add or change biographical information such as real name (A_REAL_NAME) date of birth (A_DOB) and home address (A_HOME_TOWN). The actor can also see auditions that are coming up. The actor's subscription is free and is enabled when the actor is first added to the system.

(iii) **The member** can add reviews of movies that are released. The reviews are restricted to a message of 140 characters and a 0-9 rating for the movie. The member can also update or delete the review. The member cannot see auditions or roles of movies that have not yet been released.

(iv) **Aspiring actors** can pay a subscription to be allowed to see auditions information.

(a) List the tables and columns to which each user role should have access and the type of access the role should have (create, insert, update, delete). (8 marks)

(b) (i) Write SQL to select the review text and movie name of any review for a movie in which the actor with stage name 'Nicole Kidman' plays a role. (5 marks)

(ii) Write SQL to select the stage names of all actors other than the actor with stage name 'George Clooney' that have played the role with Role_name 'Batman'. (5 marks)

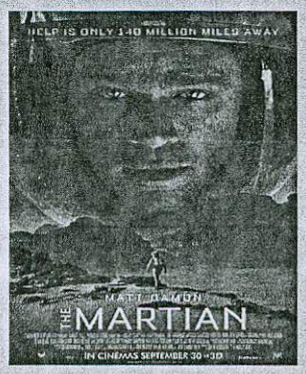
(iii) Write SQL to select the stage_name and movie_name of any actor who has played more than one role in the same movie (e.g. Batman / Bruce Wayne). (5 marks)

(c) Write SQL or PL/SQL code to ensure that an audition for a role in a movie cannot be held after the release date of the movie. (7 marks)

(d) Explain why you think the attribute MOVIE_RATING exists and in what other way could it have been provided, without storing it. (3 marks)

(e) Explain why different notation is used to describe the relationship between Actor and Role and the relationship between Movie and Role. (7 marks).

Movie information and screening times:



THE MARTIAN Censor Rating 12A

Release Date: 30th September 2015

Running time: 141 minutes.

Synopsis: Matt Damon stars in Ridley Scott's epic sci-fi about a lone astronaut fighting for survival on Mars.

Director: Ridley Scott

Screening times for Your Cinema Dublin

Friday 2nd October

Time	Screen and seating type		Screen no.
19:00	2D	Standard Allocated Seat	Screen 14
19:45	3D	Star Seating	Screen 8
20:30	2D	Standard Allocated Seat	Screen 9
21:15	3D	Standard Allocated Seat	Screen 11
22:15	2D	Standard Allocated Seat	Screen 14
22:40	3D	Standard Allocated Seat	Screen 6

Saturday 3rd October.

Time	Screen and seating type		Screen no.
11:00	2D	Standard Allocated Seat	Screen 9
11:40	3D	Standard Allocated Seat	Screen 11
12:40	2D	Standard Allocated Seat	Screen 14
14:10	2D	Standard Allocated Seat	Screen 9
14:50	3D	Standard Allocated Seat	Screen 11
15:50	2D	Standard Allocated Seat	Screen 14
17:20	2D	Standard Allocated Seat	Screen 9
18:00	3D	Standard Allocated Seat	Screen 11
19:00	2D	Standard Allocated Seat	Screen 14
19:45	3D	Star Seating	Screen 8
20:30	2D	Standard Allocated Seat	Screen 9
21:15	3D	Standard Allocated Seat	Screen 11
22:15	2D	Standard Allocated Seat	Screen 14
22:40	3D	Standard Allocated Seat	Screen 6

Figure 2 Cinema Screen times

2. When a new movie is released, the chain of cinemas called 'Your Cinema' publish a list of screen times for each of its locations. The screen times shown in Figure 2 are for the film 'The Martian' (2015) showing in the cinema location Dublin. 3D screens can also project in 2D, but if the movie is available in 3D, they project in 3D. Some screens have sophisticated Star Seating. You may assume that a movie can be uniquely identified by its name and the year it was released (e.g. The Martian 2015). No two screens start screening the same movie at exactly the same time. The censor's judgement about a movie can vary from location to location, so a film that is 12A in one place could be PG in another location.

   (a) Represent the cinema screen times in unnormalised form, first normal form, second normal form and third normal form assuming that the database is purely for the use of all locations in the 'Your Cinema' chain of cinemas and that movies can be uniquely identified by their name and the year in which they were released. (4x5 marks)

   (b) Draw an Entity Relationship diagram of the third normal form, giving the forms meaningful names. (5 marks)

   (c) Describe the meaning of External, Conceptual and Internal / Physical schemas, and state what type of schema you think is represented in your diagram from 2(b), assuming that the 'Your Cinema' chain of cinemas keeps information on screen capacity, ticket sales and availability in addition to the screening times shown in Figure 2. (5 marks)

3. (a) Write SQL to select the stage_name, and average review_rating for all actors, based on the review_rating given in the reviews, ordered from the highest to the lowest average. (10 marks)

   (b) Write SQL to select the stage_name of actors who have appeared in all of the films starting with the word "Ocean's". (10 marks)

   (c) Write SQL to select the names of people who have worked as actors (stage_name) and as directors, but not in the same movie. (10 marks)

4. Write a function to allow a member to add a review ensuring that the member and movie exists and that the movie has been released, updating the current MOVIE_RATING value in accordance with the rule that the MOVIE_RATING should always reflect the average rating for all recorded reviews unless there are fewer than ten reviews recorded for it, in which case it is undefined. The function should have the signature ADD_REVIEW (member_id, movie_name, review_text, review_rating) and return a value of 0 on success and 1 on failure. (30 marks)

SELECT column-list FROM tablename
[WHERE condition]
[ORDER BY column-list]
[GROUP BY column-name]
[HAVING condition]
SELECT column-list FROM join-expression
Join-expression =
table1 JOIN table2 ON condition | USING (column-list)
table1 LEFT JOIN table2 ON condition | USING (column-list)
Conditions : =,>,<,>=,<=,<>, BETWEEN .. AND.., IN (list), IS NULL, LIKE
Logical operators: AND, OR, NOT
Set operations: UNION, INTERSECT, MINUS

CREATE TABLE tablename
({column-definition}
[PRIMARY KEY ({column-name},)]
{[FOREIGN KEY ({column-name}) REFERENCES tablename]})

INSERT INTO tablename [{column-name,}] VALUES (data-value-list)

UPDATE tablename
[SET column-name= <data-value>] [WHERE condition]

Column-definition = column-name [CHAR [(n)] | VARCHAR(n) | NUMBER [n,p] | DATE | DATETIME] {[NOT NULL | UNIQUE | PRIMARY KEY]}

Create or replace trigger <trigger-name>
Before | after
{delete|insert|update [of column[,column]…]}
ON table-name
[referencing {old [as] <old> [new [as] <new>]
For each row
[when (condition)]]
Pl/sql_block

Procedure
CREATE [OR REPLACE] PROCEDURE procedurename [parameter1, parameter2…] IS
[constant/variable declarations]
BEGIN
Executable statements
RETURN Returnvalue
[EXCEPTION
exception handlers
END [procedurename]