Programme Code: TU856/2, TU858/2
Module Code: CMPU2007
CRN: 22391, 26460

TECHNOLOGICAL UNIVERSITY DUBLIN

CITY CAMPUS - GRANGEGORMAN

TU856 - BSc. (Honours) Degree in Computer Science
TU858 - BSc. (Honours) Degree in Computer Science International

Year 2

SEMESTER 1
EXAMINATIONS 2023/24

CMPU 2007 - Databases 1

Internal Examiner(s):
Dr. Mariana Rocha
Dr. Paul Doyle

External Examiner(s):
Ms. Sanita Tifentale (TU856/TU858)

Exam Duration: *Two hours*

Instructions To Candidates: Answer ALL questions. There is a syntax table on the last page to assist you.

Case Study – Movies database

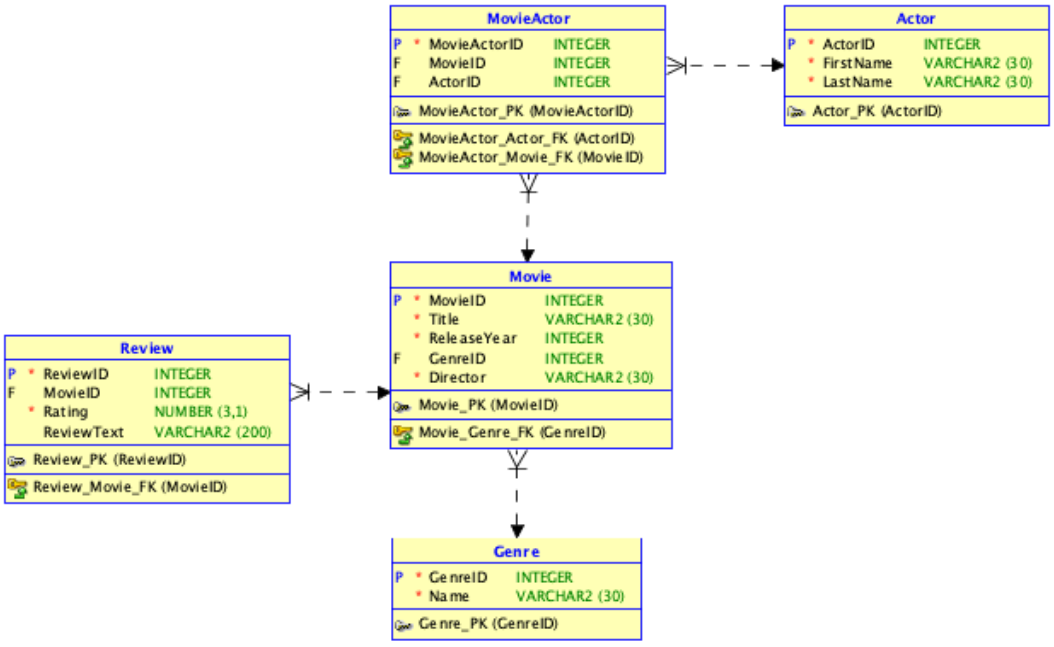
The following relational schema and interpretation will be used in subsequent questions:

A group of movie enthusiasts and developers decided to design a comprehensive database system to store and manage movie-related information. The database is designed to store data about movies, genres, actors, and reviews. It facilitates the retrieval of detailed information about movies, their associated genres, the actors who have appeared in them, and user reviews.

Users can search for movies by genre, view details about the actors who appeared in a particular movie, and read user reviews to make informed decisions about which movies to watch.

Users might also be interested in learning about all the movies featuring a particular actor, and with the MovieActor table, this becomes possible. The Review table enables users to read and contribute their movie reviews, enhancing the user experience and promoting community engagement.

Together, this group of people worked on an ERD to represent this table, resulting in the diagram below:



1. (a) Considering the Entity Relationship diagram (ERD) for the case study, clearly explain the following concepts, offering an example of each:

- Entity
- Attribute
- Relationship
- Primary Key
- Foreign Key

(10 marks)

1. (b) Write an SQL query to retrieve the titles of movies released before the year 2020, along with their respective genres.

(8 marks)

1. (c) Write an SQL query to list the full names of actors who have never been in any movies.

(8 marks)

1. (d) Write an SQL query that retrieves the average rating for each movie in the Review table, along with the movie title.

(8 marks)

1. (e) Write the SQL query to add a new column "Budget" to the Movie table. The column should be of data type DECIMAL, containing up to 10 (ten) digits with a maximum of 2 (two) digits to the right of the decimal point.

(6 marks)

2. (a) Considering the process of database normalisation:

- What are the primary objectives of normalisation?
- Describe the three main normal forms.

(10 marks)

2. (b) Using the given ERD for the Movies database as a reference, propose and describe a new entity, its attributes, and its relationships with existing entities. Make it relevant to a movie database scenario.

(10 marks)

2. (c) The following SQL queries were written as an attempt to insert data into the movie database. However, each query returned an error. Identify and explain the errors in the queries below:

```
INSERT INTO Movie
```

```
VALUES (102, 'The Shawshank Redemption', 1994, 'Frank Darabont');

INSERT INTO Actor (ActorID, FirstName, LastName)
VALUES (201, Tom, Hanks);

INSERT INTO Movie (MovieID, Title, ReleaseYear, GenreID, Director)
VALUES (101, 'Inception', 2010, 1, 'Christopher Nolan', 'christopher.nolan@gmail.com');
```

(10 marks)

3. (a) Suppose the movie, actor and genre tables have been created in a PostgreSQL database with the attributes and datatypes identified in the model given in question 1, part (a).
Write the SQL needed to add the following value constraints to the tables using the ALTER statements:

- The first character of the MovieID must be one of the following: P, L, D, A;
- The first character of GenreID must be between A and Z;
- The Name in the Genre table must be one of the following values: adventure, action, comedy, drama.

(12 marks)

3. (b) Provide an ALTER TABLE statement to add a NOT NULL constraint to the ReleaseYear attribute in the Movie table of the movie database. Explain the significance of adding this constraint and how it ensures data integrity.

(8 marks)

3. (c) Explain the concept of referential integrity in relational databases. How does it maintain data consistency, and how is it enforced?

(10 marks)

SYNTAX TABLE

***Conditions:=,>,<,>=,<=,<>*, BETWEEN .. AND.., IN *(list)*,IS NULL, IS NOT NULL, LIKE**

***Logical operators:* AND, OR, NOT**

***Set operations:* UNION, MINUS, INTERSECT**

CASE [expression]

WHEN condition_1 THEN result_1

WHEN condition_2 THEN result_2

WHEN condition_n THEN result_n

ELSE result

END

SELECT

... FROM table1 LEFT JOIN *table2*

ON *table1.field1 compopr table2.field2* |

USING *clause*

... FROM table1 RIGHT JOIN *table2*

ON *table1.field1 compopr table2.field2* |

USING *clause*

... FROM table1 INNER JOIN *table2*

ON *table1.field1 compopr table2.field2*|

USING *clause*

Key

table1, table2 The tables from which records are combined.

field1, field2 The fields to be joined.

compopr Any relational comparison operator: = < > <= >= or <>

UPDATE *tablename*

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

***Column-definition = column-name* [CHAR [(*n*)] | VARCHAR2(*n*) | NUMBER [*n*,*p*] | DATE | DATETIME] {[NOT NULL | UNIQUE | PRIMARY KEY]}**

Oracle Functions

Null Handling Functions: NVL, NVL2, NULLIF, COALESCE, CASE, DECODE.

Case Conversion functions - Accepts character input and returns a character value: UPPER, LOWER and INITCAP.
Character functions - Accepts character input and returns number or character value: CONCAT, LENGTH, SUBSTR, INSTR, LPAD, RPAD, TRIM and REPLACE.
Date functions - Date arithmetic operations return date or numeric values: MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND and TRUNC.
Group Functions: SUM([ALL | DISTINCT] expression); AVG([ALL | DISTINCT] expression); COUNT([ALL | DISTINCT] expression); COUNT(*); MAX(expression);MIN(expression)