Project 1: MovieDB

CMPE 321, Introduction to Database Systems, Spring 2023

Due: 03.04.2023, 23:59

1 Introduction

In today's world, buying movie tickets and rating movies online has become an integral part of our social lives. With the technological advancements, it has become easier to book and purchase tickets online and provide feedback on the movies we watch. However, a well-designed database system is essential to ensure the reliability, consistency, and security of such systems. Without a well-designed database system, we face the risk of having erroneous data, unsatisfied customers, and loss of revenue. This project aims to design a database system for a movie ticket booking and rating system.

Your task in this project is to design a conceptual database and a logical database based on the given requirements, taking into account various constraints that may arise during the implementation phase. These constraints may include issues such as avoiding overbooking of seats or taking already occupied slots, and providing accurate ratings for movies. In a follow-up project this semester, you will implement the database you designed in this project and an application program that uses that database. By the end of these two projects, you will have gained valuable experience in designing and implementing databases and in the practical application of SQL, while also understanding the importance of data integrity and system reliability in the real world.

2 Project Description

In this project, you will design a booking and rating system for movies. The booking mechanism is only valid for one week, which means that you have only seven days in which movies can be screened as described in more detail in the following sections. You will begin with a detailed description of the content. Then you will need to systematically go through parts of the standard database design process as you learned about in class, including conceptual and logical design. Our database should contain the following information:

1. **User** includes the following attributes; *username, password, name, surname.* Each user has a unique username and each user is either an audience or a director.

(a) **Audience** additionally have a list of bought *tickets for movie sessions* and a list of *rating platforms* to which they are subscribed.

An audience can buy tickets for different movies (or movie sessions) and can subscribe to different rating platforms such as IMDB and Letterboxd.

(b) **Directors** additionally have *nation* information and *platform id*.

- Each director **must** have **only one** *nation*.
- Each director **can** have **at most one** *platform id*.

That is, dual citizenship is not allowed and agreement with more than one platform is not possible for directors.

2. **Rating Platform** includes the following attributes: *platform id, platform name*. Both name and id must be unique.
3. **Movie Sessions** includes the following attributes: *session id, movie id, movie name, duration, genre list, average rating, director username, platform id, predecessors, theatre id, theatre name, theatre capacity, theatre district, time slot, date*. The session id must be unique.
   - Each movie's platform is the same as the platform of its director.
   - No two movie sessions can overlap in terms of theatre and the time it's screened.
   - There are four time slots for each day.
   - The duration of the movie is closely related to the time slots. The time slot attribute determines the starting time of the movie and the end time is determined by the duration. (If a movie starts at time slot 2 and has a duration of 2, the theatre is reserved for that movie during the following time slots: [2, 3]).
   - If a movie has any predecessor movies, all predecessor movies need to be watched in order to watch that movie. (See the example below: The Minions need to be watched before Minions: The Rise of Gru).
   - Each theatre id corresponds to a physical location. Hence, theatre capacity and theatre district depend solely on the theatre id.
   - Same as above; movie id determines the movie name, duration, genre list, overall rating, director username, and platform id.
   - Every movie needs to have at least one genre.
   - Every movie needs to have exactly one director.
4. **Ratings** have the following attributes: *username, movie id, rating*. A pair of username and movie id uniquely identifies a rating. Ratings will be given as floats between 0 - 5 (e.g., 4.5).
   - A user can rate the same movie only once.

- A user can rate a movie
  - if they are already subscribed to the platform that the movie can be rated.
    AND
  - if they have bought a ticket to the movie.

5. **Genre** have following attributes: *genre id, genre name*. Both name and id must be unique.
6. **Database Managers** consists of the following attributes: username and password. There exists only one database manager with a certain username. There can be at most 4 database managers registered to the system.

2.1 Sample Data

The tables below are given as examples to provide sample data for you. Please note that you have to make your own design (e.g., design and create your own tables) according to the description and constraints specified in the previous section.

Table 1: Sample User Data

username	**password**	**name**	**surname**
steven.jobs	apple123	Steven	Jobs
minion.lover	bello387	Felonius	Gru
steve.wozniak	pass4321	Ryan	Andrews
he.gongmin	passwordpass	He	Gongmin
carm.galian	madrid9897	Carmelita	Galiano
kron.helene	helenepass	Helene	Kron
arzucan.ozgur	deneme123	Arzucan	Ozgur
egemen.isguder	deneme124	Egemen	Isguder
busra.oguzoglu	deneme125	Busra	Oguzoglu
peter.weir	peter_weir879	Peter	Weir
kyle.balda	mynameiskyle9	Kyle	Balda

Table 2: Sample Audience Data

username	**list_of_bought_sessions**	**list_of_rating_platforms**
steven.jobs	50001	10130, 10131
steve.wozniak	50004, 50005	10131
arzucan.ozgur	50006	10130
egemen.isguder	50008, 50004, 50007, 50001	10132
busra.oguzoglu	50009	10131

Table 3: Sample Database Managers Data

username	**password**
manager1	managerpass1
manager2	managerpass2
manager35	managerpass35

Table 4: Sample Directors Data

username	**nationality**	**platform_id**
he.gongmin	Turkish	10130
carm.galian	Turkish	10131
kron.helene	French	10130
peter.weir	Spanish	10131
kyle.balda	German	10132

Table 5: Sample Ratings Data

username	**movie_id**	**rating**
egemen.isguder	20001	5
egemen.isguder	20005	5
egemen.isguder	20006	5
arzucan.ozgur	20004	5
busra.oguzoglu	20007	5

Table 6: Sample Genre Data

genre_id	**genre_name**
80001	Animation
80002	Comedy
80003	Adventure
80004	Real Story
80005	Thriller
80006	Drama

Table 7: Sample Rating Platforms Data

platform_id	**platform_name**
10130	IMDB
10131	Letterboxd
10132	FilmIzle
10133	Filmora
10134	BollywoodMDB

Table 8: Sample Movie Sessions Data

session_id	**movie_id**	**name**	**duration**	**genre_list**	**overall_rating**	**director_username**
50001	20001	Despicable Me	2	80001, 80002	5	kyle.balda
50002	20001	Despicable Me	2	80001, 80002	5	kyle.balda
50003	20001	Despicable Me	2	80001, 80002	5	kyle.balda
50004	20002	Catch Me If You Can	2	80003, 80004		he.gongmin
50005	20003	The Bone Collector	2	80005		carm.galian
50006	20004	Eagle Eye	2	80003	5	kron.helene
50007	20005	Minions: The Rise Of Gru	1	80001, 80002	5	kyle.balda
50008	20006	The Minions	1	80001, 80002	5	kyle.balda
50009	20007	The Truman Show	3	80002, 80006	5	peter.weir

Table 9: Sample Movie Sessions Data (cont.).

director_platform_id	**predecessors_list**	**theatre_id**	**theatre_name**	**theatre_capacity**	**district**	**time_slot**	**date**
10132		40001	Sisli_1	300	Sisli	1	3/15/23
10132		40001	Sisli_1	300	Sisli	3	3/15/23
10132		40002	Sisli_2	200	Sisli	1	3/15/23
10130		40002	Sisli_2	200	Sisli	3	3/15/23
10131		40003	Besiktas1	100	Besiktas	1	3/16/23
10130		40003	Besiktas1	100	Besiktas	3	3/16/23
10132	20001.20006	40004	Besiktas2	100	Besiktas	1	3/16/23
10132	20001	40004	Besiktas2	100	Besiktas	3	3/16/23
10131		40005	Besiktas3	500	Besiktas	1	3/16/23

2.2 Part 1: Conceptual Database Design

Your task in Part 1 is to perform the Conceptual Database Design (or ER Design) – draw ER diagrams to capture all the information, following the approach described in lectures. While there are many ER model variants, for this project, we expect you to use the ER notation from the textbook and lecture.

To receive full points for this part, you need to identify all the entity sets and relationship sets in a reasonable way. We expect there to be multiple correct solutions since the ER design is subjective. Your goal should be to reasonably capture the given information and constraints. For the entity set names, relationship set names, and attribute names that you will be using in your ER diagram, you can use the ones we have provided in Section 2.1. You can use underscores, spaces, numbers, uppercase, or lowercase letters to construct those names. It is required to use the features of ER modeling that you have learned from the lectures, including participation constraints, key constraints, weak entities, class hierarchy, and aggregation whenever necessary. We provide you with concrete sample data in Section 2.1, since these may help you understand the problem and the requirements better.

You must use computer-based / online drawing tools such as Lucidchart and diagrams.net. Handcrafted diagrams will not be accepted.

2.3 Part 2: Logical Database Design

For the second part of the project, your task is to convert the ER diagrams into relational tables, based on the set of simple rules as described in the textbook and in the lectures. You should provide the schema of each relation including the relation name, attribute names, and attribute domains.

2.4 Part 3: Write SQL DDL statements

You are required to write SQL DDL statements that create the tables that you designed in Part 2. You should specify all the constraints such as primary key, foreign key, Unique, and NOT NULL. You should turn in two files:

- createTables.sql
- dropTables.sql

Make sure that you include comments in each file.

Note: For this part, you must use a relational database management system that supports servers such as MySQL and PostgreSQL. SQLite is not allowed.

3 Submission

This project can be implemented either individually or as a team of two people. You are free to change teams in the upcoming projects.

Place all .sql files and a PDF file containing the outputs of Part 1, Part 2, and Part 3 into a folder named with the student IDs of the team members separated by an underscore (e.g. 2017400200_2018700120). Zip the folder for submission and name the .zip file with the same name. Submit the .zip file through Moodle. Only one of the team members should make the submission. However, make sure that you include the names and student IDs of both team members in the report (the pdf file).

Any other submission method is not allowed.

4 Late Submission Policy

You are allowed a total of 5 late days on the projects with no late penalties applied. You can use these 5 days as you wish. For example, you can submit the first project on time, the second project 2 days late and then the third project 3 days late. In that case, you will have to submit the fourth project on time. No late submissions for any of the project will be accepted after you use these 5 extra days. If you change your team, the team's late days used so far will be the maximum number of late days used by any of the team members.