Project Description

Board Game App

Objective : This project aims to provide you with the opportunity to design and implement a relational database and a related application. The database should be based on the provided Board Game dataset.

Composition : The project will be carried out by teams of 3 or 4 students..

Project Description

You are provided with a dataset extracted from Kaggle, composed of two files:

- Details.csv: contains the details of the set of board games
- Ratings.csv: contains the ratings of each of the games

The files can be joined using a common ***id*** column. We present in the *appendix* the attributes definitions for a better understanding:

Your ultimate purpose for this project is to design an application that exploits this database. It can be a board game center, a fan website, a board game recommendation system, or any application of your choosing. **Be creative!**

You will need to follow the project structure presented below. You will have 5 validation sessions with your professors before the final presentation, there will be an expected deliverable in each session. ***You need to validate each step with your professor during the meeting and upload the deliverable to the appropriate Moodle space at most the day before the next session.***

Project Structure

Step 1: Define Application Functionalities and Create a Mockup

Objective: Define the functionalities of your application and create a mockup of the user interface (UI). This will help you identify the entities and relationships needed for your database.

Tasks:

1. **Brainstorm application ideas**:
    - Decide on the type of application you want to build (e.g., board game center, recommendation system, fan website).
    - Define the target audience (e.g., board game enthusiasts, board game center managers).
2. **List functionalities**:
    - Identify the core functionalities of your application (e.g., search for games, rent games, recommend games, manage events).
    - Define user roles (e.g., admin, regular user) and their permissions. Make sure to have **at least two different actors**.
    - Draw a use-case diagram representing the needed functionnalities of the game.
3. **Create a mockup**:
    - Design a mockup of the UI using tools like Figma, Adobe XD, or even hand-drawn sketches.
    - Include screens for key functionalities (e.g., home page, game details page, rental page).

Deliverable:

- A document describing the application's purpose, target audience, and functionalities (with a use-case diagram).
- Mockups of the UI (at least 3 screens).

Step 2: Conceptual and Logical Database Design

Objective: Design the conceptual and logical models of your database based on the application's functionalities.

Tasks:

1. **Identify entities and relationships**:
    - List all entities (e.g., Games, Users, Rentals, Categories, Mechanics) and their attributes.
    - Define the additional entities needed for your application purpose (e.g., Rentals, inventory, comments, etc. depending on your application)
    - Define relationships between entities (e.g., a Game can have multiple Categories, a User can rent multiple Games).
2. **Create a conceptual data model (CDM)**:
    - Use an Entity-Relationship Diagram (ERD) to represent entities, attributes, and relationships.
3. **Create a logical data model (LDM)**:
    - Transform the CDM into an LDM by adding primary keys, foreign keys, and data types.
    - Normalize the database up to the **3rd Normal Form (3NF)**.
4. **Create the database schema**:
    - Write SQL scripts to create tables, define constraints (e.g., primary keys, foreign keys, NOT NULL), and establish relationships.

Deliverable:

- Conceptual Data Model (ERD).
- Logical Data Model (tables with attributes, keys, and relationships).
- Explanation of normalization steps.
- SQL scripts for database creation.

Step 3: Database Implementation and Advanced Features

Objective: Continue the implemention of the database and add advanced features.

Tasks:

1. **Populate the database**:
    - Write SQL scripts to insert sample data into the tables.
    - For this step, you may use your favorite LLM (ChatGPT, Deepseek, etc.) to generate the insertion scripts. For this purpose, you will need to write comprehensive prompts so that the LLM gives you a correct script.
    - You are not required to load the whole dataset in your database, just choose records that are somehow linked to show interesting relationships (for example, several board games of each category should be picked, so that we can see a nice visualisation)
2. **Implement advanced features**:
    - Create **views** for reporting (e.g., list of available games, user rental history).
    - Create **indexes** to optimize queries. Explain your choice of indexes.
    - Write **triggers** for automation (e.g., update inventory status when a game is rented). You need to define the computed attributes in the existing dataset and in the new entities you are adding, to automatically fill them with triggers.
    - Create **stored procedures and functions** for common operations (e.g., rent a game, return a game).

You need to create **at least three** of each object (trigger, index, etc.)

Deliverable:

- LLM prompt used for inserting sample data, and the generated insertion script.
- Documentation of triggers, stored procedures, and views.

Step 4: Application Development

Objective: Develop a basic application that interacts with the database.

Tasks:

1. **Set up the application**:
    - Choose a platform (web or desktop) and a programming language (e.g., Python, Java, PHP).
    - Connect the application to the MySQL database.
2. **Implement core functionalities**:
    - Develop at least 3 core functionalities (e.g., search for games, rent a game, view rental history).
    - Use the database's stored procedures, triggers, and views where applicable.
    - Make sure to use **transactions** in your application code to manage complex operations on the database.
3. **Test the application**:
    - Ensure the application works as expected and handles errors gracefully.

Deliverable:

- Demo of the application
- A brief report explaining how the application interacts with the database.

Step 5: Final Touches and Presentation

Objective: Finalize the project and prepare for the final presentation.

Tasks:

1. **Refine the database and application**:
    - Optimize queries and improve performance.
    - Add user authentication and role-based access control.
2. **Prepare documentation**:
    - Write a comprehensive report covering:
        - Application functionalities and UI mockups.
        - Database design (CDM, LDM, normalization).
        - Implementation details (tables, triggers, stored procedures, views).
        - Challenges faced and solutions.
3. **Prepare the presentation**:
    - Create a presentation (10-15 minutes) to demonstrate:
        - The application's functionalities.
        - The database design and implementation.
        - Key features (e.g., triggers, stored procedures, views).

Deliverable:

- Final report.
- Presentation slides.
- Demo of the application and database.

Grading Criteria

The project will be graded based on the following criteria:

Category	Weight	Details
Database Design	40%	- Correct normalization (3NF). - Proper use of PKs, FKs, and constraints. - Logical and physical data models.
Advanced SQL Features	30%	- Triggers, stored procedures, and views. - Indexes and optimization. - Proper use of transactions and locks.
Application Development	20%	- Functional application with core features. - Integration with the database.
Report and Documentation	10%	- Clear and detailed documentation. - Explanation of design choices.

Appendix

Attributes Definitions

Details.csv	
Attribute name	Description
Num	the index or serial number for the game entry in the dataset.
Id	unique identifier for each game in the database.
Primary	the name of the game.
Description	a detailed description of the game.
Yearpublished	the year the game was released or first published.
Minplayers	the minimum number of players required to play the game.
Maxplayers	the maximum number of players allowed to play the game.
Playingtime	the typical length of time required to play a full session of the game, in minutes.
Minplaytime	the minimum estimated time to complete a game session.
Maxplaytime	the maximum estimated time to complete a game session.
Minage	the minimum age recommended for players.
Boardgamecategory	lists the game's genre or thematic classification.
Boardgamemechanics	lists the gameplay mechanics that are used in the game.
Boardgamefamily	the broader themes or families associated with the game, like the components used, the regions the game represents, or any unique mechanics or settings.
Boardgameexpansion	lists the expansions or additional content for the game.
Boardgameimplementation	lists the different versions or adaptations of the game.
Boardgamedesigner	the name of the designer(s) responsible for creating the game.
Boardgameartist	the artists who contributed to the game's artwork or visual design.
Boardgamepublisher	the publishers who released the game.
Owned	the total number of copies of the game owned by people in the database.
Trading	the number of users who are currently trading this game.
Wanting	the number of people who are currently looking to acquire this game.
Wishing	the number of users who have added this game to their "wishlist" but do not necessarily own or want to trade it yet.

Ratings.csv	
Attribute name	Description
Num	the index or serial number for the game entry in the dataset.
Id	unique identifier for each game in the database.
Name	the name of the game.
Year	the year the game was released or first published.
Rank	the rank of the game based on its popularity or rating.
Average	the average rating of the game, calculated from user reviews
Bayes_average	the Bayesian average rating of the game. This is a weighted average that adjusts the score based on the number of ratings, preventing games with few ratings from having extremely high or low averages.
Users_rated	the number of users who have rated the game.
URL	the URL of the game's page on BoardGameGeek.
Thumbnail	the URL of the thumbnail image for the game.