Database for a Travel Recommendation and Review Platform

We want to design a database for a new travel recommendation and review platform. The platform aims to help users discover destinations, accommodations, attractions, and travel experiences based on reviews, ratings, and personalized recommendations.

Users will register by providing their full name, email address, and optionally their preferences such as favorite travel destinations, activities, and budget. Once registered, users can:

- Explore destinations, hotels, attractions, and travel packages, each associated with detailed descriptions, user-generated reviews, ratings (from 1 to 5), pricing, and availability.
- Submit reviews and ratings for places they have visited, including details such as date of visit, type of experience (solo, family, romantic, business), and recommendations.
- Create personalized travel itineraries that can be saved privately or shared publicly.

The platform must track the destinations visited by users, their reviews, and preferences to facilitate personalized recommendations. Recommendations can be based on factors such as similar users' ratings, common interests, popular destinations, or seasonal trends.

The database design must accommodate the following use cases:

- Users can list destinations or attractions they've rated above a certain threshold, such as places rated at least 4 out of 5.
- Users can discover new accommodations or attractions added in recent months in their favorite travel categories or regions.
- Administrators can identify inactive users (those who haven't logged in for a specified period) and retrieve their most recent travel-related activity.
- The system can recommend destinations or experiences to users based on similarities in their travel history, highly rated locations by similar travelers, or current trending destinations.

Exam requirements

The project is intended for individual students, not for team work. Therefore, each student is expected to work on the project independently.

The project is articulated on a number of tasks, all to be completed at your best. At minimum, to access the oral exam, one has to complete all tasks, and the final grade will depend on the quality of the submitted solutions and the oral exam. Submitting perfect solutions to a subset of the tasks, while not providing any solution for the remaining tasks, will not suffice to pass the exam.

The quality of solutions can range from simplistic to more sophisticated (using all the statements and more advanced functionalities illustrated during the lectures). Obviously, more sophisticated solutions will lead to a better grade.

Notice that the project requirements do not describe explicitly all the statements and the functionalities that you might use.

The solutions must not involve any external programming language.

Tasks

1. Design an Entity-Relationship Diagram using Chen's notation (Chapter 3). Additionally, provide a list of the entity types and relationship types you identified, along with the requirement(s) or use case(s) that led you to those design decisions.
2. Map the ER Diagram into a relational model by following the procedure presented in Chapter 9. For each step, write how you applied it and to which elements. Identify primary keys, foreign keys, and attributes' data domains. Provide the final relational model.
3. Create a SQL schema from the relational model, as it is shown in Chapter 6. The SQL schema must be compliant with MySQL.
4. Populate your database as much as it is reasonable. Also provide the relative CSV files and the commands to load the data in the DB.
5. Implement the use cases, by providing, for each use case, the SQL statement(s) that implements it. The SQL statements must be compliant with MySQL.

Required deliverables

(also summarized in the report template)

1. The ER Diagram
2. Mapping procedure from the ER Diagram to the relational model
3. The relational model
4. SQL schema from task 3 (that must run correctly in MySQL)
5. CSV file and relative statements to populate the DB (a script to populate the DB that must run correctly on MySQL)
6. SQL statements from task 5
7. A report that documents the design process, encountered problems, and the rationale behind the solutions provided.

To be admitted to the exam, the deliverables must be presented by email at michele.tucci@univaq.it 3 days before the exam date.