COSC 5360 Database Design

Project Description

Trip Share Database Management

Spring 2022

Project Description

In this project, you will develop ***a trip share database management system*** where authorized members can explore the world by exchanging and sharing travel information with each other. Members can post comments, rate attractions, write reviews, build itineraries, etc. The system also stores business partners' information, such as hotel booking, ticket purchasing, etc. The project is organized into four phases: conceptual design (Phase I), relational schema (Phase II), normalization (Phase III), and final report & demo (Phase IV).

The database should keep track of the following information:

- Destinations in the system are stored in a geographical hierarchy of countries, states, cities, and travel attractions. And travel attractions are further categorized into three categories: sight, restaurant, and shopping.

- Each destination has a name, description, address, rating (1.0 ~10.0), way(s) of travel, opening hours, and phone number. Further, the system records ticket price and weblink for each sight; consumption_per_capita and weblink for each restaurant.

- In the system, all countries are unique, however, some states or cities may have duplicate names in different countries. For instance, within 100-mile of Tyler, you may find a city called Paris and a Tower named Eiffel in Paris, Texas.

- Authorized members in the system can be divided into two classes: regular members and preferred members. The database stores email, username, password, ranking, address, wish-destination(s), visited-destination(s), number of followers, and number of people followed for each member. The last four attributes (wish-destination(s), visited-destination(s), number of followers, and number of people followed) should be auto-generated based on the user data.

- Members can write comments about any destination in the database. Only one comment from the same member is allowed at a time. The system stores the posting time, date, rating, and content of each comment. (No id will be created.) Make sure that the system always keeps track of who posted the comments and when they were posted. Other members in the system can like/dislike or reply to the comments.

- Once the number of followers reaches 20, a member becomes the preferred member with higher privileges, including editing the destination descriptions. The system keeps track of the modified date and the member(s) who modified it.

- Members may upload images. For each image, the database automatically generates a unique id (alphabet A−Z and digit 0−9) and saves the image link in the table.

- Any member can create a trip plan that contains a list of attractions. The system will store a unique plan id, name, purpose (food, shopping, business, family, others), start datetime, end datetime, duration, any associated members, attraction(s), and each arrival/departure datetime between attractions. The system will also estimate the potential cost for the member.

- Other members may rate the posted plan and the rating will be stored.

- Business Partners are companies that conduct commercial advertisements via the system. For each partner, the database keeps track of business name,

business type, phone number, contact person, and associated stores (restaurants or shops) in the system.

Project Questions

a) Can you think of 5 more rules (other than the one explicitly described above) that are likely to be used in the system?
b) Is the ability to model superclass/subclass relationships likely to be important in such an environment? Why or why not?
c) Justify using a Relational DBMS like Oracle for this project.

Project Phases

I. **Conceptual Design.** Answer the above questions and draw an EER diagram to accurately represent this set of requirements. Clearly specify any assumption that you are making. You can use any tools (software) to draw the EER diagram. You don't need to describe the value constraints of the attributions at this stage. (20%)
II. **Relational Schema**. Perform the following steps. (20%)
   a) Convert your Conceptual model to a Logical model that can be implemented in a relational DBMS like Oracle. During this process, you replace M-N relationships and multi-valued attributes with constructs that can be implemented in the relational DBMS. Draw EER for the logical model after your modifications. Feel free to change your conceptual model (first delivery) if needed.
   b) Convert the EER to a database design. Document your design in Database Schema format like the one we discussed in the class. Use appropriate naming conventions for all of your tables and attributes.
III. **Implementation.** (45%)
   a) Normalize all of your tables to a third normal form. Make any necessary changes to the EER. Explain why these changes needed to be made.
   b) Draw a dependency diagram for each table.
   c) Write SQL statements to create your database, tables and all other structures. Primary keys and foreign keys must be defined appropriately. The quantity constraints of the relation between the entities, which should be described in

ER diagram, are not required. Use the Create View statement to create the following views:

View 1. Retrieve all the restaurants in Tyler, TX.

View 2. Retrieve countries and the members who have visited them.

View 3. Retrieve itineraries to France.

View 4. Retrieve the country(s) and the travelers who also visited the US on the same trip (within 15 days).

View 5. Retrieve the users and the number of pictures they have uploaded individually.

View 6. Retrieve the members who posted any comments between 01/01/2021 and 1/31/2021.

d) Answer the following Queries. Feel free to use any of the views that you created in part (d).

Query 1. Retrieve the names, addresses, and estimated-prices of the 3 most popular restaurants in Tyler, TX, USA.

Query 2. Retrieve the username, status (*regular* or *preferred*), and ranking of the member who has uploaded the most pictures.

Query 3. For each country in the system, retrieve the username, address, and the number of followers of the members who live in this country and have the most followers.

Query 4. Add a new column DateOfBirth for all members along with a check constraint to check if the age is greater than and equal to 18.

Query 5. Retrieve the distinct names of the countries where its travelers also visited the US on the same trip (within 15 days).

Query 6. Retrieve the contact information of the business owner who owns the most expensive restaurant and the owner who owns the most assets in the system.

Query 7. Retrieve the names of the 5 most desirable France cities to visit.

Query 8. Retrieve the username and status of the member who either posted any comments or created any itinerary between 01/01/2021 and 1/31/2021.

Query 9. For each member who has visited all the states/provinces in France, retrieve the maximum, minimum, and average daily cost per

person, per day across all these related itineraries (**considering restaurant and attraction cost only*).

Query 10. For each country, retrieve the total number of original comments, the total number of related itineraries, the total number of members from this country, and the total number of members who have visited.

Query 11. Retrieve the names and countries of all preferred users who never visited the US.

Query 12. Retrieve the ids and emails of all preferred users who have followers from more than three countries.

IV. **Document & Demo.** Write the final term project report and demo. (15%)

a) Problem description (Copy it from the given requirement).
b) Project questions (Answer questions listed in this project).
c) EER diagram with all assumptions.
d) Relation schema after normalization. All relations must be in 3NF. The relation schema should include primary keys as well as foreign keys (if any) for all relations.
e) All requested SQL statements.
f) Dependency diagram.
g) Demo.