COMP 7115: Database Systems
Instructor: Deepak Venugopal
PROJECT - FALL 2017

1 Overview

The aim of this project is to design and implement a simple restaurant recommendation system. Students are expected to gain an understanding of all the steps involved in using a relational database for a practical application.

2 Due Date

- Final report and code is due on December 1, 2017 through ecourseware

3 Description

- Users can be identified by their email-id. Other information stored for a user includes name, date of birth and address. Each user has a star rating (1-5) indicating trustworthiness.
- A user may choose to "follow" other users.
- Each user reviews one or more restaurants.
- A review contains scores for the following items: Ambiance, Food Quality, Service, Price and Overall Experience. Each score is on a discrete scale of 1 - 5 (5 denoting the largest score). Optional free text comments are also saved.
- Each restaurant has a name and address. The same restaurant name can have multiple addresses. A restaurant belongs to exactly one of the following types: Ethnic, Fast food, Fast casual, Casual dining, Family style or Fine dining.
- A restaurant can serve multiple cuisines. Each user likes one or more cuisines.
- A restaurant accepts coupons. A coupon contains a coupon-code which is unique to a specific restaurant. The discount percentage and a specific date that the coupon is valid is stored. Users can have multiple coupons.

How to proceed?

1. Draw the ER (or EER) diagram for your system
2. Map the ER (or EER) model to a relational model
3. Normalize the relational model to 3-NF (Hint: Perhaps your model is already in 3NF)

4. Design 4 interesting[1] SQL queries for your system.

5. Implement your database and SQL queries using any existing RDBMS[2]. Connect your database to a front-end using any technology you like. Execute your queries from this front end (no fancy UI required) Some examples architectures are below.

   - MySQL database, Apache webserver, JDBC and JSP front-end
   - MySQL database, Apache webserver, ODBC and PHP front-end
   - MySQL database and python

Final Submission

A pdf of the report that contains the following items.

1. Brief introduction

2. Database design

   - An ER (or EER) diagram either hand-drawn or drawn using visio/powerpoint
   - Mapping the ER (or EER) model to a relational model
   - Normalize the model to 3-NF. If not in 3-NF explain your decision.
   - Document specific design choices or any assumptions you make. Remember that the specification given may not be complete.

3. Implementation

   - Implement your designed database using any DBMS (e.g. mysql) and load data into the database
   - Queries: Implement the SQL queries you designed through a front-end and a database connector such as JDBC/ODBC

4. Are there any limitations in your system or constraints you did not handle? How would you extend your system in the future?

   Submit all the code that you have implemented as a zip file (Please do not submit code that you did not implement but used for the project like apache code, libraries, etc.)

Demonstration

1. Each project must be demonstrated before the TA.

2. A sign-up sheet will be available later.

[1] A query that a user of this system will find useful. Typically they should have interesting joins between tables and/or aggregate functions.

[2] In case you use a non-relational system like Cassandra, that is fine as well, you still need to write SQL queries for your report but implement these queries using Cassandra

Helpful tips

1. Freeze the ER (or EER) diagram first. You don't want to change that once you begin implementation.

2. Automate several tasks by writing SQL scripts. For example, creating tables, dropping all tables in your database and inserting data into your database should be done using scripts.

3. Start working on the front-end last only after you know your database and your queries are working well.

4. Please start early!

Policies

- Please form groups of 2 for this project.

- Please do not share your code or data with other groups. Cheating will result in disciplinary action in accordance with university policies.