

COMP 7115: Database Systems
Instructor: Deepak Venugopal
PROJECT - FALL 2018

1 Overview

The aim of this project is to design and implement a simplified taxicab system (e.g. Uber). 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, 2018 through ecourseware

3 Description

Some basic requirements of the system are specified below. You can add to these requirements or make additional assumptions as needed.

- Vehicles are identified by a vehicle-id. Each vehicle of a specific type: sedan, SUV or van. A vehicle has a fixed capacity of passengers and luggage.
- Each vehicle is owned by a driver. The SSN identifies the driver, and other related attributes such as name, gender, DOB, etc. are stored.
- Customers are identified by their email-id. Other attributes of the customer such as name, gender, DOB, etc. are stored.
- Customers can request a ride and a specific vehicle is assigned to the ride. The source location name, destination location name, start time of the ride and end time of the ride is stored. The dollar amount for a ride is also stored.
- Customers can provide feedback for a ride in numerical scores for safety, customer service, cleanliness, condition of the vehicle and overall satisfaction.
- Vehicles are tracked using GPS. You can assume that there is a derived attribute that maps GPS coordinates of the vehicle to a specific location name.
- Customers are also tracked using GPS. You can assume that there is a derived attribute that maps GPS coordinates of the customer to a specific location name.
- A customer may also have one or more coupons that can be applied to a ride. Each coupon has a fixed discount percentage for the ride.

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 5 interesting¹ SQL queries that a data analyst can use in their reports. Some example queries. Drivers with highest average scores across their rides, Customers who travel most often during a specific time period, have spent the most money on rides, etc.
5. Implement your database and SQL queries using any RDBMS (mysql is preferred). 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/java command line front-end

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

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

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. Demonstrate working project to the TA.
2. I will set up a sign-up sheet later.

Grading

1. ER modeling and mapping to relational model (25 points)
2. Implementation of database in MySQL and queries (50 points)
3. Report quality (5 points)
4. Front-end (20 points)
5. You do not need to implement a Web-based UI as the front-end but are encouraged to do so. Bonus points for web-based front-end (10 points)

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.

Policies

- Please form groups of 2 for this project.
- You can discuss with other groups but please do not share your code or data with other groups. Cheating will result in disciplinary action in accordance with university policies.