CS 166 Project Description

Thursday, November 2, 2017

1 Introduction

In this project, we will model and build an internet booking system for airlines. We will use this system to track information about different airlines, the routes that they offer, passenger booking information as well as their reviews/ratings for a given flight.

The project consists of three phases: (i) requirement analysis using the ER-model, (ii) Relational schema design, and (iii) implementation. Finally, there will be a short presentation of the project to the instructors of the class at the end of the quarter.

Phase 1: ER Design

In the first phase, you will do the requirement analysis using the ER-model. Section 2 of this document contains the requirements of our internet booking system. Following these guidelines, you should generate an ER-diagram using your favorite diagram editing software. Use only the basic ER model that includes entities, relationships and attributes. Do not forget to indicate any key and participation constraints. Also, make sure to include additional documentation describing the assumptions that you made during the design process. You have to submit all your files via iLearn on the due date. Your submission should be a single zip file. Make sure to check that everything is included in your submission and it can be uncompressed without any errors.

You can make reasonable assumptions on your design, as long as:

- That you state them clearly in the documentation for this phase.
- They do not contradict the system requirements analysis we provide.

The due date for this phase is: **November 10th**.

Phase 2: Relational Schema Design

In this phase, we will provide you with a common (final) ER-diagram (so that the whole class will proceed with the same design). This final ER-diagram will be the starting point for the second phase, which involves the creation of the relational schema.

Your task in this phase will be to translate the provided ER design to a PostgreSQL relational database schema. The database schema will be in a form of a single executable SQL script (*.sql file with SQL statements). You have to submit this sql script via iLearn on the due date. The sql script should include the necessary drop statements at the beginning so it is easy to test. Check how the drop statement works and consider using the IF EXISTS statements where necessary.

In this phase, we will evaluate you for the correctness and completeness of your relational schema. You may find some constraints in the model and/or system requirement analysis that are not possible to represent or enforce in the relational schema. You may specify all these issues in the documentation for this phase. Your submission should be a single compressed file, containing all aforementioned files.

The due date for this phase is: **November 17th**.

Phase 3: Implementation

This part of the project will be performed in groups of **TWO** students. No individual submissions are allowed. Choose your partner wisely because the final evaluation is based on the group performance! In your report explicitly enumerate the tasks that each member of your group was responsible for. If one of the group members does most of the work the grade will be proportional to the effort.

Your tasks in this phase will be:

- Develop a client application using the Java Database Connector (jdbc) for psql.
- Use the client application to support specific functionality and queries for your online booking system.

In this phase, we will provide you with a create.sql script that recreates the relational schema of phase 2. You will use this schema to test and demo your application to us. Additionally, we will give you a collection of .csv files containing dummy data that are compatible with the provided relational schema. You will have to create your own .sql scripts to insert the data from the given .csv files into the database.

Finally, we will give you a skeleton code for the client application. The code will be in Java and will contain some basic functionality that will help you to communicate with the database and issue various .sql statements. You will have to implement your own code for a certain number of functions, described in more detail below:

1. **AddPassenger Function**

   Add a new passenger into the database. You should provide an interface that takes as input the information of a new passenger (i.e. passport number, full name, birth date e.t.c) and checks if the provided information are valid based on the constrains of the database schema.

2. **BookFlight Function**

   Book a flight for an existing passenger. This function will enable you to book a flight from a given origin to a given destination for an existing customer. You need to provide an interface that accepts the necessary information for booking a flight and checks if all inputs given by the user are valid based on the defined schema and the information stored in the database.

3. **TakeCustomerReview Function**

   This function will allow you, as a travel agent to note down the reviews of passengers. You should provide an interface that allows you to insert a new record of a rating for a given flight. Make sure to check for all the necessary constraints before performing the insert.

4. **ListAvailableFlightsBetweenOriginAndDestination Function**

   This function will allow you to list all available flights between two cities. A booking agent uses this information to make an informed decision when booking a given flight. You should print flight number, origin, destination, plane, and duration of flight.

5. **ListMostPopularDestinations Function**

   This function will return a list of the **k**-most popular destinations depending on the number of flights offered to that specific destination. You should print out the name of the destination city and the number of distinct flights offered to that destination. The user should provide the value of **k** during runtime.

6. **ListHighestRatedRoutes Function**

   This function will return a list of the **k**-highest rated routes based on the user ratings. You should print out the airline name, flight number, origin, destination, plane, and avg_score. The user should provide the value of **k** during runtime.

7. **ListFlightFromOriginToDestinationInOrderOfDuration Function**

   This function will return a list of a **k** flights for a given origin and destination in order of duration. You should print the airline, flight number, origin, destination, plane, and duration. The user should give the value of k during runtime.

8. **FindNumberOfAvailableSeatsForFlight**

   Find the number of empty seats for a given flight on a given date. You should print flight number, origin, destination, departure date, booked seats, total number of seats, and number of available seats.

This phase of the project is challenging, therefore we advise you to start early and allocate at least 25 hours per person to get it finished. Make sure to consider all possible scenarios for the client application and try to handle any exceptions that arise during the regular operation of your application. Do not forget that each group has to schedule a presentation to show the system running with all its functionalities to the TA. Slots for the presentation will be available online on a first come-first served basis.

For this phase you will be evaluated based on the system requirements. Your GUI and source code will also be taken into consideration in your final evaluation. Groups that implement systems with user-friendly interfaces, extra functionalities and error handling (i.e. invalid values, wrong operations, meaningful messages) will receive an extra credit. A final report about your system along with its source code has to be sent to your TA before the due date. You have to send it (documentation and final source code) using iLearn system and then send an email to your TA letting him/her know that it is available. Please keep in mind that we have already prepared a set of data, which you can load in the database once you create it.

The due date for this phase is: **December 15th**.

1.1 Grading

Your contribution to this project will be graded based on the following characteristics:

1. Phase 1 (30%)
   - Conceptual Design (ER Diagram)
2. Phase 2 (10%)
   - Logical DB Design (Relational Database Schema)
3. Phase 3 (60%). In particular, this phase will be graded as:
   - Documentation of the project including details about your assumptions (10%).
   - Implementation of SQL queries in the Client Application (40%).
   - Physical DB Design (DB performance tuning indexes) (10%).
4. Extra credit (10%)
   - Extra credit for good GUI design and interface, any dataset or schema changes/extensions, etc. (10%).

2 Requirement Analysis

1. **Airline**

An airline is responsible for offering flights to different destinations. Overall, the airline table will have the following information:

- Airline id (required)
- Airline name (required)
- Year the airline was founded, always greater than 1900 (required)
- Out of which country the airline operates (required)
- Hub airport of the airline (required)

2. **Flight/Route**

Each airline operates a certain number of routes/flights and there may exist some airlines that operate in the same route (i.e. same origin and destination). However, each route is uniquely associated with the airline that offers it. Overall, a given route/flight has the following information

- Flight number (required)
- Origin (required)
- Destination (required)
- Duration in hours, always greater than 0 and at most 24 hours (required)
- Plane Type (optional)
- Seat Number always greater than zero (required)

3. **Passenger**

A passenger is able to book flights only for the routes that are available. All available routes recur daily; therefore, for each booking we need to retain information regarding the departure date. Overall, the passenger table should retain the following information:

- Passenger id (required)
- Passport number (required)
- Full Name (required)
- Date of Birth (required)
- Country (required)

4. **Rating**

Passengers that have taken a given flight may review/rate that flight. A passenger can submit many ratings but only for different flights. Overall, a rating will have the following information:

- Rating id (required)
- Rating score in the range (0,5) (required)
- Comment (optional)