CS 425 Database Organization - Fall 2019 Course Project: Airline Flight Booking Application[*]

1 Project Timeline

The project has three deliverables. Deadlines are announced on the course Blackboard and the class. Each group will demo their application at the end of the semester. The deliverables are:

- ER-model: Each group should develop an ER-model for the application. This can be uploaded as any type of image file (please do not use esoteric formats).
- Relational schema: The second deliverable is a translation of the ER-model into a relational schema implemented as an SQL script. Besides from defining tables and constraints, this script should create indexes where appropriate. Please upload the script as a simple text file.
- Application: The last deliverable is an airline flight management application that uses the relational schema defined in the first two deliverables. This application can be either a web or desktop application.

Some of the requirements are marked as optional **bonus** requirements. You are free to not realize these requirements, but you can get extra credits by implementing them.

Every member of the group has to contribute in each phase of the project and you will be graded based on your individual contribution and on the overall project result.

2 Overview

The goal is to build an airline flight booking application. Customers can register with the application, add their payment information and personal details, and most importantly search and book flight connections.

[*] Credit: Boris Glavic, IIT

3 Data Requirements

3.1 Customer

- For each customer we should record the name of the customer, one or more addresses, credit card information. Customers are identified by their email address. A customer can have multiple credit cards and for each credit card we associate it with a payment address (one of the customers addresses). Furthermore, we record for each customer their home airport (one of the airports as described below).

3.2 Airport and Flights information

The database should record information about customers, airports, flights, prices and booking.

- **Airport**: An airport has a unique IATA location identifier (3 characters long), e.g., ORD for the O'Hare International Airport and a name. For each airport, record the country it is located in (e.g., USA), and the state (only for US and Canadian airports).

- **Airline**: An airline has a unique code (2 characters long), e.g., AA for American Airlines, a name, and a country of origin.

- **Flight**: A flight is identified by the code of the airline operating the flight, a flight number, and a date. Each flight has a departure and destination airport as well as a departure and arrival time. Furthermore, a flight has a two maximal capacity values (maximum number of seats that can be booked on this flight): one for First class seats and one for Economy class seats. For example, a flight may have 10 First class and 80 Economy class seats.

- **Price**: Each combination of flight and class (Economy or First) there exists a price. For instance, an Economy east on flight AA 66 on 2017/11/19 may cost $245 while the First class price for this flight may be $4,530. The price of a First class ticket for a flight has to be higher than the price of a Economy ticket for that flight.

- **Booking**: A flight booking is for a particular customer. A booking can consist of multiple flights. For each flight of a booking we have to record whether the seat is First of Economy class. For each flight booking we have to store which of the customer's credit cards was used to make the booking.

3.2.1 BONUS: Mileage Program

- **Mileage Program**: Airlines can have mileage programs. If a customer is registered in a mileage program from an airline, then we store a bonus mile count for the customer. For very flight from this airline the customer has booked, the customer received the number of miles of the flight as bonus miles. For that you have to either record pairwise distances between airports or lengths of flights.

4 Application Requirements

The application should support the following actions:

- A user can create an account registering with an email
- Add/modify payment/address information: a user holding an account can register/modify/delete credit cards and addresses for their account
- Search for flight connections
- Book flights

4.1 Registration

- A customer can register with an email address and name.

4.2 Payment Information and Addresses

Customers can add/modify/delete addresses and payment methods (credit cards). Addresses that are payment addresses (billing) for a credit card can not be deleted before deleting the credit card.

4.3 Search For Flight Connections

The application should allow a user to search for flight connections. The minimal information provided should be a departure and destination airport as well as a date for both the flight to the destination and the return flight. The user can select whether they want to search for a return flight or not. Additionally, the user can provide a limit for the number of connections, the maximal length (time) of the whole trip, and the price.

Only flight connections that start in the specified start and destination airports and are on the dates provided by the user should be shown. Furthermore, the connections have to fulfill all the additional requirements stated by the user. For each connection show the total price, the total length of the connection (the time difference between starting the first flight and the arrival time at the final destination), and the start and arrival time of the connection. Note that flights that are fully booked should not be considered. For each connection show the Economy class and First class price (if there are still Economy or First class seats on all the flights belonging to the connection).

Furthermore, the user can specify how results should be ordered: by price or by total flight length.

The user should be able to select a connection to retrieve additional information including all the flights that make up the connection, their start and destination airport, their length, their start and arrival time, and their price.

4.3.1 BONUS: Skyline Queries

Sometimes a user may be interested in having both flights that are both short and cheap, but they are uncertain about whether the price or length (time) of the flight is more important. In this case if is possible to apply Pareto optimization to find flight connections that are not dominated by any other connection, e.g., a connection for which no strictly better connection (both cheaper and shorter) exists. This can be modeled as follows: let's say we have a list of connections C_1 to C_n which fulfill the other criteria specified by the user. Let $Price(C)$ denote the total price of a connection and $Length(C)$ specify the total length of the connection (time from starting the first flight from the start airport of the connection until arriving at the final destination). The skyline consists of all connections C from C_1 to C_n such that does not exist a connection $C' \in \{C_1, \ldots, C_n\}$ such that $Price(C') \leq Price(C)$ and $Length(C') \leq Length(C)$ and either $Price(C') < Price(C)$ or $Length(C') < Length(C)$. Add an option for the user to specify whether only skyline flights should be shown. If this option is selected, then filter out any connections that do not belong to the skyline.

4.4 Booking Flights

From the flight search, a user can book one of the flights that are currently shown. The user has to then select a payment method and confirm the booking.

4.5 Manage Bookings

Customers can browse the bookings they have made and cancel bookings (this releases the booked seats).