

I created four new tables, which are as following:

(1) One new table is Users, which has four attributes:

username (varchar): is the primary key as no two users can have the same username

hashedPassword (VARBINARY): stores the hashed password

balance (int): the balance the users account has

salt (VARBINARY): the salt that is added to the password for one given user.

This Users table stores each user's basic information, the 'salt' here is for password verification and to help avoid storing the password string directly.

(2) Another new Table is Reservations, which has 14 attributes:

rid (int): Keeps track of all reservations by giving them an id number, not the primary key so as to incorporate indirect itineraries. (Note that reservation id is not primary key since each tuple represents one flight, and each indirect itinerary has two flights belonging to one reservation)

username (varchar): It is essential to have the username in the reservation table paying or editing other users reservations.

direct (int): To check if an itinerary is direct, 1 for direct and 0 for indirect

fid (int)

day (int): The day of departure

carrier_id (varchar): Identical to the CIDs from carrier, used to identify carriers

flight_num (varchar): Identical to the FIDs from flight, used to identify flights

origin_city (varchar): stores the origin city

dest_city (varchar): Stores the destination city

actual_time (int): Displays the duration of the trip.

price (int): To display the price of the itinerary

capacity (int): To display the number of seats/capacity of a particular itinerary.

paid (int): To see if a reservation is paid or unpaid, 1 if paid, 0 is unpaid.

cancelled (int): To see if a reservation is cancelled or not, 1 if cancelled, 0 if current.

This table stores each reservation made by the user through the application.

(3) Another new table is Capacity, which has 2 attributes:

fid (int)

seats (int): number of booked seats for that flight

This table stores the number of seats already booked in each flight, and the corresponding flight's fid.

(4) Another new table is ID, which contains 1 attribute:

rid (int)

The ID table always has a tuple, which keeps track of the latest reservation ID, is helpful when create new reservation.

I stored login status of the current session, the username that currently logged in, and the user's last search result as field.

I stored them in-memory since they change between each transaction and don't need to be retrieved across different transactions. For all other data, which we would retrieve across different sessions, I store them in database as above.

Besides, I implemented a Flight class that allows us to store data about flights from the last search more efficiently.

