Please describe and/or draw your database design. This is so we can understand your implementation as close to what you were thinking. Explain your design choices in creating new tables. Also, describe your thought process in deciding what needs to be persisted on the database and what can be implemented in-memory (not persisted on the database). Please be concise in your writeup (< half a page).

I kept the previous tables from assignment 3 (FLIGHTS, CARRIERS, MONTHS, WEEKDAYS) in the Azure database, and implemented both the Users and Reservations tables in SQL. Since both Users and Reservations had a unique ID (Primary Key), I thought it would be better if I created tables for them. Because each Reservation is unique (Primary Key rid), but multiple Reservations can still be held by each User, I implemented a Foreign Key for Reservations (username) to reference the Primary Key of Users (username).

I don't think that the Itineraries entity needed to be represented as a table because it is a combination of one or two flights. Since it doesn't have any specific unique value defining it, it was better defining it as a Java Class within the file and implemented in-memory. Each Itinerary is only relevant within each session and there is no reason to store every possible combination of two flights. Users and Reservations need to be kept past the ending of each session while Itineraries do not (and is inefficient).