

Problem Set #1 (due 9/28)

Note: In this homework, you do not need to create tables and execute queries using an actual DBMS. A written solution is sufficient.

Problem 1: Suppose you have a database modeling a very simple airline reservation system, given by the following schema:

Passenger(pID, pName, pCity, pCountry)

Airline(aID, aName, aCountry)

Airport(apCode, apCity, apCountry)

Flight(fnum, aID, deptdate, deptTime, arrTime, originAp, destinAp)

Books(pID, fnum, deptdate, price)

In this schema, each passenger is identified by aa pID, and has a name, a home city, and a home country. Airlines have an aID, a name, and a country where their headquarter is located. Airports are identified by a 3-letter airport code (e.g., JFK or LAX), and have a nearby city (e.g., New York for JFK) and a country. A city could of course have several airports. A flight has flight number (e.g., AA45 say for flight 45 run by American Airlines), an airline that operates the flight, a departure date, departure and arrival times, and origin and destination airport codes. We only model direct flights, and if a flight happens every day, there would be a separate entry for each date in the Flight table, with the same fnum but different values of deptdate. For simplicity, we assume every flight arrives on the same day it departs, and we ignore different time zones. Finally, passengers book flights; the schema does not enforce any limits on the number of tickets or seats per flight.

(a) Identify suitable foreign keys for this schema.

(b) Write statements in SQL for the following queries.

- I. List the name of any passenger who has booked a flight from Japan to the US for November 5, 2018.
- II. List the name of any passenger who has booked two flights with less than an hour transfer time, that is, they have booked a flight that departs less then one hour after their previous flight arrives.
- III. For each destination airport that can be reached directly from JFK, output the number of people who flew from JFK to that destination in 2017.
- IV. Output the name of the airline that had the largest number of passenger bookings in 2017.
- V. Output the name of any passenger who took at least two flight from Japan to the US in 2017.
- VI. Output the pIDs of any pair of passengers that took exactly the same flights in 2017. (Meaning, whenever one of them booked a flight, the other one also booked the same flight.)
- VII. Output the airport codes of any airports that can be reached from JFK using at most two flights.

(c) Write expressions in Relational Algebra for the above queries.

(d) Write either DRC or TRC queries for the above queries. Or explain the reason why you think a particular query cannot be done in DRC or TRC.

Problem 2: In this problem, you need to design a relational schema for a website that allows people who are interested in buying real estate to sign up for open houses, where they can see a property they are interested in.

In this scenario, each property is identified by an ID, and has an address consisting of a street number, street name, apartment number, city, and zipcode. In addition, we also store the neighborhood (e.g., Brooklyn Heights, Park Slope, etc.). A property also has an asking price. Each property is represented by one or more real estate agents, each identified by an ID and having a name and phone number. An open house for a property is organized by one of the real estate agents who represent the property, and has a start and end time and a date. Potential buyers need to first sign up on the web site, and we store their name and phone number. To see a property, a buyer must then make a reservation for the particular open house. (For simplicity, we assume that there is no limit on the number of people who can sign up for a particular open house.)

(a) Design a relational database schema for this website that supports the above functionality. Specify all primary and foreign key constraints, and state any assumptions you are making. You can decide which exact attributes make sense for this schema.

(b) Write SQL statements for the following queries. If your schema does not support these, you need to modify it appropriately. (For the first homework, you can use informal expression such as $\text{year(pd)} = '2017'$, where pd is a timestamp, to check if the year is 2017)

I. Output the ID of any property in Park Slope that has had open houses organized by more than 5 different agents.

II. Output the ID, street number, and street name of any property in Park Slope that has an open house on October 10, 2018.

III. Output the name of any customer who has attended an open house organized by the real estate agent with ID 61734.