

CSE 562: Project #1 (due October 5, 2006)

Submit all work in electronic form using `submit_cse562`. This is individual work.

Project

The database

You are given the following relational schema (keys underlined):

Passenger(PassId, Name, Phone)

Flight(Airline, FlightNumber, Origin, Destination, DepartureTime, ArrivalTime)

Reservation(PassId, Airline, FlightNumber, FlightDate)

Plane(Airline, FlightNumber, FlightDate, Capacity)

FlightNumber, Phone, FlightDate, DepartureTime, ArrivalTime and Capacity are nonnegative integers. Please consult the test database for the specific formats used. All the other attributes are strings.

Problem 1 (20 pts)

Identify relevant integrity constraints. Provide SQL CREATE TABLE statements to implement the above schema and the associated integrity constraints. Populate your database.

Problem 2 (50 pts)

Write the following queries in SQL, defining appropriate views and tables if necessary:

1. Find all the airlines that fly to New York but not to Buffalo.
2. For every date and airline, list the percentage of unused seats (if this percentage is not 0).
3. For every airline, list the airport with the maximum number of outgoing flights on that airline.
4. Find all the pairs of Delta flights that depart within an hour from the same airport.
5. Find the destinations that can be reached from New York within 3 hours on the same airline. Assume that a flight takes at least 1 hour and that there has to be at least 30 minutes between the flights.

Run the queries against the test database and report the results.

Problem 3 (30 pts)

Implement the following transactions using Oracle JDBC:

1. List the top 5 airlines in terms of the total number of flights.
2. Reserve a seat for a passenger, checking whether there is one still available.

Run the transactions against the test database and report the results.

You can implement the transactions as standalone Java programs or Java stored procedures.

EXTRA CREDIT

Problem 4 (25 pts)

You are given a relation with N columns of the same type. Write an SQL query that returns the tuples having the *maximum number of repetitions* of the same value. The query should have size polynomial in N .