CS 122A: Introduction to Data Management – Spring 2016, UC Irvine

Prof. Chen Li

Homework 5: More SQL (Hands-On) (100 points)

Due Date: **Thursday, May 19, 2016 11:45 PM, on EEE**

Submission

For this assignment, you need to create **a TXT file** to include **your SQL queries and results** and submit your file to the EEE dropbox. Points may be deducted if you don't follow the instructions. **Refer to the separate instruction**. SQL statements should execute correctly on MySQL. All homework assignments should have the student IDs and names of your team members. Remember that all homework assignments should be done in a group. This homework assignment should be submitted on EEE before 11:45 pm on the due date. Only one student in a group should submit the file. Everybody on the team is required to have the finally submitted version. Refer to the following table for the submission guidelines. After the 24-hour grace period, no more submission is allowed on EEE. That is, we will **not** accept assignments after that time. We will publish the solutions at that time for the next assignment. Please get all your work in on time!

Date / Time	Place	Remark
Thursday, May 19, 2016 11:45 PM	EEE dropbox	Due date
Friday, May 20, 2016 11:45 PM	EEE dropbox	24-hour grace period - **10 points will be deducted**

More SQL [40 pts]

Step 1: To continue our "SQL journey," we have provided a SQL **script** to generate and populate tables with the following schema. Make sure to run the script first:

1. **Airplane** = {registration_number:VARCHAR(10), model_number:VARCHAR(10), purchased_year:INTEGER, manufactured_year:INTEGER,capacity:INTEGER}
2. **Airport** = {IATA_code:CHAR(3), name:VARCHAR(40), airport_city:VARCHAR(20), airport_state:VARCHAR(20)}
3. **Customer** = {cid:INTEGER, ssn:CHAR(9) , gender:VARCHAR(6), email:VARCHAR(30), address_street:VARCHAR(50), address_city:VARCHAR(20), address_state:VARCHAR(20), address_zipcode:CHAR(5)}

4. **Credit_Card** = {cid:INTEGER, card_number:VARCHAR(20), expr_date:CHAR(6)}
5. **Flight** = {flight_number:VARCHAR(8), projected_departure_datetime:datetime, projected_arrival_datetime:datetime, aiplane_registration_number:VARCHAR(10), departure_airport_IATA_code:CHAR(3), actual_departure_datetime:datetime, arrival_airport_IATA_code:CHAR(3), actual_arrival_datetime:datetime}
6. **FlightAttendant** = {faid:INTEGER, phone_number:VARCHAR(20), birthdate:date, ssn:CHAR(9), job_title:VARCHAR(50), address_street:VARCHAR(50), address_city:VARCHAR(20), address_state:VARCHAR(20), address_zipcode:CHAR(5), service_year:INTEGER}
7. **Lounge** = {lid:INTEGER, location:VARCHAR(50), airport_IATA_code:CHAR(3)}
8. **MaintenanceEngineer** = {meid:INTEGER, phone_number:VARCHAR(20), birthdate:date, ssn:CHAR(9), job_title:VARCHAR(50), address_street:VARCHAR(50), address_city:VARCHAR(20), address_state:VARCHAR(20), address_zipcode:CHAR(5), skill:VARCHAR(50)}
9. **OperationStaff** = {osid:INTEGER, phone_number:VARCHAR(20), birthdate:date, ssn:CHAR(9), job_title:VARCHAR(50), address_street:VARCHAR(50), address_city:VARCHAR(20), address_state:VARCHAR(20), address_zipcode:CHAR(5), department:VARCHAR(50)}
10. **Pilot** = {pid:INTEGER, phone_number:VARCHAR(20), birthdate:date, ssn:CHAR(9), job_title:VARCHAR(50), address_street:VARCHAR(50), address_city:VARCHAR(20), address_state:VARCHAR(20), address_zipcode:CHAR(5), since:VARCHAR(50)}
11. **Customer_Reserves_Flight** = {cid:INTEGER, flight_number:VARCHAR(8), projected_departure_datetime:DATETIME, purchased_datetime:DATETIME, purchased_price:DECIMAL(7,2), quantity:INTEGER}
12. **Dish** = {lid:INTEGER, name:VARCHAR(40), price:DECIMAL(6,2)}
13. **DishOrder** = {oid:INTEGER, cid:INTEGER, lid:INTEGER, order_datetime:VARCHAR(10), total_amount:INTEGER}
14. **DishOrder_Contains_Dish** = {oid:INTEGER, lid:INTEGER, name:VARCHAR(40), quantity:INTEGER}
15. **Customer_Reserves_Flight** = {cid:INTEGER, flight_number:VARCHAR(8), projected_departure_datetime:datetime, purchased_datetime:datetime, purchased_price:decimal(7,2), quantity:INTGER}
16. **FlightAttendant_Participates_Flight** = {faid:INTEGER, flight_number:VARCHAR(8), projected_departure_datetime:datetime}
17. **MaintenanceEngineer_Maintains_Airplane** = {meid:INTEGER, Aiplane_registration_number:VARCHAR(10)}
18. **Pilot_Operates_Flight** = {pid:INTEGER, flight_number:VARCHAR(8), projected_departure_datetime:datetime}

Step 2: Write the following queries using SQL and run them on your MySQL instance to collect results.

1. [10 pts] For each Pilot, list his/her pid and duration of the maximum actual flight duration he/she has operated.

a) [7pts] SQL
b) [3pts] Results

2. [10 pts] For every Lounge, count the number of customers who have ordered from the lounge and have an American Express card. An American Express card is 15 digits long, while a Visa card is 16 digits long. Use function len() or length() to get the length of a string.
a) [7 pts] SQL
b) [3 pts] Results

3. [10 pts] Find ids of customers who have purchased from at least one lounge in every airport, and their total amount of all orders (for each customer) is above $100.
a) [7 pts] SQL
b) [3 pts] Results

4. [10 pts] List flight number and projected departure datetime of flights who have been fully booked, i.e., their total number of reservations is equal to its capacity.
a) [7 pts] SQL
b) [3 pts] Results

5. [10 pts] Currently, deleting a customer does not automatically delete the associated credit cards of the customer being deleted. Add a SQL constraint for the "Credit_card" table such that if a customer is deleted, his/her credit cards are also deleted. (We only want the statement to add the constraint, and you don't need to repeat the original "CREATE TABLE" statement.)

6. [15 pts] Write and execute a CREATE VIEW statement to create a view named Flights_offered_view that shows distinct flight numbers with their departure and destination airports. The view has the following schema:
Flights_offered_view (flight_number, departure_airport_IATA_code,
arrival_airport_IATA_code).

7. [5 pts] Can updates be performed on the view above? Justify your answer.

8. [10 pts] Write a SQL GRANT statement to give a user named “futurecustomer” read access (and only read access) to the Flights_offered_view. The user should also be allowed to give the same privilege to other users.

9. [10 pts] Create a trigger that will update the “total_amount” in the relation DishOrder whenever a dish, with its quantity, is added to that order. The trigger will increment “total_amount” by the amount “dish price * quantity”. Make sure the trigger is executed when a new row is inserted in the relation DishOrder_Contains_Dish. Write CREATE TRIGGER statement between “DELIMITER $$” and “DELIMITER;”.
DELIMITER $$

DELIMITER ;

10. [10 pts] Consider a relation scheme R(M,N,L,P,Q,R,S) with the following functional dependencies: M → N, NL → PQ, MQR→ S. Prove MLR → PS is also true.