**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**

**-----**

**progress...**

SELECT P.Pid, max(timestampdiff(second, actual\_departure\_datetime, actual\_arrival\_datetime)) as maximum\_distance FROM Pilot P, Pilot\_Operates\_Flight PO, Flight F WHERE P.pid = PO.pid AND PO.flight\_number = F.flight\_number AND PO.projected\_departure\_datetime = F.projected\_departure\_datetime group by p.pid

**-----**

**2. [10 pts] For every Lounge,n count the number of customers who have ordered from the**

**lounge and have an American Express card. A 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**

----------------------------

SELECT \*, count(D.cid)

FROM cs122a.dishorder D

where D.cid in ( SELECT CC.cid

FROM cs122a.credit\_card CC left join cs122a.customer C on CC.cid = C.cid

where length(CC.card\_number) = 15

)

Group by D.lid;

---------------------------

**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**

**--------------------------**

**progress**

select F1.cid, sum(total\_amount) as total

from (select cid, count(Distinct airport\_IATA\_Code) as c from DishOrder D, Lounge L where D.lid = L.lid group by cid) F1 , DishOrder D

where c = (select count(A.IATA\_code) from Airport A) AND D.cid = F1.cid

group by F1.cid

**-------------------------**

**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**

----------------

Combining airplane and flight to get capacity

SELECT F1.flight\_number, F1.projected\_departure\_datetime

From

(SELECT F.flight\_number, F.projected\_departure\_datetime, sum(quantity) as S, F.aiplane\_registration\_number

FROM Flight F, Customer\_Reserves\_Flight Cr

Where F.flight\_number = Cr.flight\_number

AND F.projected\_departure\_datetime = Cr.projected\_departure\_datetime

Group by F.flight\_number, F.projected\_departure\_datetime) as F1 , Airplane A

WHERE A.registration\_number = F1.aiplane\_registration\_number

AND A.capacity = F1.S

)

----------------

**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.)**

**ALTER TABLE Credit\_Card  
DROP FOREIGN KEY credit\_card\_ibfk\_1,  
ADD CONSTRAINT `credit\_card\_ibfk\_2` FOREIGN KEY (`cid`) REFERENCES `Customer` (`cid`) ON DELETE CASCADE /\* Q5 \*/;**

Alter table credit\_card

drop foreign key credit\_card\_ibfk\_1,

add constraint fk\_credit\_card\_customer

foreign key (cid)

references Customer (cid)

on delete cascade;

**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).**

**CREATE VIEW Flights\_offered\_view (flight\_number,departure\_airport\_IATA\_code, arrival\_airport\_IATA\_code)   
 AS  
 (SELECT flight\_number,departure\_airport\_IATA\_code, arrival\_airport\_IATA\_code  
 FROM Flight  
 GROUP BY flight\_number)**

**PROBABLY WRONG**

CREATE VIEW [Flights\_offered\_view] AS

SELECT DISTINCT flight\_number,   
FROM Flight

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

No since it isnt a 1 to 1 relationship. "For a view to be updatable, there must be a one-to-one relationship between the rows in the view and the rows in the underlying table.   
There are also certain other constructs that make a view un updatable."

**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.**

GRANT SELECT ON Flights\_offered\_view TO futurecustomer[GRANT OPTION]

**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 $$**

CREATE TRIGGER Update\_total\_amount

AFTER INSERT ON DishOrder\_Contains\_Dish

FOR EACH ROW

BEGIN

Update DishOrder SET total\_amount = total\_amount + Dish.price \* NEW.quantity where

New.oid = DishOrder.oid AND Dish.name = NEW.name

END;

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.**

MLR -> {MLRN}

MLR -> {MLRNPQ}

MLR -> {MLRNPQS}

MLR -> {PS}

Q.E.D.