Teori Brian Air

8a. *How can you protect the credit card information in the database from hackers?*

Encrypt the data before storing it. Keep the key

8b*. Give three advantages of using stored procedures in the database (and thereby execute them on the server) instead of writing the same functions in the frontend of the system (in for example java-script on a web-page)?*

1. Multiple programs might want to access the same information. You might want to show all departures from a specific airport in the app, website, information signs in the airport or in an api available to outside actors. Storing the procedure in the DBMS allows you to you to reuse that code everywhere.
2. Assuming the database and client are physically seperated, only the data required by the client is transmitted.
3. Incapsulation, the db-managers are free to rearrange the structure of the underlying database without any effect on the front end, as long as they also update the stored procedures. This gives more flexibility as the database ages.

9

a) -

b)No, empty set. Eftersom vi inte ännu fått en COMMIT från A så ligger inte reservation I databasen ännu.

c) The query execution is halted, waiting for A to commit their changes. After this happens B continues execution and the modification is applied.

10a

No overbooking occurs, two reservations were made for 42 passengers in total. Only one reservation can be paid and receive a ticket number due to the maximum number of passengers being 40.

10b/c

Script A	Script B
Read number of free seats If <21 then continue	
	Read number of free seats If <21 then continue
Make booking and insert into Booking	
	Make booking and insert into Booking
At this point both scripts have added rows to Booking and the Route is overbooked.	

We added a sleep statement between the free-seats check and the insertion into Booking to simulate this.

10d

Our changes:

addPayment(args)	start transaction
	addPayment(args)
	commit

This means the case in 10b/c is no longer possible as the addPayment in script A must finish completely before script B starts. Therefore, not passing the check on available seats, and therefore not overbooking. A instead deletes its reservation.

Secondary index:

We propose a secondary index on the Route table.

ARN		BER		CPH	
BER	CPH	ARN	CPH	ARN	BER
Route 11	Route 12	Route 21	Route 22	Route 31	Route 32

We anticipate many users would be interested in seeing all Routes from their desired departure airport to the desired arrival airport and then filtering through departure dates and Route codes at a later stage. A secondary index can greatly reduce execution times when extracting data based on the index values, in our case dep. and arr. airports.

Assuming all airports have equal traffic and that you are only interested in Routes from your home airport this structure narrows the search space by a factor 1/n, n being the number of total airports.