TDD12 Question Lab4
Stian Lockhart Pedersen & Felix Nyrfors

8

a) How can you protect the credit card information in the database from hackers?
   We can protect the credit card information by adding an encryption to the credit card number. This will result in the credit card numbers in the database being unreadable. To actually read the credit card numbers you will need an encryption key stored in another database which is less accessible to users and sustainers of the database. To access this database there is added security in forms of password authentication.
b) 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 webpage)?
   1. One advantage of stored procedures is that it has better performance because stored procedures only need to be compiled once and are therefore stored in the memory afterwards.
   2. Another advantage of stored procedures is the added security in the way that users can manipulate the data in the database without accessing the database. For example, a user can add a flight in our previous tables but cannot drop the actual table.
   3. The third advantage of stored procedures is the reliability. In stored procedures we always know what information we are adding to the database. We can easily verify the information we add and therefore know what is being done to the database.

9

a) Added new reservation in session A got reservation ID 3885.
b) The reservation was not visible in session B before we committed, in the beginning we only had START TRANSACTION and ADD RESERVATION. Therefore, before it was committed from A it was not visible in B.
c) When we tried to add passenger to the reservation from A in B, we got stuck loading. After a while we get the lock_timeout error message from the query in session B. This happens because session A holds the lock for the write at this reservation and has not yet committed it. Therefore, session B cannot write anything to this reservation before session A releases the lock.

10

a) Did overbooking occur when the scripts were executed? If so, why? If not, why not?
   No, there were no overbookings. This is because the first one got fully executed before the second script.
b) Can an overbooking theoretically occur? If an overbooking is possible, in what order must the lines of code in your procedures/functions be executed.
   Yes, overbooking can occur. If both scripts are in the “add passenger” and checks if there is seats left, and both have the same value one they will both execute as if it can reserve the seat both will in fact reserve the seat then subtract its amount of seats booked. This can cause overbooking.

c) Try to make the theoretical case occur in reality by simulating that multiple sessions call the procedure at the same time. To specify the order in which the lines of code are executed use the MySQL query SELECT sleep(5); which makes the session sleep for 5 seconds. Note that it is not always possible to make the theoretical case occur, if not, motivate why.
By adding sleep(5) in procedure addPayment we were able to make the case of overbooking occur. This is because both script 1 & 2 are in the else clause that adds bookings and have passed the check for available seats. But both combined do not have the possibility to add the amount of bookings.

d) We managed to lock the appropriate tables and therefore overbooking is countered:

```
LOCK TABLES booking WRITE, payment WRITE, traveller WRITE, reservation WRITE, weekly_schedule READ, d_ay READ, flight READ, route READ, y_ear READ;
CALL addPayment (@a, "Sauron",7878787878);
UNLOCK TABLES;
```

11

a) The point of secondary indexes is to be able to search for an attribute with another key then the original primary key. For example, in our database you can obtain a country by its airport code since it is the primary key. But what if you want to obtain the airport code by having the country? By adding country as a secondary index, we are eligible to obtain the airport code related to the country. There are numerous instances like this in our database but maybe the most useful one would be to add a secondary index to the credit card holder to obtain the credit card number.
ALTER TABLE payment ADD INDEX (credit_card_holder)