

LAB 4

(Parts 2 to 7 can be found on the “sql scripts” folder)

Part 8:

a)

If we store credit card information in our database as plain text and someone has access to it (a hacker, a system administrator...) the credit cards can be compromised and leaked or sold on the Internet.

To avoid this, we can encrypt the credit card information. There are many algorithms to encrypt information, but as the credit card information will be used once it's stored, we have to use a two way encryption algorithm.

A well known two way encryption algorithm is AES (with different key sizes: 128, 192 or 256 bits). AES uses the same key to encrypt and decrypt the information. This way, if we create a secure AES-256 key and encrypt all the credit card information with it, we should be safe. The key must be stored safely as if an attacker gets access to it he/she could decrypt all the encrypted information.

b)

1-. Security:

Using stored procedures can limit the access the application has to the database only allowing the necessary interactions.

2-. Performance:

The SQL server will always be faster executing stored procedures as the server has all the stored procedures in cache.

3-. Maintainability:

If a query on the database has to be changed, there is no need to change the application itself, we just have to change the stored procedure. This means that we don't need to compile the application again or force the users to download a patch.

Part 9

b)

No. As the session A is doing a transaction the changes won't be applied until the transaction is committed. And as the transaction on B started before the database changed, the information on B keeps unchanged.

c)

If we try to alter the information inserted on A, the session B gets blocked. This block continues until the transaction on A is committed, then, the changes are made on the session B but they won't be applied until it ends with a commit. This happens because the A session has a write lock on the table bookings, this lock doesn't allow to the session B to continue. The lock is freed when session A commits the changes, and with this B can continue with the task.

Part 10

a)

No. As the stored procedure `addPayment()` checks if there are enough seats on the plane, and one query has been executed slightly earlier than the other, one of the reservations always gets deleted.

b)

Yes, if two different sessions run `addPayment` at the exact same time an overbooking can be possible because they both check if there is room in the booking at the same time. If this happens, both sessions would make their booking, creating an overbooking.

c)

To make this possible we have inserted a 5 second delay after all checks have been made, just before the insert starts (`part10.sql`)

Secondary Index:

As in our database we make queries to the table users often and as this table can get really big with time, we think that an index would be really useful on this table. We usually try to find the users with the attribute “passport” so that could be a good column to put an index on so the process of finding users gets faster.