Database Management Systems

Practical Test 4B2

I. *Cars* is a table in a SQL Server database with schema *Cars[<u>CarId</u>, Brand, Model, Color, Power]*. The primary key is underlined.

CarId is the search key of the clustered index on *Cars*. The table doesn't have any other indexes.

Consider the interleaved execution below. There are no other concurrent transactions. The value of *Model* for the car with *CarId 3* is *Focus* when T1 begins execution.

Answer questions 1-3 (each question has at least one correct answer).

T1	**T2**
BEGIN TRAN	

	UPDATE Cars
	SET Model='Kuga'
	WHERE CarId=3

SELECT Model
FROM Cars
WHERE CarId=3	

	UPDATE Cars
	SET Model='Puma'
	WHERE CarId=3

COMMIT TRAN	
	COMMIT TRAN

1. T1 and T2 run under REPEATABLE READ. After the *COMMIT TRAN* statement in T2, the *Model* value for the car with *CarId* 3 is:
a. *Focus*
b. *Kuga*
c. *Puma*
d. NULL
e. None of the above answers is correct.

2. T1 runs under READ UNCOMMITTED and T2 under REPEATABLE READ. After the *COMMIT TRAN* statement in T2, the *Model* value for the car with *CarId* 3 is:
a. *Puma*
b. *Kuga*
c. *Focus*
d. NULL
e. None of the above answers is correct.

3. T1 runs under REPEATABLE READ and T2 runs under READ UNCOMMITTED. Then:
a. T2 doesn't need and exclusive lock for its first UPDATE statement.
b. T1 doesn't acquire a shared lock for its SELECT statement.
c. The *Model* value for the car with *CarId* 3 displayed from T1 is *Focus.*
d. The *Model* value for the car with *CarId* 3 displayed from T1 is *Kuga*.
e. None of the above answers is correct.

II. Create a database that manages the perfomances of the players from different teams. The entities of interest to the problem domain are: *Teams*, *Team Types*, *Sports* and *Players*. A team is characterized by name, start-up year, location and has a type. A team type also has a maximum number of members. Each team practices only one sport, that is characterized by name and description. Each player has a name, a surname, a date of birthday and a gender. A player can play for multiple teams due to a start date contract and an end date contract.

1. Write an SQL script that creates the corresponding relational data model.

2. Create a Master/Detail Form that allows one to display the teams for a given (team) type, to carry out <insert, update, delete> operations on the teams of a given (team) type. The form should have a *DataGridView* named *dgvTypes* to display the (team) types, a *DataGridView* named *dgvTeams* to display all the teams of the selected (team) type, and a button for saving added / deleted / modified teams. You must use the following classes: *DataSet, SqlDataAdapter*, *Binding Source.*

3. Create a scenario that reproduces the non-repeatable read concurrency issue on this database. Explain why the non-repeatable read occurs, and describe a solution to prevent this concurrency issue. Don't use stored procedures.

I.	1.	1p
	2.	1p
	3.	1p
II	1.	2p
	2.	2p
	3.	2p
		1 p of

Please submit a .pdf file named **Group_LastName_FirstName.pdf** (e.g., 929_Ionescu_Ana.pdf) that contains:

I. The correct answer(s) to each question (e.g., 1. A, 2. BC, 3. D).

II. The following elements:

1. The *database diagram* and the *SQL script* that creates the relational data model.
2. The *C# code* that solves the requirements and a *print screen* with the corresponding *form*. Provide *detailed comments* explaining your C# code and the design choices.
3. The *SQL script* that reproduces the *concurrency problem* and the *solution*. Provide *detailed comments* explaining your SQL code (the problem and the solution).

Please send the .pdf file to emiliapop_23@yahoo.com.