

Model for the Practical Test

I. *Customers* is a table in a SQL Server database with schema *Customers*[*CustomerID*, *FirstName*, *LastName*, *City*, *DateOfBirth*]. The primary key is underlined.

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

Consider the interleaved execution below. There are no other concurrent transactions. The value of *City* for the customer with *CustomerID* 2 is *Timisoara* when T1 begins execution.

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

T1	T2
BEGIN TRAN SELECT City FROM Customers WHERE CustomerID = 2	
	BEGIN TRAN UPDATE Customers SET City = 'Cluj-Napoca' WHERE CustomerID = 2
UPDATE Customers SET City = 'Bucuresti' WHERE CustomerID = 2	
	ROLLBACK TRAN
COMMIT TRAN	

time

1. T1 and T2 run under READ UNCOMMITTED. After the *COMMIT TRAN* statement in T1, the *City* value for the customer with *CustomerID* 2 is:

- a. *Timisoara*
- b. *Cluj-Napoca*
- c. *Bucuresti*
- d. NULL
- e. None of the above answers is correct.

2. T1 runs under READ COMMITTED and T2 under REPEATABLE READ. After the *COMMIT TRAN* statement in T1, the *City* value for the customer with *CustomerID* 2 is:

- a. *Timisoara*
- b. *Cluj-Napoca*
- c. *Bucuresti*
- d. NULL
- e. None of the above answers is correct.

3. T1 runs under REPEATABLE READ and T2 runs under READ COMMITTED. Then:

- a. T1 doesn't acquire a shared lock for its SELECT statement.
- b. T1 acquires a shared lock for its SELECT statement.
- c. T2 needs an exclusive lock for its UPDATE statement.
- d. T1 needs an exclusive lock for its UPDATE statement.
- e. None of the above answers is correct.

II. Create a database for a MiniFacebook system. The entities of interest to the problem domain are: *Users*, *Pages*, *Likes*, *Categories*, *Posts*, and *Comments*. Each user has a name, current city and date of birth. A user can like multiple pages. The system stores the date of each like. A page has a name and a category, e.g., *sports*, *movies*, *music*, etc. A category also has a category description. Users write posts and comment on existing posts. A user's post has a date, text, and number of shares. A comment is anonymous, has a text, a date, and a flag indicating whether it's a top comment for the corresponding post.

1. Write an SQL script that creates the corresponding relational data model.
2. Create a Master/Detail Form that allows one to display the posts for a given user, to carry out <insert, update, delete> operations on the posts of a given user. The form should have a *DataGridView* named *dgvUsers* to display the users, a *DataGridView* named *dgvPosts* to display all the posts of the selected user, and a button for saving added / deleted / modified posts. You must use the following classes: *DataSet*, *SqlDataAdapter*, *BindingSource*.
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
		1p of