

# DBMSs - Practical Test - SI

1h + 10min

I

Consider relation TennisPlayer[PlayerID, Name, Ranking, Country, Points] and the interleaved execution below (in SQL Server). There are no indexes on TennisPlayer and no other concurrent transactions.

There is a single row with PlayerID 1 and a single row with PlayerID 2. The Points value for the player with PlayerID 1 is 6000 when T1 begins execution. The Points value for the player with PlayerID 2 is 5000 when T1 begins execution.

Choose the correct answer(s) for multiple choice questions 1 to 3.

T1	T2
BEGIN TRAN	
	BEGIN TRAN
UPDATE TennisPlayer SET Points = Points + 200 WHERE PlayerID = 1	
	SELECT * FROM TennisPlayer WHERE PlayerID = 2
UPDATE TennisPlayer SET Points = Points - 200 WHERE PlayerID = 2	
	SELECT * FROM TennisPlayer
COMMIT TRAN	
	COMMIT TRAN

time

1

T1 and T2 run under READ UNCOMMITTED. After the COMMIT TRAN statement in T2, the Points value for the player with PlayerID 2 is:

- ☐ 5000
- ☐ 5200
- ☐ 4800
- ☐ NULL
- ☐ None of the above answers is correct.

2

T1 runs under REPEATABLE READ and T2 under READ UNCOMMITTED. After the COMMIT TRAN statement in T2, the Points value for the player with PlayerID 2 is:

- ☐ 5000
- ☐ 5200
- ☐ 4800
- ☐ NULL
- ☐ None of the above answers is correct.

T1 runs alone, in isolation under REPEATABLE READ (i.e., T2 doesn't appear in the execution above). Then:

- ☐ T1 doesn't acquire an exclusive lock for its first UPDATE statement.
- ☐ T1 needs an exclusive lock for its first UPDATE statement.
- ☐ T1 needs an exclusive lock for its second UPDATE statement.
- ☐ Exclusive locks acquired by T1 are held until T1 completes.
- ☐ None of the above answers is correct.

Create a database to track tasks for all projects in a small software development company. The entities of interest to the problem domain are: Developers, Projects, Tasks, Task Types, and Task Priorities. The company has several developers, who may work at more than one project at a time. A developer has a first name and last name. A project has a start date and an end date. Tasks are defined per project; each task is immediately assigned to a single developer who is responsible for implementing that task.

A task has a title, a description and:

- a task type (one of the following options: technical, bug or improvement);
- a status (one of the following options: started, in progress or closed);
- a task priority (critical, show-stopper, minor or trivial).

A task type has a name and description. A task priority has a name and description.

a. Write an SQL script that creates the corresponding relational data model in 3NF.

b. Create a Master/Detail Form that allows one to display the tasks for a given project, to carry out <insert, update, delete> operations on the tasks of a given project. The form should have a DataGridView named dgvProjects to display the projects, a DataGridView named dgvTasks to display all the tasks of the selected project, and a button for saving added / deleted / modified tasks. You must use the following classes: DataSet, SqlDataAdapter, BindingSource.

c. Create a scenario that reproduces the deadlock phenomenon on this database. Explain why the deadlock occurs, and describe a solution to prevent this concurrency problem. Don't use stored procedures.

I. 1	1p
2	1p
3	1p
II. a	2p
b	2p
c	2p
1p of	