Marmara University – Faculty of Engineering – Department of Computer Engineering

Fall 2017 – CSE355 Database Systems
Homework #5

(Due: 19.12.2017)

1) Consider the *Turkish Super League* database that has been e-mailed to you.

player *(playerID: int, firstName: nvarchar(25), lastName: nvarchar(25), nationality: varchar(25), birthDate: smalldatetime, age: smallint, position: varchar(25))*

team *(teamID: int, name: nvarchar(50), city: nvarchar(25))*

player_team *(playerID: int, teamID: int, season: varchar(5))*

match *(matchID: int, homeTeamID: int, visitingTeamID: int, dateOfMatch: smalldatetime, week: tinyint)*

goals *(matchID: int, playerID: int, isOwnGoal: bit, minute: tinyint)*

- Note that tables *match* and *goals* store data only for season 2013-2014.

1) [2 pts] Table creation and data insertion.

Run the following queries to create the table *transactionLog* in your database.

```
create table transactionLog (
    logID int identity(1,1) primary key,
    logTime datetime,
    logType char(1),
    beforeState nvarchar(500),
    afterState nvarchar(500),
)
```

2) [58 pts] Implement a trigger *trg_rearrange* with the followings:

- When a record is inserted into, deleted from or updated on the table *goals* (any change for *matchID*, *playerID* and/or *isOwnGoal*) and insert a relevant record into the table transactionLog.
- *transactionLog.logTime* is the time of operation.
- *transactionLog.logType* is "I" for insertion, "D" for deletion and "U" for update operation/transaction.
- *transactionLog.beforeState* is null for insertion and *transactionLog.afterState* is null for deletion. For update operation, *beforeState* is the one before the operation and *afterState* is the one after the operation.
- For the fields *beforeState* and *afterState* in table *transactionLog*, concatenate all the related fields (*matchID*, *playerID*, *isOwnGoal*, *minute*) in table *goals* and separate them by a semicolon (e.g. '306;324;0;58') and enter this data in the fields *beforeState* and *afterState*, accordingly.

3) [10 pts]

a) [5 pts] Create view playerTeam_V as follows→ List player's Name, surname and team name for all players.

b) [5 pts] Write the following query by using playerTeam_V view →
List player's Name, surname and the total number of distinct teams that they play. Results must be ordered asc according to Name and surname.

4)

[30 pts] Consider the unnormalized relation R with six attributes ABCDEF and the following functional dependencies:

AB → CDE

B → F

C → D

a) [5 pts] What is the key(s) for the relation?

b) [5 pts] What is the normal form of this relation? Explain it.

c) [20 pts] Decompose R into 3NF relations step by step if it is not in 3NF.