German University in Cairo
Media Engineering and Technology
Dr. Mervat Abouelkheir

Databases I, Winter 2022
Milestone 2

Submission: 15/12/2022 (11:59 pm)

1 Important Guidelines:

- When constructing your tables, all IDs must has an IDENTITY constraint except for the fan's national ID number.
- All alpha-numeric attributes should be defined with type varchar(20). All numeric values should be defined with type int.
- You must follow the names of the required functions, stored procedures and views. Not matching the names will result in grade loss.
- You must follow the order and data types of the parameters required for a stored procedure or function with a specific order and certain data type. Failing to follow this guideline will result in grade loss of that function/stored procedure/view.
- You must follow the order of the needed columns in case a table is required to be returned from a function or fetched from a view.
- You should do the appropriate join if you are required to implement a function/view that fetches data from different tables in your database.
- You should insert the data in the appropriate tables in your database if the required stored procedure inserts data in multiple tables in your database.
- Always think about all consequences and required actions to be done in every requirement.
- The status of any sent request can either be unhandled, accepted or rejected.
- the status of the fan (blocked or unblocked), status of the ticket (available or sold) and the status of the stadium (available or unavailable) should be represented using bit data type. For fans, 0 means blocked and 1 means unblocked. For stadiums, 0 means unavailable and 1 means available. For tickets, 0 means sold and 1 means available.
- Any compilation/syntax error in any of the requirements will result in grade loss.

2 Requirements

You are required to implement the following:

2.1 Basic Structure of the Database

a) **type**: stored procedure
name: createAllTables
input: nothing
output: nothing
description: Put the queries that creates all the tables of your database with their definition inside this procedure.

b) **type**: stored procedure
name: dropAllTables
input: nothing
output: nothing
description: Drop all tables that your database have inside this procedure.

c) **type**: stored procedure
name: dropAllProceduresFunctionsViews
input: nothing
output: nothing
description: Drop all implemented stored procedures (except this one), functions and views that you implemented in this milestone.

d) **type**: stored procedure
name: clearAllTables
input: nothing
output: nothing
description: Clear all records in all tables existing in your database.

2.2 Basic Data Retrieval

a) **type**: view
name: allAssocManagers
description: fetches the username, password, and name for all association managers.

b) **type**: view
name: allClubRepresentatives
description: fetches the username,password, name and represented club name for all club representatives.

c) **type**: view
name: allStadiumManagers
description: fetches the username, password, name and managed stadium name for all stadium managers.

d) **type**: view
name: allFans
description: Fetches the username, password, name, national id number, birth date and status (blocked or unblocked) for all fans.

e) **type**: view
name: allMatches
description: Fetches the name of the host club, the name of the guest club and the start time for all matches.

f) **type**: view
name: allTickets
description: Fetches the name of the host club, the name of the guest club, the name of the stadium that will host the match and the start time of the match for all tickets.

g) **type**: view
name: allCLubs
description: Fetches the name and location for all clubs.

h) **type**: view
name: allStadiums
description: Fetches the name ,location, capacity and status (available or unavailable) for all stadiums.

i) **type**: view
name: allRequests
description: Fetches the username of the club representative sending the request, username of the stadium manager receiving the request and the status of the request for all requests.

2.3 All Other Requirements

(i) **type**: stored procedure
name: addAssociationManager
input: varchar(20) representing a name , varchar(20) representing a user name ,varchar(20) representing a password
output: nothing
description: Adds a new association manager with the given information .

(ii) **type**: stored procedure
name: addNewMatch
input: varchar(20) representing the name of the host club , varchar(20) representing the name of the guest club, datetime representing the start time of the match and datetime representing the end time of the match
output: nothing
description: Adds a new match with the given information.

(iii) **type**: view
name: clubsWithNoMatches
description: Fetches the names of all clubs which were not assigned to any match.

(iv) **type**: stored procedure
name: deleteMatch
input: varchar(20) representing a the name of the host club and varchar(20) representing the name of the guest club
output: nothing
description: Deletes the match with the given information.

(v) **type**: stored procedure
name: deleteMatchesOnStadium
input: varchar(20) representing a name of a stadium
output: nothing
description: Deletes all matches that will be played on a stadium with the given name. The matches that have already been played on that stadium should be kept not deleted.

(vi) **type**: stored procedure
name: addClub
input: varchar(20) representing a name of a club, varchar(20) representing a location of a club
output: nothing
description: Adds a new club with the given information.

(vii) **type**: stored procedure
name: addTicket
input: varchar(20) representing the name of the host club, varchar(20) representing the name of the guest club, datetime representing the start time of the match

output: nothing
description: Adds a new ticket belonging to a match with the given information.

(viii) **type**: stored procedure
name: deleteClub
input: varchar(20) representing a name of a club,
output: nothing
description: Deletes the club with the given name.

(ix) **type**: stored procedure
name: addStadium
input: varchar(20) representing a name of a stadium, varchar(20) representing a location of a stadium, int representing a capacity of a stadium
output: nothing
description: Adds a new stadium with the given information.

(x) **type**: stored procedure
name: deleteStadium
input: varchar(20) representing a name of a stadium
output: nothing
description: Deletes stadium with the given name.

(xi) **type**: stored procedure
name: blockFan
input: varchar(20) representing a national id number of a fan
output: nothing
description: blocks the fan with the given national id number.

(xii) **type**: stored procedure
name: unblockFan
input: varchar(20) representing a national id number of a fan
output: nothing
description: Unblocks the fan with the given national id number.

(xiii) **type**: stored procedure
name: addRepresentative
input: varchar(20) representing a name ,varchar(20) representing a club name, varchar(20) representing a user name ,varchar(20) representing a password
output: nothing
description: Adds a new club representative with the given information .

(xiv) **type**: function
name: viewAvailableStadiumsOn
input: datetime
output: table
description: returns a table containing the name, location and capacity of all stadiums which are available for reservation and not already hosting a match on the given date.

(xv) **type**: stored procedure
name: addHostRequest
input: varchar(20) representing club name ,varchar(20) representing a stadium name, datetime representing the start time of a match
output: nothing
description: Adds a new request sent from the representative of the given club to the representative of the given stadium regarding the match starting at the given time which the given club is assigned to host.

(xvi) **type**: function
name: allUnassignedMatches
input: varchar(20) representing the name of a club

output: table
description: returns a table containing the info of the matches that are being hosted by the given club but have not been assigned to a stadium yet. The info should be the name of the guest club and the start time of the match.

(xvii) **type**: stored procedure
name: addStadiumManager
input: varchar(20) representing a name ,varchar(20) representing a stadium name, varchar(20) representing a user name ,varchar(20) representing a password
output: nothing
description: Adds a new stadium manager with the given information.

(xviii) **type**: function
name: allPendingRequests
input: varchar(20) representing the username of a stadium manager
output: table
description: returns a table containing the info of the requests that the given stadium manager has yet to respond to. The info should be name of the club Representative sending the request, name of the guest club competing with the sender and the start time of the match requested to be hosted.

(xix) **type**: stored procedure
name: acceptRequest
input: varchar(20) representing a username of a stadium manager ,varchar(20) representing hosting club name ,varchar(20) representing a guest club name, datetime representing the start time of a match
output: nothing
description: Accepts the already sent request with the given info.

(xx) **type**: stored procedure
name: rejectRequest
input: varchar(20) representing a username of a stadium manager ,varchar(20) representing hosting club name ,varchar(20) representing a guest club name, datetime representing the start time of a match
description: Rejects the already sent request with the given info.

(xxi) **type**: stored procedure
name: addFan
input: varchar(20) representing a name, varchar(20) representing a username, varchar(20) representing a password, varchar(20) representing a national id number, datetime representing birth date,varchar(20) representing an address and int representing a phone number
output: nothing
description: Adds a new fan with the given information .

(xxii) **type**: function
name: upcomingMatchesOfClub
input: varchar(20) representing a club name
output: table
description: returns a table containing the info of upcoming matches which the given club will play. All already played matches should not be included. The info should be the given club name, the competing club name , the starting time of the match and the name of the stadium hosting the match.

(xxiii) **type**: function
name: availableMatchesToAttend
input: datetime
output: table
description: returns a table containing the info of all upcoming matches which will be played starting from the given date and still have tickets available on sale. The info should be the host

club name, the guest club name , the start time of the match and the name of the stadium hosting the match.

(xxiv) **type**: stored procedure
name: purchaseTicket
input: varchar(20) representing the national id number of a fan,varchar(20) representing hosting club name ,varchar(20) representing the guest club name, datetime representing the start time of the match
output: nothing
description: Executes the action of a fan with the given national id number buying a ticket for the match with the given info .

(xxv) **type**: stored procedure
name: updateMatchHost
input: varchar(20) representing host lub name ,varchar(20) representing guest club name, datetime representing the start time of the match
output: nothing
description: Change the host of the given match to the guest club .

(xxvi) **type**: view
name: matchesPerTeam
description: Fetches all club names and the number of matches they have already played.

(xxvii) **type**: view
name: clubsNeverMatched
description: Fetches pair of club names (first club name and second club name) which have never played against each other.

(xxviii) **type**: function
name: clubsNeverPlayed
input: varchar(20) representing club name,
output: table
description: returns a table containing all club names which the given club has never competed against.

(xxix) **type**: function
name: matchWithHighestAttendance
input: nothing
output: table
description: returns a table containing the name of the host club and the name of the guest club of the match which sold the highest number of tickets so far.

(xxx) **type**: function
name: matchesRankedByAttendance
input: nothing
output: table
description: returns a table containing the name of the host club and the name of the guest club of all played matches sorted descendingly by the total number of tickets they have sold.

(xxxi) **type**: function
name: requestsFromClub
input: varchar(20) representing name of a stadium, varchar(20) representing name of a club
output: table
description: returns a table containing the name of the host club and the name of the guest club of all matches that are requested to be hosted on the given stadium sent by the representative of the given club.