NOVA IMS INFORMATION MANAGEMENT SCHOOL

DATABASES I/DATABASES

(2025-2026)

PROJECT: FOOTBALL STATISTICS

SPECIFICATIONS AND GUIDELINES

PART II

Description

In this phase of the final project, the groups should promote the evolution of the database that results from the first phase of the project, implementing a set of rules that make sense in the Football Statistics data model. For the starting point of the groups to be equivalent, a version of a possible representation of the data model is provided – conceptual and physical data models and the SQL script to generate the database.

Part I specification and guidelines, namely parts of the text that described rules that weren't implemented in the first phase of the project, should be considered as contextual information for Part II of the project.

The implementation of the integrity and business rules must be made using triggers complemented by other database objects (e.g., stored procedures, functions, integrity constraints) if necessary. The main goal is to produce a database enriched with a set of rules and/or objects implemented to enforce integrity and business rules and to facilitate the information extraction process.

Questions

1. Create a view that lists the players and their actual club using the information stored in the model, namely the transfers and players tables.
2. Create a function that returns a table with a list of all clubs that a player was associated with (by a transfer) using the PlayerID as an input parameter. The table to be returned must include the ClubID (ToClub in Transfers), the beginning of the contract (TransferDate) and the end of the contract (TransferDate + ContractLenght). The first contract of a player is represented by a transfer where FromClub = ToClub.
3. Implement a rule using a trigger (or triggers) such that the Actual Manager of a club is always consistent with the ClubManager record, that is:
   a) Whenever a ClubManager row is INSERTED/UPDATED it is necessary to update the Clubs table Actual Manager accordingly, i.e., the Actual Manager is updated to be the Manager for which the [StartDate, EndDate] interval includes the present date.

   b) When INSERTING a Club, the actual manager should be forced to be empty, such that it is updated as a result of an INSERT/UPDATE of the ClubManager table by the rule implementation mentioned in 3.a).

   c) It should not be possible to UPDATE the Actual Manager of Clubs table directly.

   d) A DELETE of a row in ClubManager could result in an update to NULL of the Actual Manager in Clubs if the deleted row includes the present date.

4. Implement a rule using a trigger (or triggers) to make sure that player, club and match in Lineups are consistent with the clubs participating in a match and the actual players of a club, i.e., the player must belong (at the DateOfMatch) to one of the teams participating in the match. This rule has to be implemented for INSERT and UPDATE.

5. Create a stored procedure to populate (INSERT rows) the Player Match Stats table. Your stored procedure must calculate for all matches that already finished (TotalMinutes is not NULL), for each Lineup, the stats GoalsScored, MinutesPlayed, PassesCompleted and ShotsOnTarget.

Deliverables (PART II)

1. Script with SQL code to implement the answers to the questions presented (e.g., views, triggers, functions and stored procedures) – **one (1) single text file** with ".sql" extension.
2. A text file with the identification (name and number) of all the group elements.

Guidelines (PART II)

- For the evaluation of the 2nd delivery, the code created to implement the integrity/business rules will be analyzed and their execution tested through examples created by the professors. If the objects created by the groups prevent the test samples from functioning normally, the work will be penalized. The test results will have a very significant weight in the final evaluation.
- The tests mentioned above will consider both individual record and **batch operations** in the database. Batch operations are operations (insert/update/delete) that include several rows in the same statement.
- The triggers' implementation should consider that for batch operations when in presence of mixed cases, i.e., correct rows together with incorrect rows, the DML operation (insert/update/delete) should be executed for the correct rows and not executed for the incorrect cases. If presented solutions opt to undo (rollback) all the operation when an error is detected just in part of the row set, the work will be penalized.
- Part II deadline is **December 21, 2025**. **Files delivered after the deadline will not be accepted.**