

CSCI 2141

ASSIGNMENT-3

The provided SQL script creates a simple football-related database with tables for Teams, Players, Matches, and Goals. Specifically, it includes information such as team names, player profiles (including names, birthdates, nationalities, and positions), match details (including dates and results), and goal records (including scorer and assist player information). This data is organized into tables with attributes that represent various aspects of the football world, making it suitable for database management and analysis.

Data Source: The data from the Teams and Players table are from a specific real-world data source and represents the real-life players that currently play for their teams mentioned in the tables. However, the data for the tables Matches and Goals are generated for illustrative purposes and created based on a football tournament I have personally made myself.

License Information: Once again, the data generated for Teams and Players table have specific license associated with it as the information has been sourced from an external organisation i.e. FIFA itself. So, all the Players information

followed by the Team information have specific license associated with it. On the other hand, the data for Matches and Goals tables is generated and not sourced from an external dataset as it was done out of my own interest.

Number of Tables: The database includes 4 tables :-

- 1) Teams
- 2) Players
- 3) Matches
- 4) Goals

Number of Attributes: Each table/entity has its own set of attributes. Here are all the attributes for each table.

1. Teams

- Attributes: TeamID (Primary Key), TeamName, FoundedYear, StadiumName, StadiumCapacity

2. Players

- Attributes: PlayerID (Primary Key), FirstName, LastName, DateOfBirth, Nationality, Position

3. Matches

- Attributes: MatchID (Primary Key), HomeTeamID (Foreign Key), AwayTeamID (Foreign Key), MatchDate, Result

4. Goals

- Attributes: GoalID (Primary Key), MatchID (Foreign Key), ScorerPlayerID (Foreign Key), AssistPlayerID (Foreign Key), MinuteScored

BUSINESS RULES ENFORCED BY MY DATABASE:

1) Player Age Restriction:

Business Rule: Players added should be born before 2006.

Constraint: This rule is expressed in the Players table using the CHECK constraint on the DateOfBirth column:

CHECK (DateOfBirth <= DATE '2005-12-31').

2) Result Values in Matches:

Business Rule: The result value in Matches should be limited to 'Home Win', 'Draw', or 'Away Win'.

Constraint: This rule is expressed in the Matches table using the CHECK constraint on the Result column:

CONSTRAINT valid_result CHECK (Result IN ('Home Win', 'Draw', 'Away Win')).

3) Goal Scorer Cannot Assist Himself:

Business Rule: The player who scores a goal cannot assist himself.

Constraint: This rule is expressed in the Goals table using the CHECK constraint on the ScorerPlayerID and AssistPlayerID columns:

CONSTRAINT scorer_notequal_assister CHECK (ScorerPlayerID <> AssistPlayerID).

More Information:

- The database includes data insertion for 10 teams, each with 11 players, and records of 50 matches along with corresponding goals.

- The database does not include constraints for uniqueness or non-null values on certain columns, which might be useful to ensure data integrity.
- The database attempts to handle a clean reset by dropping and recreating the database and tables if they exist.

Queries and their Purposes

Query 1: Players Info for English Outfield Players

This query retrieves information from the Players table for players who are English and play in outfield positions (not GK's). The result is ordered in ascending order based on their first names.

Query 2: Players in Teams Starting with 'Manchester'

This query performs an INNER JOIN between the Players and Teams tables, displaying information about players and their respective teams where the team's name starts with 'Manchester'. It selects specific columns from both tables for clarity.

Query 3: Players with Most Goals Scored (More than 2)

This query retrieves the first name, last name, and the goals scored for players. It uses a LEFT JOIN with the Goals table to include players who may not have scored any goals. The result is filtered to include only players with more than two goals, and it's ordered by the number of goals in descending order.

Query 4: Players from Teams Founded After 1900

This query uses a subquery in the FROM clause to find teams founded after the year 1900. It then performs an INNER JOIN with the Players table to retrieve information about players from those recently founded teams. The result includes the first name, last name, position, team name, and founding year of the team.

Query 5: Creating and Updating a View on Goals Information

This query involves multiple steps:

Step 1: Creates a view called MatchGoalView that combines information from the Players, Goals, and Matches tables, focusing on player goals.

Step 2: Runs a SELECT query on the newly created view to display information about players, matches, and goals.

Step 3: Modifies one of the underlying tables (Goals) by updating the minute at which the 3rd goal was scored.

Step 4: Re-runs the SELECT query on the view to reflect changes in the underlying tables and derived attributes. The update to the goal minute is visible in the result.

Stored Procedures Used

Stored Procedure 1: UpdatePlayerAndMatch

Procedure Purpose: This procedure updates a player's information (first name, last name, date of birth, nationality, position) and also modifies the match result if the player is part of a match.

Parameters:

playerID_param (IN): Player's ID for whom the update is performed.
newFirstName_param (IN): New first name for the player.
newLastName_param (IN): New last name for the player.
newDateOfBirth_param (IN): New date of birth for the player.
newNationality_param (IN): New nationality for the player.
newPosition_param (IN): New playing position for the player.
success_flag (OUT): An output variable indicating the success of the procedure (1 for success, 0 for failure).

Calling the procedure:

```
CALL UpdatePlayerAndMatch(  
playerID_param_value,  
'newFirstName_value',  
'newLastName_value',  
'newDateOfBirth_value',  
'newNationality_value',  
'newPosition_value',  
@success_flag );  
-- Check the success flag  
SELECT @success_flag AS UpdatePlayerSuccess;
```

Stored Procedure 2: TransferPlayer

Procedure Purpose: This procedure updates a player's team (transfers the player) and removes the player's goal contributions for both scored and assisted goals.

Parameters:

playerID_param (IN): Player's ID who is being transferred.

newTeamID_param (IN): ID of the new team to which the player is being transferred.

success_flag (OUT): An output variable indicating the success of the procedure (1 for success, 0 for failure).

Calling the procedure:

```
CALL TransferPlayer(  
playerID_param_value,  
newTeamID_param_value,  
@success_flag );  
-- Check the success flag  
SELECT @success_flag AS TransferPlayerSuccess;
```