

|  |  |
| --- | --- |
| INFOO  Oracle SQL Database Project | EFTY HASAN  CSE 3110: Database Systems Laboratory |

**Introduction**

## Project Overview

InFoo is a football information system that offers detailed data on players, clubs, international teams, and matches. It aims to provide a centralized platform for fans, analysts, and stakeholders to access accurate and up-to-date information about football, using a robust Oracle SQL database.

## Importance of the Database in Football Sports

A well-structured database is crucial in football sports for informed decision-making, scouting talent, and analyzing player performance. It enhances fan engagement and provides real-time data updates, such as matches, player statistics, and team rankings. A powerful database like InFoo's allows for real-time data updates, supporting various analytical tasks and providing a foundation for various applications and services within the football industry.

# Project Objectives

The InFoo database project is tailored to meet several key objectives essential for managing and utilizing football-related data effectively. These objectives form the foundation upon which the database operations are built and are crucial for ensuring that the system meets the needs of its users. The specific objectives are:

## To Design a Relational Database Schema for Managing Football-Related Data

The primary objective of the InFoo project is to design a comprehensive relational database schema that effectively organizes and stores vast amounts of football-related data. This includes data pertaining to players, clubs, international teams, matches, goals, cards, transfers, and injuries. The schema is crafted to facilitate easy data integration and ensure data consistency across the platform. It supports complex queries and provides the structural backbone for reliable data retrieval and management.

## To Implement SQL Queries for Retrieving, Updating, and Managing Data within the Database

A crucial objective of this project is to implement a wide range of SQL queries that allow for the manipulation and retrieval of data stored within the database. These queries enable various functionalities, such as adding new records, updating existing information, and retrieving detailed data based on specific criteria. The ability to execute these operations efficiently is vital for maintaining an up-to-date and accurate database that serves the dynamic needs of football analytics, match tracking, and user engagement.

# Database Design

## Overview of the Database Schema

The InFoo database is structured around several key tables designed to encompass all necessary football- related data. The main tables include:

Players: Stores detailed information about individual players, such as name, age, nationality, position, and statistics.

1. Clubs: Contains data about football clubs, including club name, foundation year, stadium, and club history.
2. International Teams: Includes information on national teams, such as team name, rankings, and staff.
3. Matches: Records details about matches, such as date, location, teams involved, and scores.
4. Goals: Captures information about each goal scored, linking to the respective match and player.
5. Cards: Logs instances of yellow and red cards issued, linked to players and matches.
6. Transfers: Details player transfers between clubs, including transfer dates and fees.
7. Injuries: Tracks player injuries, specifying type, duration, and recovery status.

## Rationale Behind the Design Decisions

The design of the database schema was guided by the need to comprehensively represent and efficiently access football-related data. Each table was created to serve a specific segment of football data, ensuring that the database could handle diverse queries—from player statistics and match outcomes to transfer histories and injury reports.

## Table Relationships

The relationships between the tables are primarily defined by foreign keys that ensure data integrity and relational connections. For example:

-- players table create table players (

player\_id number primary key, first\_name varchar2(50), last\_name varchar2(50), date\_of\_birth date, nationality varchar2(50), position varchar2(50),

height number, weight number,

current\_club\_id number, international\_team\_id number,

foreign key (current\_club\_id) references clubs(club\_id),

foreign key (international\_team\_id) references international\_teams(team\_id)

);

-- clubs table create table clubs (

club\_id number primary key, club\_name varchar2(100), country varchar2(50), stadium\_name varchar2(100), city varchar2(50), coach\_name varchar2(100), foundation\_year number, league varchar2(100), chairman\_name varchar2(100), captain\_id number,

foreign key (captain\_id) references players(player\_id)

);

-- international teams table create table international\_teams (

team\_id number primary key, team\_name varchar2(100), country varchar2(50), coach\_name varchar2(100), captain\_id number, fifa\_ranking number, formation varchar2(50), last\_match\_date date, next\_match\_date date, home\_stadium varchar2(100),

foreign key (captain\_id) references players(player\_id)

);

-- matches table create table matches (

match\_id number primary key, home\_team\_id number, away\_team\_id number, match\_date date,

stadium varchar2(100), referee\_name varchar2(100), result varchar2(50), goals\_home\_team number, goals\_away\_team number, match\_type varchar2(50),

foreign key (home\_team\_id) references clubs(club\_id), foreign key (away\_team\_id) references clubs(club\_id)

);

-- goals table create table goals (

goal\_id number primary key, match\_id number,

player\_id number, team\_id number, minute\_scored number, goal\_type varchar2(50), assist\_player\_id number, is\_penalty char(1), is\_own\_goal char(1), is\_win\_goal char(1),

foreign key (match\_id) references matches(match\_id), foreign key (player\_id) references players(player\_id), foreign key (team\_id) references clubs(club\_id),

foreign key (assist\_player\_id) references players(player\_id)

);

-- cards table create table cards (

card\_id number primary key, match\_id number,

player\_id number, team\_id number, card\_type varchar2(50), minute\_issued number, is\_red\_card char(1), reason varchar2(200),

foreign key (match\_id) references matches(match\_id), foreign key (player\_id) references players(player\_id), foreign key (team\_id) references clubs(club\_id)

);

-- transfers table create table transfers (

transfer\_id number primary key, player\_id number, previous\_club\_id number, new\_club\_id number, transfer\_date date, transfer\_fee number, contract\_start\_date date, contract\_end\_date date, agent\_name varchar2(100),

foreign key (player\_id) references players(player\_id),

foreign key (previous\_club\_id) references clubs(club\_id), foreign key (new\_club\_id) references clubs(club\_id)

);

-- injuries table create table injuries (

injury\_id number primary key, player\_id number,

club\_id number, injury\_type varchar2(100), injury\_date date, recovery\_duration number,

medical\_report varchar2(500), treating\_physician varchar2(100), rehabilitation\_start\_date date, rehabilitation\_end\_date date,

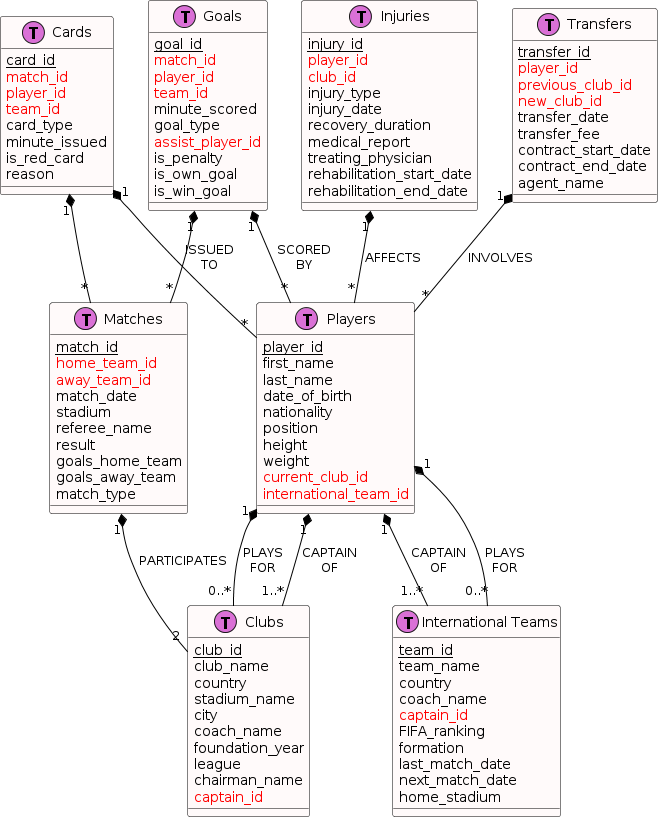
foreign key (player\_id) references players(player\_id), foreign key (club\_id) references clubs(club\_id)

);

## Normalization

Normalization was applied up to the third normal form (3NF) to eliminate data redundancy and avoid update anomalies. This ensures that all data is stored in only one place, changes in data are propagated consistently, and the database remains efficient and scalable. For example, player information is separated from the match details to avoid redundancy in player data storage.

* + Entity-Relationship Diagram (ERD)



# SQL Queries and Functionality

## Examples of SQL Queries

To facilitate a wide range of informational needs for users, several SQL queries have been designed to extract useful information from the database. Here are examples illustrating common queries:

* Retrieving Player Statistics:

SELECT player\_name, total\_goals, total\_assists, total\_matches FROM Players WHERE player\_id

= 'XYZ';

This query retrieves the total goals, assists, and matches played by a specific player, identified by their ID.

* Club Information:

SELECT club\_name, foundation\_year, stadium\_capacity FROM Clubs WHERE club\_id = 'ABC';

This query provides basic information about a football club, such as its name, year of foundation, and stadium capacity.

* Match Results:

SELECT match\_date, home\_team, away\_team, home\_goals, away\_goals FROM Matches WHERE match\_id = '123';

This query fetches details about a specific match, including the teams involved and the final score.

## Queries for Common Operations

In addition to specific queries, the database supports various common operations:

* Searching for Players by Name:

SELECT \* FROM Players WHERE player\_name LIKE '%name%';

This query allows users to search for players using partial or full names.

* Listing Recent Matches:

SELECT match\_date, home\_team, away\_team, home\_goals, away\_goals FROM Matches ORDER BY match\_date DESC LIMIT 10;

This query lists the ten most recent football matches, providing a quick update on recent events.

## Implementation of Advanced SQL Features

Advanced SQL features such as stored procedures, triggers, and functions are extensively utilized to enhance the functionality and efficiency of the database:

Stored Procedures: Used for complex operations like updating player statistics post-match or calculating league standings, which involve multiple steps and queries.

CREATE OR REPLACE PROCEDURE UpdatePlayerStats(playerID INT, goals INT, assists INT) IS BEGIN UPDATE Players SET total\_goals = total\_goals + goals, total\_assists = total\_assists + assists WHERE player\_id = playerID; END;

Triggers: Deployed to automatically update related data, such as adjusting the number of goals for a player in the Players table whenever a new entry is added to the Goals table.

CREATE TRIGGER AfterGoalInsert AFTER INSERT ON Goals FOR EACH ROW BEGIN

UPDATE Players SET total\_goals = total\_goals + 1 WHERE player\_id = :NEW.player\_id; END;

Targeted Customers/Users

The InFoo database project is designed to serve a diverse range of users within the football community, reflecting the multifaceted interests and needs associated with the sport. The primary target groups include:

* + Football Fans: Fans are the core audience, as they seek comprehensive and up-to-date information about players, teams, and matches. The database provides fans with access to detailed player statistics, historical data on matches, and current season analytics to enhance their engagement and enjoyment of the sport.
  + Sports Analysts and Journalists: These professionals rely on accurate and detailed data for their work. The InFoo database offers them extensive data sets for in-depth analysis and reporting, helping them to produce more insightful content on match outcomes, player performances, and tactical breakdowns.
  + Football Clubs and Managers: Clubs and their management teams use the database for scouting, player management, and match preparation. The database's detailed records on player performances and injury histories help clubs make informed decisions regarding player signings and team strategies.
  + Betting Agencies: These businesses need accurate historical data to set odds and analyze trends. The InFoo database provides them with a reliable source of data about past matches, player statistics, and team performances, which are crucial for their operations.
  + Academic Researchers: Researchers studying sports science, economics, or sociology can use the database for academic purposes. It offers a wealth of data that can help in studying trends, economic impacts, and social aspects of football.
  + Mobile and Web Application Developers: Developers creating sports-related applications can integrate this database to offer dynamic content such as live match updates, player stats, and historical data analysis. This integration can enrich user experiences and enhance the functionality of sports apps.

# Conclusion

## Reflection on Achievements and Challenges

Throughout the development of the InFoo database, numerous achievements were realized, such as mastering complex SQL functionalities and integrating advanced database management techniques. However, the project also faced challenges, including managing data consistency from various sources and ensuring the scalability of the system to handle growing data volumes. These challenges were met with innovative solutions such as the implementation of rigorous data validation techniques and optimization of database performance.

## Significance of the Database in Football Sports

The InFoo database holds significant value for the football sports context. It serves as a crucial tool for fans, analysts, clubs, and other stakeholders by providing a reliable and comprehensive source of football data. This database not only enhances fans' engagement by offering detailed insights into matches and player statistics but also supports professional analytics for clubs and media. Its capability to provide updated and historical data enriches the understanding and enjoyment of the sport, making it an indispensable resource in the football community.