[PROJECT 1]

[DBD281]

Gianni Snyders, Tinashe Ndawu & Leslie Khanye

Contents

[*1.* *Introduction* 2](#_Toc131636377)

[*2.* *Entity Relationship Diagram* 3](#_Toc131636378)

[*3.* *Normalisation Process* 5](#_Toc131636379)

[*4.* *Entity Identification and Explanation* 6](#_Toc131636380)

[*5.* *List of Queries* 7](#_Toc131636381)

[*6.* *Physical Database Implementation* 8](#_Toc131636382)

[**a.** **TABLES:** 8](#_Toc131636383)

[**b.** **INDEXES:** 8](#_Toc131636384)

[**c.** **CONSTRAINTS** 8](#_Toc131636385)

[*7.* *Views* 9](#_Toc131636386)

[*8.* *Stored Procedures, Triggers & Logins* 10](#_Toc131636387)

[a. Stored Procedures 10](#_Toc131636388)

[b. Triggers 10](#_Toc131636389)

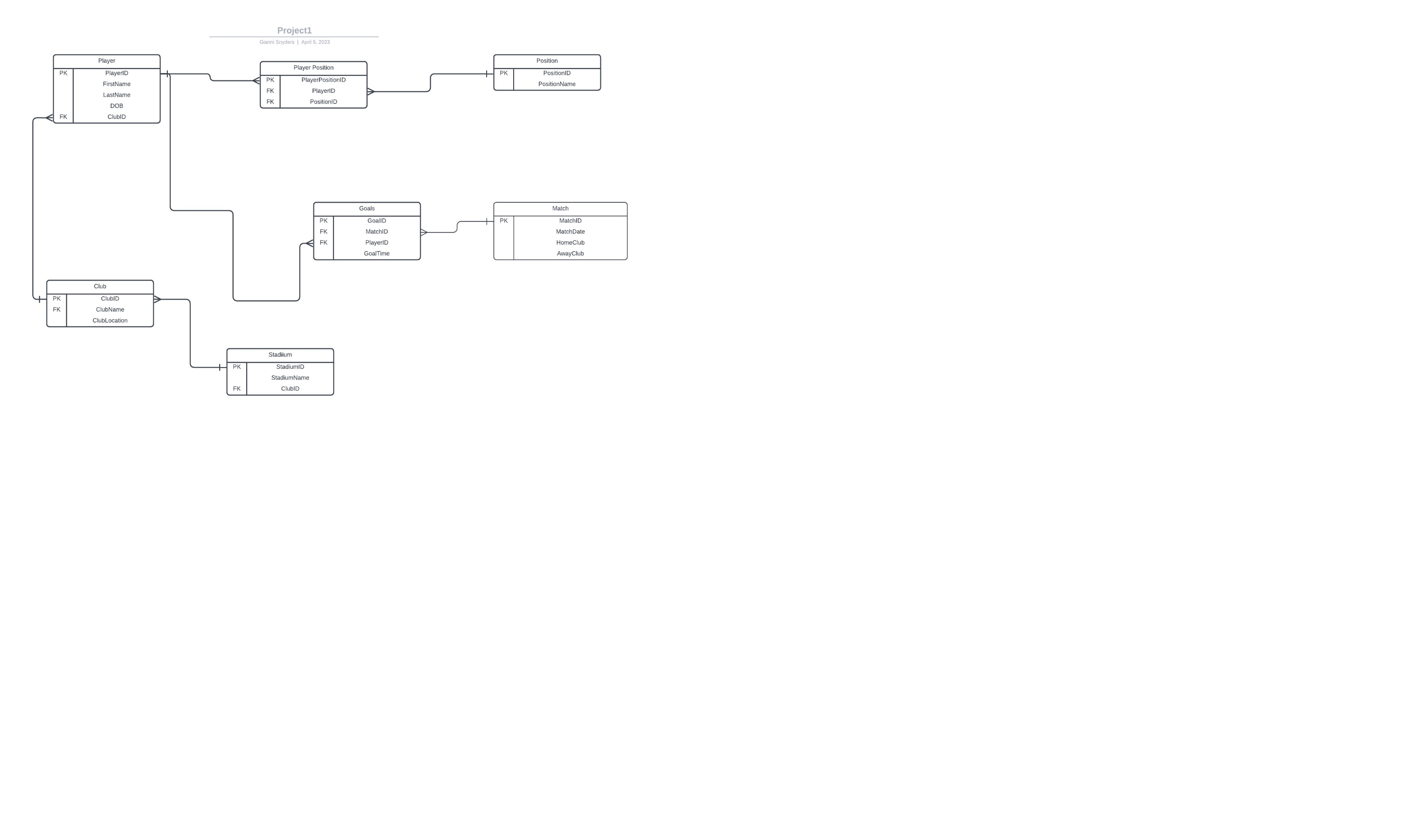
[c. Logins 10](#_Toc131636390)

# Introduction

FIFA (Fédération Internationale de Football Association) is the international governing body for association football, futsal, and beach soccer. It was founded in 1904 and is headquartered in Zurich, Switzerland.

FIFA is responsible for organizing major international soccer tournaments, including the World Cup and many other leagues and tournaments, which is the most prestigious competition in the world. FIFA needed us to create a newly updated database for them as theirs were no longer manageable and in a good working position as a database should be. We were asked to create tables for each individuals club, the players details such as their first name, what position they play in, a table named match where it stores the match date and whether they are playing home or away, a table that’s named goal to keep track of their performances and lastly a table named stadium to state whether what stadium they are playing at. FIFA were struggling to accurately track and manage players performance data, including statistics such as goals scored because of the enormous amount of information produced during matches and the need to effectively store and retrieve this information. The governing body database should be able to handle player registration and membership data, including personal information, contact details, medical history, and other relevant information. It should also provide a system for managing player documentation, such as player contracts, insurance, and other legal and administrative requirements which their previous database failed to do. Retrieval of data was also impossible, and our newly developed database will take that into consideration, and they will be able to retrieve data such as retrieving all matches played at a specific stadium, retrieving all players that plays in a specific position, all players who hasn’t scored in a specific season, etc.

# Entity Relationship Diagram



# Normalisation Process

First Normal Form (1NF): Each table has a primary key, and all attributes are atomic (indivisible).

Tables: Club, Stadium, Position, PlayerPosition, Player, Match, Goal.

Second Normal Form (2NF): All non-key attributes are dependent on the primary key and not on any subset of the primary key.

Tables: Club, Stadium, Position, PlayerPosition, Player, Match, Goal.

Third Normal Form (3NF): All non-key attributes are dependent only on the primary key and not on any other non-key attributes.

Tables: Club, Stadium, Position, Player, Match, Goal.

The normalization steps above have helped to ensure data integrity and minimize redundancy within the database. By breaking down the data into smaller, more manageable tables, we have avoided data duplication and eliminated the need for complex, error-prone update operations. This leads to a more efficient and robust database design that can easily accommodate changes and growth in the future.

# Entity Identification and Explanation

Club Table: This table stores information about soccer clubs such as the ClubID, ClubName, and ClubLocation.

Stadium Table: This table stores information about the stadiums where the matches are played. It includes the StadiumID, StadiumName, and the ClubID of the club that owns the stadium.

Position Table: This table stores information about different player positions, such as goalkeeper, defender, midfielder, and striker. It includes the PositionID and PositionName.

PlayerPosition Table: This table stores the relationship between players and their positions. It includes the PlayerID and the PositionID. It is a brigde entity.

Player Table: This table stores information about individual players, such as the PlayerID, FirstName, LastName, DateOfBirth, and ClubID.

Match Table: This table stores information about soccer matches, such as the MatchID, HomeClubID, AwayClubID, MatchDate, and StadiumID.

Goal Table: This table stores information about goals scored in a match. It includes the GoalID, PlayerID, MatchID, GoalTime, and GoalType.

# List of Queries

1. Retrieve all matches played by a specific club.
2. Retrieve all players who play in a specific position.
3. Retrieve all players who have not scored any goals in a specific season.
4. Retrieve the number of goals scored by a specific player in a specific season.
5. Retrieve all matches played in a specific stadium.

# Physical Database Implementation

## **TABLES:**

* TABLE Club
* TABLE Player
* TABLE Position
* TABLE PlayerPosition
* TABLE Match
* TABLE Goal
* CREATE TABLE Stadium

## **INDEXES:**

* Creating an index on the Club table to improve performance on queries that filter by ClubName
* Creating an index on the Player table to improve performance on queries that filter by LastName
* Creating an index on the Match table to improve performance on queries that filter by MatchDate
* Creating an index on the Goal table to improve performance on queries that filter by GoalTime
* Creating an index on the Stadium table to improve performance on queries that filter by StadiumName

## **CONSTRAINTS**

* Adding a NOT NULL constraint on ClubName in the Club table
* Adding a FOREIGN KEY constraint on AwayClubID in the Match table to reference the Club table.

# Views

1. --A view that shows the matches played by each club in the Match table--

CREATE VIEW vw\_ClubMatches AS

SELECT ClubID, COUNT(\*) AS MatchesPlayed

FROM (

SELECT HomeClubID AS ClubID

FROM Match

UNION ALL

SELECT AwayClubID AS ClubID

FROM Match

) AS ClubMatches

GROUP BY ClubID;

1. -- Returns a list of all players along with their positions--

CREATE VIEW PlayerPositions AS

SELECT Player.FirstName, Player.LastName, Position.PositionName

FROM Player

LEFT JOIN PlayerPosition ON Player.PlayerID = PlayerPosition.PlayerID

LEFT JOIN Position ON PlayerPosition.PositionID = Position.PositionID;

# Stored Procedures, Triggers & Logins

## Stored Procedures

* Returns the total number of goals scored and matches played by a specific player.
* Adds new matches to the system.

## Triggers

* Updates the club's last modified date whenever a record is updated named UpdateClubRecord1.
* automatically adds a player to a club when their club ID is updated named AddPlayerToClub.

## Logins

* We created a login to get full access.
* And we created a read only access.