DBMS MINIPROJECT

PROJECT TITLE : Football Transfer Market

SRS

UE21CS351A: Database Management System
MINI PROJECT USER REQUIREMENT SPECIFICATION

TEAM ID: G28
P N KOUSHAL KAARTHIEK: PES2UG21CS372
PARTHIV NAMBIAR: PES2UG21CS366

Table of Contents

Introduction:

- **Purpose of the project:** The main goal we aim to achieve through this project is to form a relational database management system for a football transfer market. Through this system accessing, viewing and gathering the current information about a player is made simpler while also storing all required details pertaining to transfers happening or finished amongst clubs .
- **Scope of this project:** An efficient way of storing all data pertaining to the transfer market can be achieved using this project. This program can be implemented on all levels of the competitions in football. Using a R-DBMS approach, this project will be normalized to industry grade(upto 3rd normal form) and hence, will be relatively simple to implement. The project promises user-friendliness and provides data abstraction through different levels of access and privileges. User accounts will be given different roles based on their credibility.

Project Description:

Project Overview:

This Football Transfer Market Management system is built to provide clubs with a platform to facilitate players and managers transfers along with their required information. It has the following key functionalities:

Major Project Functionalities:

1. Club admin:

Functionalities: 1. to view all the players, manager and their attributes
2. add players/manager to the club
3. remove players/manager to the club
4. renew their contracts with players/managers
5. can view the transfer market.
6. can make bids in transfer market

2.Transfer market admin: (Transfer Fees)

Functionalities: 1.Able to view available players/managers.
2. add players/managers to the market.
3. remove players/managers to the market.
4.can view bids from diff clubs but not edit.
5. Assigns estimated value to players/manager.

3.Player/Manager:

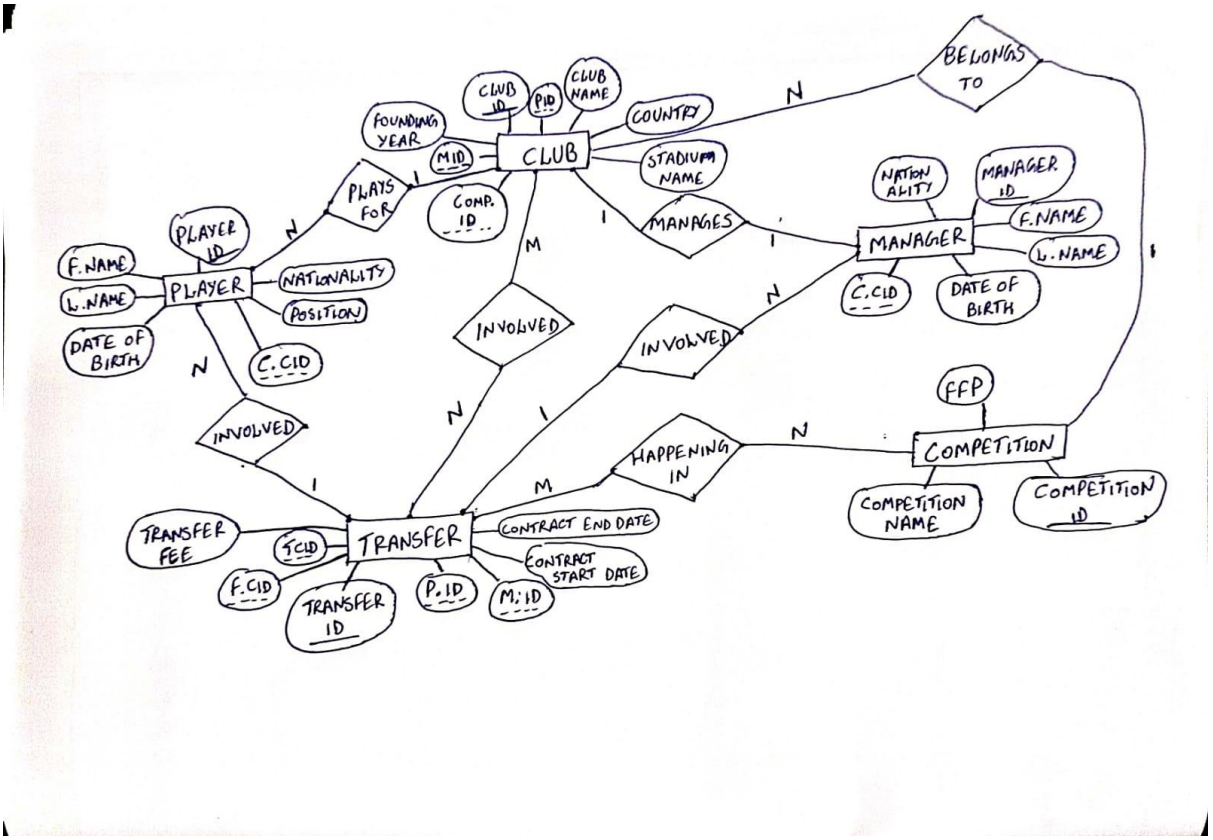
Functionalities: 1.View all the clubs on the transfermarket.
2.Can put in a transfer request to a specific club.

System Features and Functional Requirements:

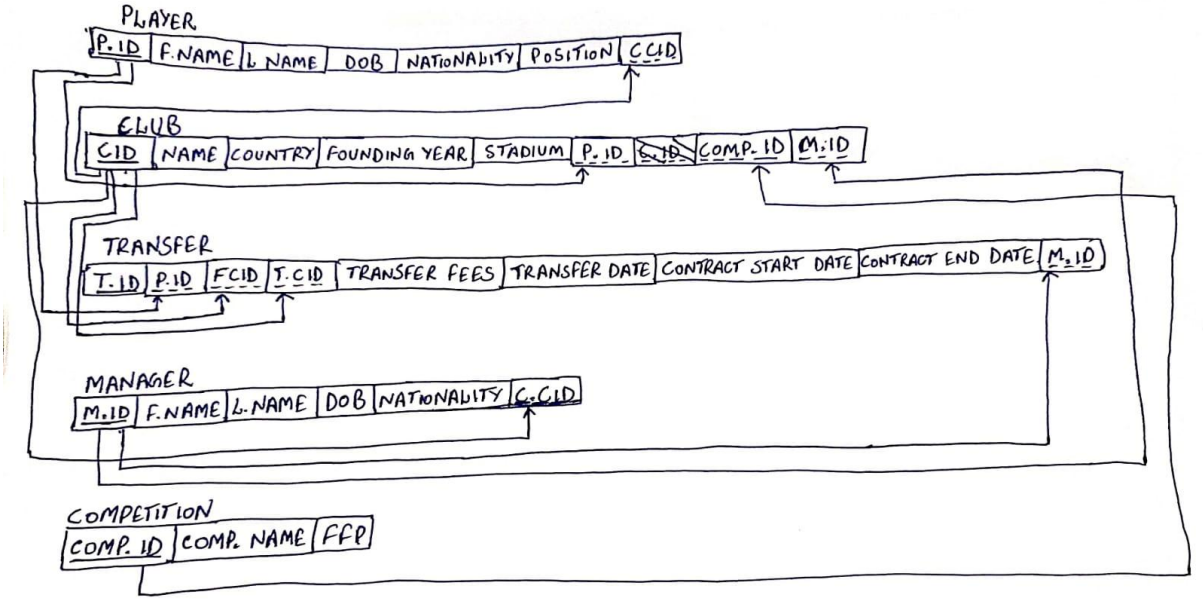
- **Initiating a Transfer:** When a transfer is agreed upon between two clubs, the admin can update the system where the player is moved from one club to another in the system and it is shown accordingly to end users with viewing accessibility only. All details related to the transfer will also be updated so as to make it convenient to fetch any information(such as transfer fee, the new club of the player and so forth) at any point of time.
  Functional requirements:
  For initiating a transfer the required entities will be Transfer which will be connected to Competition.The Players' name,ID ,their new Club ID and the Competition they have been assigned to will be the required inputs.
- **Searching based on filters:** Clubs have specific standards for both players and managers. Our system will provide a platform so that they may browse through these filters and make an informed decision based on their requirements.
  Functional requirements:
  For Clubs the required entities for searching will be Players and Managers which in turn will contain different attributes based on which they can be filtered. The input will be the name of the player or manager and their respective ID as assigned in the system.

- **Addition, Removal, Updation, Viewing of Players and Managers:** At any point of time, a club may choose to bring in new players in their squad from their youth academies or terminate their contract with that player or the player may request an official rename. In such instances, we have provided addition, removal, updation and viewing of players and managers for the club admin. Using these features manipulation of the club's data is made easier.
  Functional Requirements:
  The entities used here are clubs,players/managers as the club admin will be browsing through all the members of the clubs and actively changing certain information as per requirement. The input will be the player or manager's name and their ID.

Entity Relationship Diagram



Relational Schema



Trigger definition

Trigger Name: club_insert_trigger

```
DELIMITER //
CREATE TRIGGER club_insert_trigger
AFTER INSERT ON club
FOR EACH ROW
BEGIN
    DECLARE new_expendable_budget DECIMAL(10, 2);
    SET new_expendable_budget = NEW.budget * 0.8;
    UPDATE club
    SET expendable_budget = new_expendable_budget
    WHERE club_id = NEW.club_id;
END;
//
DELIMITER ;
```

//Add sql statement which fires this trigger

```
INSERT INTO club (club_name, founding_year, stadium_name, country, comp_id, budget)
VALUES ('Manchester United', 1878, 'Old Trafford', 'England', 2, 200000000.00);
```

Procedure definition

Procedure Name: get_club_by_country

```
DELIMITER ;;
CREATE PROCEDURE `get_club_by_country`(IN search_country VARCHAR(50))
BEGIN
   SELECT * FROM `club` WHERE `country` = search_country;
END ;;
DELIMITER ;
```

Queries – any 4 sample queries

1.Here we are selecting clubs from a specific country 'Spain'

```
SELECT * FROM club WHERE country = 'Spain';
```

```
mysql> SELECT * FROM club WHERE country = 'Spain';
+---------+--------------+--------------+-------------------+---------+---------+--------+-------------------+
| club_id | club_name    | founding_year | stadium_name      | country | comp_id | budget | expendable_budget |
+---------+--------------+--------------+-------------------+---------+---------+--------+-------------------+
|       2 | barcelona    |          1886 | Camp Nou          | Spain   |    NULL |   0.00 |              0.00 |
|       8 | FC Barcelona |          1899 | Camp Nou          | Spain   |    NULL |   0.00 |              0.00 |
|       9 | Real Madrid  |          1902 | Santiago Bernabeu | Spain   |    NULL |   0.00 |              0.00 |
+---------+--------------+--------------+-------------------+---------+---------+--------+-------------------+
3 rows in set (0.00 sec)
```

2.Select clubs with a budget greater than a certain amount

```
SELECT * FROM club WHERE budget > 5000000.00;
```

```
mysql> SELECT * FROM club WHERE budget > 5000000.00;
+---------+--------------+--------------+----------------+-----------------+---------+-------------+-------------------+
| club_id | club_name    | founding_year | stadium_name   | country         | comp_id | budget      | expendable_budget |
+---------+--------------+--------------+----------------+-----------------+---------+-------------+-------------------+
|       1 | Example Club |          1950 | Example Stadium | Example Country |       1 | 10000000.00 |        8000000.00 |
|       3 | Third Club   |          1980 | Third Stadium   | Third Country   |       3 | 12000000.00 |        9600000.00 |
|       7 | Another Club |          1965 | Another Stadium | Another Country |       2 | 15000000.00 |       12000000.00 |
|      16 | New Club 2   |          1995 | New Stadium 2   | New Country 2   |       2 |  7500000.00 |        6000000.00 |
+---------+--------------+--------------+----------------+-----------------+---------+-------------+-------------------+
4 rows in set (0.00 sec)
```

3.Retrieve Contracts Ending After 2023-12-31:

SELECT * FROM contract WHERE contract_end_date > '2023-12-31';

```
mysql> SELECT * FROM contract WHERE contract_end_date > '2023-12-31';
+-------------+---------------------+-------------------+-----+---------+-------------+---------+
| contract_id | contract_start_date | contract_end_date | ffp | comp_id | transfer_id | club_id |
+-------------+---------------------+-------------------+-----+---------+-------------+---------+
|           4 | 2023-08-30          | 2025-06-30        |   1 |       1 |           1 |    NULL |
|           5 | 2023-09-15          | 2025-06-30        |   2 |       2 |           2 |    NULL |
|           6 | 2023-08-20          | 2025-06-30        |   3 |       3 |           3 |    NULL |
+-------------+---------------------+-------------------+-----+---------+-------------+---------+
3 rows in set (0.01 sec)
```

4.Counting the number of players in each position

SELECT position, COUNT(*) as player_count

FROM player

GROUP BY position;

```
mysql> SELECT position, COUNT(*) as player_count
    -> FROM player
    -> GROUP BY position;
+--------------+--------------+
| position     | player_count |
+--------------+--------------+
| Goalkeeper   |            1 |
| New Position |            1 |
| Forward      |            5 |
+--------------+--------------+
3 rows in set (0.01 sec)
```