CPSC 3300 21SQ

Personal Database Project

Soccer Referee Organization DB

Michael Moffat

# Table of Contents

Contents

[1 Table of Contents 2](#_Toc60134489)

[3 Overview 3](#_Toc60134490)

[3.1 Purpose 3](#_Toc60134491)

[3.2 Key Users 3](#_Toc60134492)

[3.3 Key data to store 3](#_Toc60134493)

[3.4 Key Features 3](#_Toc60134494)

[4 Data Model 4](#_Toc60134495)

[4.1 Relationships 4](#_Toc60134496)

[4.1.1 One to One 4](#_Toc60134497)

[4.1.2 One to Many 4](#_Toc60134498)

[4.1.3 Many to Many 4](#_Toc60134499)

[4.2 Constraints 4](#_Toc60134500)

[4.3 Business Rules 4](#_Toc60134501)

[5 Relational Database Model (RDM) 5](#_Toc60134502)

[5.1 Tables 5](#_Toc60134503)

[5.2 Table Definition 5](#_Toc60134504)

[6 Entity Relationship Diagram (ERD) 6](#_Toc60134505)

# Overview

## Purpose

Write between 1 and 3 paragraphs describing the purpose of your database project.

Target your customer as your audience.

The intention of this project is to build a database that could be used for an organization that assigns soccer referees to local soccer matches. The database will store information on soccer matches and referees in the organization. Things like the time, location, level of match, and teams will be recorded for each match. Referees can be assigned to different types of matches based on their level of experience. Information relevant after the game such as final score and penalties given will also be held. The database will serve to facilitate the referee’s personal sign up for matches.

## Key Users

List and describe each type of user of the database by role and actions they will perform

Most databases will have a generic user, and an administrator user. Some databases have more

Role is the type of user “Administrator” “Customer”

Action is what they can do. Example Administrator Add new Categories, Delete Flights

|  |  |
| --- | --- |
| Role (Who uses it) | Action (What they can do) |
| Referee | Register for matches |
| Referee Administrator | Add and manipulate referees |
| League Administrator | Add new matches, delete matches |
| DB Administrator | Add new categories and requirements |

## Key data to store

List and describe the key data and description.

What makes the data you store interesting, special, and worth storing in a database

|  |  |
| --- | --- |
| Data | Description |
| Referees | Can be added to matches |
| Matches | Can be assigned different referees based on type |

## Key Features

Describe the key features of the system

Databases are about providing information. What information will your database provide?

Example:

FK1 List Destinations Show the list of Destinations the Airline can fly to

FK2 Manage Flights Create and manage flights the airline uses

|  |  |  |
| --- | --- | --- |
| ID | Feature | Description |
| KF1 | List matches | Show the list of matches that need (a) referee(s) |
| KF2 | Assign to match | Enter a certain referee to a certain match |
| KF3 | Manage matches | Create and alter scheduled matches |

# Data Model

## Business Rules

List Known Business Rules

A Business rule maps to what the customer describes about their business. The rule should be phrased in the way the customer would describe it. The description is the explanation of the rule. Rules are generally grouping of concepts that the customer describes together.

|  |  |  |
| --- | --- | --- |
| Rule ID | Rule | Description |
| BR1 | Referees must be able to attend the games they sign up for | A referee can only work one game at a time |
| BR2 | Teams can only play games against teams at their tier | A game must consist of two teams that are the same tier |
| BR3 | A referee must be certified to work a game | There are various tiers of matches and a certification enables a referees to sign up for matches of levels endorsed by their certification |
| BR4 | Game can only have two teams | A game must have two team and no more than two teams |
| BR5 | Each game must be one tier | The two teams in a given game must be of the same tier |
| BR6 | A team cannot play itself | Game consists of two different teams |

## Relationships

List known Relationships

Each project must have at least 2 Many to Many Table.

### One to One (Try to discover a 1:1 relationship)

### One to Many (Most tables are 1:M)

* One tier can have many teams but each team has one tier
* One certification can have many referees, but each referee has one certification

### Many to Many (You must have 2 MM tables)

* One referee can sign up for many games and some games can be refereed by multiple referees
* One certification can referee many tiers, and one level can be managed by many certifications

## Constraints

List known Constraints for data

Shall have a minimum of 3 constraints

* The score of a game must be a whole number
* Date must be after 2020
* Start time must be before end time

# Relational Database Model (RDM)

## Tables

List Key tables for the system and purpose

|  |  |
| --- | --- |
| Table | Description |
| REFEREE | Represent each referee in the system |
| GAME | Represent each scheduled match between 2 teams in the system. Records score of teams 1 and 2 |
| OFFICIATION | Bridging table between REFEREE and MATCH |
| TEAM | Represent each team in the system |
| TIER | Define the type of team/game based on age or skill of teams. Records whether assistant referees are needed for games. |
| CERTIFICATION | Represents the TIERS that a referee can be assigned too. Allows for the certification of referees to be easily changed, and ability to manage which tiers are part of which certification easily rather than assigning tiers to referees |
| ENDORSEMNT | Bridging table between CERTIFICATION and TIER |
| PLAYER | Represents the players of the teams in the system |
| RECORDING | Represents a record of an action performed by a player in a game |
| LOCATION | Represents a location where games are held |

## Table Definition

For each table list the Table, Attributes, description, data type, constraint, keys, nullable

REFEREE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| REFEREE\_ID | PK | auto increment int | Positive | No |
| CERTIFICATION\_ID | Certification of referee | int | Positive | No |
| REFEREE\_NAME\_FIRST | First name of referee | string | No more than 15 characters | Yes |
| REFEREE\_NAME\_FIRST | Last name of referee | string | No more than 15 characters | Yes |

GAME:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| GAME\_ID | Primary Key | auto increment int | Positive | No |
| TIER\_ID | Level of match Determines which referees can be assigned to match | int | Positive | No |
| GAME\_TEAM\_1 | One of 2 opponents in the game (FK) | int | Positive | no |
| GAME\_TEAM\_2 | One of 2 opponents in the game (FK) | int | Positive | no |
| GAME\_START\_TIME | Start time of match | DATETIME | 00:00(a/p)m | No |
| GAME\_END\_TIME | End time of match | DATETIME | 00:00(a/p)m | No |
| GAME\_LOCATION | Location of game | int | No more than 50 characters | no |
| GAME\_TEAM1\_GOALS | Goals scored by team 1 | int | Less than 100 | Yes |
| GAME\_TEAM2\_GOALS | Goals scored by team 2 | int | Less than 100 | Yes |

OFFICIATION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| GAME\_ID | game assigned to referee | int | Positive | No |
| REFEREE\_ID | Referee assigned to game | int | Positive | No |

TEAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| TEAM\_ID | PK | int | Positive | No |
| TEAM\_NAME | Name of team | String | No more the 50 characters | No |
| TIER\_ID | Level of team | int | Positive | No |

PLAYER:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| PLAYEER\_ID | PK | int | Positive | No |
| TEAM\_ID | Player team | Int | positive | yes |
| PLAYER\_FIRST\_NAME | Player name | string | No more than 15 characters | No |
| PLAYER\_LAST\_NAME | Player name | string | No more than 15 characters | No |
| PLAYER\_SUSPENDED | Indicates if the player is not allowed to play in games | Boolean | True/false | no |

RECORDING:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| RECORDING\_ID | PK | Auto increment int | positive | No |
| GAME\_ID | Game of recoding | Int | Positive | No |
| PLAYER\_ID | Player who recoding is about | Int | Positive | NO |
| RECORDING\_TYPE | What is being recorded | String | “Goal” or “Yellow Card” or “Red Card” | no |

TIER:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| TIER\_ID | PK | int | Positive, less than 10^3 | No |
| TIER\_NAME | Name of tier | String | No more the n50 characters |  |
| TIER\_ASSISTANTS\_NEEDED | Whether or not an assistant referee in needed for matches of this level | Boolean | True / False | No |
| TIER\_GAME\_DURATION\_MINUTES | Minutes per game | Int | Positive, less than 90 | no |

CERTIFICATION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| CERTIFICATION\_ID | PK | int | Positive | No |
| CERTIFICATION\_NAME | Names the level of certification | String | No more than 50 characters | no |

ENDORSEMENT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| CERTIFICATION\_ID | Certification for a group of levels | int | Positive | No |
| TIER\_ID | Level that can work certification | int | Positive | No |

LOCATION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Type | Constraint | Nullable |
| LOCATION\_ID | PK | Auto increment int | positive | No |
| LOCATION\_TITLE | General Name of location | String | No more than 50 characters | No |
| LOCATION\_ADDRESS | Address of location | String | Nom more that 150 characters | Yes |
| LOCATION\_FIELD\_NUIMBER | Specific field location at general location | Int | No more than 20 | yes |

# Entity Relationship Diagram (ERD)

Using Crows foot notation, diagram the entire Conceptual Model

You may need to adjust the model to fit to the document betteAdd Cardinality, add Strength to the ERD



# Queries

Give examples of an interesting queries your database will answer?

**What games need to be assigned referees*?*** *i.e. what games have no referee, are in a tier that needs assistants and does not have at least 1 assistant or has an assistant but not a center.*

**What Players have received a red card or are suspended?** *This list may be used to know what players are not allowed to play in any further games.*

**How many referees are certified for each tier?** *Useful assessing if the assignment of tiers to certifications provides an acceptable distribution of responsibility across the tiers and assessing the capacity for handling a large increase in games for particular tiers.*

**What is the maximum number of goals scored by a player on each team that has scored?**  *Might be reported to the soccer league at the end of the season to give out awards.*

## Select Query with Inner Join across at least three tables

What query will you have that joins across at least 3 tables?

How many referees are certified for each tier?

## Select Query with Left Outer Join finding missing data from the left table

What query will you have that looks for missing data?

*What games need to be assigned referees?*

What tables will it join?

*Referee, officiation, game, tier*

What is the data null data you will look for?

*NULL OFFICIATION.REFEREE IDs*

*(only games without any referee will be found using the outer join method; the other cases don’t depend on the outer join)*

# Views

Your project must have at least 1 view that de-normalizes data as part of a select from the view

All data return in the result must come exclusively from views

## SQL to create the View

Describe at least one View you will need that denormalizes the data for your final report

* View all the games including the teams, location, referee(s), and score
* View all players including their team, tier of team and whether they are suspended
* View all certifications and all tiers for each certification as well as whether assistants are needed for the tier
* View all recordings including games in which they occurred and players
* View all locations including field numbers and addresses

# Stored Procedures

All data must be inserted exclusively with Stored Procedures using natural values (not IDs)

## SQL for Stored Procedure

List the names of the stored procedures to add data with the parameters

TIER\_ADD (TIER\_NAME, TIER\_ASSISTANTS\_NEEDED, TIER\_GAME\_DURATION\_MINUTES)

CERTIFICATION\_ADD (CERTIFICATION\_NAME)

INDORSEMNT\_ADD (CERTIFICATION\_NAME, TIER\_NAME)

TEAM\_ADD (TEAM\_NAME, TIER\_NAME)

PLAYER\_ADD (PLAYER\_FIRST\_NAME, PLAYER\_LAST\_NAME, TEAM\_NAME, PLAYER\_SUSPENDED)

REFEREE\_ADD (REFEREE\_FIRST\_NAME, REFEREE\_LAST\_NAME, CERTIFICATION\_NAME)

LOCATION\_ADD (LOCATION\_TITLE, LOCATION\_ADDRESS, LOCATION\_FIELD\_NUMBER)

GAME\_ADD (TIER\_NAME, TEAM\_1\_NAME, TEAM\_2\_NAME, GAME\_START\_TIME, GAME\_END\_TIME, TEAM\_1\_SCORE, TEAM\_2\_SCORE)

OFFICIATION\_ADD (TEAM\_1\_NAME, TEAM\_2\_NAME, GAME\_START\_TIME, OFFICIATION\_POSITION)

RECORDING\_ADD (TEAM\_1\_NAME, TEAM\_2\_NAME, GAME\_START\_TIME, PLAYER\_NAME\_FIRST, PLAYER\_NAME\_LAST, RECORDING\_TYPE)

# Wrapping Stored Procedures (Create and Show Result)

Your database will wrap all commands inside two stored procedures.

F) Data Result:  You will create a single Stored Procedure named:SP\_SHOW\_RESULT that will run all the proper select queries exclusively from your Views to generate a data visualization that matches the interesting features you database supports

## SP\_CREATE\_DATA

This single Stored Procedure will call other stored procedures to do the following:

SP\_CREATE\_DATA

Drop Tables: SP\_DROP\_TABLES

Create Tables: SP\_CREATE\_TABLES

Create Views: SP\_CREATE\_VIEWS

Populate Data: SP\_POPULATE\_DATA

## SP\_SHOW\_RESULT

This Stored Procedure will return results from your interesting views. It will not call any tables directly only views.

SP\_SHOW\_RESULT

SELECT \* from VIEW\_ABC;

SELCT \* FROM VIEW\_XYZ;

# Final Project Requirements

## Document Requirements

1. Document must be well written, formatted, and professional
2. Your system must have a minimum of 6 required tables in 4FN form
3. Must have at least two M:M relationships
4. Must use Views for all data returned in the final result

## SQL Scripts

A) SQL Script: You will submit fully functioning SQL script for the Database design. All commands must be wrapped inside a stored procedures. The overall stored procedure that will execute the commands shall be named: SP\_CREATE\_DATA

B) Data Views:  You will create Views that will show the denormalized data from the proposal document using natural values. (Mike instead of Person\_ID etc.) These views should be interesting views, and must be used in your report for Data Result (F), and the only way data is gathered for Data Result (F)

C) Stored Procedures: You will create Stored Procedures that will Insert and Update data in all tables using natural values (Mike instead of Person\_ID etc.)

D) Data Population:  You will Populate all data in the database exclusively using Stored Procedures. Each table must have at least two rows, and all attributes populated.

E) Calling Data population: You will create a stored procedure that populates all the data.  This stored procedure will call all other data population stored procedures using only natural values.

F) Data Result:  You will create a single Stored Procedure named:SP\_SHOW\_RESULT that will run all the proper select queries exclusively from your Views to generate results

G) All SQL Scripts will run to completion, with no errors

H) All SQL Scripts will contain comments for documentation

## Grading of SQL

Grading of the execution of the SQL will be the following procedure

1. Open the single SQL file
2. Execute the entire loaded SQL
3. Run SP\_CREATE\_DATA
4. Run SP\_SHOW\_RESULT
5. Review the SQL