**Principles of Database Design   
Milestone 2 (Team Lake)**

Kyungchan Im, Aaron Galicia, Levi Guerengomba

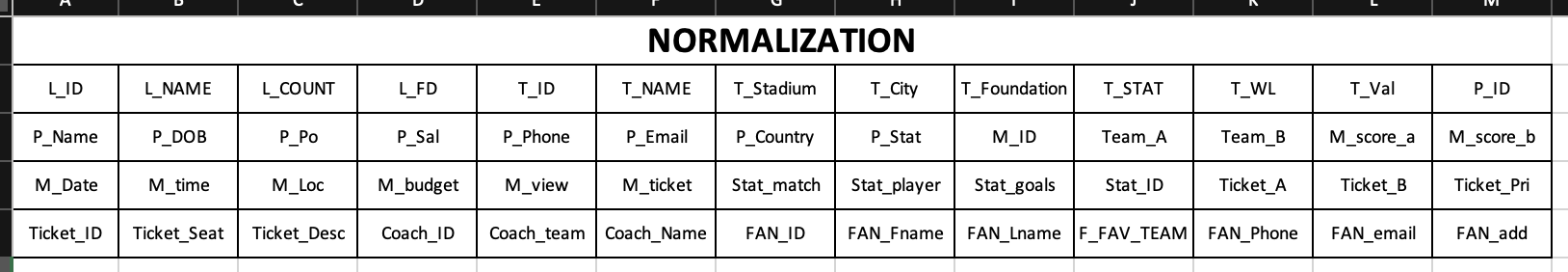
1. An explanation of the data types used for each attribute in your entities.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Explanation** |
| LEAGUE\_NAME | Varchar (50) | League Name is a character value |
| LEAGUE\_ID | INT | League ID is an int to give each its own primary key |
| LEAGUE\_Country | Vartchar(50) | League Country is a character value to tell us where the league is from |
| LEAGUE\_Foundation\_Date | INT | League Foundation Date is a an int to let us know when the league was founded |
| TEAM\_ID | INT | Team ID is an int to give each team its own primary key |
| TEAM\_Name | Varchar(50) | Team Name is a character value to identify each team |
| TEAM\_Stadium | Varchar(50) | Team Stadium is a character value to let us know where the team plays |
| TEAM\_City | Varchar(50) | Team City is a character value to let us know where the team is located |
| Team\_Foundation | INT | Team Foundation Date is an int to let us know wjeen the tea, was founded |
| PLAYERS\_ID | INT | This is an int primary key to identify every single player |
| PLAYER\_DOB | INT | This is an int to let us kniow whne the player was born |
| PLAYER\_Name | Varchar(50) | This is a character value to let us know the Player Name |
| PLAYER\_POS | Varchar(50) | This is a character value to let us know the positions |
| PLAYER\_Salary | INT | This is an int that tells how much thd deplayer is making |
| Player STAT | INT | This is an int that lets us how good the player is |
| Player Country | Varchar(50) | This is a character Value to let us know what country the player is from |
| Player\_email | varchar(60) | This is a character value of data. Email |
| Player\_phone | varchar(45) | This is a character value of data. Phone number |
| Match\_ID | INT | This is int primary to identify every match |
| Home Team | INT | This is a home team id value |
| Away Team | INT | This is a away team id value |
| Match Score | INT | This is an int to let us know the score of the match |
| Match Date | INT | This is an int to let us know the date of the match |
| Match Time | INT | This is an int to let us know the time of the match |
| Match Location | Varcha(50) | This is a charaxter value to let us know the location of the match |
| Match Marketing | INT | This is an int that lets us know the budget of the marketing for the match |
| Match Viewership | INT | This is an int to let us know how many peope viewed thr match |
| Match Tickets | INT | This is an int to let us know how many tickets were sold |
| Number of Goals | INT | This is an in to let us know how many goals were scored |
| Ticket ID | INT | This is an int primary key to identify each ticket |
| Ticket Price | INT | This is an int to let us know how much the ticket is going to be |
| Ticket\Seat | VarChar(50) | This is a character value to let us know where the seat is located |
| Coach ID | INT | This is an int primary key that lets us identify every coach |
| Coach\_Name | Varchar(50) | This is a character value that let us know the name of every coach |
| Fan ID | INT | This is an int primary key to let us identify every fan |
| Fan First Name | Varchar(50) | This is a character value to let us know the Fan’s first name |
| Fan Last Name | Varchar(50) | This is a character value to let us know the Fan’s first name |
| Fan Favorite Team | varchar(50) | This is a character value to let us know the Fan’s favoite team. |
| Fan Phone | INT | This is an int to let us know the fan’s phone number |
| Fan Email | Varcha(50) | This is a character value to let us know the fan’s email |
| Fan Address | Varchar(1000 | This us a charafter vlaue to let us know where the fan lives. |

1. An explanation of the concepts for entity integrity and referential integrity that influence the performance of the MySQL server. (Aaron)  
     
    Entity integrity makes sure that primary keys are unique in their tables and that the primary key will never be null. Referential integrity ensures that the values in a foreign key column will be found in the originating table with the primary key. Ensuring both entity integrity and referential integrity is crucial for enhancing the performance of a MySQL server. These constraints maintain the accuracy and consistency of data, preventing errors and inconsistencies that can adversely impact the database's performance. Additionally, when the data structure is well-defined and consistent, well-established integrity constraints can assist in query optimization and boost the database's performance.
2. An explanation of how the first, second, and third normal forms can improve. Draw the dependency diagram for each entity in your project (Chris).

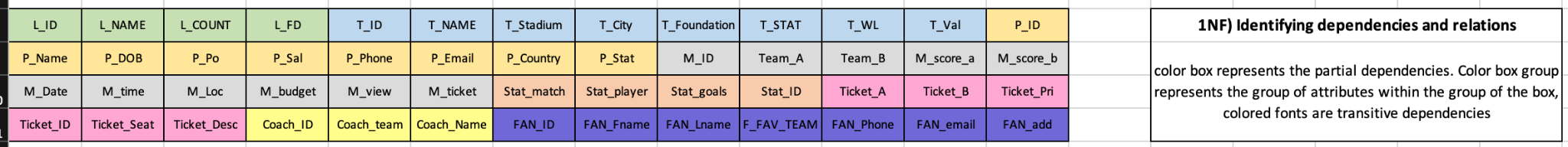
Our project is based on UEFA football database. Depends on the user guide and requirement, we identified the attributes required for the database. Once we defined and list the requirement attributes, we were able to separate attributes within its own group of entities. In FRD process, most of the work is accomplished, and here’s detail process.x

1. Identify the attributes for the UEFA database

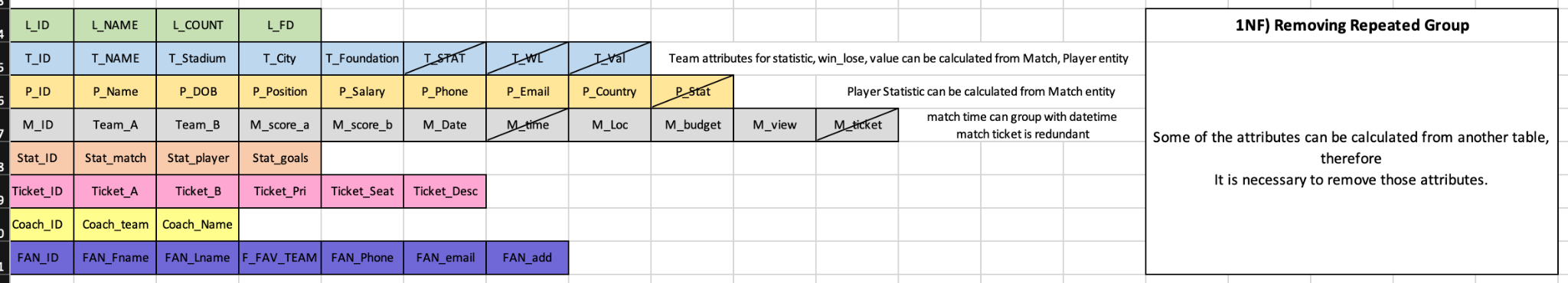


* The database allows user to query the any data that is related to the player, team, match, and more. We identified what attributes are required to search user desired data, so above the table is attributes for the database.

1. First Normal Form: Removing redundancy and define dependencies

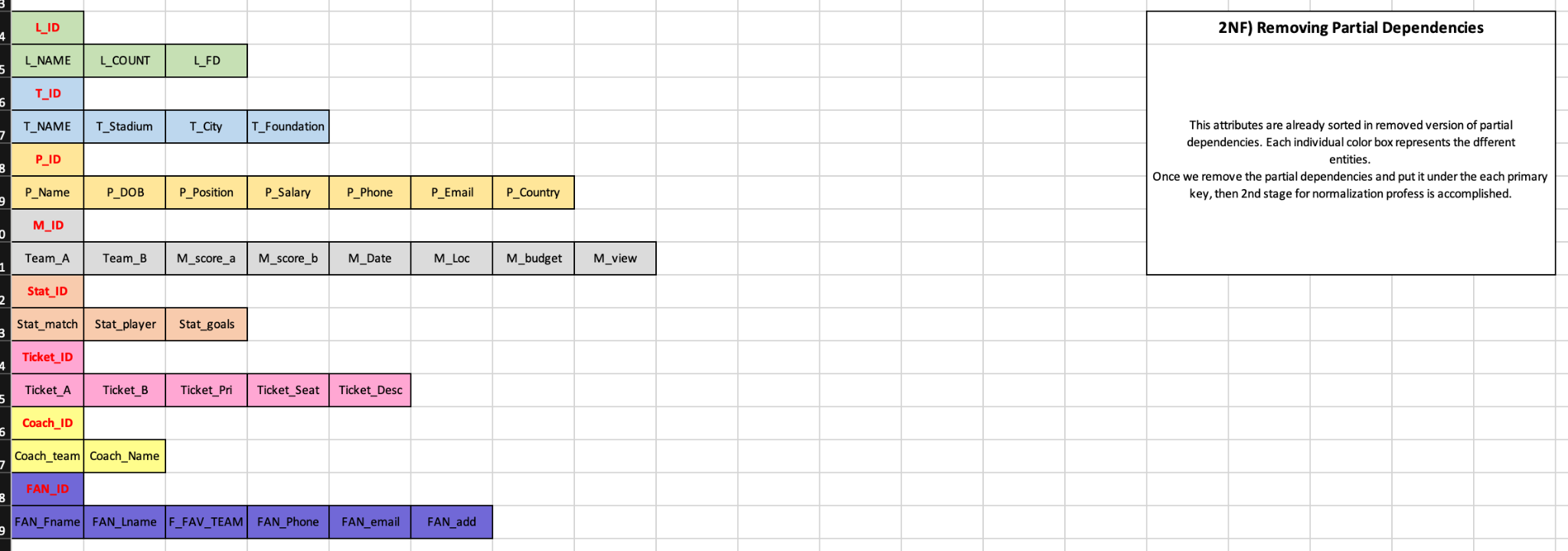


* We identified partial dependencies with the individual attribute ids, so id’s can define and become a role as primary key.



* Once we identified all the partial dependencies and transitive dependencies, we removed some of redundant attributes (attributes that can be calculated from another attribute or entities)

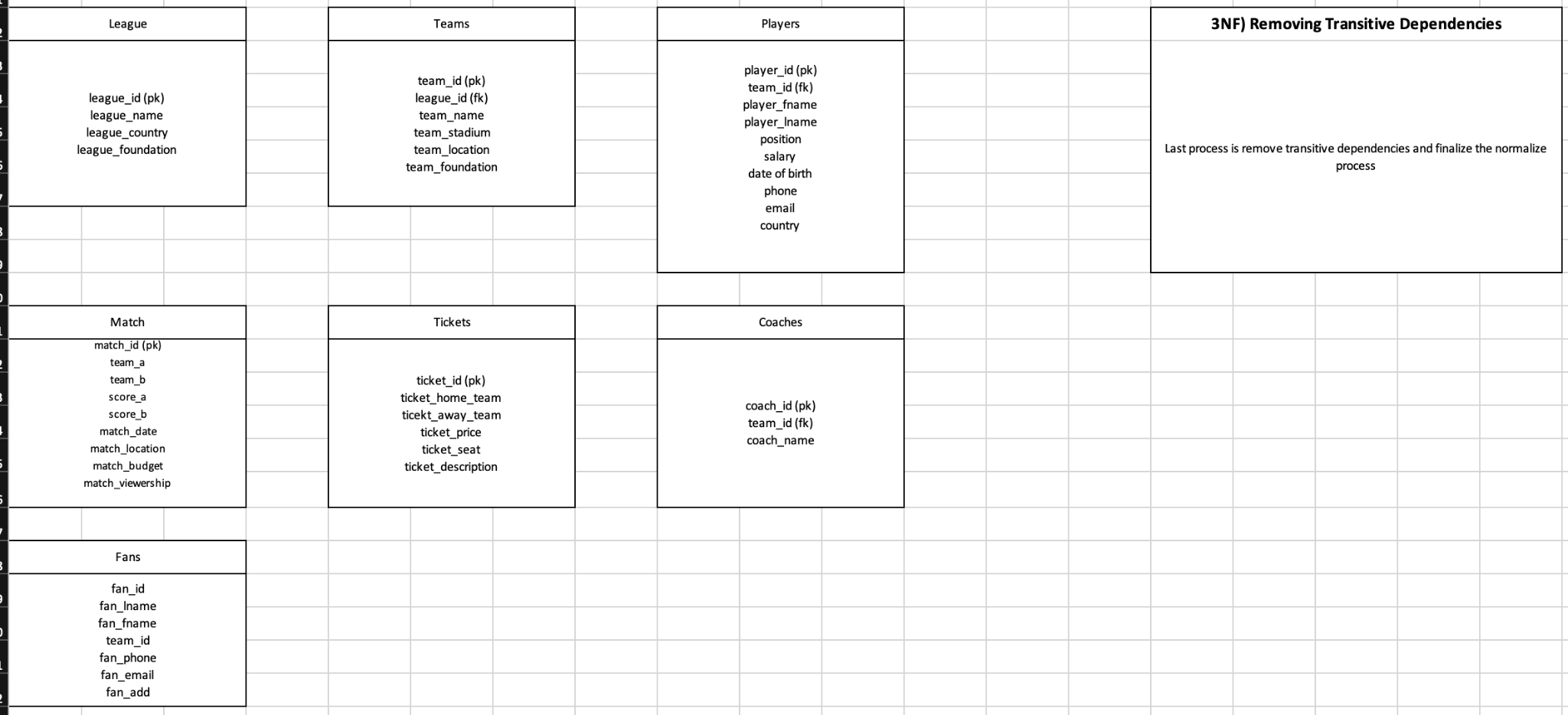
1. Second Normal Form: Removing Partial Dependencies



* Partial dependencies are already sorted and arranged with its own primary keys. Our team checked one more time to make sure there is no redundant attributes.

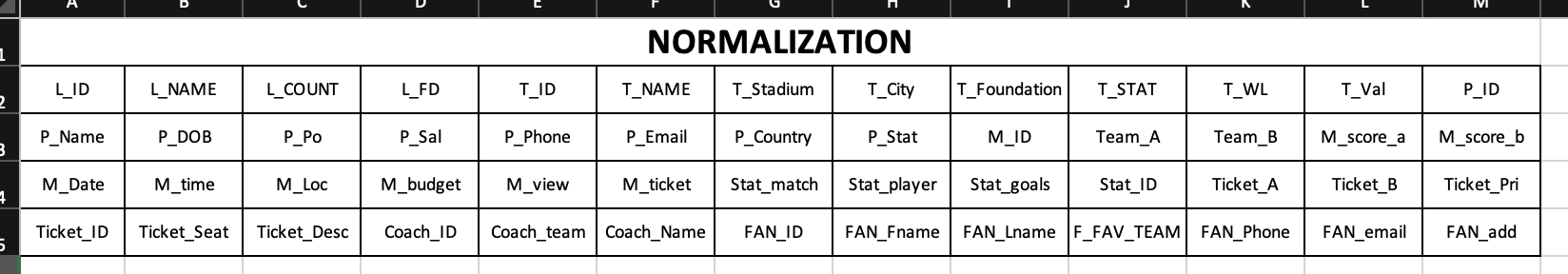
1. Third Normal Form: Remove Transitive Dependencies and define individual entities

* We removed transitive dependencies and created individual entities so that can hold its the attributes.

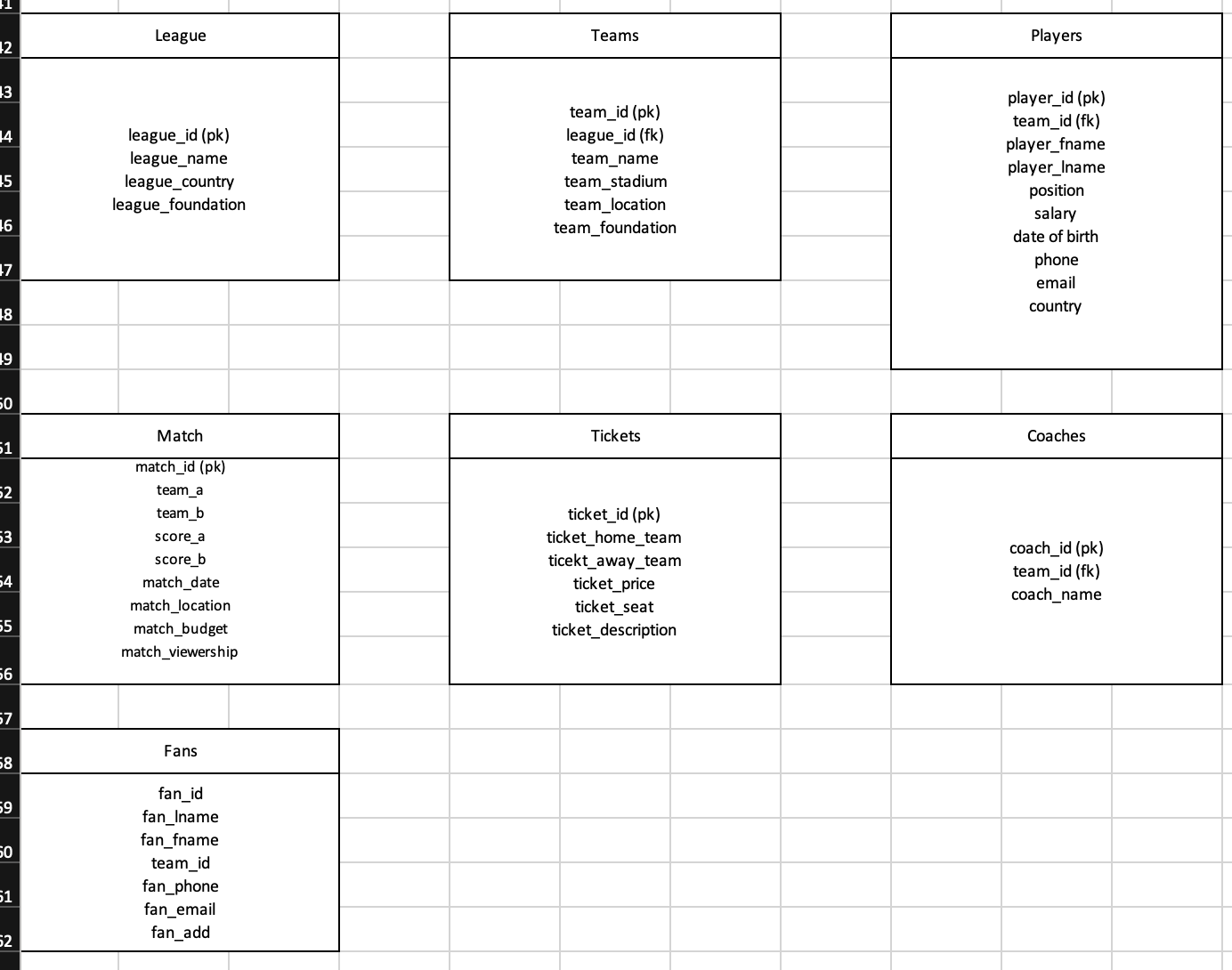


4. Screenshots before and after normalization, making sure to explain the process and the advantages of normalizing the database. If the database happens to already be normalized, explain why this is true (Chris)

**Before Normalization**



**After Normalization**



Normalization was already done in our FRD, however, we demonstrated the normalization process to figure out what attributes are needed and what entities we have to define. Within the proces, our team was able to get detailed understanding of the UEFA database and the process of normalization. There are some conflict within the defining foreign key or relations. FRD helped a lot for us to guide and identify the user requirement, so it is true that the normalization is already accomplished within the process of writing a FRD. Before normalization, the tables are just attributes, and after normalization we can identify the entities and its attributes.

5. A recommendation and implementation of two additional improvements to the database structure, showing screenshots, and explanations detailing the improvement tasks and the gains. (Aaron)

Improvements that were made to the database structure included removing attributes from entities that could be made into its own entity. For example, tickets sold were originally an attribute of every match. This has been changed into a new entity that takes a foreign key from a match. Additionally, the ticket entity is a foreign key to a Fan going to the match. Another adjustment was creating a Statistics entity. Originally, the stats were just a static attribute for each player. Now, every match will also have a statistic entity for each player and has attributes of the stats for the match. These changes have allowed the database to be more specific and organized.

6. Develop your data dictionary for the database (Aaron).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table Name | Attribute Name | Contents | Type | Format | Range | Required | PK/FK | FK Referenced Table |
| LEAGUES | LEAGUE\_ID | League ID | INT | 0001 | 0-9999 | Y | PK |  |
|  | LEAGUE\_NAME | League Name | VARCHAR | Xxxx |  | Y |  |  |
|  | LEAGUE\_COUNTRY | League’s Home Country | VARCHAR | Xxxx |  | Y |  |  |
| TEAMS | TEAM\_ID | Team ID | INT | 0001 | 0-9999 | Y | PK |  |
|  | LEAGUE\_ID | League ID | INT | 0001 | 0-9999 | Y | FK | LEAGUE |
|  | TEAM\_NAME | Team Name | VARCHAR | Xxxx |  | Y |  |  |
| PLAYERS | PLAYER\_ID | Player ID | INT | 0001 | 0-9999 | Y | PK |  |
|  | PLAYER\_FNAME | First Name | VARCHAR | Xxxx |  | Y |  |  |
|  | PLAYER\_LNAME | Last Name | VARCHAR | Xxxx |  | Y |  |  |
|  | TEAM\_ID | Team ID | INT | 0001 | 0-9999 | Y | FK | TEAM |
|  | PLAYER\_POS | Player Position | VARCHAR | xxxx |  | Y |  |  |
|  | PLAYER\_VAL | Player Contract Value | VARCHAR | 9999 | 0-99999999 |  |  |  |
|  | PLAYER\_PHONE | Phone Number | VARCHAR | (123)456-7891 |  |  |  |  |
|  | PLAYER\_EMAIL | Player Email | VARCHAR | XX@x |  |  |  |  |
| MATCH | MATCH\_ID | Match ID | INT | 0000 | 0-9999 | Y | PK |  |
|  | TEAM\_ID | Team ID | INT | 0001 | 0-9999 | Y | FK | TEAM |
|  | TEAM\_ID | Team ID | INT | 0001 | 0-9999 | Y | FK | TEAM |
|  | MATCH\_SCORE\_A | Team A Score | INT | 0001 | 0-9999 | Y |  |  |
|  | MATCH\_SCORE\_B | Team B Score | INT | 0001 | 0-9999 | Y |  |  |
|  | MATCH\_DATE | Match Date | DATE |  |  | Y |  |  |
|  | MATCH\_LOCATION | Match Location | VARCHAR | Xxxx |  | Y |  |  |
|  | MATCH\_TIME | Match Total Time | INT | 0001 |  | Y |  |  |
|  | MATCH\_BUDGET | Marketing Budget | INT | 99999 | 0-99999999 | Y |  |  |
| COACH | COACH\_ID | Coach ID | INT | 0001 | 0-9999 | Y | YK |  |
|  | COACH\_NAME | Coach Name | VARCHAR | Xxxx |  | Y |  |  |
|  | TEAM\_ID | Team ID | INT | 0001 | 0-9999 | Y | FK | TEAM |
| FAN | FAN\_ID | Fan ID | INT | 0001 | 0-9999 | Y | YK |  |
|  | FAN\_FNAME | First Name | VARCHAR | Xxxx |  | Y |  |  |
|  | FAN\_LNAME | Last Name | VARCHAR | Xxxx |  | Y |  |  |
|  | FAN\_TEAM | Favorite Team | INT | 0001 | 0-9999 |  | FK | TEAM |
|  | FAN\_PHONE | Phone Number |  | (123)456-7891 |  |  |  |  |
|  | FAN\_EMAIL | Email Address | VARCHAR | Xxxx@xxxx |  |  |  |  |
|  | FAN\_ADDRESS | Home Address | VARCHAR | Xxxx |  |  |  |  |
| STATISTIC | STAT\_ID | Statistic ID | INT | 0001 | 0-9999 | Y |  |  |
|  | MATCH\_ID | Match ID | INT | 0001 | 0-9999 | Y | FK | MATCH |
|  | PLAYER\_ID | Player ID | INT | 0001 | 0-9999 | Y | FK | PLAYER |
|  | STAT\_GOALS | Goals Scored | INT | 0001 | 0-9999 | Y |  |  |
|  | STAT\_RATE | Player Match Rating | INT | 0001 | 0-9999 | Y |  |  |

7. A list of all the duties assigned to each team member and a status update.

Kyungchan Im - Normalization process, Quality management, Overall review

Aaron Galicia - Data Dictionary, Entity and Referential Integrity Explanations, development ERD

Levi Guerengomba - Identifying attributes, quality management, development ERD

8. A list of all references used.

None.