

IPL FANTASY LEAGUE

Abhinav Reddy Boddu	Rohit Gowlapalli	Gnana Prakash Punnavajhala
2021101034	2021101113	2021111027

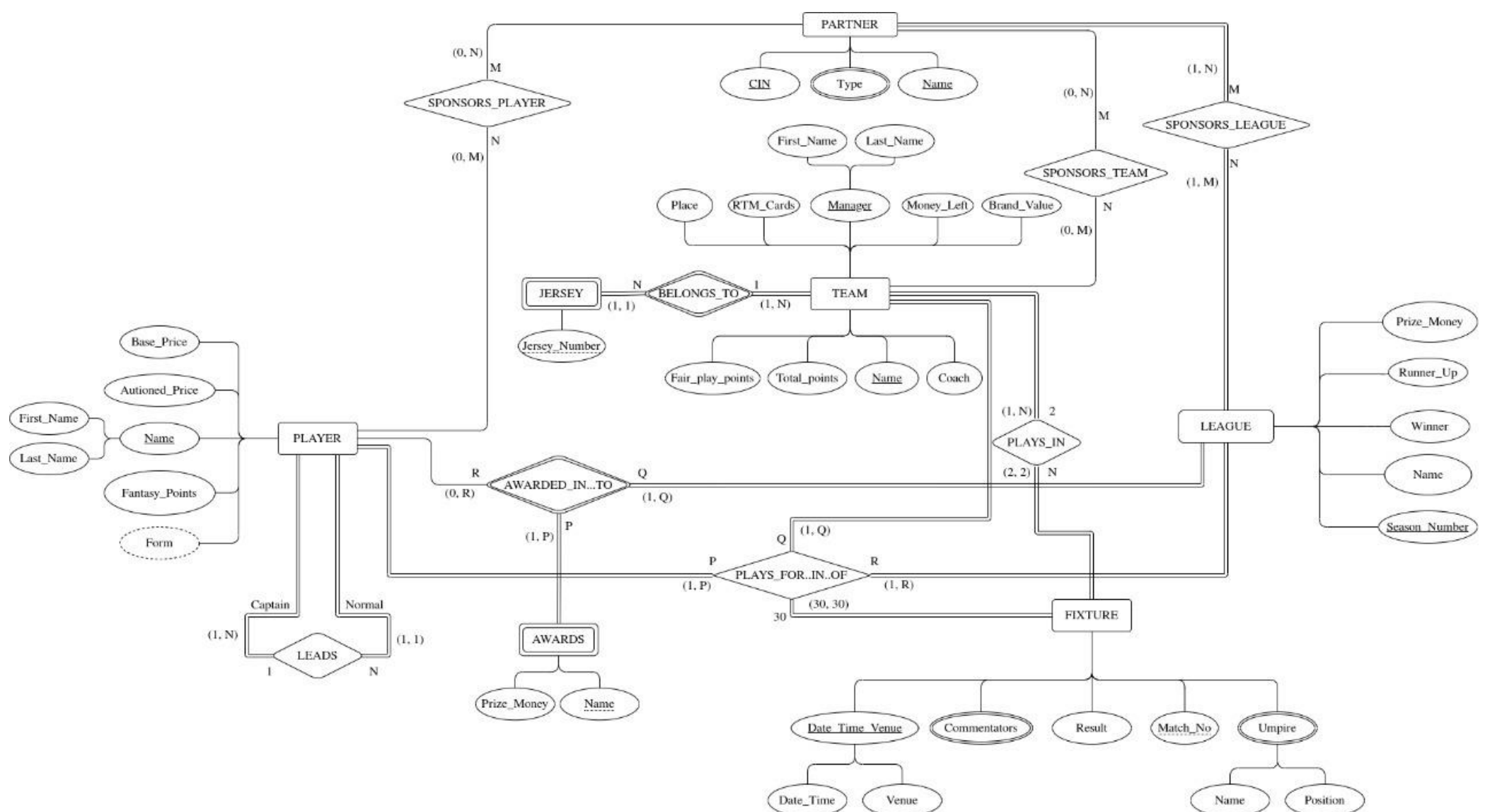
Modifications in ER Model:

- We added the dependency of Form on Fantasy_Points to show the attribute from which Form is derived from.
- We also removed the composite attribute Name having sub-attributes First_Name and Last_Name and added only the Name attribute to the player as the primary key attribute.

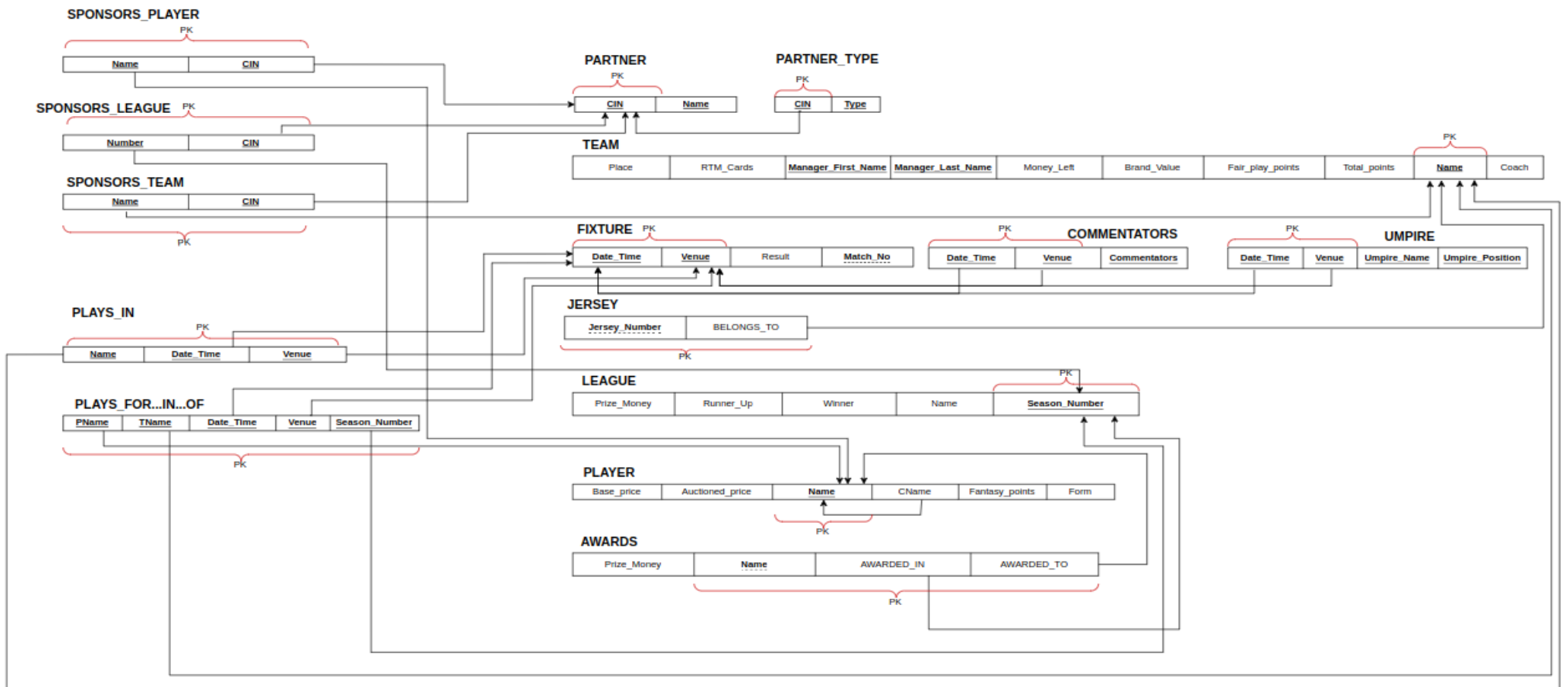
ER Model to Relational Model:

Below steps describe the steps of an algorithm for ER-to-relational mapping. The **LEAGUE** ER Schema is shown below, and the corresponding **LEAGUE** relational database schema followed by it to illustrate the mapping steps. Our mapping will create tables with simple single-valued attributes. The relational model constraints which include primary keys and referential integrity constraints on the relations will also be specified in the mapping results.

Entity Relationship Schema



Relational Database Schema



Step 1: Mapping of Regular Entity Types:

For each regular (strong entity) type E in the ER schema we created a relation R that includes all the simple attributes of E . We included only the simple component attributes of a composite attribute. We chose one of the key attributes of E as the primary key of R . If the chosen key of E is composite, then the set of simple attributes that formed it were together denoted as the primary key of R . In our case,

- The composite attribute $Date_Time_Venue$ of “FIXTURE” was converted to simple attributes $Date_Time$ and $Venue$
- The composite attribute $Manager$ of “TEAM” was converted to simple attributes $Manager_First_Name$ and $Manager_Last_Name$

Step 2: Mapping of Weak Entity Types:

For each weak entity type W in the ER schema with owner entity type E , we created a relation R and included all its simple attributes (or simple components of composite attributes) of W as attributes of R . In addition, we included the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s) as foreign key attributes of R (this takes care of mapping the identifying relationship type of W). In our case,

- The relation “JERSEY” was created with $Jersey_Number$ as a partial key and $Belongs_To_Team$ as a foreign key referencing Name of “TEAM”.
- The relation “AWARDS” was created with $Prize_Money$ as simple attribute and $Name$ as partial key. The identifying relationships are $AWARDED_IN$ and $AWARDED_TO$ where $AWARDED_IN$ is a foreign key referencing Season Number of “LEAGUE” and $AWARDED_TO$ is a foreign key referencing Name of “PLAYER”.

Step 3: Mapping of Binary 1:1 Relationship Types:

No Binary 1:1 Relationship Types exists in our schema

Step 4: Mapping of Binary 1:N Relationship Types:

We have employed the **foreign key approach** which identifies the relation S that represents the participating entity type at the N -side of the relationship type for each regular binary 1:N Relationship type R and includes the primary key of the relation T that represents the other entity type participating in R as foreign key in S . In our case,

- $CName$ has been added as a foreign key to “PLAYER” referencing Name of “PLAYER” to denote the “LEADS” relation such that player “ $CName$ ” LEADS a player “ $Name$ ”
- The “PLAYS_IN” relation was created with $Name$ as a foreign key referencing Name of “TEAM”, $Date_Time$ as a foreign key referencing Date_Time of “FIXTURE” and $Venue$ as a foreign key referencing Venue of “FIXTURE”

Step 5: Mapping of Binary M:N Relationship Types:

We have employed the **Relationship relation / cross-reference approach**. For each binary M:N relationship type R, we created a new relation S to represent R. We included the primary keys of the relations that represent the participating entity types as foreign key attributes in S; their combination will form the primary key of S. Also, we included any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S. In our case,

- The relation “SPONSORS_PLAYER” was created with Name as a foreign key referencing Name of “PLAYER” and CIN as a foreign key referencing CIN of “PARTNER”
- The relation “SPONSORS_LEAGUE” was created with Season as a foreign key referencing Season Number of “LEAGUE” and CIN as a foreign key referencing CIN of “PARTNER”
- The relation “SPONSORS_TEAM” was created with Name as a foreign key referencing Name of “TEAM” and CIN as a foreign key referencing CIN of “PARTNER”

Step 6: Mapping of Multivalued Attributes:

For each multivalued attribute A, we created a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute. The primary key of R is the combination of A and K. If the multivalued attribute is composite, we included its simple components. In our case,

- The relation “PARTNER_TYPE” was created with CIN as a foreign key referencing CIN of “PARTNER” and Type as key attribute
- The relation “COMMENTATORS” was created with Date_Time as a foreign key referencing Date Time of “FIXTURE”, Venue as a foreign key referencing Venue of “FIXTURE” and Commentators as key attribute.
- The relation “UMPIRE” was created with Date_Time as a foreign key referencing Date Time of “FIXTURE”, Venue as a foreign key referencing Venue of “FIXTURE” and Umpire_Name and Umpire_Position as key attributes.

Step 7: Mapping of N-ary Relationship Types:

We have employed the **Relationship relation / cross-reference approach**. For each n-ary relationship type R, where $n > 2$, we created a new relationship relation S to represent R. We included the primary keys of the relations that represent the participating entity types as foreign key attributes in S. Also, we included any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S. In our case,

- The relation “PLAYS_FOR...IN...OF” was created with PName as a foreign key referencing Name of “PLAYER”, TName as a foreign key referencing Name of “TEAM”, Date_Time and Venue as foreign keys referencing Date Time and Venue of “FIXTURE” respectively and Season_Number as a foreign key referencing Season Number of “LEAGUE”

Step 8: Options for mapping Specialization or Generalization:

No Subclasses exist in our Schema

Step 9: Mapping of Union Types (Categories):

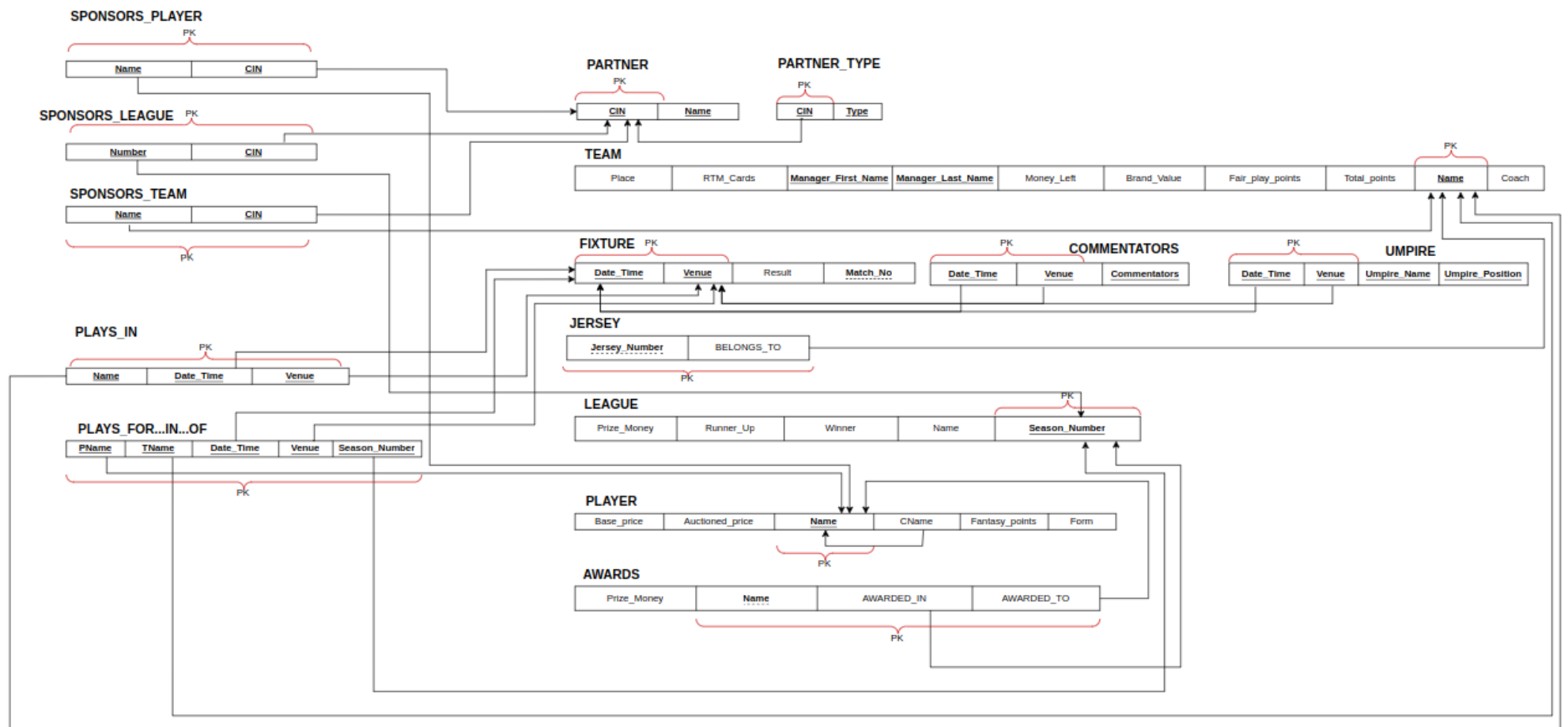
No Union type exists in our Schema

Normalization:

Conversion of Relational Model to 1NF:

Relation schema is in 1NF if the values in domain of each attribute are atomic. **The relational model is already in 1NF** as new relations for Multivalued attributes were created in Step 6 and Composite attributes were converted to Atomic (simple) attributes in Step 1.

1NF

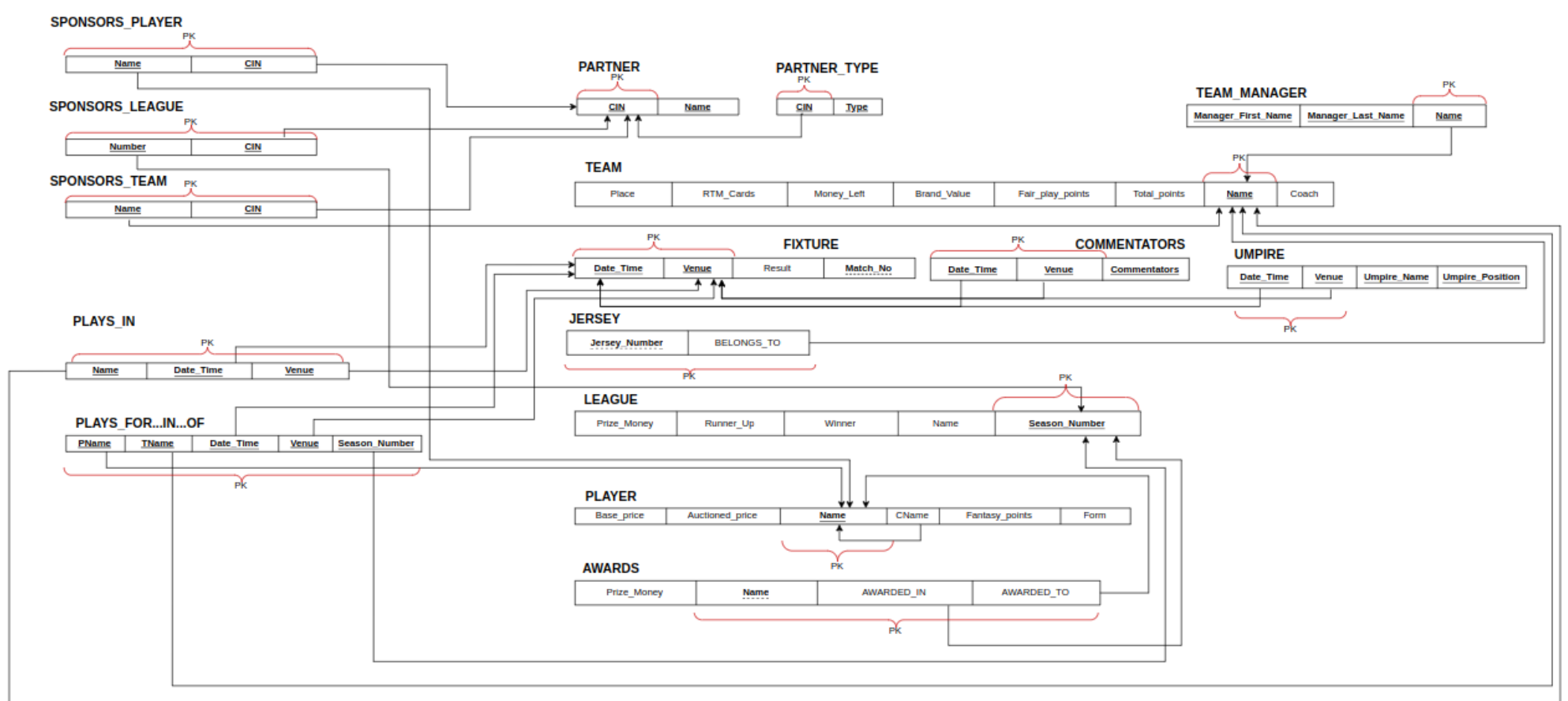


Conversion of Relational Model to 2NF:

A relation schema is in 2NF if every non-prime attribute A in R is fully functionally dependent on every key of R. In our case,

- The superkey {Name, Manager_First_Name, Manager_Last_Name} of “TEAM” was broken down into just Name as the only primary key and another relation “TEAM_MANAGER” was created with Name as foreign key referencing Name of “TEAM” with {Manager_First_Name, Manager_Last_Name} as key attributes. This was done because both the subsets {Name} and {Manager_First_Name, Manager_Last_Name} could uniquely identify each of the non-prime attributes of “TEAM” which violates the guidelines of 2NF.

2NF



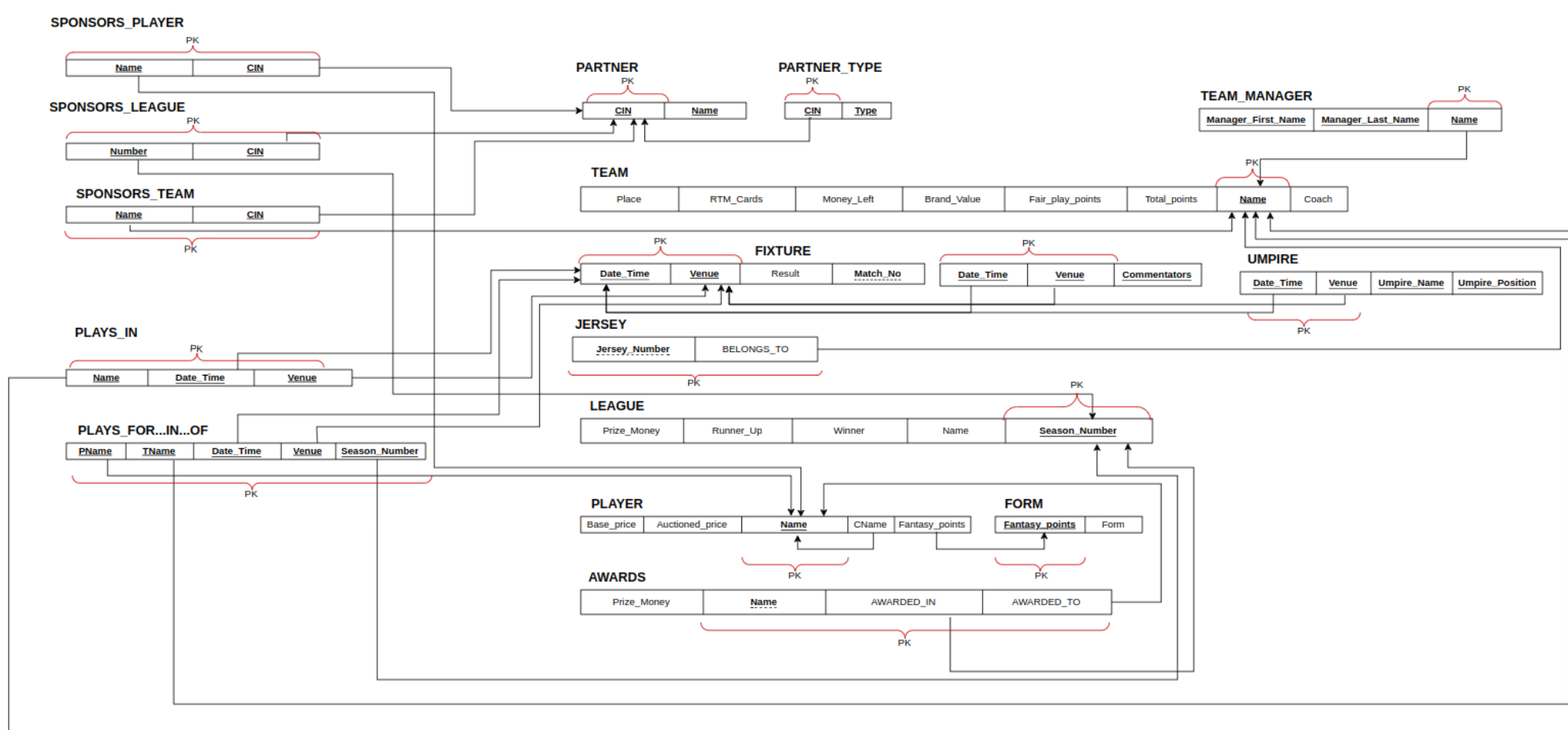
Conversion of Relational Model to 3NF:

A relation schema is in 3NF if all non-trivial dependencies in F+ are of the form $X \rightarrow A$ with either

- X is a superkey
- A is a prime attribute

In our case, there was a relation from Fantasy_Points to Form (derived attribute) of "PLAYER" and Fantasy_Points is not a prime attribute. Thus, the relation "FORM" was created with Fantasy_Points of "PLAYER" as a foreign key referencing Fantasy_Points of "FORM" and Form as a simple attribute.

3NF



Links to the relational Model:

- Relational Model & 1NF:
https://drive.google.com/file/d/1TYSx456RcqdYZ1rNxR_tNuy5NpFExArG/view?usp=sharing
- 2NF:
https://drive.google.com/file/d/1Wnum_BBXM2HJxa9qniFJg0fyoWuLQMit/view?usp=sharing
- 3NF:
<https://drive.google.com/file/d/1AAkcPeoEr2s3ngNmSuyGnUOPTNj0XILQ/view?usp=sharing>