Indian Premier League Database

Saketh Vemula - 2022114014
Asish Bharadwaj - 2022101100
Madireddy Ananya - 2022101102
Chanukya SVSK - 2022101120
Viswanath Vuppala - 2022101084

1 ER Diagram to Relational Model

We describe the steps of an algorithm for ER-to-relational mapping. The IPL ER diagram is shown in figure 1, and the corresponding IPL relational database schema is shown in figure 8 to illustrate the mapping steps.

2 Steps:

1. **Mapping of Regular Entity Types**
   For each regular(strong) entity type E in the ER Schema, create a relation R that includes all the simple attributes pf E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as the primary key for R. If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R. The foreign key and relationship attributes, if any, are not included yet; they will be added during subsequent steps. If multiple keys were identified for E during the conceptual design, the information describing the attributes that form each additional key is kept in order to specify additional (unique) keys of relation R.

   Following Regular Entity Types are mapped:
   Broadcasters, Season, Match, Team, Sponsors, Venue, Umpire, Player

2. **Mapping of Weak Entity Types**
   For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of W as attributes of R. In addition, include as foreign key attributes of R, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of mapping the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

   Following Weak Entity Types are mapped:
   Toss, Staff, Batsman_Stats, Bowler_Stats.

3. **Mapping of Binary 1:1 Relationship Types**
   For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. Choose one of the relations—S, say—and include as a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S. Include all the simple attributes (or simple components of composite attributes) of the 1:1 relationship type R as attributes of S.
   In our case we only have one relation TOSS_Decision in this category. We include Match_ID as foreign key in Toss.
   This Method is known as **Foreign Key Approach**.

   We don't have any 1:1 Binary Relationship Types in our Database.

4. **Mapping of Binary 1:N Relationship Types**
   We use **Foreign Key Approach** for this step. For each regular binary 1:N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; we do this because each entity instance on the N-side is related to at most one entity instance on the 1-side of the relationship type. Include any simple attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S.
   Also represent the foreign key relationship lines.

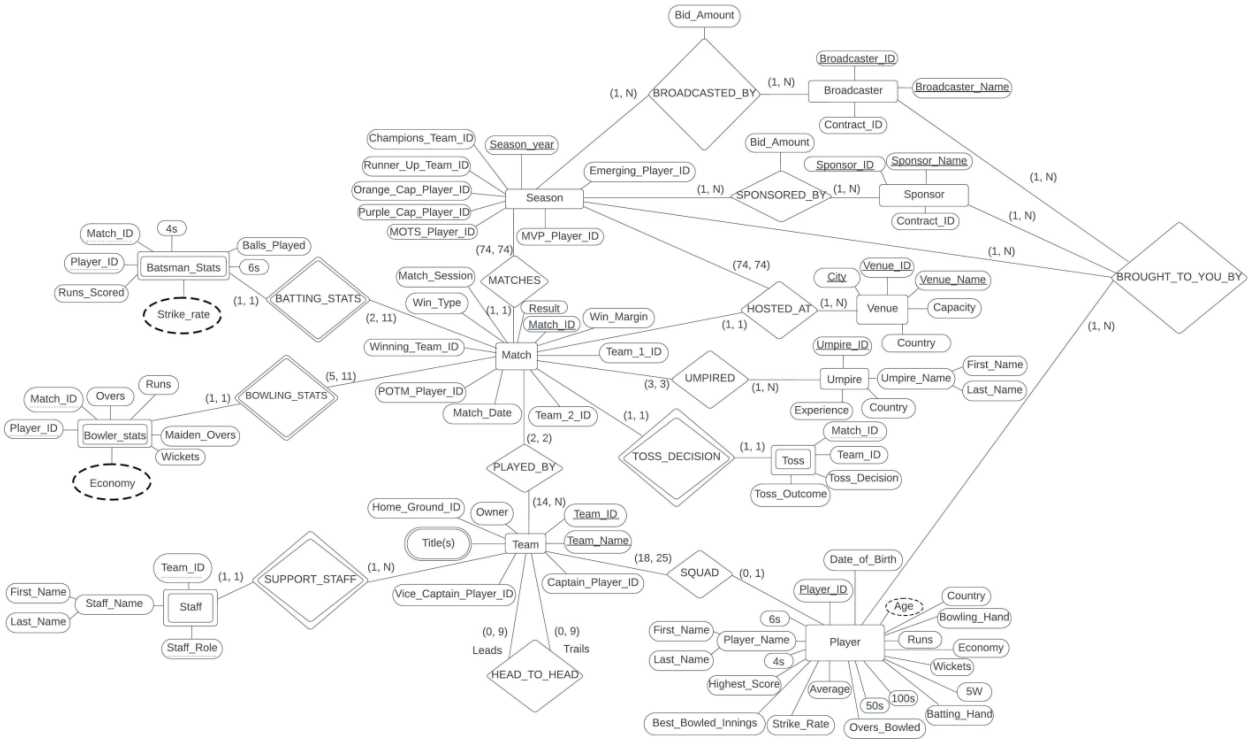

Figure 1: ER Diagram

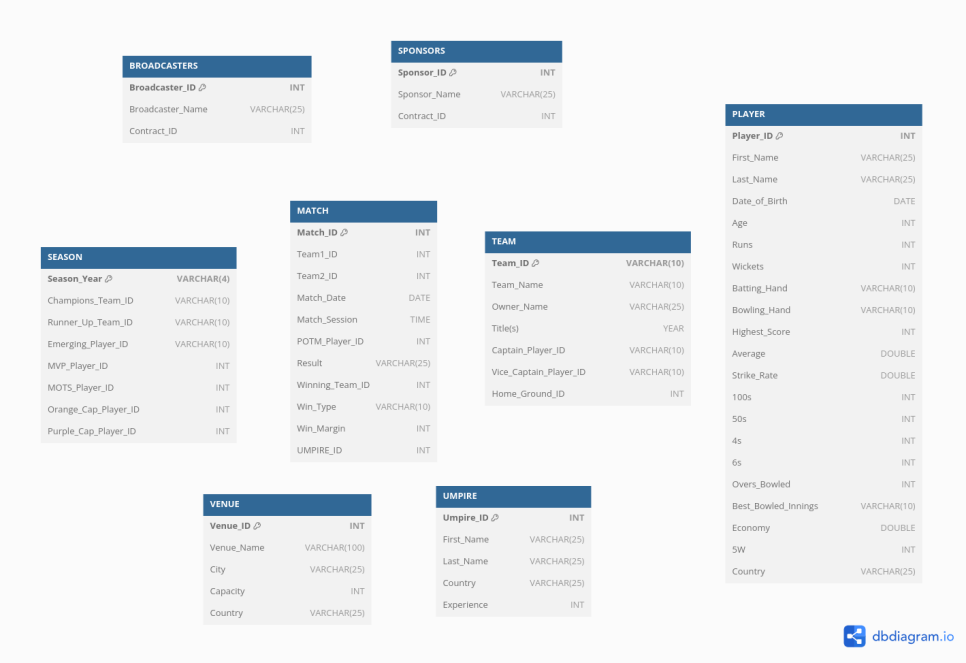

Figure 2: Relational Model after Step 1

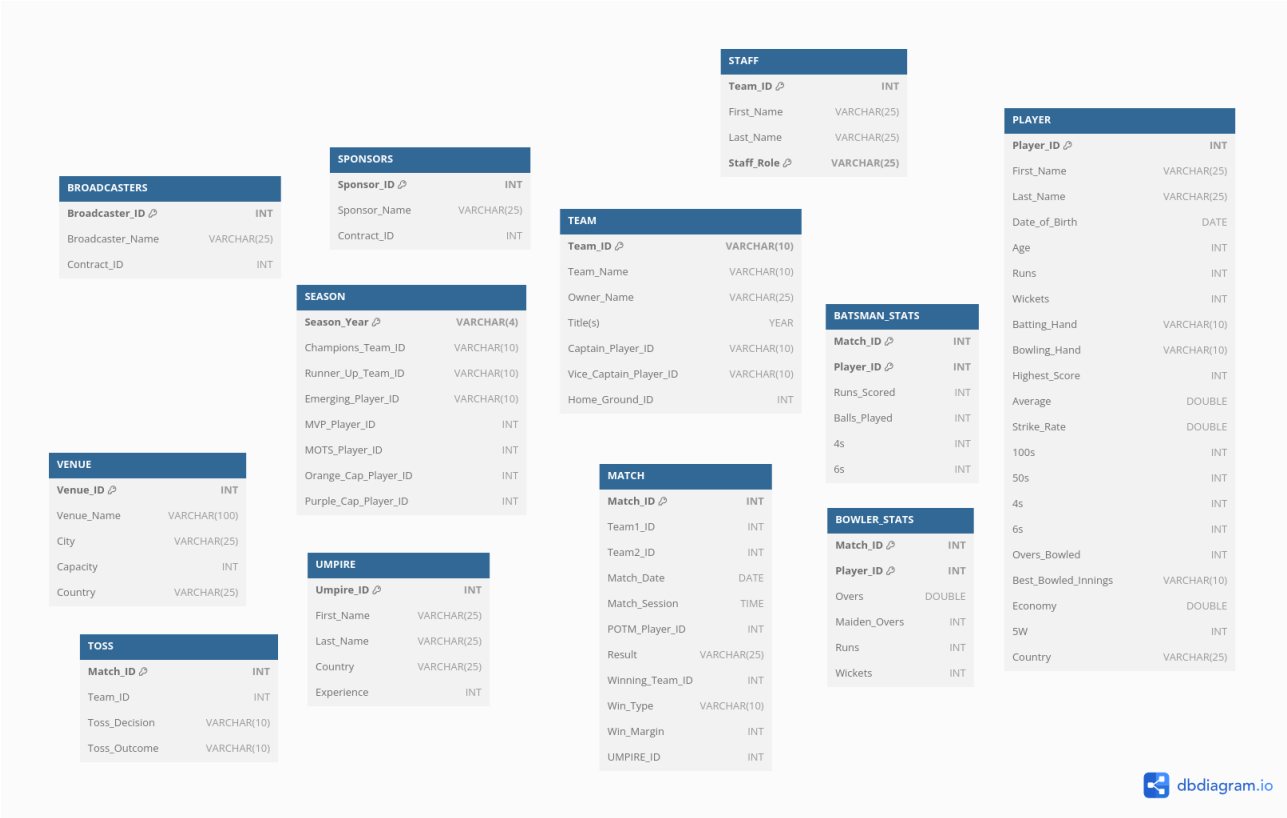

Figure 3: Relational Model after Step 2

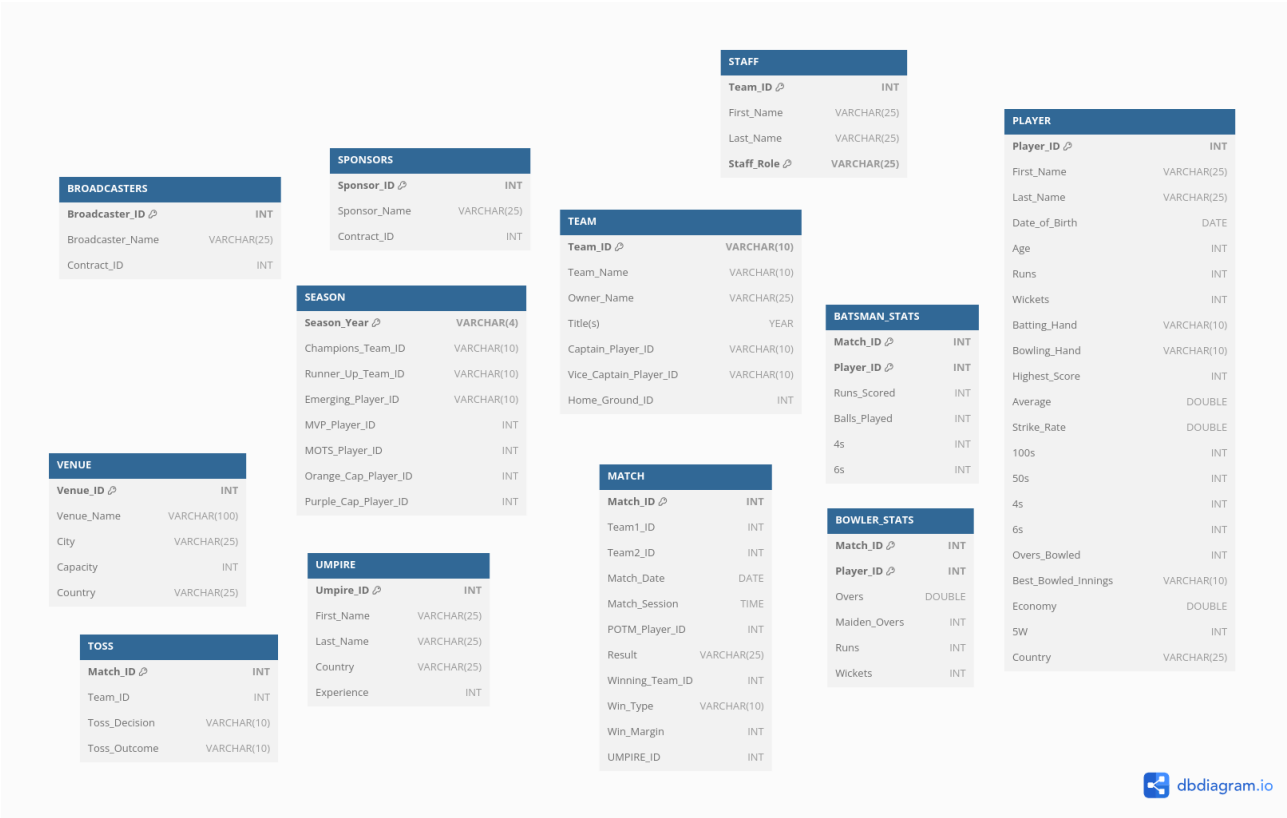

Figure 4: Relational Model after Step 3

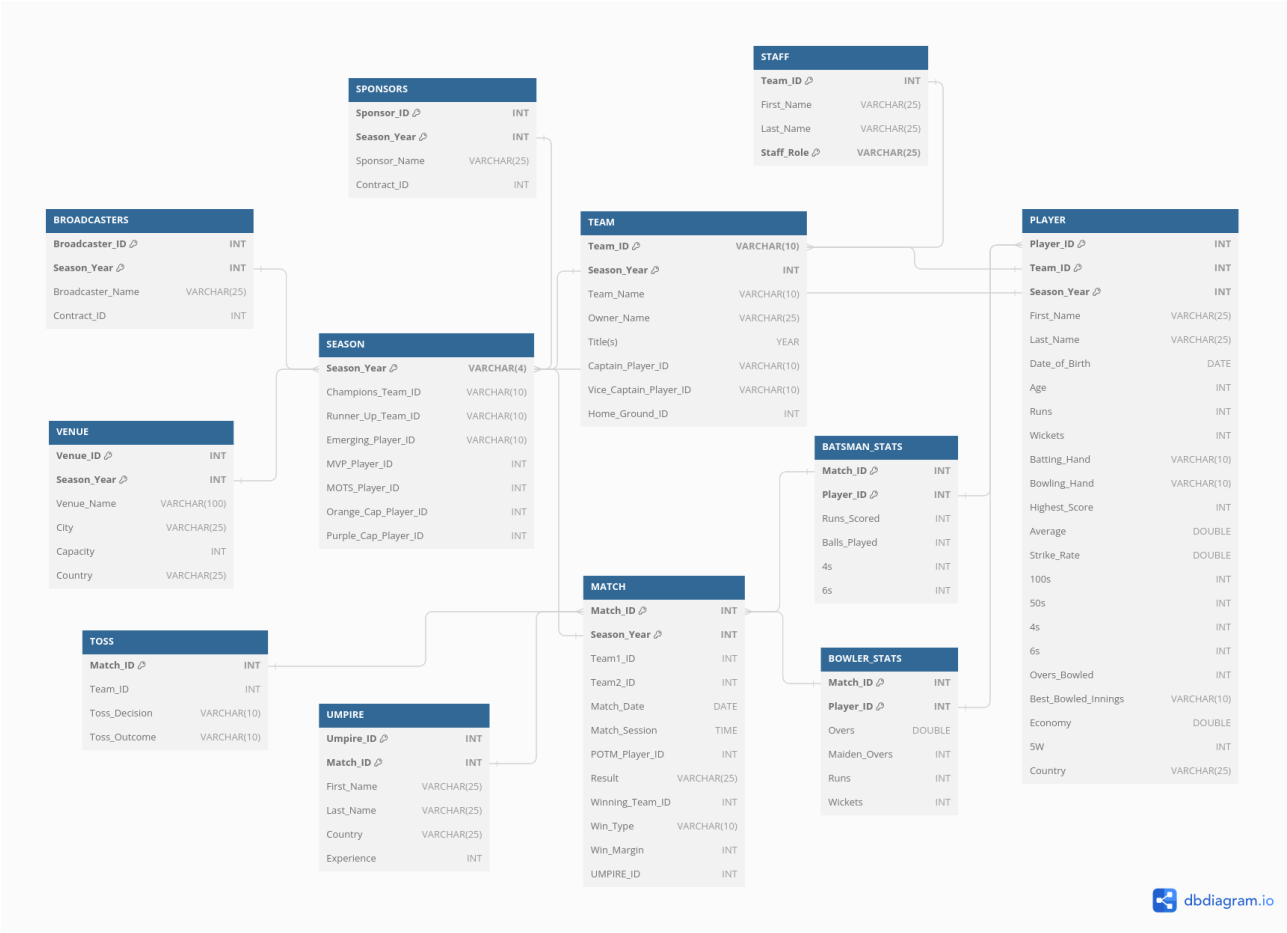

Figure 5: Relational Model after Step 4

The included mappings are shown in figure 5 along with new attributes included.
These include:

(a) Adding attribute Season_Year into relations Broadcasters, Sponsors, Venue, Team, Match, Player.

(b) Adding attribute Match_ID into relation Umpire.

(c) Adding attribute Team_ID into relation Player.

5. **Mapping of Binary M:N Relationship Types**
In the traditional relational model with no multi-valued attributes, the only option for M:N relationships is the relationship relation (cross-reference) option. For each binary M:N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

   We skip this as there is no binary relationship type of M:N CR in our ER Diagram.

6. **Mapping of N-ary Relationship Types**
We use the relationship relation option. For each nary relationship type R, where n greater than 2, create a new relationship relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types. Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S. The primary key of S is usually a combination of all the foreign keys that reference the relations representing the participating entity types. However, if the cardinality constraints on any of the entity types E participating in R is 1, then the primary key of S should not include the foreign key attribute that references the relation E-dash corresponding to E.

   We introduce two new relations:
Brought_To_You_By, Hosted_At.

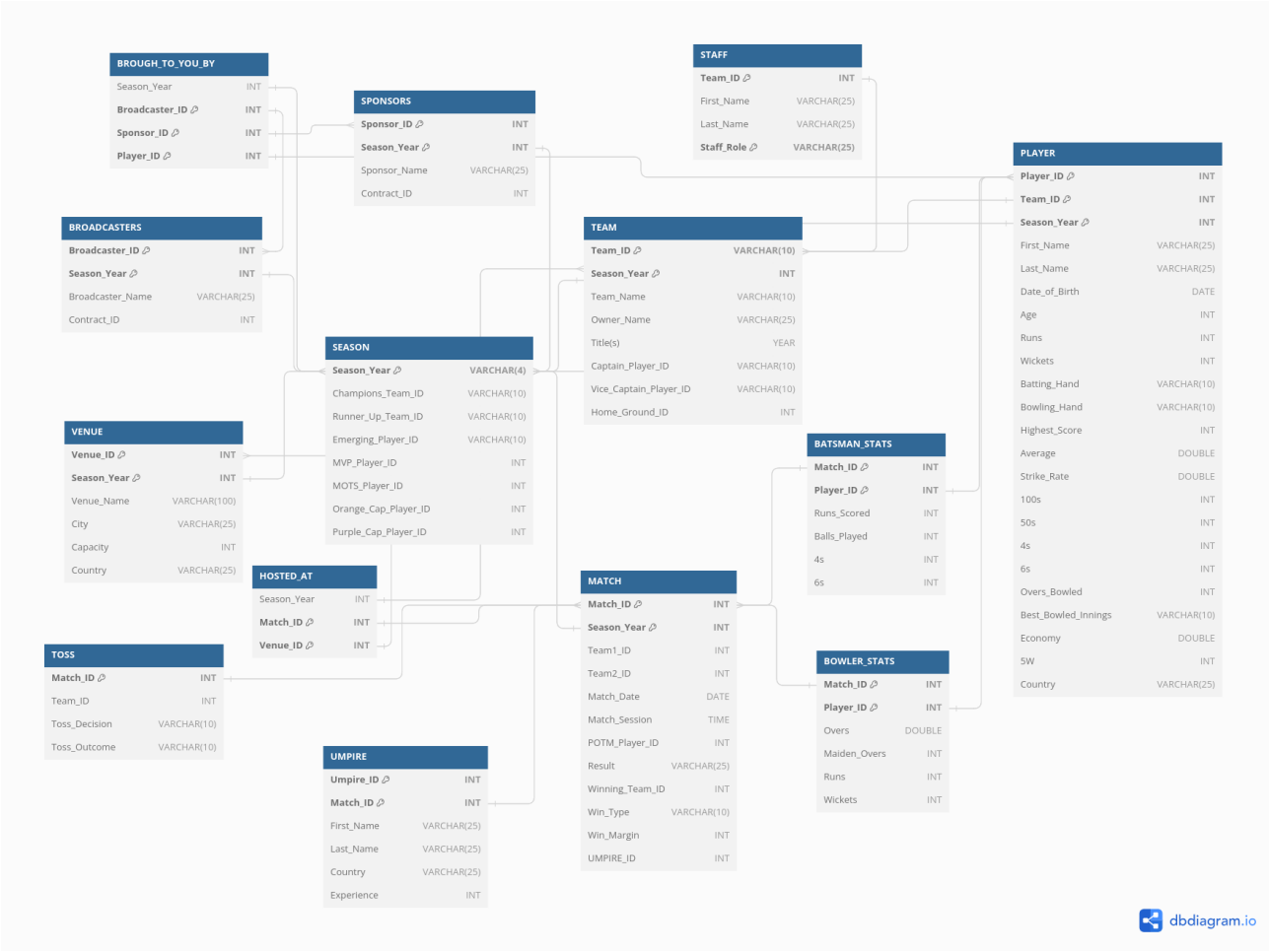

Figure 6: Relational Model after Step 6

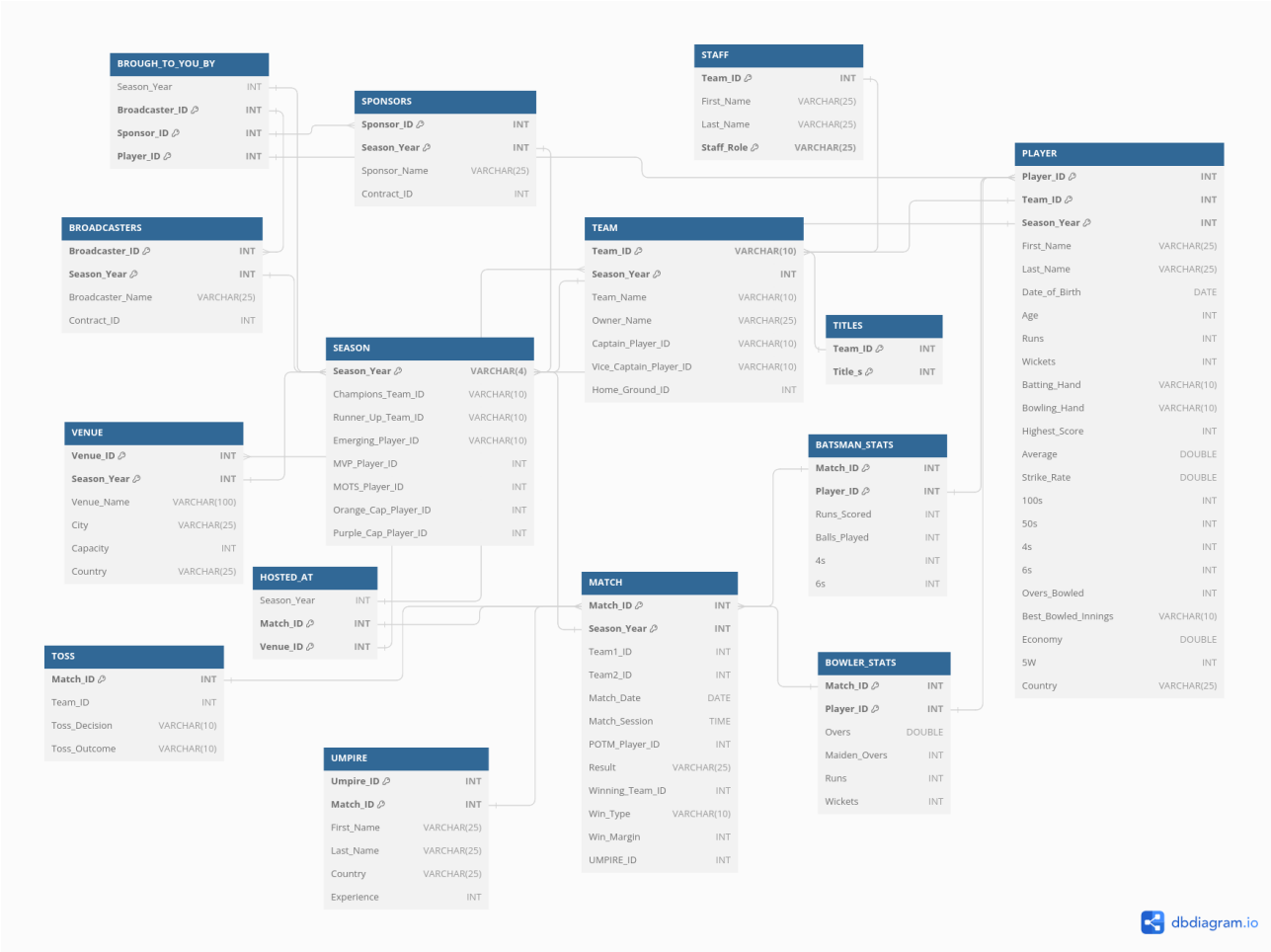

Figure 7: Relational Model After Step 7

7. **Mapping of Multi-valued Attributes**
   For each multivalued attribute A, create 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 include its simple components.

   Multivalued attribute Titles is included as a new relation Titles, with primary key consisting of Team_ID and Title_s.

8. **Including Attributes of Relationship Types**
   Finally we include attributes of relationship types if they aren't.

   In our case we include attribute Bid_Amount into relations Sponsors and Broadcasters.

3 Conversion to 1NF

1NF disallows multivalued attributes, compostie attributes, and their combination. The only attributes values permitted by 1NF are single atomic values.

In our relational model,

1. Brought_To_You_By has three multivalued attributes:
   Broadcasters_ID, Sponsor_ID, Player_ID.

2. Hosted_At has two multivalued attributes:
   Match_ID and Venue_ID.

Used the method of Removing Attributes for this purpose.

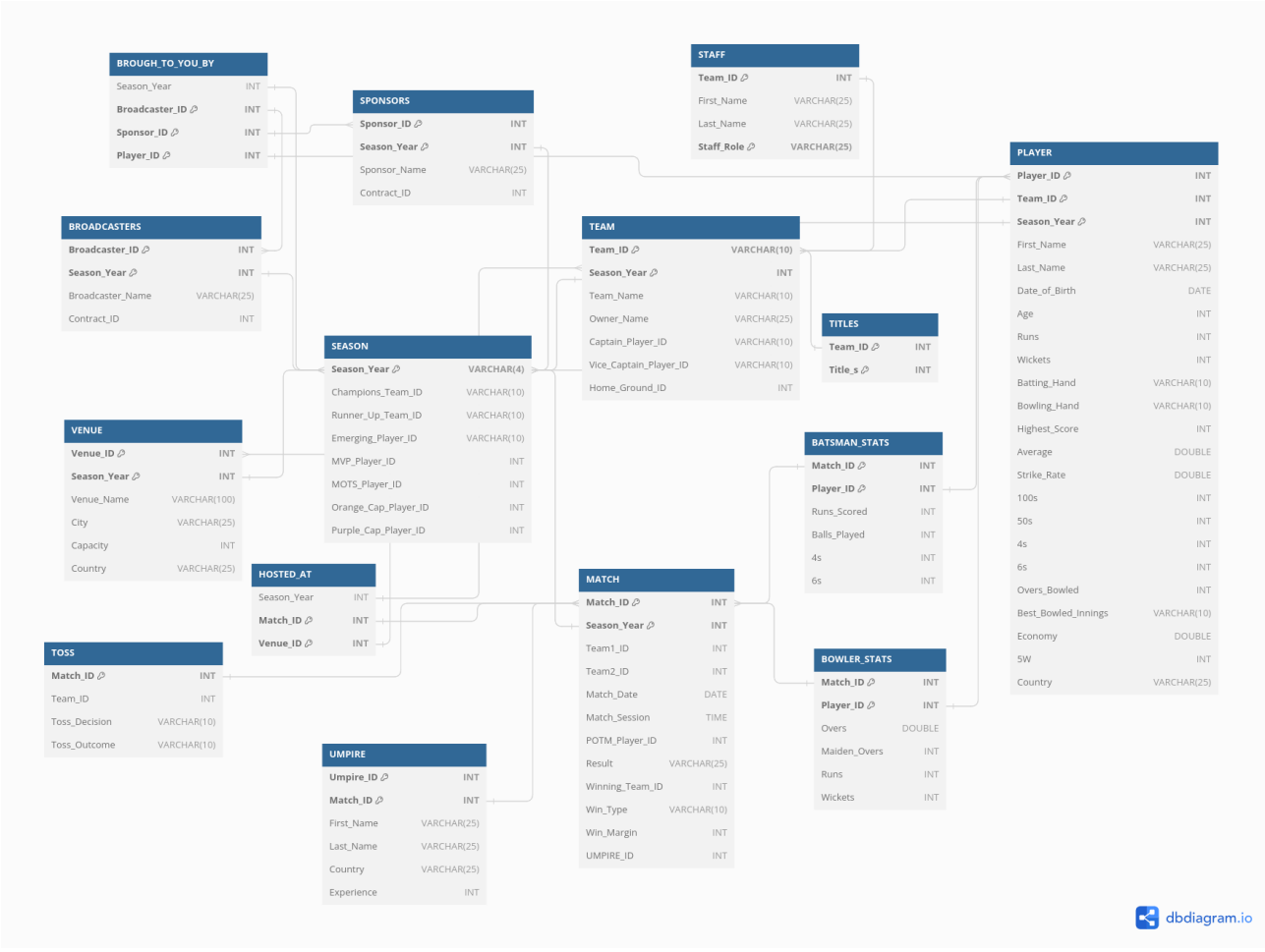

Figure 8: Final Relational Model

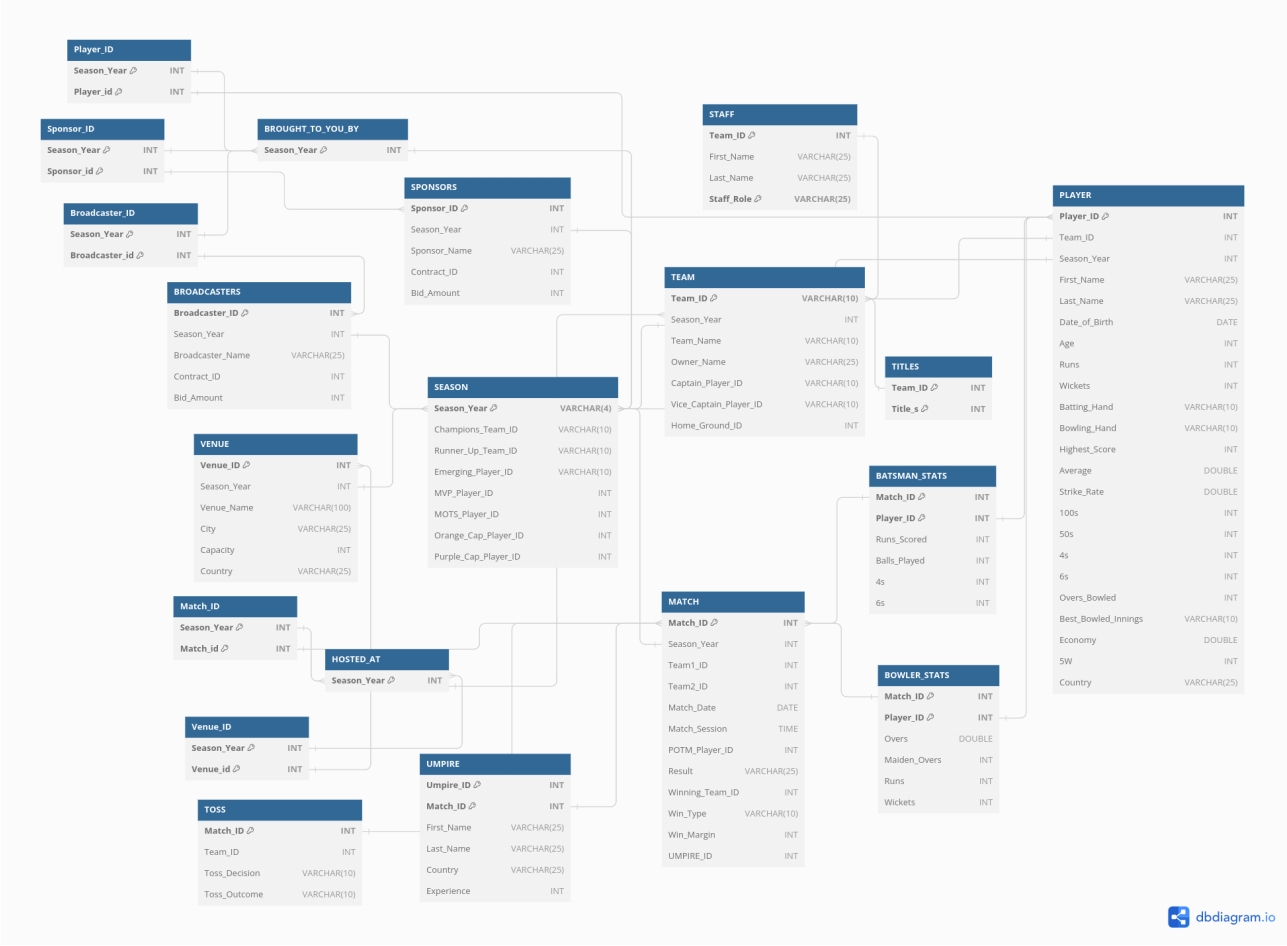
Player_ID
Season_Year INT
Player_id INT
Sponsor_ID
Season_Year INT
Sponsor_id INT
BROUGHT_TO_YOU_BY
Season_Year INT
STAFF
Team_ID INT
First_Name VARCHAR(25)
Last_Name VARCHAR(25)
Staff_Role VARCHAR(25)
SPONSORS
Sponsor_ID INT
Season_Year INT
Sponsor_Name VARCHAR(25)
Contract_ID INT
Bid_Amount INT
Broadcaster_ID
Season_Year INT
Broadcaster_id INT
PLAYER
Player_ID INT
Team_ID INT
Season_Year INT
First_Name VARCHAR(25)
Last_Name VARCHAR(25)
Date_of_Birth DATE
Age INT
Runs INT
Wickets INT
Batting_Hand VARCHAR(10)
Bowling_Hand VARCHAR(10)
Highest_Score INT
Average DOUBLE
Strike_Rate DOUBLE
100s INT
50s INT
4s INT
6s INT
Overs_Bowled INT
Best_Bowled_Innings VARCHAR(10)
Economy DOUBLE
5W INT
Country VARCHAR(25)
BROADCASTERS
Broadcaster_ID INT
Season_Year INT
Broadcaster_Name VARCHAR(25)
Contract_ID INT
Bid_Amount INT
TEAM
Team_ID VARCHAR(10)
Season_Year INT
Team_Name VARCHAR(10)
Owner_Name VARCHAR(25)
Captain_Player_ID VARCHAR(10)
Vice_Captain_Player_ID VARCHAR(10)
Home_Ground_ID INT
TITLES
Team_ID INT
Title_s INT
SEASON
Season_Year VARCHAR(4)
Champions_Team_ID VARCHAR(10)
Runner_Up_Team_ID VARCHAR(10)
Emerging_Player_ID VARCHAR(10)
MVP_Player_ID INT
MOTS_Player_ID INT
Orange_Cap_Player_ID INT
Purple_Cap_Player_ID INT
VENUE
Venue_ID INT
Season_Year INT
Venue_Name VARCHAR(100)
City VARCHAR(25)
Capacity INT
Country VARCHAR(25)
BATSMAN_STATS
Match_ID INT
Player_ID INT
Runs_Scored INT
Balls_Played INT
4s INT
6s INT
Match_ID
Season_Year INT
Match_id INT
MATCH
Match_ID INT
Season_Year INT
Team1_ID INT
Team2_ID INT
Match_Date DATE
Match_Session TIME
POTM_Player_ID INT
Result VARCHAR(25)
Winning_Team_ID INT
Win_Type VARCHAR(10)
Win_Margin INT
UMPIRE_ID INT
HOSTED_AT
Season_Year INT
BOWLER_STATS
Match_ID INT
Player_ID INT
Overs DOUBLE
Maiden_Overs INT
Runs INT
Wickets INT
Venue_ID
Season_Year INT
Venue_id INT
UMPIRE
Umpire_ID INT
Match_ID INT
First_Name VARCHAR(25)
Last_Name VARCHAR(25)
Country VARCHAR(25)
Experience INT
TOSS
Match_ID INT
Team_ID INT
Toss_Decision VARCHAR(10)
Toss_Outcome VARCHAR(10)
dbdiagram.io

Figure 9: 1NF

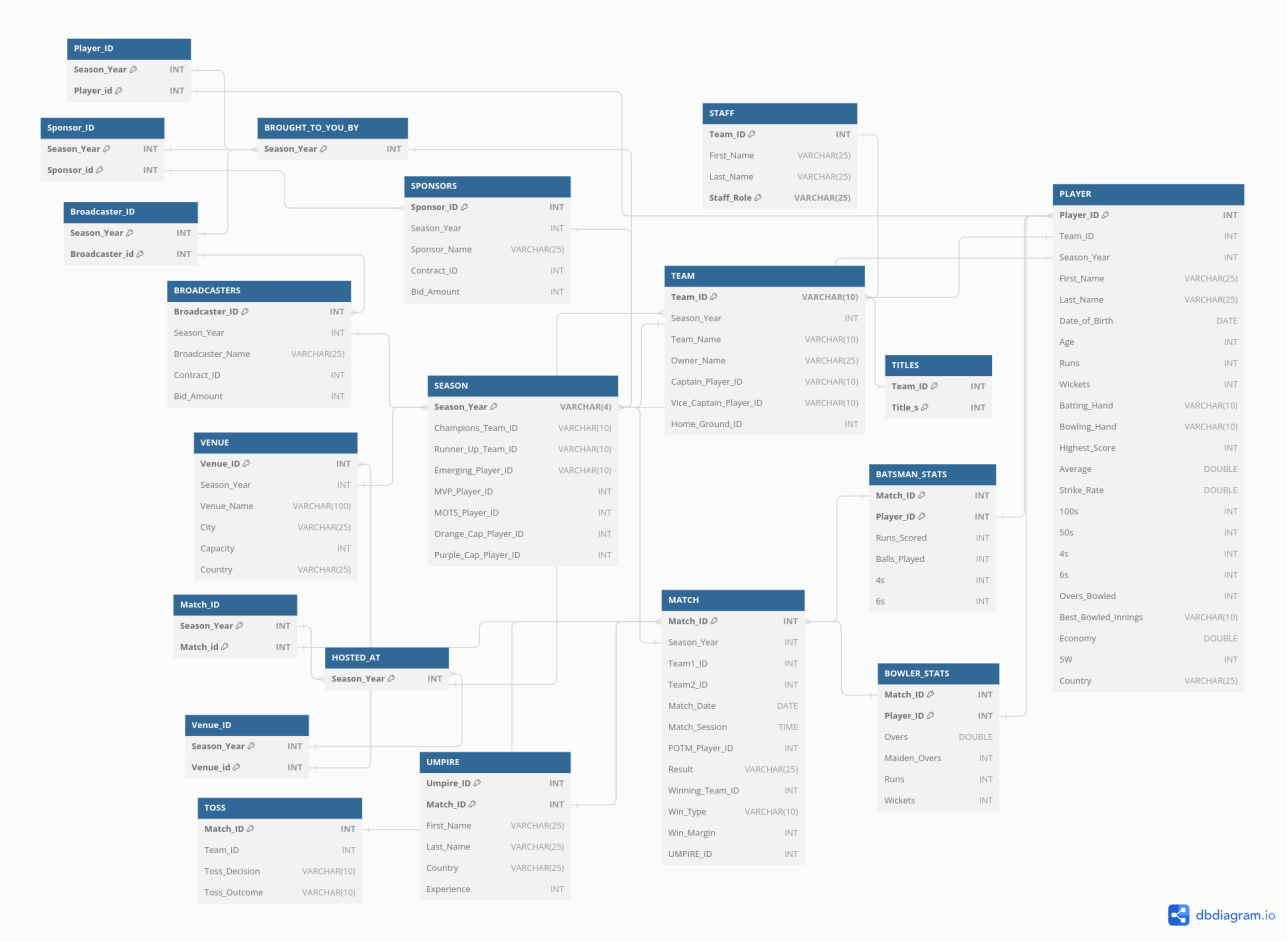

Figure 10: Final Relational Model after Normalization

4 Conversion to 2NF

A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

In our 1NF, we don't have any nonprime attribute which is not fully functionally dependent on its primary key.

Hence the Relational Model is already in 2NF.

5 Conversion to 3NF

To convert the given 2NF into Third Normal Form (3NF), we need to ensure that there are no transitive dependencies—that is, no non-prime attribute is transitively dependent on the primary key.

Since there are transitive dependencies in our relations, the Model is already in Third Normal Form.