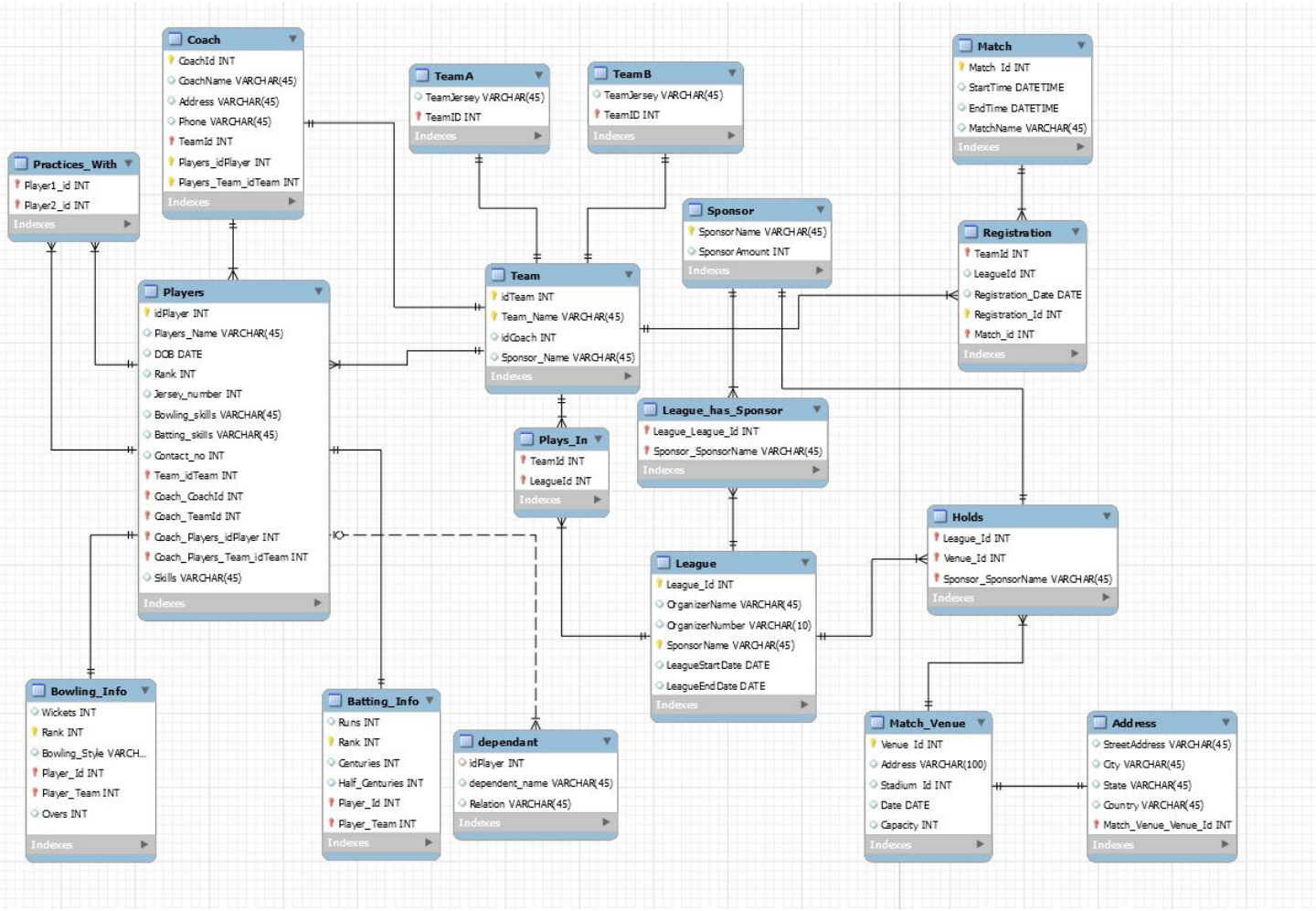
Indian Premier League

ALPHA: Prateek Vats
BETA: Arihant Jain

Entity Relationship Diagram



OVERVIEW

Here we have an overview of **Indian Premier League**(IPL). IPL is a professional Twenty20 cricket league in India contested during April and May of every year by teams representing Indian cities. The league was founded by the Board of Control for Cricket in India (BCCI) in 2007, and is regarded as the brainchild of Lalit Modi, the founder and former commissioner of the league.

ER diagram shown above illustrates a registration process in the IPL for each team , creating a relationship between **team** and **match**. After completion of match registration process, a team will be provided with a registration ID which is a primary key. Registration id and the other attributes 'registration_date', 'League_Id' are stored in the '**Registration** table.

Each team will have one **Coach**. Primary key of a coach is his coach ID. A coach will work with the team and its players for a match.

Players are the main attribute of a team. Each **player** has the following: a rank and one or multiple skills in batting and bowling. 'Dependent(s)' is someone who the player takes along during a league and is a weak entity, since there can be players who may not take a dependent along.

Each league will have matches which will be held between two teams:Team A and Team B. A team **plays in** a league. Each **league** has an organizer whose contact information is being stored, and can be held in multiple **venues**.

Venue_id is the primary key for each Venue and it also has an attribute that stores the date of the reservation.

In order to help with the league, financially, we need to have sponsor(s). There can be multiple sponsors for a league. For example, Pepsi, Vivo, Coca-Cola.

List of Entities

1. Player - It is the main entity in the tournament, and contains the following attributes:

- Player Name (Composite attribute - First Name, Last Name, Nick Name)
- Address
- Contact Number
- Player Rank
- Date of birth
- Player ID (Primary Key)
- Skills (Multivalued Attribute)
- Coach id
- Team ID (Foreign Key)
- Jersey_number

Constraints

- Primary Key Constraint - Player_Id is the primary key and cannot be null (not referring to null constraint)
- Referential Integrity Constraint (Foreign key constraint) - Team_Id and Coach_Id are foreign keys referencing Team and Coach relationships, respectively.
- Domain Constraint - Applies on each attribute. The datatypes for each attribute must be as shown in the schema.

2.Team - It has the following attributes.

- Team Id (Primary Key)
- Team_Name
- Coach ID
- Sponsor Name

Constraints

- Primary Key Constraint -Team ID is the primary key and cannot be null (not referring to null constraint)
- Domain Constraint - Applies on each attribute. The datatypes for each attribute must be as shown in the schema.

3.Coach - It has the following attributes.

- CoachId (Primary Key)
- CoachName
- Address
- Phone
- Team Id(Foreign Key)

Constraints

- Primary Key Constraint - CoachId is the primary key and cannot be null (not referring to null constraint)
- Domain Constraint - Applies on each attribute. The datatypes for each attribute must be as shown in the schema.

4.Team A - It has the following attributes:

- Team_Jersey
- Team ID (Primary Key)

Constraints

- Primary Key Constraint -Team A ID is the primary key and cannot be null (not referring to null constraint)
- Referential integrity Constraint (Foreign key constraint) -Team ID is a foreign key and must not be null.
- Domain Constraint - Applies on TeamJersey, value must be varchar.

5. Team A - It has the following attributes:

- Team_Jersey
- Team ID (Primary Key)

Constraints

- Primary Key Constraint -Team A ID is the primary key and cannot be null (not referring to null constraint)
- Referential integrity Constraint (Foreign key constraint) -Team ID is a foreign key and must not be null.
- Domain Constraint - Applies on TeamJersey, value must be varchar.

6.League - It has the following attributes:

- Leage Id (Primary Key)
- League Organizer Name
- League Organizer Number
- Sponsor's Name (Foreign Key)
- League Start Date
- League End Date

Constraints

- Primary Key Constraint - League ID is the primary key and cannot be null (not referring to null constraint)
- Referential integrity Constraint (Foreign key constraint) – Sponsors Name is a foreign key and must not be null.
- Domain Constraint - Applies on each Sponsor Name, Organizer Name, Organizer Number, League Start Date and League End Date.

7.Sponsor - It has the following attributes:

- Sponsor Name (Primary Key)
- Sponsor Amount

Constraints

- Primary Key Constraint - SponsorName is the primary key and cannot be null (not referring to null constraint)
- Domain Constraint - Applies on each Sponsor Amount, must be of type INT.

8. Match_Venue - It has the following attributes:

- Venue Id (Primary Key)
- Address (Multi Valued Attribute – Country, City, State, Address, Zipcode)
- Stadium_Id
- Date
- Capacity

Constraints

- Primary Key Constraint – Venue_Id is the primary key and cannot be null (not referring to null constraint)
- Domain Constraint - Applies on each Venue-Id, must be of type INT.

9. Match - It has the following attributes:

- Match Id (Primary Key)
- Match Name
- Start Time
- End Time

Constraints

- Primary Key Constraint – Match Id is the primary key and cannot be null (not referring to null constraint)
- Domain Constraint – Match Id is INT, Starttime and end time are datetime, and Matchname is varchar.

10. Weak Entity Type: Batting - It has the following attributes:

- Runs
- Rank (Primary Key)
- Centuries
- Half_Centuries
- Player id (Foreign Key)
- Player Team name (Foreign Key)

Constraints

- Primary Key Constraint –Rank is the primary key and cannot be null (not referring to null constraint)
- Referential integrity Constraint (Foreign key constraint) -TeamId & PlayerId are foreign keys and must not be null.
- Domain Constraint - Applies on each (like centuries & half centuries are INT so they must be integer) , batting style is varchar

11. Weak Entity Type: Bowling - It has the following attributes:

- Wickets
- Rank (Primary Key)
- Bowling Style
- Player Player Id (Foreign Key)
- Player team name(Foreign Key)
- Overs

Constraints

- Primary Key Constraint - Rank is the primary key and cannot be null (not referring to null constraint)
- Referential integrity Constraint (Foreign key constraint) -TeamId & PlayerId are foreign keys and must not be null.
- Domain Constraint – Wickets is INT, Rank is INT, Bowling style is varchar, Overs is INT

12. Weak Entity Type: Dependent - It has the following attributes:

- Dependent name
- Relation
- Player Id (Foreign Key)

Constraint

- Referential integrity Constraint (Foreign key constraint) – Player_id is a foreign key and must not be null.

List of relationship type

1.Plays_In - It has the following attributes

- TeamId (Foreign Key)
- LeagueId (Foreign Key)

Constraints

- Referential integrity Constraint (Foreign key constraint) -TeamId & LeagueId are foreign keys and must not be null.
- Domain Constraint - Applies on each – each value accepted must be of the type expected

2.Holds - It has the following attributes:

- League_Id(Foreign Key)
- Sponsor_name(Foreign Key)
- Venue_id (Foreign Key)

Constraints

- Referential integrity Constraint (Foreign key constraint) - LeagueId, Sponsor_Name, Venue_id (Foreign Key) are foreign keys and should not be null.
- Domain Constraint - each value accepted must be of the type expected

3.Practice_with - It has the following attributes.

- Player1_id (Foreign Key)
- Player2_id (Foreign Key)

Constraints

- Referential integrity Constraint (Foreign key constraint) - All are foreign keys and must not be null.
- Domain Constraint - Applies on each - each value accepted must be of the type expected

4.Registration - It has the following attributes:

- Registration_id (Primary Key)
- TeamId (Foreign Key)
- Match_id (Foreign Key)
- Registration_date

Constraint -

- Primary Key Constraint - Registration_Id is the primary key and cannot be null (not referring to null constraint)
- Referential integrity Constraint (Foreign key constraint) -TeamId and Match_id is a foreign key and must not be null.
- Domain Constraint - Applies on each (Registration-fees is INT only accepts type INT, Registration_date should be date type)

5.League_has_Sponsor - It has the following attributes:

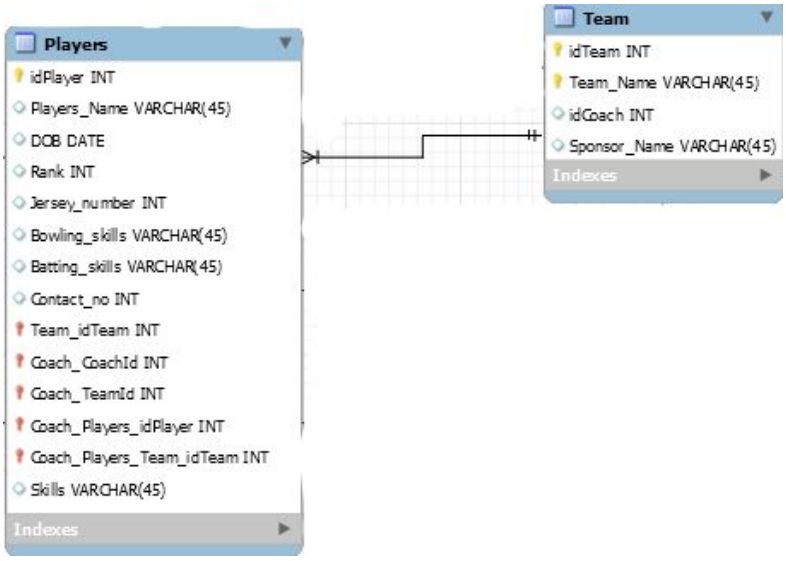
- League_Id (Foreign Key)
- SponsorName (Foreign Key)

Constraint -

- Referential integrity Constraint (Foreign key constraint) -League_Id and SponsorName is a foreign key and must not be null.
- Domain Constraint - Applies on each - each value accepted must be of the type expected

ALPHA's Contribution

1. A many-one relationship type with total participation on the one side

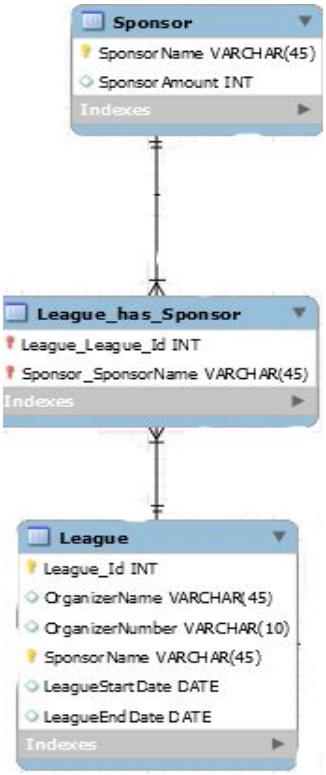


This diagram represents a many to one Relationship.(Players and Team)

Multiple Players play for a Team and without a player a team cannot be formed.

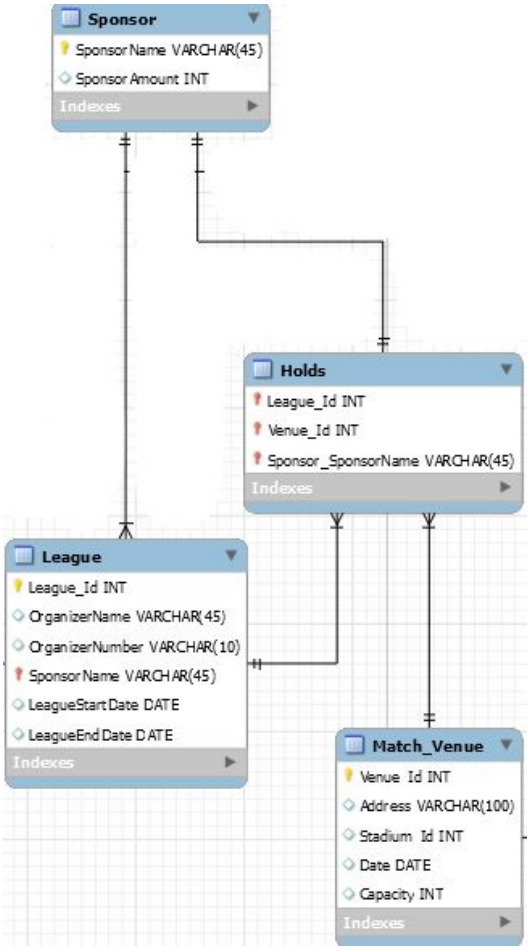
E.g. if we take cricket, it has 11 players which together form a team that is why a team is totally dependent on players.

2. Many to Many



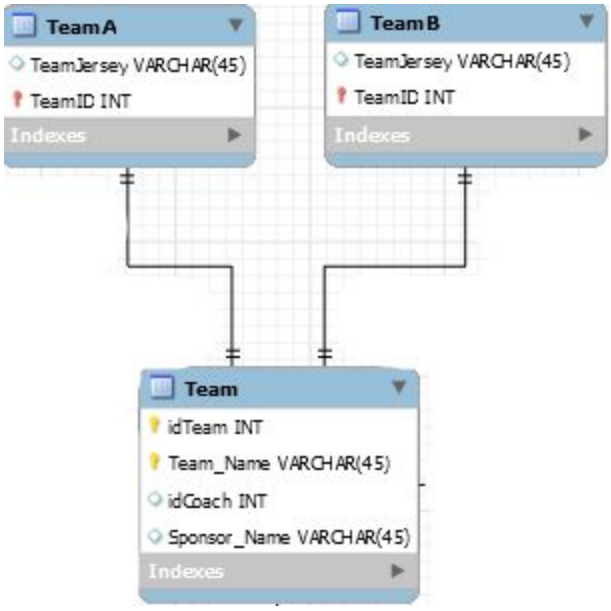
Here a certain league can have multiple sponsors, and a certain sponsor can sponsor many different leagues. Which builds a many to many relationship between the sponsor and the league.

3. Ternary Relationship



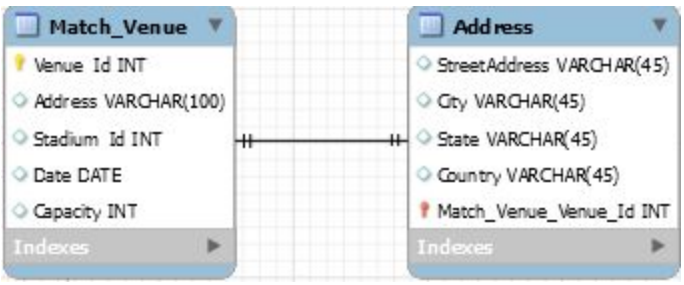
Here the ternary relationship is between **League, Match_venue, Sponsor**

4. A disjoint inheritance hierarchy



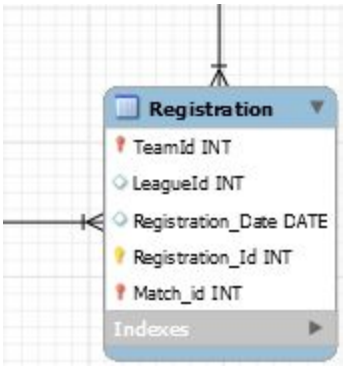
This is a Disjoint Set hierarchy. Team A and Team B inherit the entity - Team. A player cannot be a part of Team A and Team B together resulting in a mutually exclusive hierarchy.

5. A composite attribute



Here I have represented the address field in Match_venue as a multivalued attribute consisting of – StreetAddress, City, State, Country

6. A domain constraint

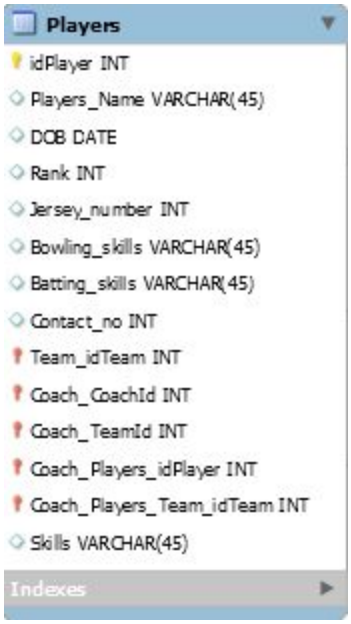


A domain constraint is defined as constraint values that are permitted for an attribute. Here are the following constraints-

TeamId, Match_id are Integer values

Registration_Date only accepts date values

7. Multivalued Attributes

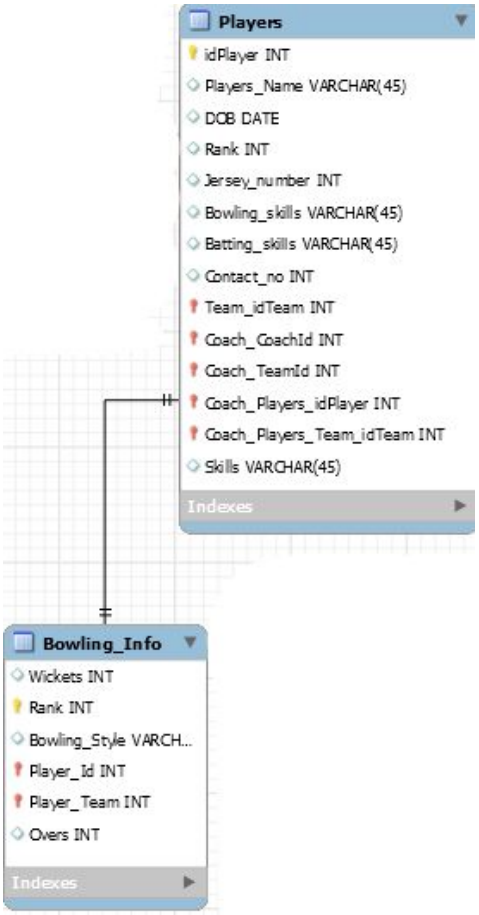


A multivalued attribute is the attribute which can have multiple values at a time. In the player entity, we have Bowling_skills, and Batting_skills. E.g a player could bowling skill as leg spin, off-spin,etc.

Player could have batting skills as middle order batsmen, opening, lower order batsman, right handed, left handed,etc.

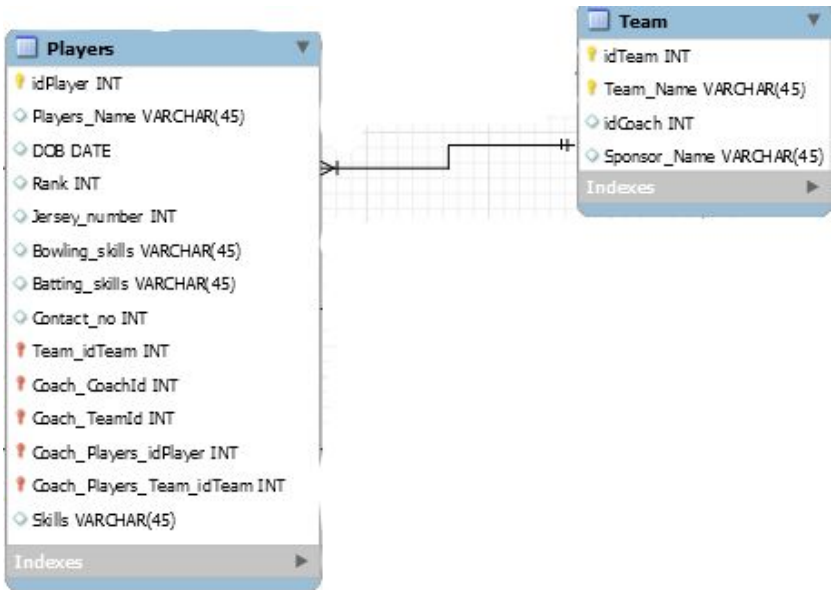
BETA's Contribution

1. A one-one relationship type with total participation on at least one side



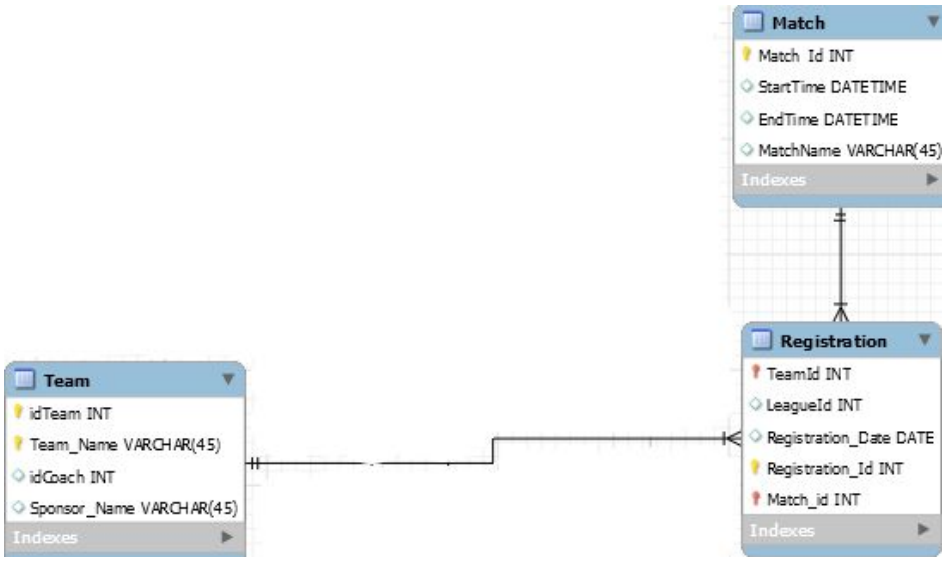
This is a one-one relationship between Players and the Bowling_Info. Each player has bowling stats and information, and each player has only one bowling stats and information.

2. A one-many relationship type with partial participation on both sides



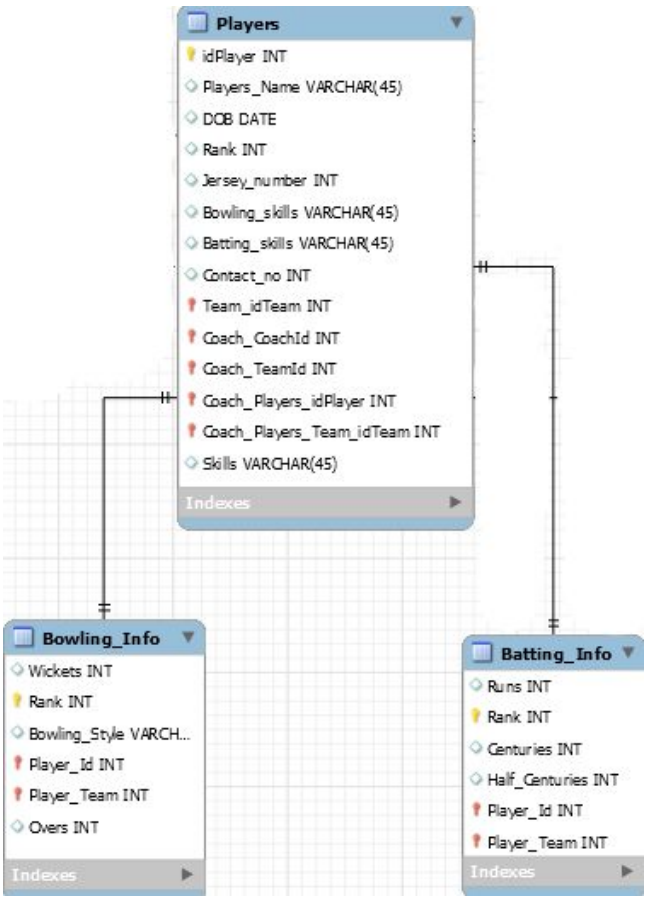
The diagram shows a one-many relationship with partial participation on both sides. One team has multiple players and it is not necessary for each player to be a part of a team, they can be not selected due to various reasons(injuries, off-form, other). Also, there can be change of number of players or in players (though the least number must be 11, the upper bound can be anything)

3. A relationship with an attribute(s)



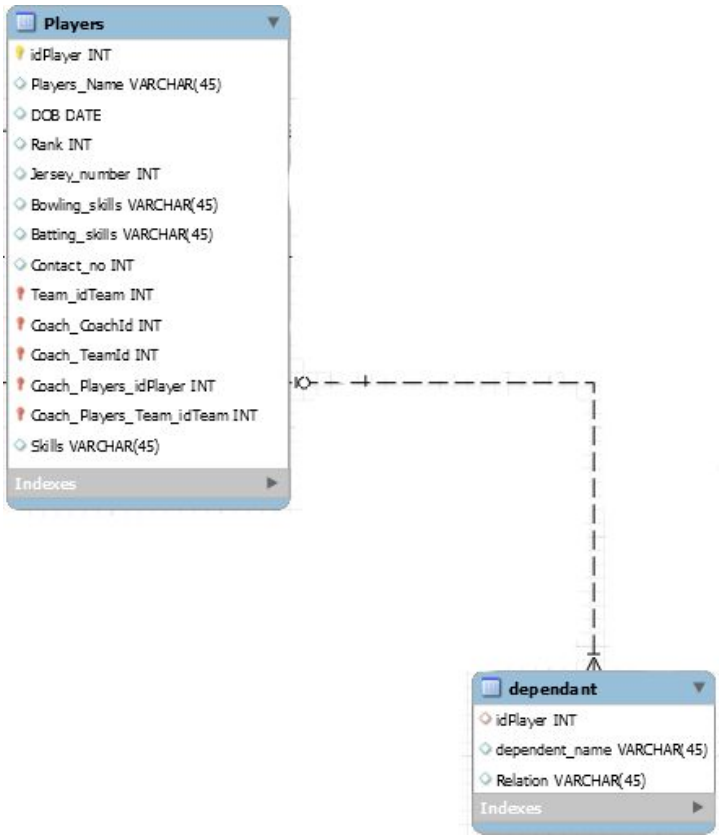
Here, registration is a relationship between Team and Match with the attributes Registration_Id, League_Id. It also contains the foreign keys TeamId and Match_id.

4. An overlapping inheritance hierarchy



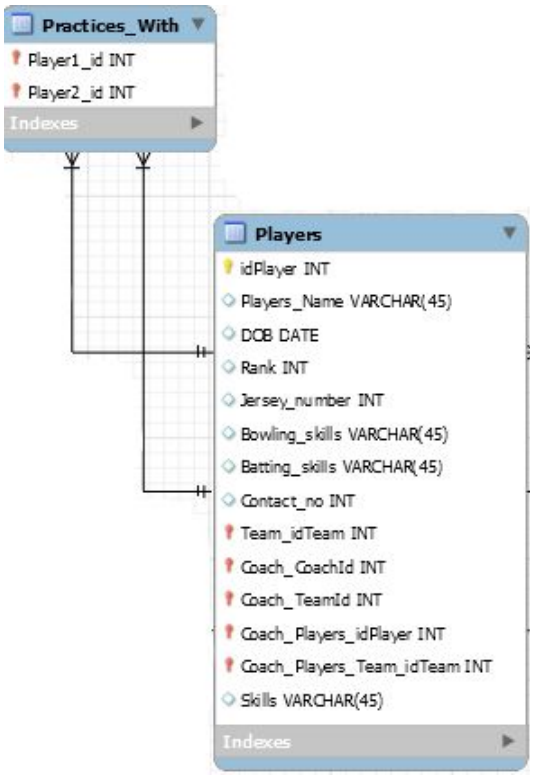
Here, Bowling_Info and Batting_Info are inherited from Players, A player can be a batsman, a bowler or both. So, a player can have attributes from each or from both(overlapping inheritance).

5. A weak-entity type



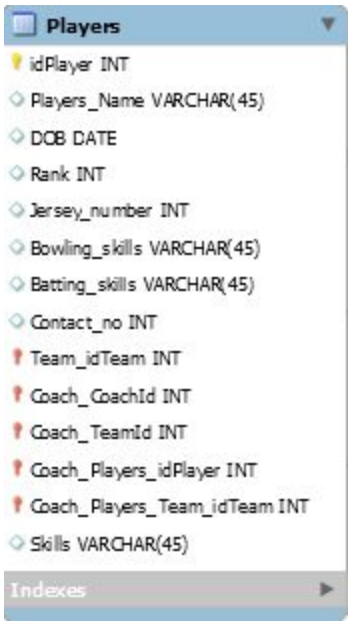
A weak entity has no primary key. Here the dependant is recognised through a player. A player can have multiple dependants and also it is possible that a player has no dependant.

6. A reflexive relationship type



A reflexive relation is when an entity has a relationship with a subset of itself. Here, a player can practice with another player. The assumption is that only two players practice together.

7. A check (integrity constraint on an attribute).



Here, check is applied on the Rank, i.e. 'Rank' > 0 AND 'Rank' < 1000. If rank of a player is more than 1000 then they are not qualified to play in the league. Check is used so that the integrity of the data remains intact, by filtering out faulty data.