Project Team 4

Database Topic:

Cricket League Data Management System

Database Purpose:

The purpose of this database is to maintain a major cricket league (Indian Premier League) data for managing the players, teams, staff, and other statistics like schedule, match results, player leaderboard, and points table. This database captures minute details like ball-to-ball score, bowling speed which could enable sports analysts and team coaches to provide strategic tips to the players. The league partners can also leverage this data for updating the scoreboard and other statistics on the league's website and mobile application.

Business Problems Addressed:

- Allow maintenance and tracking of data on Players, Teams, Staff, Venue
- Provide data on ball-to-ball scores and other details like bowling speed, runs, extra-runs, wickets
- Maintain and track several statistics of players like centuries, half-centuries, five wicket hauls, maiden overs
- Help provide history of past and current edition match results, toss results, team scores, points table
- Understand statistics like team performances at home venue
- Venue Insights like level highest and lowest totals
- Provide details on various player level performances in batting and bowling powerplays
- Maintain data on various awards distributed to players on match level and edition level

Business Rules:

- Each match may have only one venue
- Each match will involve two teams
- Each match will have only 22 players and 11 from each team
- Each innings should have up to 20 overs
- Each match has 2 innings
- Each innings may have up to 10 wickets

- Each team in a match gets a result of win/ lose/ draw
- Each player may have a subcategory (batters- left hand, right hand… bowlers-spin, fast, medium)
- Each player will be mapped to one team
- Each venue may have a home team
- Every team may have multiple staff
- Each over may have only 6 legal deliveries
- Each delivery has a bowler and a batter
- Each run type may have 0-6 runs usually
- Each staff member can be mapped to only one team

Design Decisions:

Entity Name	Purpose of the Entity	Relation with other Entities
Schedule	The Schedule entity is included because a sports league will have a schedule for every edition and it will include the details of the match number, the two teams that are involved in the match, date of the match, start time and end time of the match, venue for the match and the edition of the league.	The Schedule entity has two non-identifying many-to-one relationship with the Team entity as two teams will be involved in a match. The two foreign keys of Team entity are present in this entity There is a one-to-one non-identifying relationship with MatchResult entity Schedule has one-to-many non-identifying relationships using the primary key MatchID with the Delivery entity Schedule also has many-to-one non-identifying relationship with the venue entity

Venue	Every match must take place in a venue and without a venue there will be no game. So, the Venue entity is included. It has the details of venue name, venue Id, the city of the venue, the capacity of the venue, home team id (to which team the venue belongs to) and the country the venue belongs to.	Venue entity has one-to-many identifying relationship VenueStatistics entity using the primary key VenueID. It has one-to-many non-identifying with the Schedule entity which has the primary key of Venue as the foreign key in it. It cannot be empty in Schedule entity. It also has a many-to-one non-identifying relationship with the Team entity.
MatchResult	After every match, depending on the runs scored the winner is decided. The match results are stored in the MatchResult entity, and it has the attributes Toss winner, match id, team1 score, team 2 score, team 1 wickets, team 2 wickets, winning team, result type	MatchResult entity has two non-identifying many-to-one relationships with the Team entity because two teams will be involved in a match with result. It has one-to-one non-identifying relationship with the Schedule entity using the MatchID which is the primary key for both the entities. It is also involved with one-to-many identifying relationship with MatchAwards entity.
PointsTable	As the edition goes on, the points table will be updated according to the results of the matches. This helps in tracking the winner, runners, and positioning of the teams after every match. It also helps in deciding the team that goes to playoffs by checking the net run rate in case of tie in points which is the criteria that changes according to the winning margins. PointsTable has the information of the team Id, rank, edition, net run rate and the points.	PointsTable entity has non-identifying many-to-one relationship with the Team entity using the primary key of the Team entity TeamID which is a foreign key in the PointsTable.

Team	In a cricket league, the main components are the matches between the teams, and it is usually between two teams. So, the purpose of the Team entity is to store all the team's details participating in the league. It has the Team id, team name, owners, primary sponsors, team worth, the debut edition of the team, captain and vice-captain of the team in the league.	Team entity has two non-identifying one -to-many relationship with the Schedule entity by means of the TeamID which is the primary key of Team entity and foreign key in the Schedule entity. It also has two non-identifying one -to-many relationship with the MatchResult entity as two teams' involvement is mandatory in a result scenario. And the primary key TeamID in Team is foreign key in MatchResult as WinnerTeam. It also involved in one-to-many non-identifying relationship with PointsTable entity, Venue entity and Player entity by means of the TeamID which is the foreign key in both Player and PointsTable entities. And two one to one relation with player entity. It has one more one-to-many non-identifying relationship with Staff entity as each staff will be associated with a team.
MatchAwards	After each match depending on the performance of the players in the match, the awards will be presented to them and to maintain a track of all such awards the MatchAwards entity is designed. It has the attributes of the MatchID, category of the award (Player of the Match, Super Striker etc), and the player ID associated with the award.	The MatchAwards entity has identifying many-to-one relationship with the MatchResult using the MatchID as primary key in both the entities. It is also involved in the non-identifying many-to-one relationship with the Player entity and category entity using the primary key of Player entity as foreign key in the entity.

Player	Every team has a bunch of players and each player has a profile to be stored. This entity has details like player id, first name, last name, categoryID (batter, bowler, allrounder), sub-categoryID, date of birth, the team he/she belongs to, country and the Date of birth of the player.	Player entity is the base entity for the different kind of players (Batters, bowlers, wicket keepers) and is involved in the identifying one-to-one relationship with all the three entities Batter, Bowler, and WicketKeeper entities. All the three uses the Player ID which is primary key in the PlayerID as the foreign key in Bowler, Batter, and WicketKeeper entities. Player entity is also involved in the one-to many non-identifying relationships with the TournamentAwards, PlayerLeaderboard, Delivery and MatchAwards entities. It uses the primary key as the foreign keys in those entities. The key PlayerID cannot be empty in any of those entities. It is in one many-to-one non-identifying relationship and two one-to-one relationship with the Team entity. It uses the TeamID which is the primary key in the Team entity as the foreign key in the Player entity. It also has many-to-one non identifying relationship with category entity

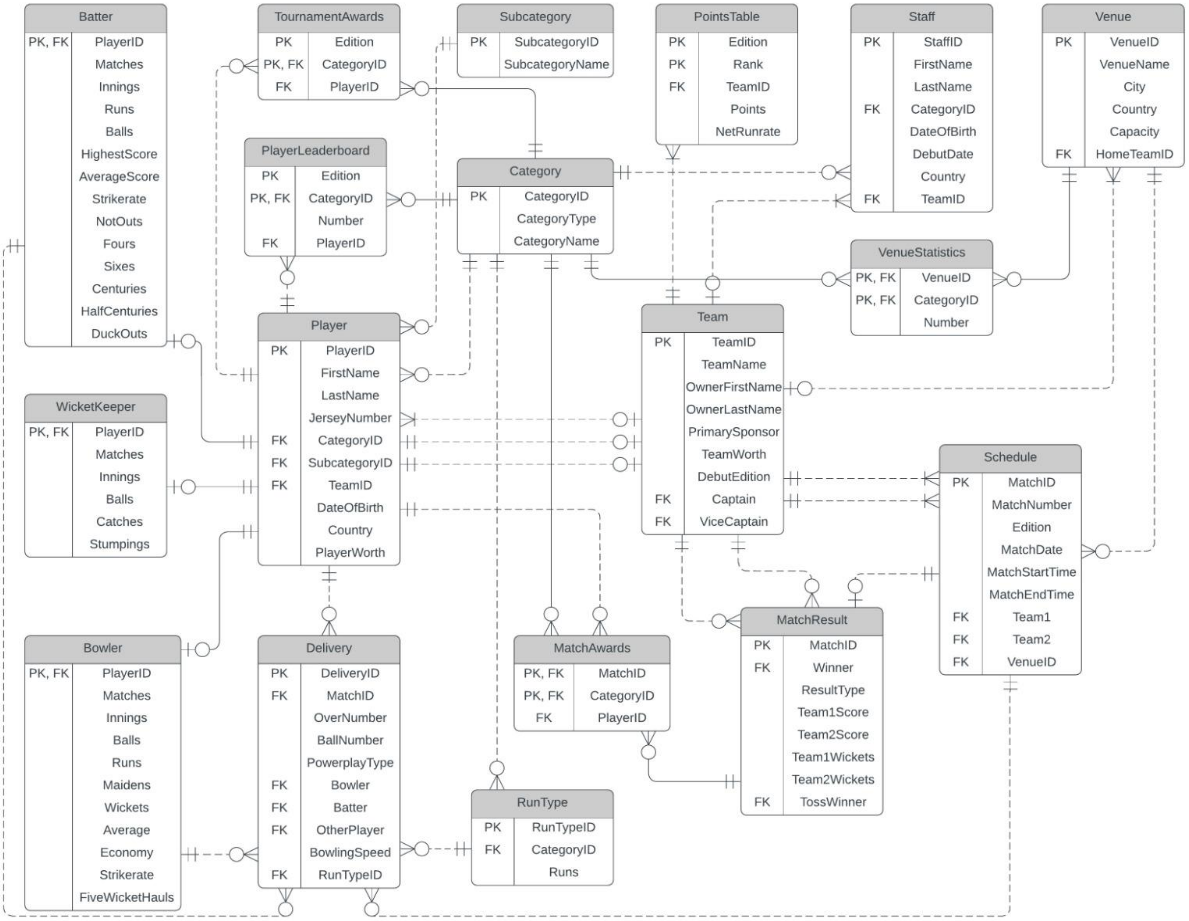
Batter	In the players, one of the categories is batters. They have a major role in the game and consists of some unique details which make them unique from the rest and they are highest individual score, average score, number of not outs, fours and sixes count, centuries (100+ runs), half centuries (50+ runs), duck outs (0 runs) and other information like player id, matches, innings played, total runs and number of balls faced.	Batter entity has one-to-one identifying relationship with the Player entity by the means of the primary key of the Player entity which is both primary and foreign key in the Batter entity. It is involved in non-identifying one-to-many relationship with the Delivery entity.
Bowler	One more type of the players is the bowler, by using the bowler entity. It is a derived entity from the Players entity. it helps in recording the information and the statistics of the bowler. It has the information of the bowler id, matches, runs conceded, balls bowled, wickets taken, maidens (0 runs conceded overs), average of the runs conceded per wicket, average of balls delivered per wicket which is also called strike rate, and the five wicket hauls.	Bowler entity is in one-to-one identifying relationship with the Player entity using the primary key of the Bowler entity as the primary key of the Player entity. And involved in non-identifying one-to-many relationship with the Delivery entity.
WicketKeeper	The last type of the player is the wicketkeeper player. The purpose of the WicketKeeper entity is to handle the information of the wicket keepers in the tournament. It has the information of the Player id, matches played, the number of catches, stumpings count, and the number of balls.	WicketKeeper entity has identifying one-to-one relation with the Player entity, as every wicket keeper is a player. The Player ID can never be empty or null.

PlayerLeaderboard	PlayerLeaderboard entity is used to maintain the list of the leading run scorers, wicket takers etc. in the league. It has the information of the category of the leaderboard, edition of the league, number of the category related field and the player id.	PlayerLeaderboard entity is involved in the many-to-one relationship non-identifying with the Player entity using the PlayerID as the foreign key in the entity. It also has many-to-one identifying relationship with category entity.
TournamentAwards	TournamentAwards entity is used to track the awards in every edition of the league using attributes like edition, categoryID, and the player ID.	The entity is in many-to-one non-identifying relationship with the Player entity. It also has many-to-one identifying relationship with category entity.
VenueStatistics	This entity is required to maintain the statistics for each venue like highest score, average score etc. It can help in predicting the results during the match using all the previous stats. It has the attributes of venue id, categoryID, and the numbers.	The VenueStatistics entity is in the identifying many-to-one relationship with the Venue entity and Category entity. It is by the means of the VenueID which the primary key in both the entities.

Staff	Every team has their own staff and there are different categories of them fitness coach, bowling coach, batting coach, medic, analyst, and the purpose of this entity is to handle the information of them. It has the staff id, staff first name, last name, team id, birthdate, debut date and their country.	The Staff entity relationship is with the Team entity which is non-identifying many-to-one relationship. The primary key of the Staff entity (StaffID) is the foreign key in the Team entity. It also has many-to-one non identifying relationship with category entity.
RunType	One of the main components of the cricket game is the runs and there are many ways to score a run (it is ones, twos, threes, running between wickets, boundaries like fours and sixes, extras like wide, no-ball, leg byes, byes, out). This run type also has the information about dismissals/wickets. It has the attributes of the run type id, category of the runs and the runs for every category.	The RunType entity is involved in the non-identifying one-to-many relationship with the Delivery entity. It uses the primary key of the RunType (RunTypeID) as the foreign key in the Delivery entity. It also has many-to-one non identifying relationship with category entity.
Delivery	One more integral part of the game is the delivery which often results in the runs and helps the team in setting a target. So, to track the match delivery-to-delivery, the Delivery entity will help in fulfilling the requirement. It has the attributes such as delivery id, match id, over number, ball number, bowler name, powerplay type, batsman facing the delivery, the speed of the delivery by the bowler and the run type to calculate the runs.	The Delivery entity is participated in non-identifying many-to-one relationship with the RunType entity using the primary key of the RunType entity as the foreign key of the Delivery entity. It is involved in non-identifying relationship many-to-one with Bowler, Batter, and Player entities where the primary keys of those entities are foreign keys in Delivery entity. It is also part of the many-to-one non identifying relationship with the Schedule entity.

Category	It has the different categories information for the data related to entities -VenueStatistics, Player, Staff, TournamentAwards, RunType, PlayerLeaderBoard, MatchAwards	It has non-identifying one to many relationships with RunType, Player, Staff entities. It also has 3 identifying one to many relationships with the MatchAwards, VenueStatistics and TournamentAwards
Subcategory	It has the information related to all the sub-categorical information for the categories mentioned in the Player entity	Maintains non-identifying one-to-many relationship with the Player entity

ER Diagram:



Batter	
PK, FK	PlayerID
	Matches
	Innings
	Runs
	Balls
	HighestScore
	AverageScore
	Strikerate
	NotOuts
	Fours
	Sixes
	Centuries
	HalfCenturies
	DuckOuts

TournamentAwards	
PK	Edition
PK, FK	CategoryID
FK	PlayerID

Subcategory	
PK	SubcategoryID
	SubcategoryName

PointsTable	
PK	Edition
PK	Rank
FK	TeamID
	Points
	NetRunrate

Staff	
PK	StaffID
	FirstName
	LastName
FK	CategoryID
	DateOfBirth
	DebutDate
	Country
FK	TeamID

Venue	
PK	VenueID
	VenueName
	City
	Country
	Capacity
FK	HomeTeamID

PlayerLeaderboard	
PK	Edition
PK, FK	CategoryID
	Number
FK	PlayerID

Category	
PK	CategoryID
	CategoryType
	CategoryName

VenueStatistics	
PK, FK	VenueID
PK, FK	CategoryID
	Number

Player	
PK	PlayerID
	FirstName
	LastName
	JerseyNumber
FK	CategoryID
FK	SubcategoryID
FK	TeamID
	DateOfBirth
	Country
	PlayerWorth

Team	
PK	TeamID
	TeamName
	OwnerFirstName
	OwnerLastName
	PrimarySponsor
	TeamWorth
	DebutEdition
FK	Captain
FK	ViceCaptain

WicketKeeper	
PK, FK	PlayerID
	Matches
	Innings
	Balls
	Catches
	Stumpings

Schedule	
PK	MatchID
	MatchNumber
	Edition
	MatchDate
	MatchStartTime
	MatchEndTime
FK	Team1
FK	Team2
FK	VenueID

Bowler	
PK, FK	PlayerID
	Matches
	Innings
	Balls
	Runs
	Maidens
	Wickets
	Average
	Economy
	Strikerate
	FiveWicketHauls

Delivery	
PK	DeliveryID
FK	MatchID
	OverNumber
	BallNumber
	PowerplayType
FK	Bowler
FK	Batter
FK	OtherPlayer
	BowlingSpeed
FK	RunTypeID

MatchAwards	
PK, FK	MatchID
PK, FK	CategoryID
FK	PlayerID

MatchResult	
PK	MatchID
FK	Winner
	ResultType
	Team1Score
	Team2Score
	Team1Wickets
	Team2Wickets
FK	TossWinner

RunType	
PK	RunTypeID
FK	CategoryID
	Runs