Description of the Indian Premier League Database

Domain Description

The IPL tournament consists of individual T20 matches. A match in the tournament is uniquely determined by an ID.

A T20 cricket match is a 20-over per side game played between two teams. A coin toss is performed at the beginning of the game and the team winning the toss chooses whether to bat or to field. A game ends with one of the teams winning by either a certain number of runs or a certain number of wickets. Apart from the above details, we would like to maintain the match date and the match time.

A team is uniquely identified by its team ID. Each team in the IPL represents a city in India (For example, Chennai Super Kings represents Chennai, Delhi Capitals represent Delhi, etc). A team consists of players and other support staff who play the role of coach, analyst, etc. The team is owned by multiple entities each of whom have a particular stake.

A team consists of multiple players. Within a team the player may occupy the post of a captain or a vice-captain. For the team to be balanced the team requires a good mix of varied skills of batting, bowling, wicket-keeping, all-rounder. IPL has limitations that a team can only consist of 4 non-Indian players, hence the nationality of the player also plays an important role in the team formation. The player can be uniquely identified by his player_id, however for convenience we would also like to maintain his name and age.

Each match requires 4 match officials : 3 umpires and a referee. Each of the umpires has a unique rank (First, Second and Third). The first and second umpires are on-field whereas the third umpire comes into play only in case of a review from either team regarding a decision made by the on-field umpires.

Each match official is uniquely identified by official ID. We also want to maintain the official's name, age and number of years of work experience. The match official is provided with accommodation.

A match is held at one of the many cricket stadiums in India or in the UAE. Each stadium is uniquely identified by its ID. Apart from this, we would like to maintain the name of the stadium, the location and its capacity. The location of the stadium consists of country, state and city. Each stadium is usually the home ground for one of the teams participating in the tournament.

A T20 match consists of two innings. Each innings consists of 20-overs. Each over has 6 balls. Therefore, each ball of a match is uniquely identified by innings number, over number and ball

number in the over. In each ball, one of the two batsmen (striker / non-striker) faces the bowler assigned to bowl that over. Each ball can result in the following outcomes,

1) A wicket which can be of many types (bowled, caught, run-out) involving a number of other team members of the fielding side apart from the bowler.
2) Runs scored by the striker (example : 0, 4, 6)
3) Extra runs (example : wide, no-ball)

Apart from these we also maintain the batting team, bowling team, who was the bowler, who were the batsmen.

The various teams playing in the IPL are funded financially or through products by various companies called sponsors. The sponsors in return get advertisement by having logos on the jerseys of the players. A team can have multiple sponsors whereas one sponsor is usually associated with a single team. The sponsors have a unique identification number.

Every IPL team is owned by certain people who pay for the salaries of the players and the staff. They earn money through the selling of tickets. Each owner has a stake in the team which indicates his financial involvement in the team. We also store the owner id and the name.

The support staff forms the backbone of the team and provides required support to enhance the team's potential. The support staff have a unique identification and a name. There are several diverse areas of support required for IPL teams. These roles include physio, analyst, nutritionist, coach etc.

The players also require good accommodation after a tiring match. For the same they are usually provided accommodation close to the stadium in a hotel. The hotel also has a unique hotel id. The capacity is usually a good indicator of the likelihood of the hotel being vacant at a particular time and hence we also want to maintain that along with the hotel address.

Entities and Relationships

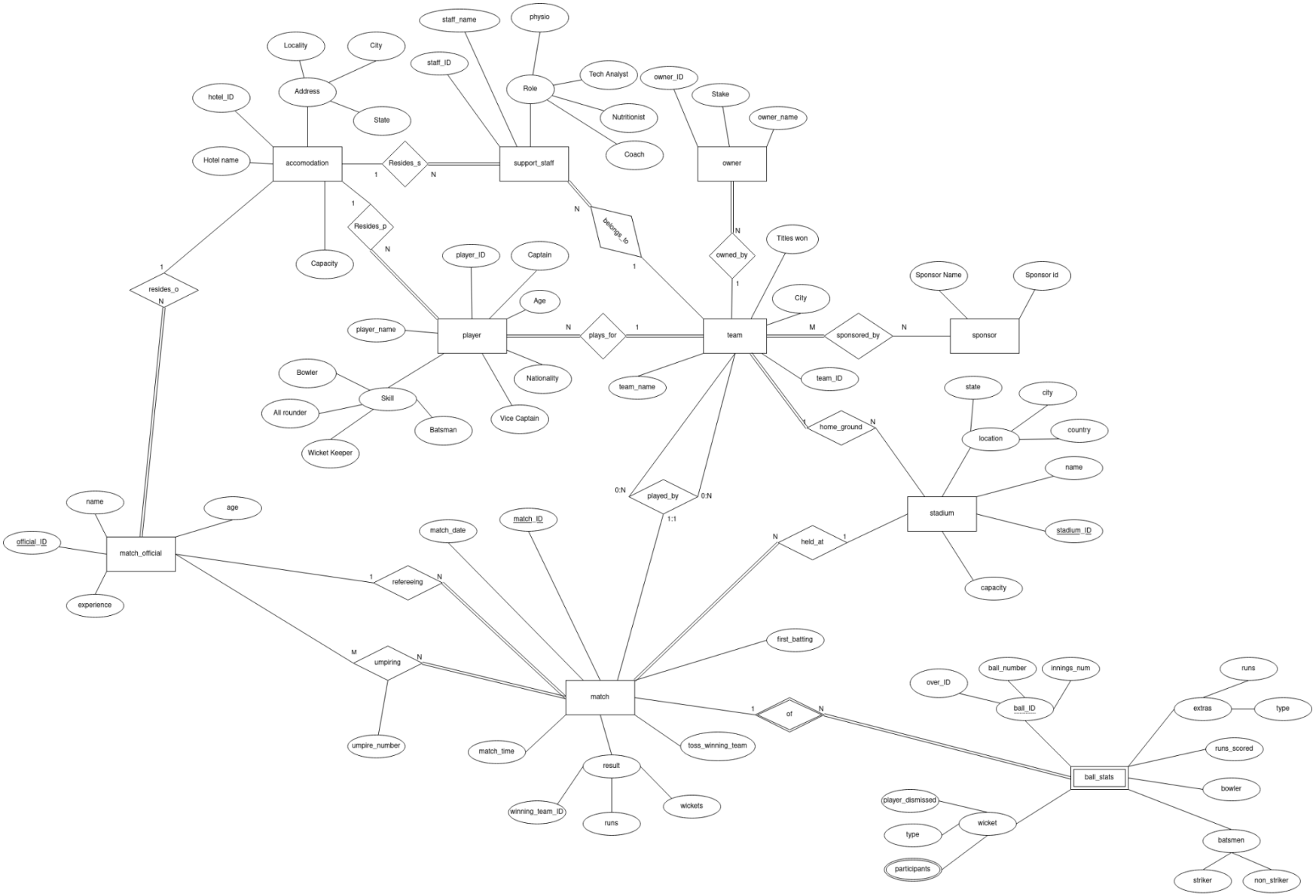
Entities

1. Match_official :
    a. official_ID
    b. name
    c. age
    d. experience
2. Match:
    a. match_ID
    b. first_batting
    c. match_date
    d. match_time
    e. toss_winning_team
    f. result(winning_team_ID, runs, wickets)

3. Ball_stats :
    a. ball_ID(over_ID, ball_number, innings_num)
    b. batsmen(striker, non-striker)
    c. bowler
    d. runs_scored
    e. extras(type, runs)
    f. wicket(player_dismissed, type, participants)
4. Stadium:
    a. stadium_ID
    b. name
    c. country
    d. location(state, city, capacity)
5. Team:
    a. team_ID
    b. team_name
    c. titles won
    d. city
6. Player
    a. player_id
    b. captain
    c. age
    d. nationality
    e. vice-captain
    f. player_name
    g. skill (batsman, wicket-keeper, all-rounder, bowler)
7. Sponsor
    a. sponsor ID
    b. sponsor Name
8. Owner
    a. owner ID
    b. stake
    c. owner Name
9. Accomodation
    a. hotel ID
    b. hotel name
    c. address(Locality, CIty, State)
    d. capacity
10. Support Staff
    a. staff id
    b. staff name
    c. role(physio, tech analyst, nutritionist, coach)

Relationships

1. Refereeing:
    a. match_official
    b. match
2. Umpiring:
    a. match_official
    b. match
    c. umpire_number
3. Held_at:
    a. match
    b. venue
4. Of:
    a. match
    b. ball_stats
5. Plays_for:
    a. team
    b. player
6. Resides_p:
    a. accommodation
    b. player
7. Sponsored_by:
    a. team
    b. sponsor
8. Owned_by:
    a. team
    b. owner
9. Belongs_to:
    a. support_staff
    b. team
10. Resides_s:
    a. accommodation
    b. support_staff
11. Resides_o
    a. accommodation
    b. match_official
12. Home_ground
    a. team
    b. stadium
13. Played_by
    a. match
    b. team
    c. Team


staff_name
physio
Locality
City
staff_ID
Address
Role
Tech Analyst
owner_ID
hotel_ID
State
Stake
Nutritionist
owner_name
Hotel name
accomodation
Resides_s
support_staff
Coach
owner
1
N
Resides_p
belongs_to
Capacity
owned_by
Titles won
player_ID
Captain
Sponsor Name
Sponsor id
resides_o
Age
City
player_name
player
plays_for
team
sponsored_by
sponsor
Bowler
Nationality
team_name
team_ID
Skill
All rounder
Batsman
Vice Captain
Wicket Keeper
state
city
home_ground
location
country
name
0:N
played_by
1:1
stadium
stadium_ID
official_ID
age
match_date
match_ID
held_at
capacity
match_official
refereeing
experience
first_batting
umpiring
over_ID
ball_number
innings_num
runs
match
of
extras
type
ball_ID
umpire_number
match_time
result
toss_winning_team
runs_scored
ball_stats
bowler
player_dismissed
wicket
batsmen
winning_team_ID
runs
wickets
type
participants
striker
non_striker

ER Diagram

Brief Description of Entities and Relationships

We have a team entity which represents the teams participating in the IPL. Each team in the IPL represents a city in India (For example, Kolkata Knight Riders represents Kolkata, Mumbai Indians represent Mumbai, etc). A team consists of players and other support staff who play the role of coach, analyst, etc. For example, Hardik Pandya is a player in the Mumbai Indians Team, and Anil Kumble is a coach in the Punjab Kings team.

A team is owned by certain owners, each of whom have a name and a certain stake in the team, which are the attributes of the owner entity. For example, Shilpa Shetty is an owner of the Rajasthan Royals and has 11.7% stake.

There are many companies which sponsor teams in the IPL. The sponsor entity showcases these companies. For example, Chennai Super Kings is sponsored by India Cements. There are companies like Jio which sponsor multiple teams in the IPL and hence the relation between sponsor and team is one to many.

We have a player entity which has various attributes like name, age etc. There are also binary attributes representing whether he/she is a captain, vice-captain etc. The skill is a composite attribute representing whether the player is primarily a batsman, bowler or wicket-keeper. For example, Virat Kohli or Rohit Sharma or Sachin will have 1 in captain and batsman in skill whereas MS Dhoni will have wicket-keeper in his skill attribute.

We assume the whole team resides in the same hotel , that is why we have a many to one relationship between accomodation and player with many players of the same team residing in the same place.

Teams can have one or more home stadiums, for example RCB has the Chinnaswamy stadium as their only home ground whereas Kings XI Punjab have the PCA Mohali Stadium and HPCA Dharamshala stadium as their home grounds. Since every team compulsorily has a home stadium , there is a total participation of team entities in the home_ground relation.

The support staff entity has a composite attribute **role** regarding the work in the team. For example, Nitin Patel (support staff for MI) will have physio in the role attribute. A team usually has multiple support staff so we have a many to one relationship belongs_to between support staff and team entities. Also each support staff will always be a part of some team , that is why we have used total participation in the relationship.

We have a match entity which represents a specific match conducted/to-be conducted in the tournament (the entity is named as ‘Matches’ in the schema so as to avoid collision with the SQL Keyword ‘Match’). Each match is uniquely defined by a match_id which acts as it’s primary key. The attributes first_batting and toss_winning_team are both necessary according to the rules specified.

For example, in a match between RCB and MI, MI can win the toss first and choose to bowl first in which case, toss_winning_team is MI and first_batting is RCB. The result of the match is defined by the team which wins and also by how many runs or wickets. For example, the result of the above match could be MI winning by 4 wickets, or RCB winning by 5 runs. To signify the teams playing in every match, we decided to use a ternary relationship played_by. Played_by is a recursive ternary relationship between two teams and a match. The participation constraint on the team side is 0:N as a team need not take part in any match (to handle special cases) and can take part in at most N matches. On the match side, the participation constraint is 1:1 because each match appears in 1 and exactly 1 entry of the played_by relationship. To denote the venue for every match, we have a played_at relationship between match and stadium entity sets. This relation is N-to-1 with total participation at the match side. This is because many matches can be held at a given venue and each match has to be played at a venue.

To track the statistics per ball, we have a weak entity called ball_stats. This has to be a weak entity because it's existence is dependent on that of the entity match. For example, if a match does not occur at all (cancelled or rained out), no ball_stats entry can exist for that match. We have correspondingly defined a 1-to-N relation "of" between a match and a ball_stat with total participation on the ball_stat side.
A ball in a given match is uniquely identified by the over number, the ball number within the over and the innings number. For example, we could have the 3rd ball of the 15th over of the first innings. These three attributes together are the key for the weak entity ball_stats. The other attributes of ball_stats are defined in such a way that all possible results that can occur in a given ball are encapsulated. For example, if a ball results in a run out wicket of the striker, which involves 3 fielders (can occur in very rare cases), the attribute wicket.player_dismissed would contain the ID of the striker, wicket.type would be 'run out' and wicket.participants would contain the IDs of the 3 fielders involved in the wicket.
Note : We earlier had a "participating_teams" composite attribute consisting of two components i.e. "batting_team" and "bowling_team" where the "bowling_team" attribute was made a derived attribute. The reasoning behind this was that, from the match entity corresponding to the ball_stats entity we can get the two teams participating in that match, therefore, defining one of the batting_team or bowling_team uniquely defines the other. But we later realized that since ball_stats is a weak entity of match, we can entirely do away with the participating_teams composite attribute. The information about which team is batting and bowling in a given ball is uniquely defined from the two teams participating in the match, the first_batting attribute in the match entity set and the innings number in the ball_stats entity set.

We have a stadium entity to hold the various venues where a cricket match could be held throughout the tournament. The attributes are self-explanatory.

We also have an entity for a match official. Match officials are used in a match as either umpires or match referees. In a given match, we have 1 match official but multiple umpires could be involved, correspondingly we have 1-to-N and M-to-N relationships to signify the same. In the umpiring relationship, we also accommodate an attribute which refers to the umpire number for that match official. For example, Chris Broad could act as the 3rd umpire in a given match, in which case, the

umpire number attribute would be 3. We also have a relationship that defines where a match official is accommodated similar to the other *resides* relations described above.

Schema

```
create database if not exists ipl;
use ipl;

-- SET FOREIGN KEY CHECKS = OFF;
DROP TABLE IF EXISTS Team;
DROP TABLE IF EXISTS Accommodation;
DROP TABLE IF EXISTS Player;
DROP TABLE IF EXISTS Support_staff;
DROP TABLE IF EXISTS Owner;
DROP TABLE IF EXISTS Sponsor;
DROP TABLE IF EXISTS Sponsored_by;
DROP TABLE IF EXISTS Matches;
DROP TABLE IF EXISTS Match_official;
DROP TABLE IF EXISTS Stadium;
DROP TABLE IF EXISTS Participants_in_wicket;
DROP TABLE IF EXISTS Ball_stats;
DROP TABLE IF EXISTS Umpires;
-- SET FOREIGN KEY CHECKS = ON;

CREATE TABLE Team
(
team_name CHAR(40) DEFAULT NULL,
team_id INT NOT NULL,
city CHAR(30) DEFAULT NULL,
titles_won INT DEFAULT NULL,
PRIMARY KEY(team_id)
);

CREATE TABLE Accommodation
(
hotel_name CHAR(40) DEFAULT NULL,
hotel_id INT NOT NULL,
locality CHAR(30) DEFAULT NULL,
city CHAR(30) DEFAULT NULL,
state CHAR(30) DEFAULT NULL,
capacity INT DEFAULT NULL,
PRIMARY KEY(hotel_id)
);

CREATE TABLE Player
(
```

```
player_name CHAR(40) DEFAULT NULL,
nationality CHAR(40) DEFAULT NULL,
player_id INT NOT NULL,
age INT DEFAULT NULL,
captain BIT DEFAULT 0,
vice_captain BIT DEFAULT 0,
batsman BIT DEFAULT 0,
wicket_keeper BIT DEFAULT 0,
all_rounder BIT DEFAULT 0,
bowler BIT DEFAULT 0,
team_id INT DEFAULT NULL,
hotel_id INT DEFAULT NULL,
PRIMARY KEY(player_id),
FOREIGN KEY(team_id) REFERENCES Team (team_id),
FOREIGN KEY(hotel_id) REFERENCES Accommodation (hotel_id)
);

CREATE TABLE Support_staff
(
staff_name CHAR(40) DEFAULT NULL,
staff_id INT NOT NULL,
role CHAR(40) DEFAULT NULL,
team_id INT DEFAULT NULL,
hotel_id INT DEFAULT NULL,
CONSTRAINT chk_role CHECK (role IN ('physio', 'tech_analyst', 'nutritionist', 'coach')),
PRIMARY KEY(staff_id),
FOREIGN KEY(team_id) REFERENCES Team (team_id),
FOREIGN KEY(hotel_id) REFERENCES Accommodation (hotel_id)
);

CREATE TABLE Owner
(
owner_name CHAR(40) DEFAULT NULL,
owner_id INT NOT NULL,
stake DECIMAL(5,2) DEFAULT NULL,
team_id INT DEFAULT NULL,
PRIMARY KEY(owner_id),
FOREIGN KEY(team_id) REFERENCES Team (team_id)
);

CREATE TABLE Sponsor
(
sponsor_name CHAR(40) DEFAULT NULL,
sponsor_id INT NOT NULL,
```

```
PRIMARY KEY(sponsor_id)
);

CREATE TABLE Sponsored_by
(
team_id INT NOT NULL,
sponsor_id INT NOT NULL,
FOREIGN KEY(team_id) REFERENCES Team (team_id),
FOREIGN KEY(sponsor_id) REFERENCES Sponsor (sponsor_id),
PRIMARY KEY(team_id, sponsor_id)
);

CREATE TABLE Match_official
(
age INT DEFAULT NULL,
name CHAR(40) DEFAULT NULL,
official_id INT NOT NULL,
experience INT DEFAULT NULL,
accomodated_at INT DEFAULT NULL,
FOREIGN KEY(accomodated_at ) REFERENCES Accommodation (hotel_id),
PRIMARY KEY(official_id)
);

CREATE TABLE Stadium
(
stadium_id INT NOT NULL,
home_ground_team_id INT DEFAULT NULL,
name CHAR(40) DEFAULT NULL,
capacity INT DEFAULT NULL,
location_state CHAR(40) DEFAULT NULL,
location_city CHAR(40) DEFAULT NULL,
location_country CHAR(40) DEFAULT NULL,
FOREIGN KEY(home_ground_team_id) REFERENCES Team (team_id),
PRIMARY KEY(stadium_id)
);

CREATE TABLE Matches
(
match_date DATE DEFAULT NULL,
match_id INT NOT NULL,
first_batting INT DEFAULT NULL,
toss_winning_team INT DEFAULT NULL,
result_runs INT DEFAULT NULL,
result_wickets INT DEFAULT NULL,
```

```
result_winning_team_id INT DEFAULT NULL,
played_by_team_id1 INT DEFAULT NULL,
played_by_team_id2 INT DEFAULT NULL,
match_time TIME DEFAULT NULL,
held_at_stadium INT NOT NULL,
referee INT NOT NULL,
FOREIGN KEY(first_batting) REFERENCES Team (team_id),
FOREIGN KEY(toss_winning_team) REFERENCES Team (team_id),
FOREIGN KEY(result_winning_team_id) REFERENCES Team (team_id),
FOREIGN KEY(played_by_team_id1) REFERENCES Team (team_id),
FOREIGN KEY(played_by_team_id2) REFERENCES Team (team_id),
FOREIGN KEY(held_at_stadium) REFERENCES Stadium (stadium_id),
FOREIGN KEY(referee) REFERENCES Match_offical (official_id),
PRIMARY KEY(match_id)
);

create TABLE Ball_stats
(
match_id INT NOT NULL,
over_id INT NOT NULL,
ball_number INT NOT NULL,
innings_num INT NOT NULL,
extras_runs INT DEFAULT 0,
extras_type CHAR(40) DEFAULT NULL,
CONSTRAINT chk_extra CHECK (extras_type IN ('no_ball', 'wide', 'bye', 'leg_bye', 'penalty_run')),
runs_scored INT DEFAULT 0,
bowler INT NOT NULL,
striker INT NOT NULL,
non_striker INT NOT NULL,
batting_team INT NOT NULL,
bowling_team INT NOT NULL,
wicket_type CHAR(40) DEFAULT NULL,
CONSTRAINT chk_wicket CHECK (wicket_type IN ('bowled', 'caught', 'LBW', 'run_out', 'stumped',
'retired', 'hit_the_ball_twice', 'hit_wicket', 'obstructing_the_field', 'timed_out')),
wicket_player_dismissed INT DEFAULT NULL,
FOREIGN KEY(bowler) REFERENCES Player (player_id),
FOREIGN KEY(match_id) REFERENCES Matches (match_id),
FOREIGN KEY(striker) REFERENCES Player (player_id),
FOREIGN KEY(non_striker) REFERENCES Player (player_id),
FOREIGN KEY(batting_team) REFERENCES Team (team_id),
FOREIGN KEY(bowling_team) REFERENCES Team (team_id),
FOREIGN KEY(wicket_player_dismissed) REFERENCES Player(player_id),
PRIMARY KEY(match_id, over_id, ball_number, innings_num)
);
```

```
CREATE TABLE Participants_in_wicket
(
match_id INT NOT NULL,
over_id INT NOT NULL,
ball_number INT NOT NULL,
innings_num INT NOT NULL,
participant INT NOT NULL,
FOREIGN KEY(match_id) REFERENCES Matches (match_id),
FOREIGN KEY(over_id) REFERENCES Ball_stats (over_id),
FOREIGN KEY(ball_number) REFERENCES Ball_stats (ball_number),
FOREIGN KEY(innings_num) REFERENCES Ball_stats (innings_num),
FOREIGN KEY(participant) REFERENCES Player (player_id),
PRIMARY KEY(match_id, over_id, ball_number, innings_num, participant)
);

CREATE TABLE Umpires
(
official_id INT NOT NULL,
match_id INT NOT NULL,
umpire_number INT NOT NULL,
CONSTRAINT chk_ump CHECK (umpire_number IN (1, 2, 3)),
FOREIGN KEY(official_id) REFERENCES Match_official (official_id),
FOREIGN KEY(match_id) REFERENCES Matches (match_id),
PRIMARY KEY(official_id, match_id)
);
```