Pie-in-the-Sky

IPL Match Bidding App

Pie-in-the-Sky is a mobile app that is used for bidding for IPL matches legally. Any registered user can bid for any of the IPL matches listed in it. New users or bidders need to register themselves into the app by providing their mobile phone number, email id, and password. Admin will maintain the match roster and keep updating other details in the system.

The app shows the match details which include the playing team, the venue of the match, and the current standing of the teams on the points table. It will display the winner at the end of the match and update the team standings in the tournament and bidder points table. System will send updates to the bidders whenever required. It will also generate the bidders' leaderboard.

App functionalities:

- *Predict Winner*

   The app allows the user to predict the winner of the match before the toss happens for the match on which the user is predicting. This is dynamic as the matches can have a different start time. Start time will also be influenced by disruptions like rains and other unforeseen circumstances. Users will not be able to see what others have predicted. Users can change the team bids only till the toss happens. Once the toss happens everything freezes for that match.

- *Point System*

   For every win, users get points. There are no negative points, meaning if the user loses the bid, he or she does not lose his/her points. The Point system is very dynamic.

   At the start of the tournament when every team is at zero points, every user who wins the prediction wins 2 points.

   If the difference in the points between two teams playing, is <= 6, but > 0, then the person who predicts:

   - Team with higher points will win, gets 2 points
   - Team with lower points will win, gets 3 points

   If the difference in the points between two teams playing is > 6, then the person who predicts:

   - Team with higher points will win, gets 2 points
   - Team with lower points will win, gets 5 points

- *Leaderboard*

    At every time the user will be able to see his/her points and his/her position in the overall user standings. He/she will also be able to see the top 3 leader positions
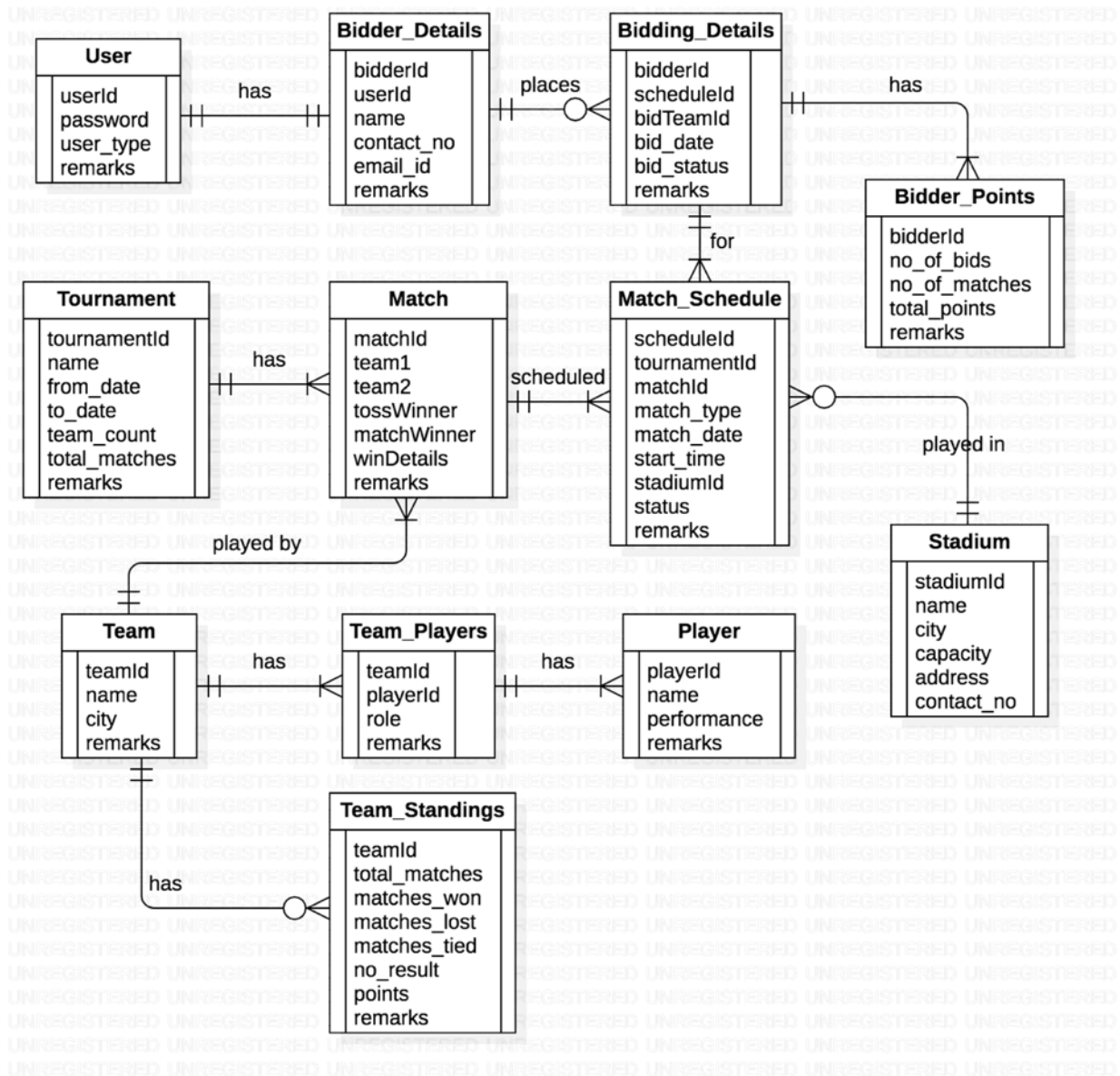
A bidder can do the following things using this app:

- can see all the match schedules (teams, date & time of the match, location).
- bid on a team for a match before the start of the match
- can predict the match winner only till its toss occurs (Note that match start time might change due to weather conditions)
- can bid for any number of matches
- after bidding on a team, (s)he would be able to change his/her team before the match starts
- bidding cannot be changed once the respective match starts
- can cancel the bid on a match; will not lose any points
- At any time, the bidder will be able to see his/her points and his/her position in the overall bidder standings.
- can see top 3 leader positions
- can see team standings anytime (i.e. their points tally)
- cannot see any details of other bidders

Admin can do following activities:

- manage tournament (tournament id, duration, description)
- manage teams (description of players and team)
- schedule and reschedule matches. Each team will play only once with the remaining teams
- edit details of match and stadium
- update match statistics (date and match result of all the earlier matches)
- declare the result of the match along with their scores
- declare winner and loser along with their points
- update team statistics (team and player performance)
- update overview at the end of the match
- view all the bidders bidding on a particular team and the %age of users supporting a team

ER Diagram

User
- userId
- password
- user_type
- remarks

has

Bidder_Details
- bidderId
- userId
- name
- contact_no
- email_id
- remarks

places

Bidding_Details
- bidderId
- scheduleId
- bidTeamId
- bid_date
- bid_status
- remarks

has

Bidder_Points
- bidderId
- no_of_bids
- no_of_matches
- total_points
- remarks

for

Tournament
- tournamentId
- name
- from_date
- to_date
- team_count
- total_matches
- remarks

has

Match
- matchId
- team1
- team2
- tossWinner
- matchWinner
- winDetails
- remarks

scheduled

Match_Schedule
- scheduleId
- tournamentId
- matchId
- match_type
- match_date
- start_time
- stadiumId
- status
- remarks

played in

Stadium
- stadiumId
- name
- city
- capacity
- address
- contact_no

played by

Team
- teamId
- name
- city
- remarks

has

Team_Players
- teamId
- playerId
- role
- remarks

has

Player
- playerId
- name
- performance
- remarks

has

Team_Standings
- teamId
- total_matches
- matches_won
- matches_lost
- matches_tied
- no_result
- points
- remarks

List of Tables

1. **Table: IPL_User**

Column name	Data type	Comments
UserId	VARCHAR	Primary Key - Unique id for the User
Password	VARCHAR	Password to for the User
User_type	VARCHAR	Admin or Bidder
Remarks	VARCHAR	Only null values

2. **Table: IPL_Stadium**

Column name	Data type	Comments
StadiumId	NUMBER	Primary key - Unique ID's for stadium
Stadium_name	VARCHAR	Unique, Not null - Name of the Stadium
City	VARCHAR	City name of the Stadium present
Capacity	NUMBER	Capacity of the stadium
Address	VARCHAR	Address of the stadium
Contact_no	NUMBER	Contact number of the stadium

3. **Table: IPL_Team**

Column name	Data type	Comments
TeamId	NUMBER	Primary key - Unique team id for all the teams
Team_name	VARCHAR	Unique, Not null - Team name for all the teams
Team_city	VARCHAR	Origin city of the team
Remarks	VARCHAR	Short form of the team names

4. **Table: IPL_Player**

Column name	Data type	Comments
PlayerId	NUMBER	Primary key
Player_name	VARCHAR	Unique, Not null
Performance_dtls	VARCHAR	Performance details
Remarks	VARCHAR	Remarks like a top performer or second best performer

5. **Table: IPL_Team_players**

Column name	Data type	Comments
TeamId	NUMBER	Composite Primary key
PlayerId	NUMBER	Composite Primary key
Player_role	VARCHAR	Captain, Batsman, Bowler, WK, etc.
Remarks	VARCHAR	Team names that he has been played for.

6. **Table: IPL_Tournament**

Column name	Data type	Comments
TournamentId	NUMBER	Primary key
Tournament_name	VARCHAR	Not null
From_date	DATE	Tournament starting date
To_date	DATE	Tournament ending date
Team_count	NUMBER	Total team count of the seasons
Total_matches	NUMBER	Total matches the team had played for each season's
Remarks	VARCHAR	Champions teams name of the seasons

7. **Table: IPL_Match**

Column name	Data type	Comments
MatchId	NUMBER	Primary key
TeamId1	NUMBER	FK from Team table. Not null
TeamId2	NUMBER	FK from Team table. Not null
TossWinner	NUMBER	Team no. 1 or 2
MatchWinner	NUMBER	Team no. 1 or 2
WinDetails	VARCHAR	Team 1 or 2 Won by XX runs or X wickets, Match tied.
Remarks	VARCHAR	E.g. Match canceled due to rain.

8. **Table: IPL_Match_Schedule**

Column name	Data type	Comments
ScheduleId	NUMBER	Primary key
TournamentId	NUMBER	FK from Tournament table.
MatchId	NUMBER	FK from Match table.
Match_type	VARCHAR	League, Knock out, Final, etc.
Match_date	DATE	This date should be within the from and to dates of the tournament.
Start_time	TIME	
StadiumId	NUMBER	FK from Stadium table
Status	VARCHAR	Scheduled, Completed, Cancelled, etc.
Remarks	VARCHAR	Reasons for the march cancellation.

9. **Table: IPL_Bidder_Details**

Column name	Data type	Comments
BidderId	NUMBER	Primary key

Column name	Data type	Comments
UserId	NUMBER	FK from User table.
Bidder_name	VARCHAR	Not null
Contact_no	NUMBER	Not null
Emailid	VARCHAR	Email id of the bidders'
Remarks	VARCHAR	Null values

10. Table: IPL_Bidding_Details

Column name	Data type	Comments
BidderId	NUMBER	FK from Bidder table. Composite Primary key
ScheduleId*	VARCHAR	FK from Match_Schedule table. Composite Primary key.
BidTeam	NUMBER	One of the team-ids of the match (1 or 2). Composite primary key.
BidDate	DATETIME	Exact date & time of placing the bid. Update this column if a bidder re-bids on the same team for the same match. Composite Primary key.
BidStatus	VARCHAR	Bid, Cancelled, Won, Lost

* FK from Match_Schedule to bid for only matches that are 'scheduled'.

11. Table: IPL_Bidder_Points

Column name	Data type	Comments
BidderId	NUMBER	FK from Bidder table. Primary key
TournamentId	NUMBER	FK from Tournament table.
No_of_bids	NUMBER	Total no. of bids placed by a bidder. Updated after completion of the match on which s/he placed the bid.
No_of_matches	NUMBER	Total no. of matches on which s/he placed the bid. Updated as above.
Total_points	NUMBER	Not null. Default 0

12. Table: IPL_Team_Standings

Column name	Data type	Comments
TeamId	NUMBER	FK from Team table. Primary key
TournamentId	NUMBER	FK from Tournament table.
Matches_played	NUMBER	Not null. Default 0 - Number of matches played
Matches_won	NUMBER	Not null. Default 0 - Number of matches won
Matches_lost	NUMBER	Not null. Default 0 - Number of matches lost
Matches_tied	NUMBER	Default 0 - Number of matches tied
No_result	NUMBER	Default 0 - Number of matches that have no results
Points	NUMBER	Not null. Default 0 - Total points of the team
Remarks	VARCHAR	mentioned champions

Instructions:

1. Create these tables in the database by running the database script provided
2. The script also has statements to insert appropriate data into all of these tables
3. Test the successful execution of the script by selecting some rows from a few tables
4. Clearly understand the structure of each table and relationships among them
5. Insert/update appropriate rows into relevant tables if you need to get more rows in the output to verify your answers

Problem Statement:

The problem statement is to use the SQL queries to find the various insight from the above-given data. And also write your insights based on the results that you will get from the queries which you will be using.

Questions – Write SQL queries to get data for the following requirements:

1. Show the percentage of wins of each bidder in the order of highest to lowest percentage.

2. Display the number of matches conducted at each stadium with the stadium name and city.

3. In a given stadium, what is the percentage of wins by a team which has won the toss?

4. Show the total bids along with the bid team and team name.

5. Show the team id who won the match as per the win details.

6. Display total matches played, total matches won and total matches lost by the team along with its team name.

7. Display the bowlers for the Mumbai Indians team.

8. How many all-rounders are there in each team, Display the teams with more than 4 all-rounders in descending order.

9. Write a query to get the total bidders points for each bidding status of those bidders who bid on CSK when it won the match in M. Chinnaswamy Stadium bidding year-wise.
   Note the total bidders' points in descending order and the year is bidding year.
   Display columns: bidding status, bid date as year, total bidder's points

10. Extract the Bowlers and All Rounders those are in the 5 highest number of wickets.
    Note
          1. use the performance_dtls column from ipl_player to get the total number of wickets
          2. Do not use the limit method because it might not give appropriate results when players have the same number of wickets
          3. Do not use joins in any cases.
          4. Display the following columns teamn_name, player_name, and player_role.

11. show the percentage of toss wins of each bidder and display the results in descending order based on the percentage

12. find the IPL season which has min duration and max duration.
    Output columns should be like the below:
    Tournment_ID, Tourment_name, Duration column, Duration

13. Write a query to display to calculate the total points month-wise for the 2017 bid year. sort the results based on total points in descending order and month-wise in ascending order.
    Note: Display the following columns:
    1. Bidder ID, 2. Bidder Name, 3. bid date as Year, 4. bid date as Month, 5. Total points
    Only use joins for the above query queries.

14. Write a query for the above question using sub queries by having the same constraints as the above question.

15. Write a query to get the top 3 and bottom 3 bidders based on the total bidding points for the 2018 bidding year.
    Output columns should be:
    like:
    Bidder Id, Ranks (optional), Total points, Highest_3_Bidders --> columns contains name of bidder, Lowest_3_Bidders --> columns contains name of bidder;

16. Create two tables called Student_details and Student_details_backup.

Table 1: Attributes	Table 2: Attributes
Student id, Student name, mail id, mobile no.	Student id, student name, mail id, mobile no.