IPL Project – Problem Statement

In this project, you have to perform the job of a sports analyst. You are given two datasets related to IPL (Indian Premier League) cricket matches. One dataset contains ball-by-ball data and the other contains match-wise data. You have to import the datasets into an SQL database and perform the tasks given in this assignment to find important insights from this dataset.

About the Data

The first CSV file is for ball-by-ball data and it has information of all the 193468 balls bowled between the years 2008 and 2020. It has 17 columns and below is the details of those 17 columns:

Column title	Description
id	Unqiue Match ID as per ESPNCricinfo
city	City in which stadium is located
date	Date on which match is held
player_of_match	Player awarded with best performance
venue	Stadium name
neutral_venue	Is the venue neutral i.e. is not homeground to the playing teams
team1	Team 1
team2	Team 2
toss_winner	Team who won the toss
toss_decision	Decision of the toss winner
winner	Match wiining team
result	Result based on victory by runs or by wickets
result_margin	Margin of wickets or runs
eliminator	Was a superover bowled or not
method	Was DL (duckworth lewis) method applied
umpire1	First umpire
umpire2	Second umpire

The second file contains match-wise data and has data of 816 IPL matches. This table has 17 columns and below is a short description of the columns in this table:

Column title	Description
id	Unqiue Match ID as per ESPNCricinfo
city	City in which stadium is located
date	Date on which match is held
player_of_match	Player awarded with best performance
venue	Stadium name
neutral_venue	Is the venue neutral i.e. is not homeground to the playing teams
team1	Team 1
team2	Team 2
toss_winner	Team who won the toss
toss_decision	Decision of the toss winner
winner	Match wiining team
result	Result based on victory by runs or by wickets
result_margin	Margin of wickets or runs
eliminator	Was a superover bowled or not
method	Was DL (duckworth lewis) method applied
umpire1	First umpire
umpire2	Second umpire

Write queries for the following tasks:

1. Create a table named ‘matches’ with appropriate data types for columns

2. Create a table named ‘deliveries’ with appropriate data types for columns

3. Import data from csv file ’IPL_matches.csv’ attached in resources to ‘matches’

4. Import data from csv file ’IPL_Ball.csv’ attached in resources to ‘deliveries’

5. Select the top 20 rows of the *deliveries* table.

6. Select the top 20 rows of the *matches* table.

7. Fetch data of all the matches played on 2nd May 2013.

8. Fetch data of all the matches where the margin of victory is more than 100 runs.

9. Fetch data of all the matches where the final scores of both teams tied and order it in descending order of the date.

10. Get the count of cities that have hosted an IPL match.

11. Create table *deliveries_v02* with all the columns of *deliveries* and an additional column *ball_result* containing value *boundary*, *dot* or *other* depending on the *total_run* (boundary for >= 4, dot for 0 and other for any other number).

12. Write a query to fetch the total number of boundaries and dot balls.

13. Write a query to fetch the total number of boundaries scored by each team.

14. Write a query to fetch the total number of dot balls bowled by each team

15. Write a query to fetch the total number of dismissals by dismissal kinds

16. Write a query to get the top 5 bowlers who conceded maximum extra runs

17. Write a query to create a table named *deliveries_v03* with all the columns of *deliveries_v02* table and two additional column (named *venue* and *match_date*) of *venue* and *date* from table *matches*

18. Write a query to fetch the total runs scored for each venue and order it in the descending order of total runs scored.

19. Write a query to fetch the year-wise total runs scored at *Eden Gardens* and order it in the descending order of total runs scored.

20. Get unique team1 names from the *matches* table, you will notice that there are two entries for *Rising Pune Supergiant* one with *Rising Pune Supergiant* and another one with *Rising Pune Supergiant**s***. Your task is to create a *matches_corrected* table with two additional columns *team1_corr* and *team2_corr* containing team names with replacing *Rising Pune Supergiant**s*** with *Rising Pune Supergiant*. Now analyse these newly created columns.

21. Create a new table deliveries_v04 with the first column as ball_id containing information of match_id, inning, over and ball separated by '-' (For ex. 335982-1-0-1 match_id-inning-over-ball) and rest of the columns same as deliveries_v03)

22. Compare the total count of rows and total count of distinct ball_id in deliveries_v04.

23. Create table deliveries_v05 with all columns of deliveries_v04 and an additional column for row number partition over ball_id.

24. Use the r_num created in deliveries_v05 to identify instances where ball_id is repeating.

25. Use subqueries to fetch data of all the ball_id which are repeating.

NOTE: - All the queries listed in the problem statement are solved in pgAdmin (PostgreSQL).