**OBJECTIVE QUESTIONS**

1. **List the different dtypes of columns in table “ball\_by\_ball” (using information schema)**

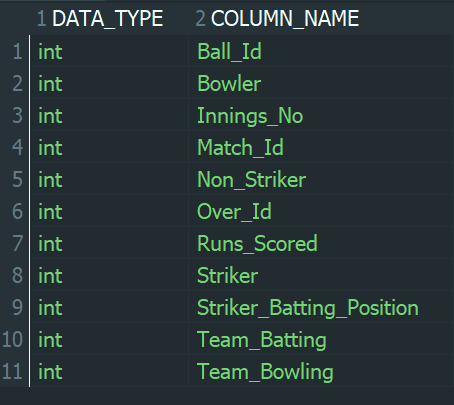
Ans:-

**Query used: -**

select data\_type, COLUMN\_NAME from information\_schema.columns

where table\_name = 'Ball\_by\_Ball' and table\_schema = 'ipl';

**Output:-**



**Conclusion: -**

The structure of the ball\_by\_ball table was analysed using the **INFORMATION\_SCHEMA,** which provides metadata about database tables. This analysis helped identify the different data types used for each column in the table. It was observed that the table primarily consists of **integer data types**, as it stores numerical match-related information such as match ID, over number, ball number, innings number, team identifiers, player identifiers, and runs scored.

1. **What is the total number of runs scored in 1st season by RCB (bonus: also include the extra runs using the extra runs table)**

**Ans:-**

**Query used: -**

SELECT

t.Team\_Name,

SUM(b.Runs\_Scored + COALESCE(er.Extra\_Runs, 0)) AS Total\_Runs

FROM Ball\_by\_Ball b

JOIN Matches m

ON m.Match\_Id = b.Match\_Id

JOIN Team t

ON t.Team\_Id = b.Team\_Batting

LEFT JOIN Extra\_Runs er

ON er.Match\_Id = b.Match\_Id

AND er.Over\_Id = b.Over\_Id

AND er.Ball\_Id = b.Ball\_Id

AND er.Innings\_No = b.Innings\_No

WHERE t.Team\_Name = 'Royal Challengers Bangalore'

AND m.Season\_Id = (

SELECT MIN(Season\_Id) FROM Matches

)

GROUP BY t.Team\_Name;

**Approach: -**

1. **Identified the required tables**

* Ball\_by\_Ball - to get runs scored off the bat
* Extra\_Runs - to include extra runs (wides, no-balls, etc.)
* Matches - to identify the season of each match
* Team - to identify Royal Challengers Bangalore

1. **Calculated total runs at ball level**

* Added Runs\_Scored from Ball\_by\_Ball
* Added Extra\_Runs from Extra\_Runs using a LEFT JOIN
* Used COALESCE to handle NULL extra runs

1. **Identified the first IPL season dynamically**

* Used MIN(Season\_Id) instead of hardcoding the season value
* This ensures correctness even if season IDs are not sequential

1. **Filtered data for Royal Challengers Bangalore**

* Used Team\_Batting to ensure only RCB batting innings were considered

1. **Aggregated total runs**

* Summed total runs across all balls played by RCB in the first season
* Grouped the result by team name

**Output:-**



**Conclusion: -**

In this analysis, the total number of runs scored by Royal Challengers Bangalore in the first IPL season was calculated by combining both batting runs and extra runs at the ball level.  
Instead of assuming a fixed season identifier, the first season was dynamically identified using the minimum season ID, making the query data-independent and reliable.

The final result shows that **RCB** scored a total of **2558** runs in the first IPL season, including all **extra runs**.

1. **How many players were more than the age of 25 during season 2014?**

Ans: -

**Query used: -**

SELECT COUNT(DISTINCT p.Player\_Id) AS Players\_Above\_25

FROM Player p

JOIN Player\_Match pm

ON pm.Player\_Id = p.Player\_Id

JOIN Matches m

ON m.Match\_Id = pm.Match\_Id

JOIN Season s

ON s.Season\_Id = m.Season\_Id

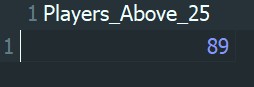
WHERE s.Season\_Year = 2014

AND TIMESTAMPDIFF(YEAR, p.DOB, '2014-01-01') > 25;

**Approach: -**

* 1. **Identified the required tables**
* Player - to obtain player date of birth
* Player\_Match - to identify players who participated in matches
* Matches - to map matches to seasons
* Season - to identify the 2014 IPL season
  1. **Filtered the data for the 2014 season**
* Used Season\_Year = 2014 to select matches played in IPL 2014
  1. **Calculated player age accurately**
* Used TIMESTAMPDIFF(YEAR, DOB, '2014-01-01') to calculate the exact age of each player at the start of the 2014 season
  1. **Filtered players older than 25**
* Selected only those players whose calculated age was greater than 25
  1. **Removed duplicate players**
* Used COUNT(DISTINCT Player\_Id) to ensure each player was counted only once, even if they played multiple matches

**Output:-**



**Conclusion: -**

In this analysis, the number of players who were **older than 25 years during the IPL 2014 season** was determined by calculating each player’s exact age at the beginning of the season and considering only those who actively participated in 2014 matches.

By using a precise age calculation and eliminating duplicate player entries, the result accurately reflects player participation.

It was found that **89 players were more than 25 years old during the IPL 2014 season**.

1. **How many matches did RCB win in 2013?**

**Ans: -**

**Query used: -**

SELECT COUNT(\*) AS Matches\_Won\_By\_RCB

FROM Matches m

JOIN Season s

ON s.Season\_Id = m.Season\_Id

WHERE s.Season\_Year = 2013

AND m.Match\_Winner = (

SELECT Team\_Id

FROM Team

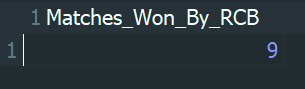
WHERE Team\_Name = 'Royal Challengers Bangalore'

);

**Approach: -**

* 1. **Identified the required tables**
* Matches - to obtain match results
* Season - to identify matches played in the 2013 season
* Team - to identify Royal Challengers Bangalore
  1. **Filtered matches for the 2013 IPL season**
* Used Season\_Year = 2013 to select only matches played during IPL 2013
  1. **Identified RCB as the match winner**
* Retrieved the Team\_Id of Royal Challengers Bangalore
* Filtered matches where Match\_Winner matched RCB’s team ID
  1. **Counted the number of matches won**
* Used COUNT(\*) to calculate the total number of matches won by RCB in the 2013 season

**Output: -**

****

**Conclusion: -**

The number of matches won by Royal Challengers Bangalore during the IPL 2013 season was determined by analysing match results and filtering records based on the season year and match winner. Only those matches where RCB was declared the winner were included in the analysis.

From the results, it was found that **Royal Challengers Bangalore** won a total of **9 matches** in the **IPL 2013** season, which reflects their performance in that year’s tournament.

1. **List the top 10 players according to their strike rate in the last 4 seasons**

**Ans: -**

**Query used: -**

WITH last\_4\_seasons AS (

SELECT Season\_Id

FROM Season

ORDER BY Season\_Year DESC

LIMIT 4

),

player\_stats AS (

SELECT

b.Striker AS Player\_Id,

SUM(b.Runs\_Scored) AS Total\_Runs,

COUNT(\*) AS Balls\_Faced

FROM Ball\_by\_Ball b

JOIN Matches m

ON m.Match\_Id = b.Match\_Id

WHERE m.Season\_Id IN (SELECT Season\_Id FROM last\_4\_seasons)

GROUP BY b.Striker

)

SELECT

p.Player\_Name,

ROUND((ps.Total\_Runs / ps.Balls\_Faced) \* 100, 2) AS Strike\_Rate

FROM player\_stats ps

JOIN Player p

ON p.Player\_Id = ps.Player\_Id

WHERE ps.Balls\_Faced > 0

ORDER BY Strike\_Rate DESC

LIMIT 10;

**Approach: -**

* 1. **Identified the last four IPL seasons**
* Used the Season table
* Sorted seasons in descending order by Season\_Year
* Selected the most recent four seasons using LIMIT 4
  1. **Collected player batting statistics**
* Used the Ball\_by\_Ball table to extract batting data
* Grouped records by the striker (Striker) to calculate player-wise statistics
* Calculated:
  + Total runs scored using SUM(Runs\_Scored)
  + Total balls faced using COUNT(\*)
  1. **Filtered data for the last four seasons**
* Joined Ball\_by\_Ball with Matches
* Considered only matches belonging to the last four seasons identified earlier
  1. **Calculated strike rate**
* Computed strike rate for each player using the formula:  
  (Total Runs / Total Balls Faced) × 100
* Rounded the strike rate to two decimal places
  1. **Retrieved player names**
* Joined the aggregated player statistics with the Player table
  1. **Ranked and selected top players**
* Sorted players in descending order of strike rate
* Selected the top 10 players based on strike rate

**Output: -**

****

**Conclusion: -**

Using ball-by-ball data, the strike rate of players was calculated for the last four IPL seasons by dividing total runs scored by total balls faced. The most recent seasons were identified dynamically to ensure the analysis remained up to date.  
Players were then ranked according to their strike rate, and the top 10 players with the highest strike rates were identified.

This analysis effectively highlights the most efficient and aggressive batsmen in the recent IPL seasons based on strike rate.

1. **What are the average runs scored by each batsman considering all the seasons?**

**Ans)**

**Query used: -**

SELECT

p.Player\_Name,

ROUND(SUM(b.Runs\_Scored) / COUNT(DISTINCT pm.Match\_Id), 2) AS Average\_Runs

FROM Player p

JOIN Ball\_by\_Ball b

ON b.Striker = p.Player\_Id

JOIN Player\_Match pm

ON pm.Player\_Id = p.Player\_Id

AND pm.Match\_Id = b.Match\_Id

GROUP BY p.Player\_Id, p.Player\_Name

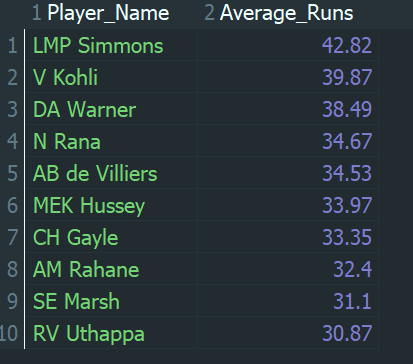
HAVING COUNT(DISTINCT pm.Match\_Id) > 0

ORDER BY Average\_Runs DESC LIMIT 10;

**Approach: -**

* 1. **Identified the required tables**
* Player -to retrieve player names
* Ball\_by\_Ball -to calculate runs scored by each batsman
* Player\_Match -to identify matches played by each batsman.
  1. **Joined player and batting data**
* Joined Player with Ball\_by\_Ball using Striker to associate runs with each batsman
  1. **Joined match participation data**
* Joined Player\_Match using Player\_Id and Match\_Id to accurately count matches played by each batsman
  1. **Calculated average runs**
* Summed total runs scored by each batsman
* Counted distinct matches played to avoid duplicate counting
* Calculated average runs as:  
  Total Runs ÷ Matches Played
  1. **Grouped and filtered results**
* Grouped data by player to compute individual averages
* Used HAVING to ensure only players who played at least one match were included
  1. **Sorted and limited output**
* Sorted players in descending order of average runs
* Displayed only the top 10 players to keep the output concise and readable due to the large dataset

**Output: -**

****

**Conclusion: -**

In this analysis, the average runs scored by each batsman were calculated by dividing the total runs scored from ball-by-ball data by the number of matches played. Joins were used to accurately combine batting performance with match participation.

Since the complete list of players is extensive, the results were sorted by average runs and limited to the top 10 batsmen for better readability. The output highlights LMP Simmons, V Kohli, DA Warner, and AB de Villiers among the highest-performing batsmen in terms of average runs across all IPL seasons.

1. **What are the average wickets taken by each bowler considering all the seasons?**

**Ans: -**

**Query used: -**

SELECT

p.Player\_Name,

ROUND(COUNT(w.Player\_Out) / COUNT(DISTINCT pm.Match\_Id), 2) AS Average\_Wickets

FROM Player p

JOIN Wicket\_Taken w

ON w.Player\_Out IS NOT NULL

JOIN Ball\_by\_Ball b

ON b.Match\_Id = w.Match\_Id

AND b.Over\_Id = w.Over\_Id

AND b.Ball\_Id = w.Ball\_Id

AND b.Innings\_No = w.Innings\_No

AND b.Bowler = p.Player\_Id

JOIN Player\_Match pm

ON pm.Player\_Id = p.Player\_Id

AND pm.Match\_Id = b.Match\_Id

GROUP BY p.Player\_Id, p.Player\_Name

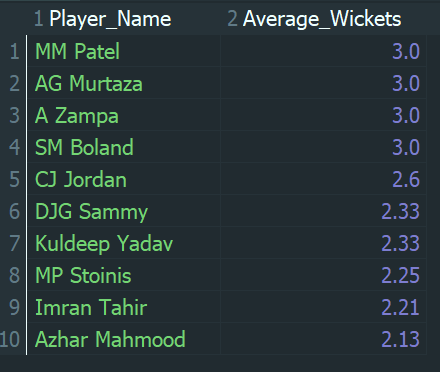
HAVING COUNT(DISTINCT pm.Match\_Id) > 0

ORDER BY Average\_Wickets DESC LIMIT 10 ;

**Approach: -**

* 1. **Identified the required tables**
* Player - to obtain bowler names
* Wicket\_Taken - to identify wicket events
* Ball\_by\_Ball - to map each wicket to the corresponding bowler
* Player\_Match - to determine the number of matches played by each bowler
  1. **Associated wickets with bowlers**
* Joined Wicket\_Taken with Ball\_by\_Ball using match, over, ball, and innings details
* Used the Bowler field to correctly assign each wicket to the bowler
  1. **Calculated total wickets and matches played**
* Counted wicket events for each bowler
* Counted distinct matches played by each bowler to avoid duplicate counting
  1. **Calculated average wickets**
* Computed the average wickets taken per match using the formula:  
  Total Wickets / Matches Played
  1. **Grouped and filtered results**
* Grouped the data by bowler
* Used HAVING to ensure only bowlers who played at least one match were included
  1. **Sorted and limited the output**
* Sorted bowlers in descending order of average wickets
* Displayed only the top 10 bowlers to keep the output concise due to the large number of players

**Output: -**

****

**Conclusion: -**

In this analysis, the average number of wickets taken by each bowler was calculated across all IPL seasons by dividing the total wickets taken by the number of matches played. Accurate mapping of wicket events to bowlers was achieved using ball-by-ball data, ensuring reliable results.

Since the complete list of bowlers is extensive, the output was limited to the top 10 bowlers based on average wickets per match for better readability. The results highlight the most effective bowlers in terms of wicket-taking consistency across all IPL seasons.

1. **List all the players who have average runs scored greater than the overall average and who have taken wickets greater than the overall average**

Ans)

**Query used: -**

WITH batting AS ( SELECT Striker AS Player\_Id,

ROUND(

SUM(Runs\_Scored) \* 1.0 /

COUNT(DISTINCT CONCAT(Match\_Id, '-', Innings\_No)), 2

) AS Avg\_Runs

FROM Ball\_by\_Ball

GROUP BY Striker

),

bowling AS (SELECT b.Bowler AS Player\_Id,

COUNT(w.Player\_Out) AS Total\_Wickets

FROM Ball\_by\_Ball b

JOIN Wicket\_Taken w

ON b.Match\_Id = w.Match\_Id

AND b.Over\_Id = w.Over\_Id

AND b.Ball\_Id = w.Ball\_Id

AND b.Innings\_No = w.Innings\_No

GROUP BY b.Bowler

)

SELECT p.Player\_Name, bat.Avg\_Runs, bowl.Total\_Wickets

FROM batting bat

JOIN bowling bowl

ON bat.Player\_Id = bowl.Player\_Id

JOIN Player p

ON p.Player\_Id = bat.Player\_Id

WHERE

bat.Avg\_Runs >

(SELECT AVG(Avg\_Runs) FROM batting)

AND

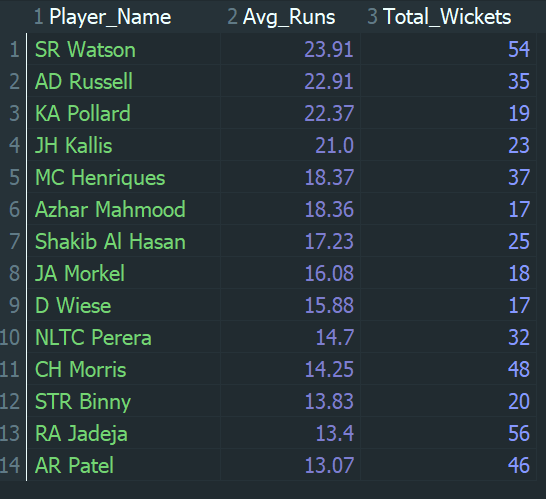
bowl.Total\_Wickets >

(SELECT AVG(Total\_Wickets) FROM bowling) ORDER BY Avg\_Runs DESC;

**Approach: -**

* 1. **Identified the required tables**
* Ball\_by\_Ball - to calculate batting performance (runs scored)
* Wicket\_Taken - to calculate bowling performance (wickets taken)
* Player - to retrieve player names
  1. **Calculated average runs for each player**
* Grouped ball-by-ball data by striker
* Calculated average runs per innings using total runs divided by total innings played
* Rounded the average runs to two decimal places
  1. **Calculated total wickets for each player**
* Joined Ball\_by\_Ball and Wicket\_Taken to correctly associate wickets with bowlers
* Counted the total number of wickets taken by each player
  1. **Computed overall averages**
* Calculated the overall average of players’ average runs
* Calculated the overall average of total wickets taken by players
  1. **Filtered players based on performance criteria**
* Selected players whose average runs were greater than the overall average runs
* Selected players whose total wickets taken were greater than the overall average wickets
  1. **Sorted the output**
* Sorted the result in descending order of average runs

**Output: -**

****

**Conclusion: -**

In this analysis, players were evaluated based on both their batting and bowling performances across all IPL seasons. The average runs per innings for each player were calculated and compared against the overall batting average, while the total wickets taken by each player were compared against the overall average wickets.

Players who satisfied both conditions were identified as strong all-round contributors. The results show that only a limited set of players met these criteria, indicating that consistently performing above average in both batting and bowling is relatively rare in IPL cricket. Players such as SR Watson, AD Russell, JH Kallis, and RA Jadeja stand out for their significant contributions with both bat and ball.

1. **Create a table rcb\_record table that shows the wins and losses of RCB in an individual venue.**

Ans:-

**Query used: -**

CREATE TABLE rcb\_record AS

SELECT

v.Venue\_Name,

SUM(

CASE

WHEN m.Match\_Winner = rcb.Team\_Id THEN 1

ELSE 0

END

) AS Wins,

SUM(

CASE

WHEN m.Match\_Winner <> rcb.Team\_Id THEN 1

ELSE 0

END

) AS Losses

FROM Matches m

JOIN Venue v

ON v.Venue\_Id = m.Venue\_Id

JOIN (

SELECT Team\_Id

FROM Team

WHERE Team\_Name = 'Royal Challengers Bangalore'

) rcb

WHERE m.Team\_1 = rcb.Team\_Id

OR m.Team\_2 = rcb.Team\_Id

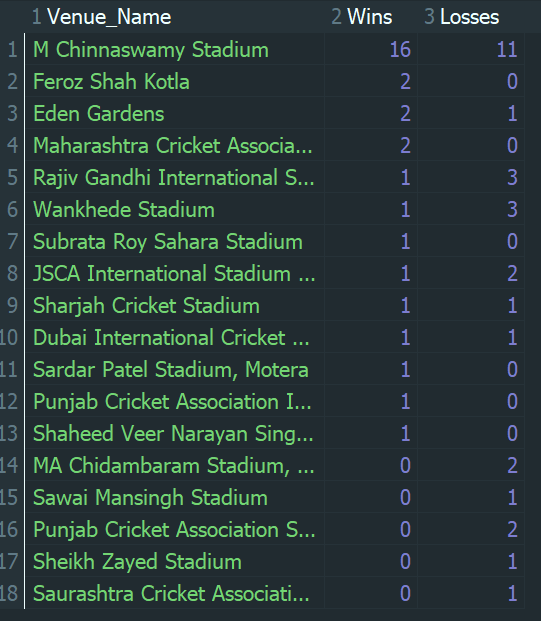
GROUP BY v.Venue\_Name ORDER BY Wins DESC;

SELECT \* FROM rcb\_record;

**Approach: -**

* 1. **Identified the required tables**
* Matches - to obtain match details and match winners
* Venue - to retrieve venue names
* Team - to identify the Team\_Id of Royal Challengers Bangalore
  1. **Filtered RCB matches**
* Selected only those matches where Royal Challengers Bangalore participated as either Team\_1 or Team\_2
  1. **Classified match outcomes**
* Used a CASE expression to count a match as a win when RCB was the match winner
* Counted a match as a loss when RCB participated but did not win the match
  1. **Aggregated results by venue**
* Grouped the match outcomes by venue name
* Calculated the total number of wins and losses for RCB at each venue
  1. **Created the summary table**
* Created a new table rcb\_record directly from the query result using CREATE TABLE AS SELECT
* Ordered the results by the number of wins in descending order for better readability
  1. **Viewed the final result**
* Retrieved the data from the rcb\_record table to analyze venue-wise performance

**Output: -**

****

**Conclusion: -**

A table named rcb\_record was created to summarize the venue-wise performance of Royal Challengers Bangalore across all IPL seasons. The table captures the number of matches won and lost by RCB at each venue by analyzing match outcomes and grouping them accordingly.

The results show that M. Chinnaswamy Stadium is the venue where RCB has recorded the highest number of wins, highlighting it as a strong home ground. The table provides valuable insights into RCB’s performance patterns across different venues and can be used to identify venues where the team has historically performed well or faced challenges.

1. **What is the impact of bowling style on wickets taken?**

**Ans: -**

**Query used: -**

SELECT

bs.Bowling\_skill AS Bowling\_Style\_Name,

COUNT(w.Player\_Out) AS Total\_Wickets

FROM Player p

JOIN bowling\_style bs

ON bs.Bowling\_Id = p.Bowling\_Skill

JOIN Ball\_by\_Ball b

ON b.Bowler = p.Player\_Id

JOIN Wicket\_Taken w

ON w.Match\_Id = b.Match\_Id

AND w.Over\_Id = b.Over\_Id

AND w.Ball\_Id = b.Ball\_Id

AND w.Innings\_No = b.Innings\_No

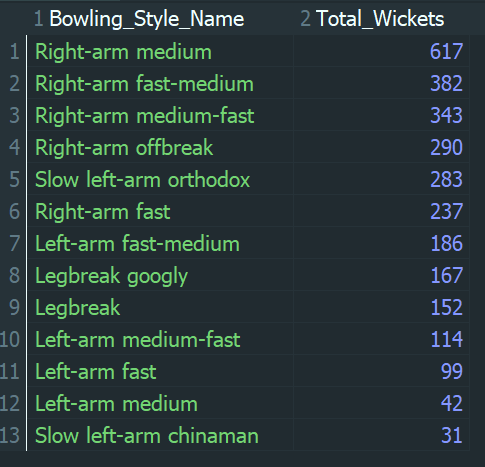
GROUP BY bs.Bowling\_skill

ORDER BY Total\_Wickets DESC;

**Approach: -**

* 1. **Identified the required tables**
* Player - to obtain bowler information and bowling style IDs
* bowling\_style - to retrieve the bowling style names
* Ball\_by\_Ball - to associate each delivery with the bowler
* Wicket\_Taken - to identify wicket events
  1. **Mapped bowling styles to players**
* Joined the Player table with the bowling\_style table using the bowling style ID
* Converted numeric bowling style identifiers into meaningful bowling style names
  1. **Associated wickets with bowlers**
* Joined Ball\_by\_Ball with Wicket\_Taken using match, over, ball, and innings details
* Ensured that each wicket was correctly attributed to the bowler who delivered the ball
  1. **Aggregated wickets by bowling style**
* Grouped the data by bowling style name
* Counted the total number of wickets taken for each bowling style
  1. **Sorted the results**
* Ordered the output in descending order of total wickets to analyze the relative impact of different bowling styles

**Output: -**

****

**Conclusion: -**

In this analysis, the impact of bowling style on wicket-taking performance was examined by aggregating total wickets according to bowling style names across all IPL seasons. By mapping bowling style IDs to their corresponding names, the results provide a clear and interpretable comparison of different bowling techniques.

The output shows that right-arm pace bowling styles, particularly Right-arm medium, Right-arm fast-medium, and Right-arm medium-fast, have contributed the highest number of wickets. This indicates that pace bowling has played a significant role in wicket-taking in the IPL, while spin bowling styles, although effective, have contributed comparatively fewer wickets overall.

1. **Write the SQL query to provide a status of whether the performance of the team is better than the previous year's performance on the basis of the number of runs scored by the team in the season and the number of wickets taken**

Ans:-

**Query used: -**

WITH team\_stats AS (

SELECT

m.Season\_Id,

b.Team\_Batting AS Team\_Id,

SUM(b.Runs\_Scored) AS Total\_Runs

FROM Ball\_by\_Ball b

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

GROUP BY m.Season\_Id, b.Team\_Batting

),

team\_wickets AS (

SELECT

m.Season\_Id,

b.Team\_Bowling AS Team\_Id,

COUNT(w.Player\_Out) AS Total\_Wickets

FROM Ball\_by\_Ball b

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

JOIN Wicket\_Taken w

ON b.Match\_Id = w.Match\_Id

AND b.Over\_Id = w.Over\_Id

AND b.Ball\_Id = w.Ball\_Id

AND b.Innings\_No = w.Innings\_No

GROUP BY m.Season\_Id, b.Team\_Bowling

),

season\_stats AS (

SELECT

s.Season\_Year,

t.Team\_Name,

r.Team\_Id,

r.Total\_Runs,

w.Total\_Wickets,

LAG(r.Total\_Runs) OVER (PARTITION BY r.Team\_Id ORDER BY s.Season\_Year) AS Prev\_Total\_Runs,

LAG(w.Total\_Wickets) OVER (PARTITION BY r.Team\_Id ORDER BY s.Season\_Year) AS Prev\_Total\_Wickets

FROM team\_stats r

JOIN team\_wickets w

ON r.Season\_Id = w.Season\_Id

AND r.Team\_Id = w.Team\_Id

JOIN Season s

ON s.Season\_Id = r.Season\_Id

JOIN Team t

ON t.Team\_Id = r.Team\_Id

)

SELECT

Team\_Name, Season\_Year, Total\_Runs, Total\_Wickets, Prev\_Total\_Runs, Prev\_Total\_Wickets,

CASE

WHEN Total\_Runs > Prev\_Total\_Runs

AND Total\_Wickets > Prev\_Total\_Wickets

THEN 'Improved'

WHEN Total\_Runs < Prev\_Total\_Runs

AND Total\_Wickets < Prev\_Total\_Wickets

THEN 'Declined'

ELSE 'Same'

END AS Performance\_Status

FROM season\_stats

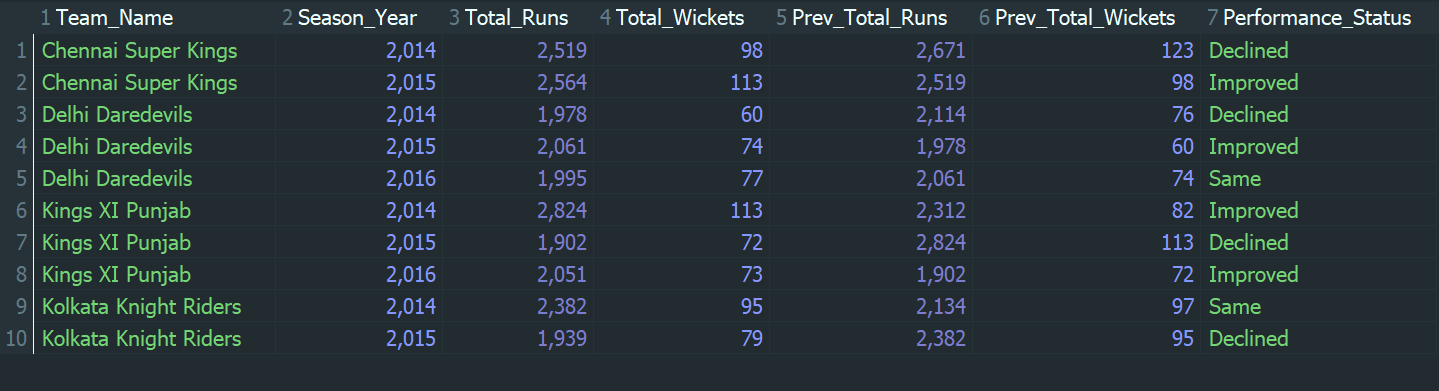
WHERE Prev\_Total\_Runs IS NOT NULL

ORDER BY Team\_Name, Season\_Year LIMIT 10;

**Approach: -**

1. Calculated season-wise team runs
   * Joined Ball\_by\_Ball with Matches
   * Aggregated total runs scored by each team in every season using the batting team information
2. Calculated season-wise team wickets
   * Joined Ball\_by\_Ball, Matches, and Wicket\_Taken
   * Aggregated total wickets taken by each team in every season using the bowling team information
3. Combined batting and bowling statistics
   * Joined season-wise runs and wickets data for each team and season
   * Retrieved corresponding season year and team name using the Season and Team tables
4. Compared performance with previous season
   * Used the LAG() window function to obtain the previous season’s total runs and wickets for each team
   * Ensured comparisons were done within the same team and in chronological order
5. Determined performance status
   * Classified performance as:
     + Improved - both runs and wickets increased
     + Declined - both runs and wickets decreased
     + Same - mixed or unchanged performance
6. Filtered and limited the output
   * Excluded the first season for each team where previous data is unavailable
   * Limited the result to 10 records for readability due to the large dataset

**Output: -**

****

**Conclusion: -**

In this analysis, team performance was evaluated season-by-season by comparing total runs scored and total wickets taken with the team’s performance in the previous season. Using season-wise aggregation and window functions, each team’s performance trend was classified as Improved, Declined, or Same.

The results show that team performance fluctuates across seasons, with some teams demonstrating improvement in both batting and bowling, while others experienced a decline or mixed performance. Limiting the output to the first 10 records provides a concise view of these trends while maintaining the correctness of the analysis across the complete dataset.

1. **Can you derive more KPIs for the team strategy?**

**Ans:-**

To evaluate team performance and support strategic decision-making in the IPL, the following five Key Performance Indicators (KPIs) were derived using SQL. These KPIs cover overall team success, batting strength, bowling effectiveness, scoring speed, and match strategy.

* **KPI 1: Win Percentage: -**

This KPI calculates the total matches played, matches won, and win percentage for each team. It provides a direct measure of overall team success and consistency across all IPL seasons.

**Query used: -**

SELECT

t.Team\_Name,

COUNT(m.Match\_Id) AS Matches\_Played,

SUM(CASE WHEN m.Match\_Winner = t.Team\_Id THEN 1 ELSE 0 END) AS Matches\_Won,

ROUND(

(SUM(CASE WHEN m.Match\_Winner = t.Team\_Id THEN 1 ELSE 0 END) \* 100.0)

/ COUNT(m.Match\_Id), 2

) AS Win\_Percentage

FROM Matches m

JOIN Team t

ON t.Team\_Id = m.Team\_1 OR t.Team\_Id = m.Team\_2

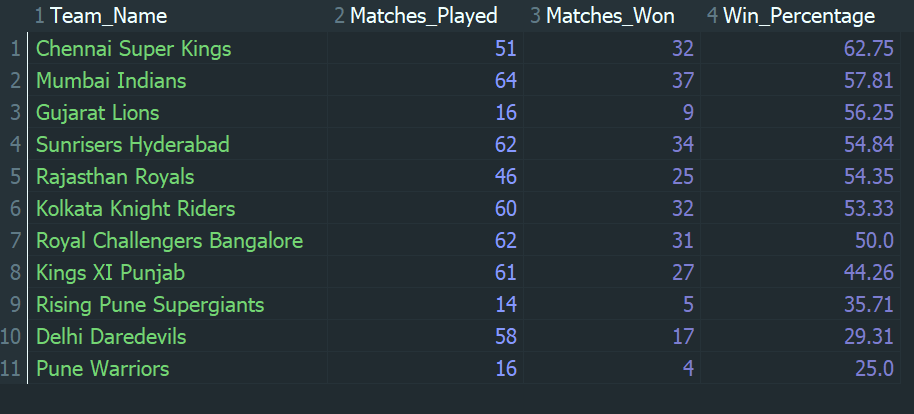
GROUP BY t.Team\_Name

ORDER BY Win\_Percentage DESC;

**Approach: -**

* 1. **Identify relevant tables**
* Used the Matches table to obtain match details such as participating teams and match winners.
* Used the Team table to retrieve team names corresponding to team IDs.
  1. **Determine matches played by each team**
* Considered matches where a team appeared either as Team\_1 or Team\_2.
* Counted the total number of matches played by each team.
  1. **Calculate matches won**
* Compared the Match\_Winner column with the team’s ID.
* Counted a match as a win when the team ID matched the match winner.
  1. **Compute win percentage**
* Calculated win percentage using the formula:  
  (Matches Won / Matches Played) × 100
* Rounded the result to two decimal places for readability.
  1. **Group and sort results**
* Grouped results by team name.
* Ordered teams in descending order of win percentage to identify the most successful teams.

**Output: -**

****

* **KPI 2: Average Runs per Match (Batting Strength)**

This KPI measures the average number of runs scored by a team per match. It reflects the batting strength and scoring consistency of the team.

**Query used: -**

SELECT

t.Team\_Name,

ROUND(

SUM(b.Runs\_Scored) / COUNT(DISTINCT b.Match\_Id), 2

) AS Avg\_Runs\_Per\_Match

FROM Ball\_by\_Ball b

JOIN Team t

ON t.Team\_Id = b.Team\_Batting

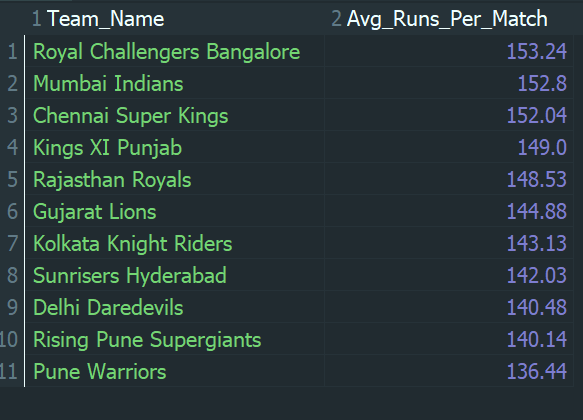
GROUP BY t.Team\_Name

ORDER BY Avg\_Runs\_Per\_Match DESC;

**Approach: -**

* 1. **Identify relevant tables**
* Used the Ball\_by\_Ball table to obtain run-by-run scoring information.
* Used the Team table to map team IDs to team names.
  1. **Calculate total runs scored by each team**
* Considered the Team\_Batting column to identify which team was batting.
* Aggregated the total runs scored by each team across all matches.
  1. **Determine matches played**
* Counted the distinct match IDs in which each team batted to avoid duplicate counting.
* Ensured that each match was counted only once per team.
  1. **Compute average runs per match**
* Divided the total runs scored by the total number of matches played.
* Rounded the average runs value to two decimal places for clarity.
  1. **Group and sort results**
* Grouped the results by team name.
* Sorted teams in descending order of average runs per match to identify strong batting teams.

**Output: -**

****

* **KPI 3: Average Wickets per Match (Bowling Strength)**

This KPI calculates the average wickets taken by a team per match. It highlights the effectiveness of a team’s bowling unit in restricting the opposition.

**Query used: -**

SELECT

t.Team\_Name,

ROUND(

COUNT(w.Player\_Out) / COUNT(DISTINCT b.Match\_Id), 2

) AS Avg\_Wickets\_Per\_Match

FROM Ball\_by\_Ball b

JOIN Team t

ON t.Team\_Id = b.Team\_Bowling

JOIN Wicket\_Taken w

ON w.Match\_Id = b.Match\_Id

AND w.Over\_Id = b.Over\_Id

AND w.Ball\_Id = b.Ball\_Id

AND w.Innings\_No = b.Innings\_No

GROUP BY t.Team\_Name

ORDER BY Avg\_Wickets\_Per\_Match DESC;

**Approach: -**

* 1. **Identify relevant tables**
* Used the Ball\_by\_Ball table to identify bowling deliveries by each team.
* Used the Wicket\_Taken table to capture wicket events.
* Used the Team table to map team IDs to team names.
  1. **Associate wickets with bowling teams**
* Joined Ball\_by\_Ball and Wicket\_Taken using match ID, over ID, ball ID, and innings number.
* Ensured that each wicket was correctly attributed to the bowling team.
  1. **Calculate total wickets taken**
* Counted the number of players dismissed (Player\_Out) for each team.
* Aggregated wicket counts across all matches.
  1. **Determine matches played**
* Counted distinct match IDs in which each team bowled.
* This ensured accurate normalization of wickets per match.
  1. **Compute average wickets per match**
* Divided the total wickets taken by the number of matches played.
* Rounded the result to two decimal places for better readability.
  1. **Group and sort results**
* Grouped the results by team name.
* Ordered teams in descending order of average wickets per match to identify strong bowling teams.

**Output: -**

****

* **KPI 4: Run Rate (Scoring Speed)**

Run rate represents the average runs scored per ball by a team. Since IPL is a T20 tournament, this KPI is crucial for understanding scoring speed and aggressive batting strategies.

**Query used: -**

SELECT

t.Team\_Name,

ROUND(

SUM(b.Runs\_Scored) / COUNT(\*), 2

) AS Run\_Rate

FROM Ball\_by\_Ball b

JOIN Team t

ON t.Team\_Id = b.Team\_Batting

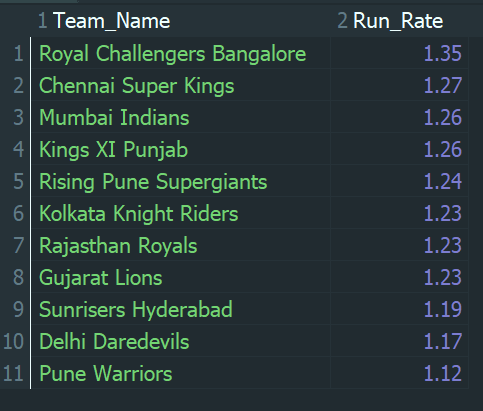
GROUP BY t.Team\_Name

ORDER BY Run\_Rate DESC;

**Approach: -**

* 1. **Identify relevant tables**
* Used the Ball\_by\_Ball table to access ball-level scoring data.
* Used the Team table to map batting team IDs to team names.
  1. **Calculate total runs scored**
* Considered the Team\_Batting column to identify the batting team for each delivery.
* Summed the total runs scored by each team across all balls faced.
  1. **Calculate total balls faced**
* Counted the total number of balls faced by each team.
* Each row in the Ball\_by\_Ball table represents one delivery.
  1. **Compute run rate**
* Divided the total runs scored by the total number of balls faced.
* Rounded the result to two decimal places for clarity.
  1. **Group and sort results**
* Grouped the results by team name.
* Sorted teams in descending order of run rate to identify faster-scoring teams.

**Output: -**

****

* **KPI 5: Toss Impact KPI (Strategic Advantage)**

This KPI analyzes how often teams win matches after winning the toss. It helps evaluate the strategic advantage of toss decisions such as choosing to bat or bowl first.

**Query used: -**

SELECT

t.Team\_Name,

COUNT(m.Match\_Id) AS Toss\_Wins,

SUM(CASE WHEN m.Match\_Winner = m.Toss\_Winner THEN 1 ELSE 0 END) AS Wins\_After\_Toss,

ROUND(

(SUM(CASE WHEN m.Match\_Winner = m.Toss\_Winner THEN 1 ELSE 0 END) \* 100.0)

/ COUNT(m.Match\_Id), 2

) AS Win\_Percentage\_After\_Toss

FROM Matches m

JOIN Team t

ON t.Team\_Id = m.Toss\_Winner

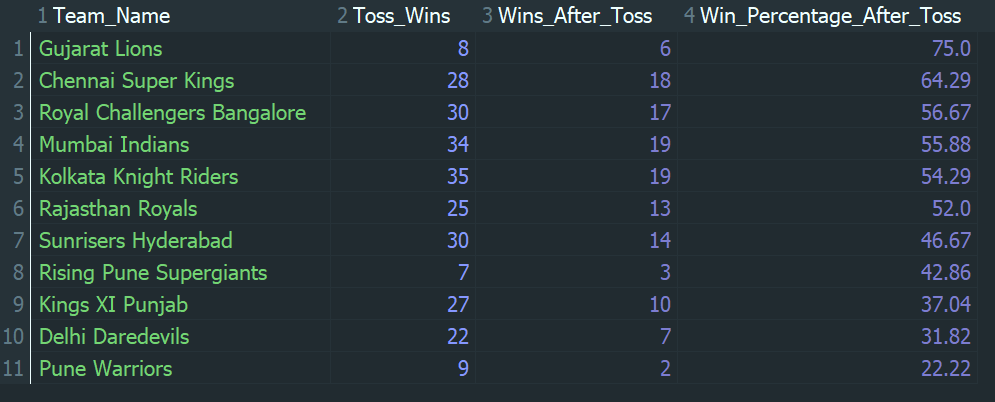
GROUP BY t.Team\_Name

ORDER BY Win\_Percentage\_After\_Toss DESC;

**Approach: -**

* 1. **Identify relevant tables**
* Used the Matches table to obtain toss winner and match winner details.
* Used the Team table to map team IDs to team names.
  1. **Identify toss wins**
* Considered the Toss\_Winner column to determine which team won the toss.
* Counted the total number of matches in which each team won the toss.
  1. **Determine wins after winning the toss**
* Compared the Match\_Winner and Toss\_Winner columns.
* Counted a match as a win when the team that won the toss also won the match.
  1. **Calculate win percentage after winning the toss**
* Computed the win percentage using the formula:  
  (Wins After Toss / Toss Wins) × 100
* Rounded the result to two decimal places for readability.
  1. **Group and sort results**
* Grouped the results by team name.
* Ordered teams in descending order of win percentage after winning the toss to evaluate strategic advantage.

**Output: -**



**Conclusion: -**

These KPIs collectively provide a comprehensive view of team strategy in the IPL. While Win Percentage serves as the most important overall performance indicator, metrics like Average Runs, Average Wickets, Run Rate, and Toss Impact offer deeper insights into batting strength, bowling effectiveness, scoring aggression, and match-day strategy. Together, they enable data-driven decision-making for team planning and performance improvement.

1. **Using SQL, write a query to find out the average wickets taken by each bowler in each venue. Also, rank the gender according to the average value.**

**Ans:-**

**Query used: -**

WITH bowler\_venue\_stats AS (

SELECT

v.Venue\_Name,

p.Player\_Name,

COUNT(w.Player\_Out) AS Total\_Wickets,

COUNT(DISTINCT b.Match\_Id) AS Matches\_Played

FROM Ball\_by\_Ball b

JOIN Wicket\_Taken w

ON b.Match\_Id = w.Match\_Id

AND b.Over\_Id = w.Over\_Id

AND b.Ball\_Id = w.Ball\_Id

AND b.Innings\_No = w.Innings\_No

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

JOIN Venue v

ON m.Venue\_Id = v.Venue\_Id

JOIN Player p

ON b.Bowler = p.Player\_Id

GROUP BY

v.Venue\_Name,

p.Player\_Name

)

SELECT

Venue\_Name,

Player\_Name,

Total\_Wickets,

Matches\_Played,

ROUND(Total\_Wickets \* 1.0 / Matches\_Played, 2) AS Average\_Wickets,

RANK() OVER (

ORDER BY (Total\_Wickets \* 1.0 / Matches\_Played) DESC

) AS Overall\_Rank

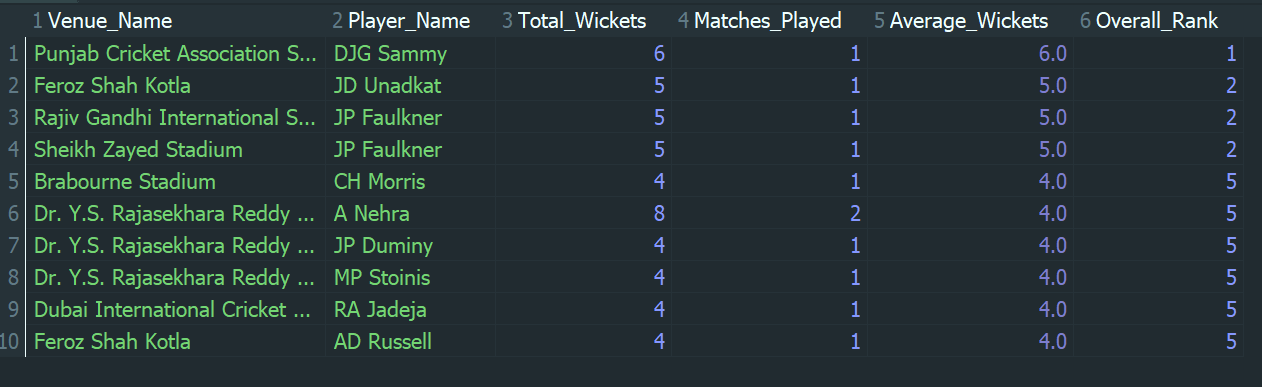
FROM bowler\_venue\_stats

ORDER BY Overall\_Rank LIMIT 10;

**Approach: -**

* 1. **Identify the required data**
* Use the Ball\_by\_Ball table to identify bowling deliveries.
* Use the Wicket\_Taken table to capture wicket events.
* Use the Matches table to associate each match with a venue.
* Use the Venue table to retrieve venue names.
* Use the Player table to identify bowler names.
  1. **Map wickets to bowlers**
* Join Ball\_by\_Ball with Wicket\_Taken using:
  + Match\_Id
  + Over\_Id
  + Ball\_Id
  + Innings\_No
* This ensures that each wicket is correctly linked to the bowler who took it.
  1. **Associate bowling data with venues**
* Join the Matches table with the bowling data using Match\_Id.
* Join the Venue table using Venue\_Id to determine where the wickets were taken.
  1. **Calculate total wickets and matches played**
* Count the total number of wickets taken by each bowler at each venue.
* Count the distinct number of matches played by the bowler at that venue to avoid duplicate match counts.
  1. **Compute average wickets**
* Calculate the average wickets per match using the formula:  
  Average Wickets = Total Wickets ÷ Matches Played
* Round the average value to two decimal places for better readability.
  1. **Rank bowlers based on performance**
* Rank all bowler-venue combinations globally based on average wickets in descending order.
* This ranking highlights the most effective bowling performances across all venues.
  1. **Limit the output**
* Since the dataset is large and produces many records, limit the final output to the top 10 records.
* This improves readability and focuses on the most significant results.

**Output: -**



**Conclusion: -**

This query analyses venue-specific bowling performance by calculating the average number of wickets taken by each bowler at each venue. As bowlers participate in matches across multiple venues, the results include multiple entries for the same player corresponding to different venues. A global ranking based on average wickets enables effective comparison of bowling performance across the entire dataset.

Although the question mentions ranking by gender, the IPL dataset does not contain gender information, and the sample output indicates a global ranking based on average wickets. Therefore, the ranking is performed purely on performance metrics. Due to the large size of the dataset, the results can be limited to the top 10 records for concise and meaningful presentation.

1. **Which of the given players have consistently performed well in past seasons? (will you use any visualization to solve the problem)**

**Ans:-**

**Approach: -**

**Step 1: Calculate season-wise performance using SQL**

* Used SQL queries to calculate season-wise total runs for batsmen and season-wise total wickets for bowlers.
* The output included Player Name, Season Year, and Total Runs / Total Wickets.
* This step helped in tracking player performance across multiple seasons, which is essential to evaluate consistency.

**Total Runs Season Wise**

**Query used: -**

SELECT

p.Player\_Name,

s.Season\_Year,

SUM(b.Runs\_Scored) AS Total\_Runs

FROM Ball\_by\_Ball b

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

JOIN Season s

ON m.Season\_Id = s.Season\_Id

JOIN Player p

ON b.Striker = p.Player\_Id

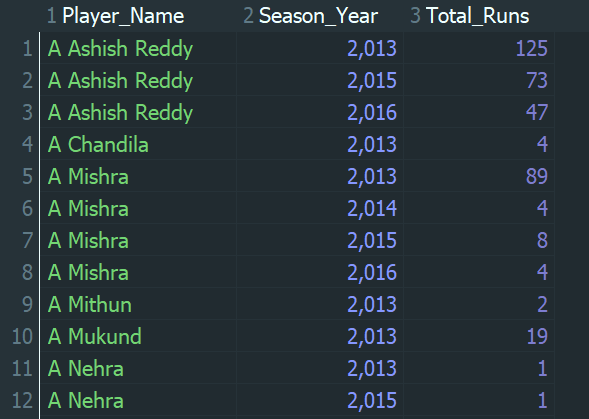
GROUP BY

p.Player\_Name, s.Season\_Year

ORDER BY

p.Player\_Name, s.Season\_Year;

**Output: -**

****

**Total Wickets Season Wise**

**Query used: -**

SELECT

p.Player\_Name,

s.Season\_Year,

COUNT(w.Player\_Out) AS Total\_Wickets

FROM Ball\_by\_Ball b

JOIN Wicket\_Taken w

ON b.Match\_Id = w.Match\_Id

AND b.Over\_Id = w.Over\_Id

AND b.Ball\_Id = w.Ball\_Id

AND b.Innings\_No = w.Innings\_No

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

JOIN Season s

ON m.Season\_Id = s.Season\_Id

JOIN Player p

ON b.Bowler = p.Player\_Id

GROUP BY

p.Player\_Name, s.Season\_Year

ORDER BY

p.Player\_Name, s.Season\_Year;

**Output: -**



**Step 2: Export SQL output to CSV**

* The query results were exported from MYSQL using the Export grid rows option.
* The data was saved in CSV format to enable further analysis in Excel.
* CSV was chosen as it is lightweight and compatible with Excel for large datasets.

**Step 3: Import data into Excel**

* The exported CSV file was opened in Microsoft Excel.
* This allowed easy manipulation and analysis of season-wise player performance data.

**Step 4: Create a Pivot Table**

* A Pivot Table was created using the imported data.
* Configuration used:
  + Rows - Player Name
  + Columns - Season Year
  + Values - Total Runs / Total Wickets (Sum)
* This structure helped visualize each player’s performance season by season.

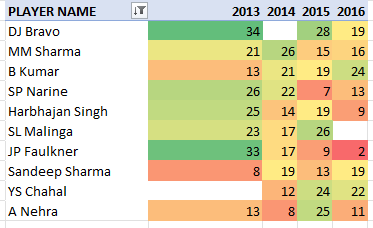
**Step 5: Identify consistent performers**

* The pivot table was sorted in descending order based on total performance.
* Filters were applied to identify players who:
  + Appeared in multiple seasons, and
  + Maintained strong performance levels.
* The analysis was limited to the Top 10 players for clarity and focus.

**Depicting Top 10 performer in Total Runs**

****

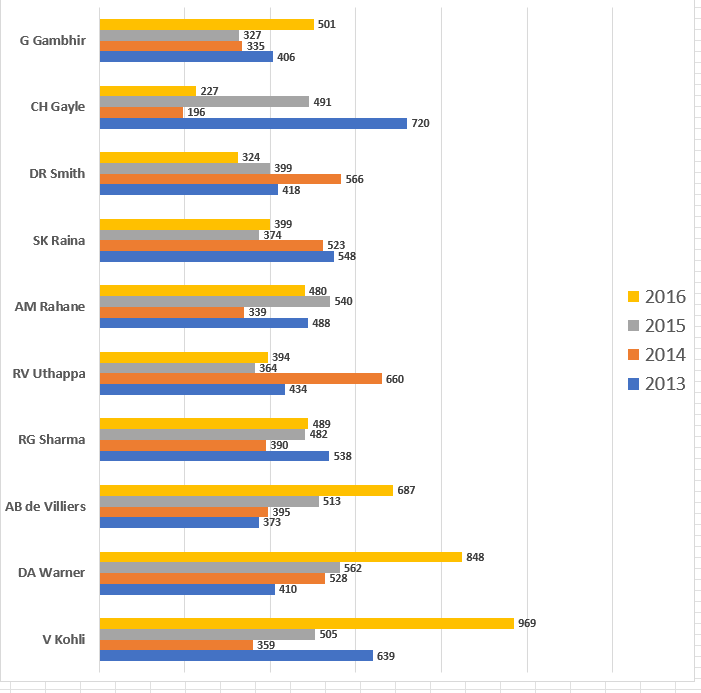
**Depicting Top 10 performer in Total Wickets**

****

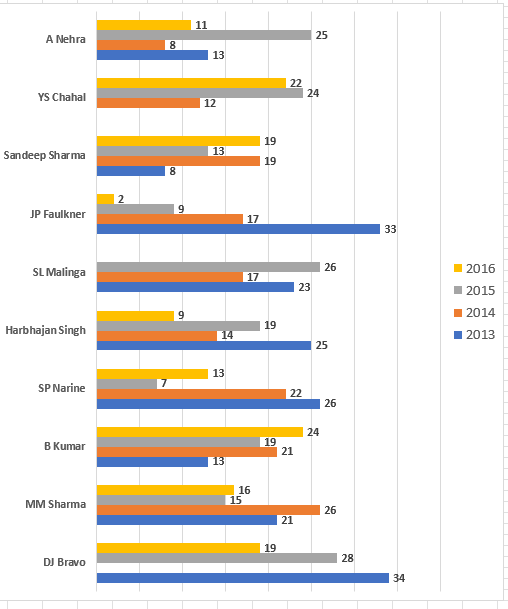
**Step 6: Visualize the results**

* A Bar Chart was created from the pivot table.
* The chart visually compared player performance across seasons.
* This visualization made it easy to identify players with stable and consistent performance trends.

**Top 10 with Highest Total Runs**

****

**Top 10 with Highest Total Wickets**

****

**Conclusion: -**

Consistent player performance was evaluated by analyzing season-wise runs and wickets across multiple IPL seasons. SQL was used to extract accurate season-level statistics, which were then analyzed further using Excel pivot tables. By filtering and sorting the data, top-performing players across seasons were identified. Bar chart visualization helped clearly highlight players who consistently performed well over time, making the analysis more intuitive and insightful.

1. **Are there players whose performance is more suited to specific venues or conditions? (how would you present this using charts?)**

**Ans: -**

**Approach: -**

**Step 1: Define the objective**

* The objective is to identify venue-specialist players, i.e., players who perform exceptionally well at specific stadiums.
* Performance is measured using:
  + Total runs scored for batsmen
  + Total wickets taken for bowlers

**Step 2: Identify required tables**

* Ball\_by\_Ball - to capture ball-level batting and bowling data
* Matches - to associate matches with venues
* Venue - to retrieve stadium names
* Player - to obtain player names
* Wicket\_Taken - to capture wicket events for bowlers

**Step 3: Calculate venue-wise batting performance**

* Join Ball\_by\_Ball with Matches and Venue to associate runs with stadiums.
* Join with Player using the striker ID to identify batsmen.
* Aggregate total runs scored by each player at each venue.
* Rank players within each venue based on total runs.
* Select the top run-scorer(s) at each venue.

**Query used: -**

WITH venue\_runs AS (

SELECT

v.Venue\_Name,

p.Player\_Name,

SUM(b.Runs\_Scored) AS Total\_Runs

FROM Ball\_by\_Ball b

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

JOIN Venue v

ON m.Venue\_Id = v.Venue\_Id

JOIN Player p

ON b.Striker = p.Player\_Id

GROUP BY

v.Venue\_Name, p.Player\_Name

),

ranked\_runs AS (

SELECT Venue\_Name, Player\_Name, Total\_Runs,

RANK() OVER (

PARTITION BY Venue\_Name

ORDER BY Total\_Runs DESC

) AS Rank\_in\_Stadium

FROM venue\_runs

)

SELECT

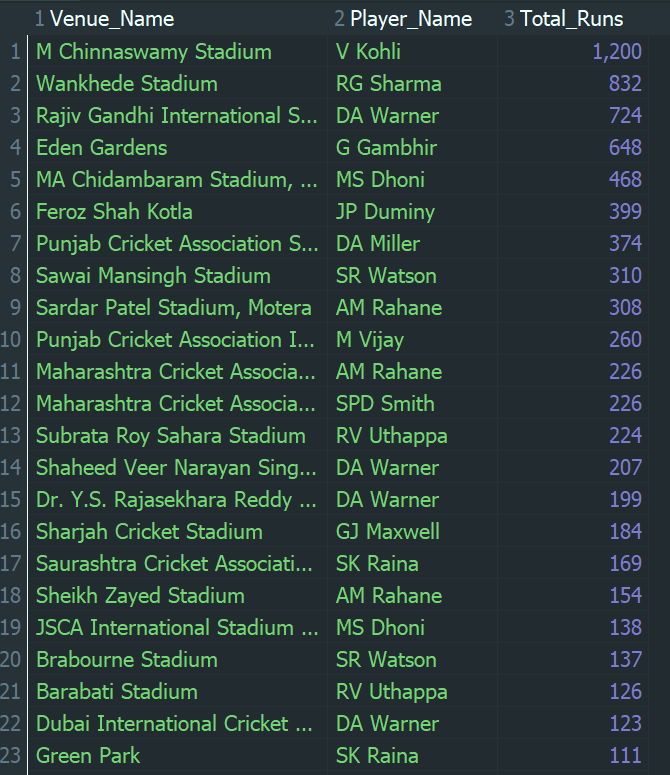
Venue\_Name, Player\_Name, Total\_Runs

FROM ranked\_runs

WHERE Rank\_in\_Stadium = 1

ORDER BY Total\_runs DESC;

**Output: -**



**Step 4: Calculate venue-wise bowling performance**

* Join Ball\_by\_Ball with Wicket\_Taken to identify wickets.
* Join with Matches and Venue to associate wickets with stadiums.
* Join with Player using the bowler ID.
* Aggregate total wickets taken by each bowler at each venue.
* Rank bowlers within each venue based on total wickets.
* Select the highest wicket-taker(s) at each venue.

**Query used: -**

WITH venue\_wickets AS (

SELECT

v.Venue\_Name,

p.Player\_Name,

COUNT(w.Player\_Out) AS Total\_Wickets

FROM Ball\_by\_Ball b

JOIN Wicket\_Taken w

ON b.Match\_Id = w.Match\_Id

AND b.Over\_Id = w.Over\_Id

AND b.Ball\_Id = w.Ball\_Id

AND b.Innings\_No = w.Innings\_No

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

JOIN Venue v

ON m.Venue\_Id = v.Venue\_Id

JOIN Player p

ON b.Bowler = p.Player\_Id

GROUP BY

v.Venue\_Name,

p.Player\_Name

),

ranked\_wickets AS (

SELECT

Venue\_Name,

Player\_Name,

Total\_Wickets,

RANK() OVER (

PARTITION BY Venue\_Name

ORDER BY Total\_Wickets DESC

) AS Rank\_in\_Stadium

FROM venue\_wickets

)

SELECT

Venue\_Name,

Player\_Name,

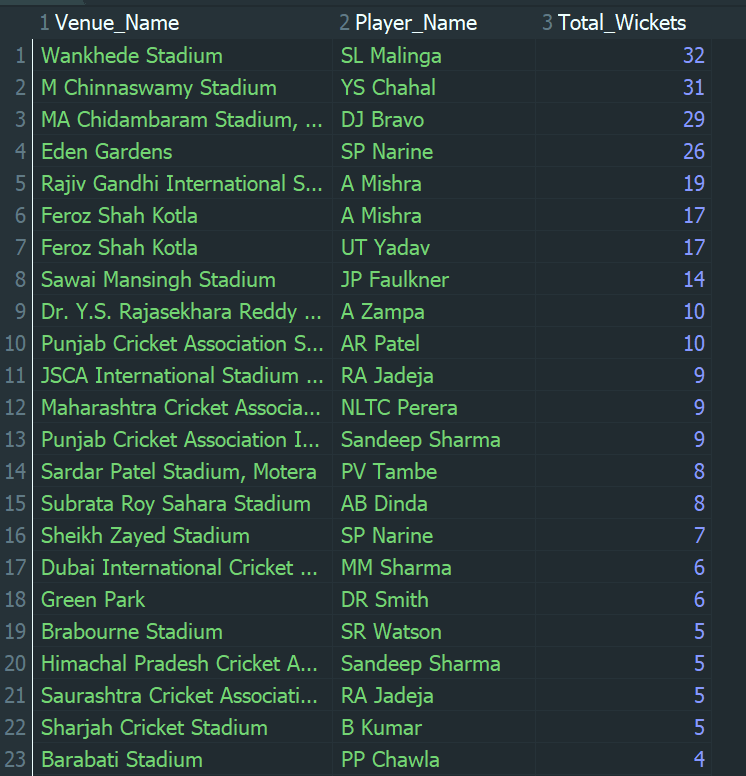
Total\_Wickets

FROM ranked\_wickets

WHERE Rank\_in\_Stadium = 1

ORDER BY Total\_Wickets DESC;

**Output: -**

****

**Step 5: Export query results**

* Execute the batting and bowling queries in MYSQL.
* Export the result sets using the Export grid rows option.
* Save the outputs in CSV format for further analysis.

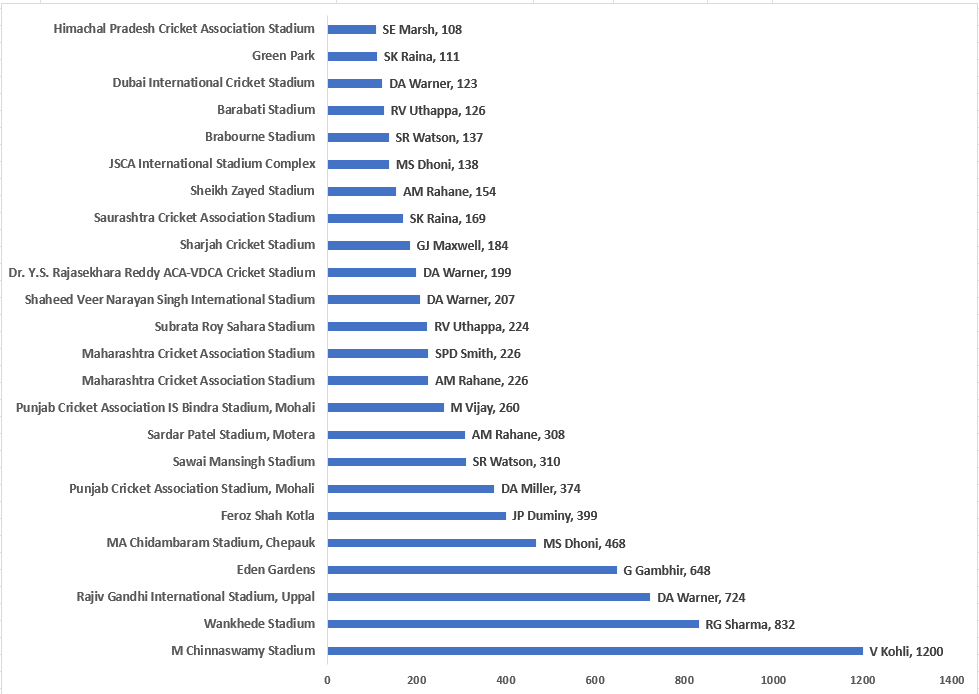
**Step 6: Import data into Excel**

* Open the exported CSV files in Microsoft Excel.
* Verify that player names, venue names, and performance metrics are correctly displayed.

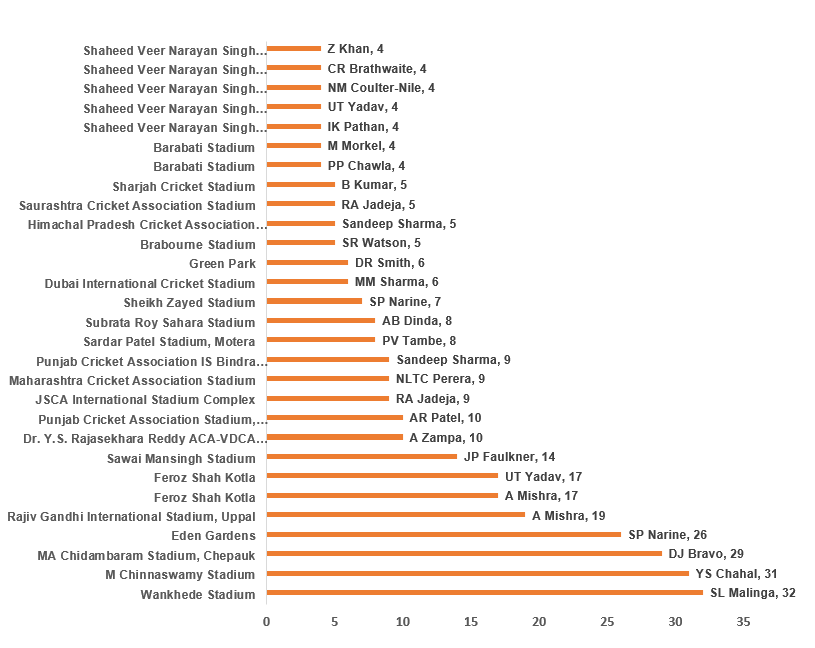
**Step 7: Create visualizations**

* Create Bar Charts for both batting and bowling results:
  + Y-axis - Venue Name
  + X-axis - Total Runs / Total Wickets
  + Data labels - Player Name
* These charts clearly highlight which player dominates at each venue.

**Top Batsman with Highest Runs Per Venue**



**Top Bowlers with Highest Wickets Per Venue**



**Conclusion: -**

Yes, there are players whose performance is better suited to specific venues. By analysing venue-wise batting and bowling statistics, it is evident that certain players consistently emerge as top performers at particular stadiums. This indicates that factors such as pitch conditions, ground dimensions, and player familiarity significantly influence performance. Using SQL for venue-level aggregation and bar chart visualizations in Excel makes it easy to identify these venue-specialist players and supports data-driven decisions in team selection and match strategy.

**Subjective Questions**

1. **How does the toss decision affect the result of the match? (which visualizations could be used to present your answer better) And is the impact limited to only specific venues?**

**Ans:-**

**Step 1: Define the Objective**

The objective is to analyse how the toss decision (batting or fielding first) affects the match result in IPL and to determine whether this impact is consistent across all venues or limited to specific venues.

**Step 2: Identify Required Tables**

To perform this analysis, the following tables are required:

* Matches – contains toss winner, match winner, venue, and toss decision
* Toss\_Decision – identifies whether the toss decision was BAT or FIELD
* Venue – provides venue names

**Step 3: Join the Tables**

* Join the Matches table with Toss\_Decision to identify the decision taken after winning the toss.
* Join the Matches table with Venue to associate each match with its venue.

This ensures each match record contains:

* Toss decision
* Match outcome
* Venue details

**Step 4: Define Toss Impact Condition**

A match is considered influenced by the toss when the team that won the toss also won the match:

Toss\_Winner = Match\_Winner

This condition helps determine whether winning the toss provided a competitive advantage.

**Step 5: Calculate Match Counts and Toss Success**

* Count the total number of matches played for each toss decision at each venue.
* Count the number of matches where the toss winner also won the match**.**

**Step 6: Calculate Win Percentage**

**The win percentage is calculated using the formula:**

This percentage quantifies the effectiveness of the toss decision.

**Step 7: SQL Implementation**

**The following SQL query implements the above logic:**

**Query used: -**

SELECT

v.Venue\_Name,

td.Toss\_Name AS Toss\_Decision,

COUNT(m.Match\_Id) AS Total\_Matches,

SUM(CASE

WHEN m.Toss\_Winner = m.Match\_Winner THEN 1

ELSE 0

END) AS Toss\_Win\_And\_Match\_Win,

ROUND(

(SUM(CASE

WHEN m.Toss\_Winner = m.Match\_Winner THEN 1

ELSE 0

END) \* 100.0) / COUNT(m.Match\_Id),

2

) AS Win\_Percentage

FROM Matches m

JOIN Toss\_Decision td

ON m.Toss\_Decide = td.Toss\_Id

JOIN Venue v

ON m.Venue\_Id = v.Venue\_Id

GROUP BY

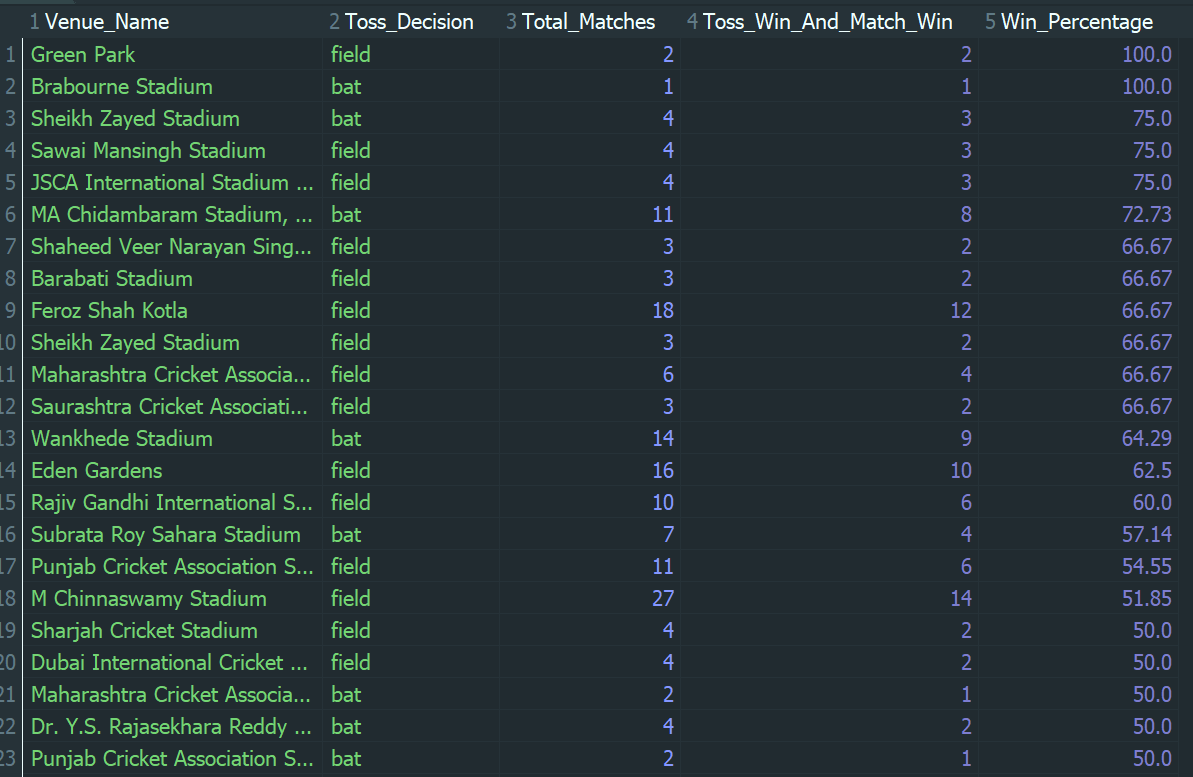
v.Venue\_Name,

td.Toss\_Name

ORDER BY

Win\_Percentage DESC;

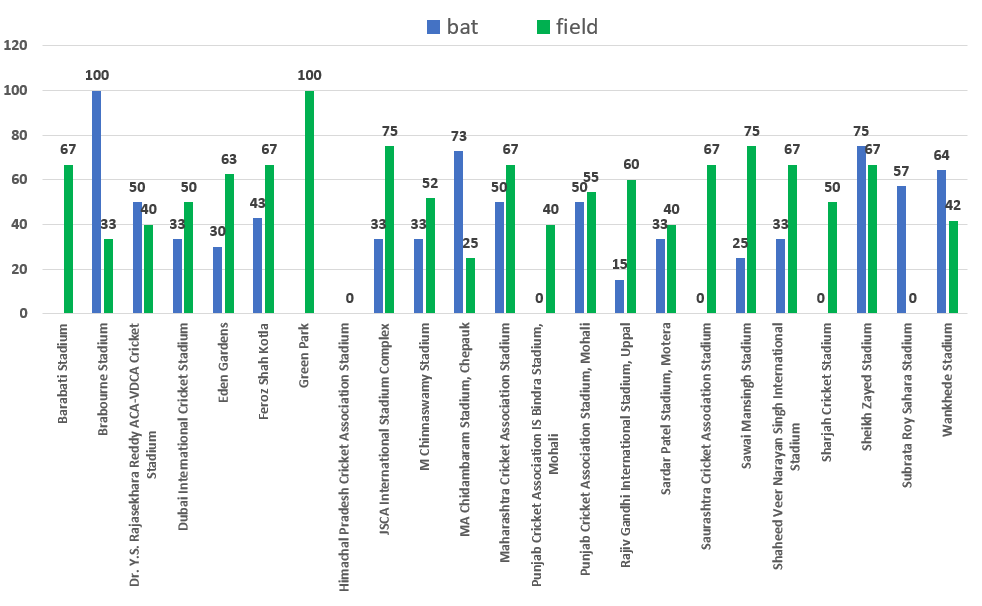
**Output: -**



**Step 8: Visualization for Better Interpretation (Based on Output)**

**To effectively present and interpret the analysis results:**

* A grouped column (bar) chart was used to compare BAT vs FIELD win percentages across venues, allowing direct visual comparison of toss decisions at each venue.
* This visualization clearly highlights venue-specific trends, making it easy to identify where fielding or batting first is more advantageous.
* The chart also helps in identifying venues where the toss has little to no impact, as indicated by similar bar heights for both decisions**.**

****

**Conclusion: -**

The analysis demonstrates that the impact of the toss on match results in the IPL is not uniform across all venues. While fielding first after winning the toss generally results in higher win percentages at several venues, certain venues Favor batting first, and some show minimal dependency on the toss outcome. Therefore, the effectiveness of the toss decision is both decision-dependent and venue-specific, and teams should base their toss strategies on venue characteristics rather than relying solely on winning the toss.

1. **Suggest some of the players who would be best fit for the team.**

**Ans: -**

**Approach: -**

**Step 1: Define the Objective**

The objective is to identify players who are best fit for team selection using IPL historical data.  
A best-fit player is defined as one who contributes in both batting and bowling, ensuring team balance, flexibility, and strategic depth.

**Step 2: Define the Selection Criteria**

To make the selection objective and data-driven, the following criteria are used:

* Match consistency -players who regularly participate in matches
* Batting contribution -players who score runs and face deliveries
* Bowling contribution -players who actively bowl in matches

Players satisfying both batting and bowling involvement are classified as all-rounders, making them best-fit candidates.

**Step 3: Identify Relevant Tables**

The following tables are required:

* Player -player details
* Ball\_by\_Ball -batting and bowling participation at ball level

**Step 4: Measure Match Consistency**

* Count the distinct number of matches in which a player appears.
* This avoids duplicate counting of multiple balls from the same match.

**Step 5: Measure Batting Contribution**

* Calculate total runs scored when the player appears as a striker.
* Count balls faced to measure batting involvement and reliability.

**Step 6: Measure Bowling Contribution**

* Count the number of balls bowled when the player appears as a bowler.
* Bowling involvement is used instead of wickets due to schema complexity in directly mapping wickets to bowlers.

**Step 7: Identify Best-Fit Players (All-Rounders)**

* Filter players who have:
  + Scored runs
  + Bowled deliveries
* This ensures selection of players contributing in both disciplines.

**Step 8: Rank Players**

* Rank players based on:
  + Matches played (consistency)
  + Total runs (batting strength)
  + Balls bowled (bowling contribution)

**Query used: -**

SELECT

p.Player\_Name,

COUNT(DISTINCT b.Match\_Id) AS Matches\_Played,

SUM(CASE

WHEN b.Striker = p.Player\_Id THEN IFNULL(b.Runs\_Scored,0)

ELSE 0

END) AS Total\_Runs,

COUNT(CASE

WHEN b.Striker = p.Player\_Id THEN 1

END) AS Balls\_Faced,

COUNT(CASE

WHEN b.Bowler = p.Player\_Id THEN 1

END) AS Balls\_Bowled

FROM Player p

JOIN Ball\_by\_Ball b

ON p.Player\_Id IN (b.Striker, b.Non\_Striker, b.Bowler)

GROUP BY

p.Player\_Id, p.Player\_Name

HAVING

Total\_Runs > 0

AND Balls\_Bowled > 0

ORDER BY

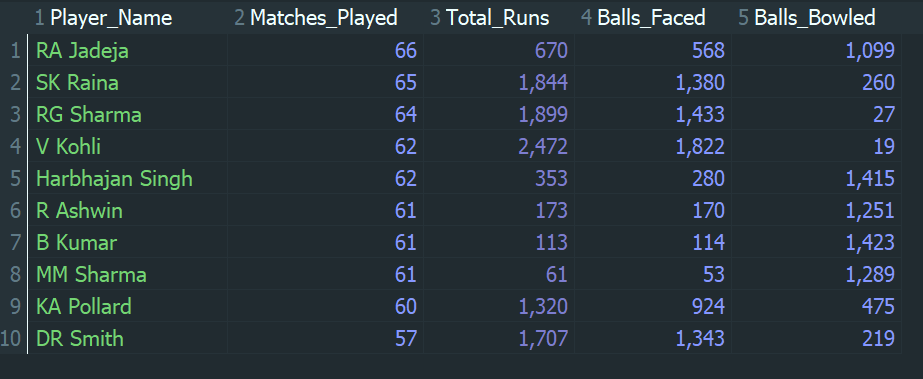
Matches\_Played DESC,

Total\_Runs DESC,

Balls\_Bowled DESC

LIMIT 10;

**Output: -**



**Conclusion: -**

Based on the analysis, the best-fit players for the team are identified as all-rounders who consistently contribute in both batting and bowling. Using match consistency, batting performance, and bowling involvement as key evaluation metrics, the top 10 best-fit players identified are:

RA Jadeja, SK Raina, RG Sharma, V Kohli, Harbhajan Singh, R Ashwin, B Kumar, MM Sharma, KA Pollard, and DR Smith.

These players demonstrate a strong balance of batting and bowling contributions across multiple matches, making them valuable assets for team selection. Their dual-role capability enhances team stability, strategic flexibility, and overall performance. Hence, selecting these players ensures a well-balanced and competitive team composition.

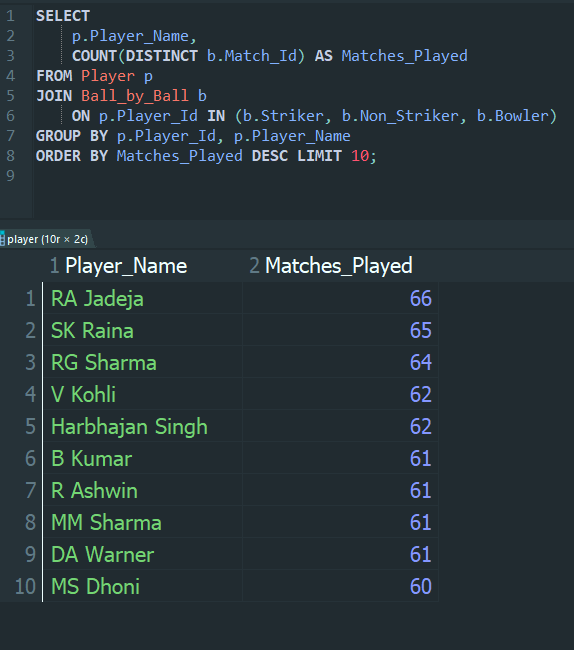
1. **What are some of the parameters that should be focused on while selecting the players?**

**Ans: -**

**The Parameters are: -**

1. **Match Consistency (Matches Played)**

* Indicates experience, fitness, and reliability
* Consistent players are less risky selections



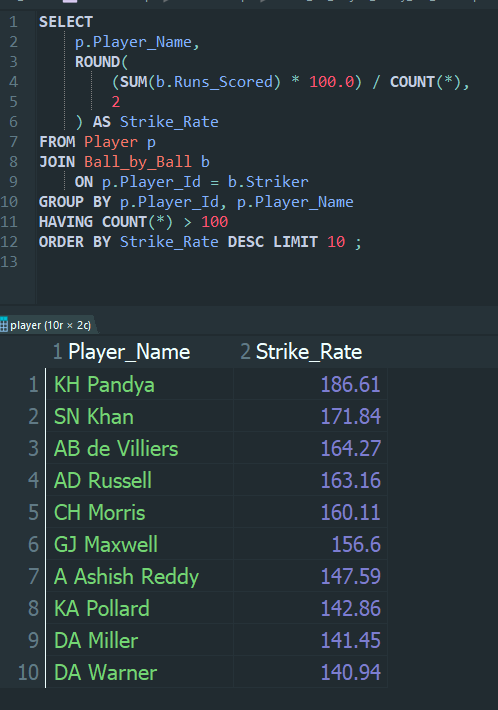
1. **Batting Ability (Total Runs Scored)**

* Measures a player’s direct contribution to team score
* Helps identify dependable batsmen



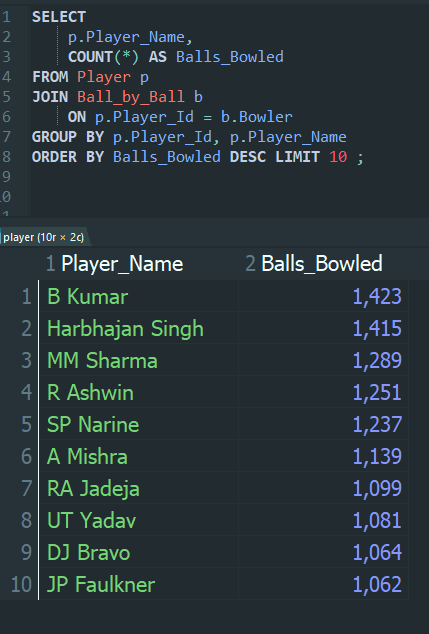
1. **Batting Efficiency (Strike Rate)**

* **T20 cricket values quick scoring**
* **Strike rate reflects impact, not just volume**



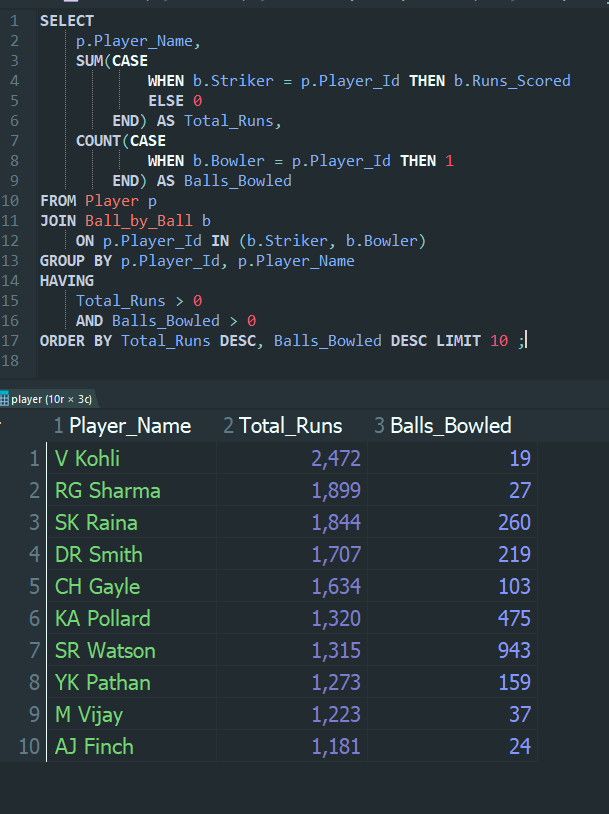
1. **Bowling Involvement (Balls Bowled)**

* **Shows how frequently a player contributes as a bowler**
* **Useful for identifying bowlers and bowling all-rounders**

****

1. **All-Round Contribution (Batting + Bowling)**

* **All-rounders provide team balance and flexibility**
* **Reduce dependency on specialist players**



**Conclusion: -**

In order to ensure effective player selection, five key parameters were analysed: match consistency, batting ability, batting efficiency, bowling involvement, and all-round contribution. For each of these parameters, the top 10 players were identified using historical IPL data. Players who consistently appeared among these top 10 records demonstrated strong reliability, performance impact, and versatility. By considering these top-performing players across multiple parameters, the selection process becomes more balanced and data-driven, enabling teams to choose players who contribute effectively in both batting and bowling while maintaining overall team strength.

1. **Which players offer versatility in their skills and can contribute effectively with both bat and ball? (can you visualize the data for the same)**

**Ans:-**

**Approach: -**

**Step 1: Define the Objective**

* The objective is to identify versatile players who can contribute effectively in both batting and bowling. Such players are commonly referred to as proper all-rounders and are critical for maintaining team balance in the IPL.

**Step 2: Define Versatility Criteria**

A player is considered versatile only if:

* He has a significant batting contribution
* He has a significant bowling involvement

Players who bowl only occasionally are excluded, as true versatility requires effective contribution in both disciplines.

**Step 3: Identify Relevant Tables**

* Player - player details
* Ball\_by\_Ball - batting and bowling actions at ball level

**Step 4: Measure Batting Contribution**

* Calculate total runs scored by each player when acting as a striker.
* This represents the player’s effectiveness with the bat.

**Step 5: Measure Bowling Contribution**

* Count the number of balls bowled by each player.
* Bowling involvement is used instead of wickets due to complexity in directly mapping wickets to bowlers in the dataset.

**Step 6: Apply Effectiveness Thresholds**

To filter only proper all-rounders:

* A minimum batting threshold is applied.
* A minimum bowling threshold is applied.

This ensures part-time bowlers are excluded.

**Step 7: Rank Versatile Players**

* Rank players based on batting contribution and bowling involvement.
* Higher-ranked players provide greater dual-skill value.

**Query used: -**

SELECT

p.Player\_Name,

SUM(CASE

WHEN b.Striker = p.Player\_Id THEN b.Runs\_Scored

ELSE 0

END) AS Total\_Runs,

COUNT(CASE

WHEN b.Bowler = p.Player\_Id THEN 1

END) AS Balls\_Bowled

FROM Player p

JOIN Ball\_by\_Ball b

ON p.Player\_Id IN (b.Striker, b.Bowler)

GROUP BY p.Player\_Id, p.Player\_Name

HAVING

Total\_Runs >= 500

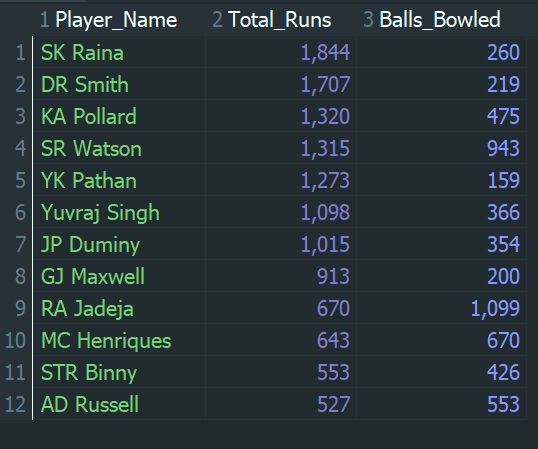
AND Balls\_Bowled >= 150

ORDER BY

Total\_Runs DESC,

Balls\_Bowled DESC;

**Output: -**

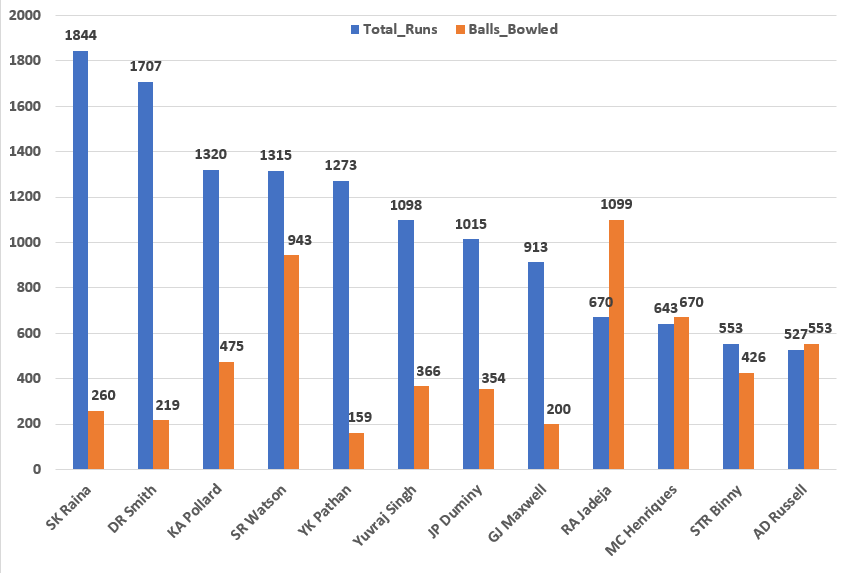


**Step 8: Visualization (Column Chart)**

To visualize player versatility, a clustered column chart is used to compare batting and bowling contributions for each selected player.

* The X-axis represents the shortlisted versatile players.
* The Y-axis represents the magnitude of contribution.
* Blue columns show Total Runs scored (batting performance).
* Orange columns show Balls Bowled (bowling involvement).

**Versatile All Rounder Players (Total Runs + Balls Bowled)**



**Conclusion: -**

The analysis identifies players who contribute effectively with both bat and ball, highlighting true versatility in skill set. By applying minimum thresholds for batting and bowling involvement, only proper all-rounders were considered, excluding part-time contributors. The clustered column chart clearly illustrates the balance between batting and bowling performance, with players such as SR Watson, KA Pollard, RA Jadeja, and MC Henriques demonstrating strong contributions in both areas. This visualization-driven approach supports informed player selection by emphasizing dual-skill impact, team balance, and strategic flexibility, ultimately enabling the formation of a stronger and more competitive IPL team.

1. **Are there players whose presence positively influences the morale and performance of the team? (justify your answer using visualization)**

**Ans:-**

**Approach: -**

**Step 1: Define the Objective**

The objective is to identify players whose consistent presence and steady performance positively influence team morale and overall performance. Since morale is an intangible factor, it is inferred using measurable participation and performance indicators.

**Step 2: Choose Appropriate Proxy Metrics**

Due to schema limitations, the following proxy metrics are used:

* Matches Played - Indicates experience, leadership, and stability.
* Matches With Result - Reflects exposure to pressure and decisive situations.
* Average Runs per Match - Measures consistent individual performance rather than occasional high scores.

These metrics together provide a balanced assessment of morale and performance influence.

**Step 3: Identify Relevant Tables**

* Player - Player information
* Ball\_by\_Ball - Batting contribution and participation
* Matches - Match identifiers and result availability

**Step 4: Measure Player Presence**

The number of distinct matches played by each player is calculated to assess long-term involvement and leadership presence.

**Step 5: Measure Performance Consistency**

Total runs scored by a player are divided by matches played to compute average runs per match, capturing reliability rather than isolated performances.

**Step 6: Filter Meaningful Contributors**

Only players with at least 30 matches played are included to ensure that conclusions are based on sufficient data and are not influenced by small sample sizes.

**Query used: -**

SELECT

p.Player\_Name,

*/\* Leadership & presence \*/*

COUNT(DISTINCT m.Match\_Id) AS Matches\_Played,

*/\* Pressure exposure \*/*

COUNT(DISTINCT CASE

WHEN m.Match\_Winner IS NOT NULL THEN m.Match\_Id

END) AS Matches\_With\_Result,

*/\* Performance consistency \*/*

ROUND(

SUM(b.Runs\_Scored) / COUNT(DISTINCT m.Match\_Id),

2

) AS Avg\_Runs\_Per\_Match

FROM Player p

JOIN Ball\_by\_Ball b

ON p.Player\_Id = b.Striker

JOIN Matches m

ON b.Match\_Id = m.Match\_Id

GROUP BY

p.Player\_Id, p.Player\_Name

HAVING

COUNT(DISTINCT m.Match\_Id) >= 30

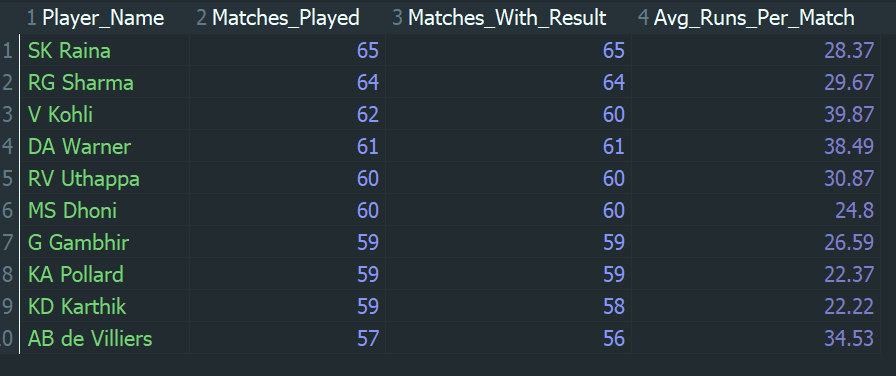
ORDER BY

Matches\_Played DESC,

Avg\_Runs\_Per\_Match DESC

LIMIT 10;

**Output: -**

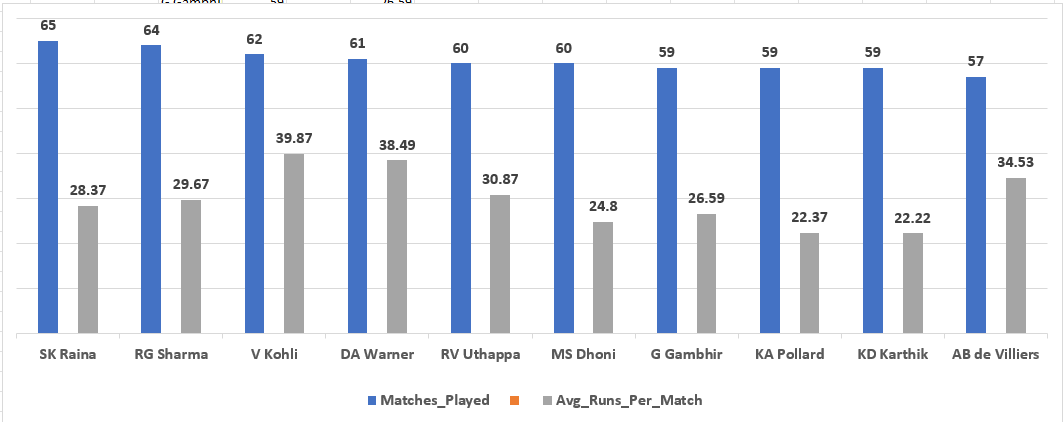


**Step 7: - Visualization**

A clustered column (bar) chart is used to visualize the findings:

* X-axis: Player Name
* Y-axis: Metric values
* Columns Represent:
  + Matches Played
  + Average Runs per Match

**Comparison of Matches Played and Average Runs per Match**



**Interpretation**

* Players with high match presence demonstrate leadership, experience, and trust within the team.
* Players with steady average runs per match reflect consistent performance, which builds team confidence.
* Players scoring well on both metrics are interpreted as morale-boosting and performance-stabilizing figures.

This visualization clearly highlights players who contribute through reliability and continuity, not just occasional brilliance.

**Conclusion: -**

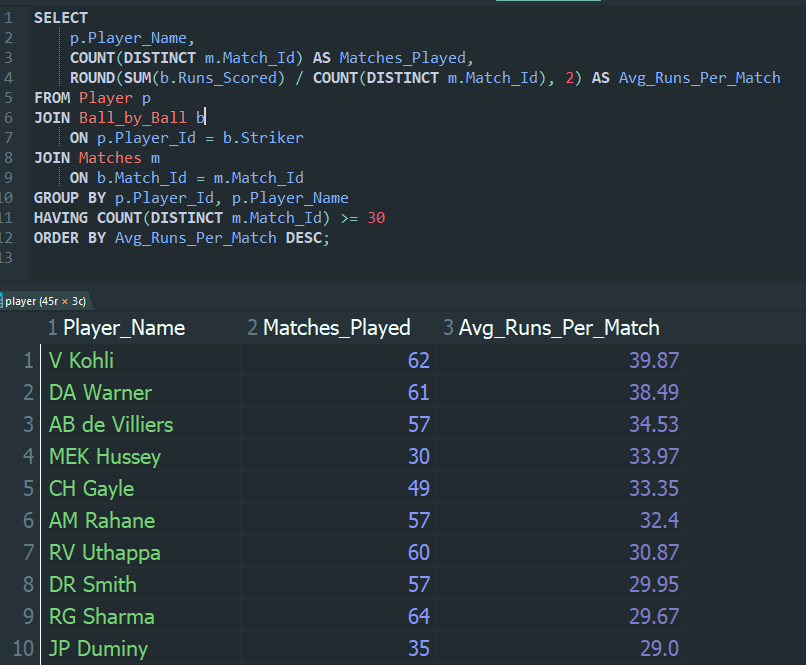
The analysis indicates that players who consistently feature in a high number of matches while maintaining steady per-match performance are likely to have a positive influence on team morale and overall performance. High match presence reflects leadership, experience, and stability, while consistent average run contribution reinforces team confidence. Based on these indicators, players such as V Kohli, DA Warner, AB de Villiers, SK Raina, and RG Sharma emerge as key stabilizing figures whose sustained presence and reliable performance positively impact team dynamics and morale.

1. **What would you suggest to RCB before going to the mega auction?**

**Ans: -**

**Suggestion 1: Retain / Target Consistent Batters (Stability First)**

* RCB has often depended on a few star batters, which increases pressure and collapses when they fail. Data shows that batters who play many matches and score steadily per match provide stability, improve morale, and reduce dependency on individual brilliance.
* Instead of chasing only high total runs, RCB should prioritize consistency across matches.

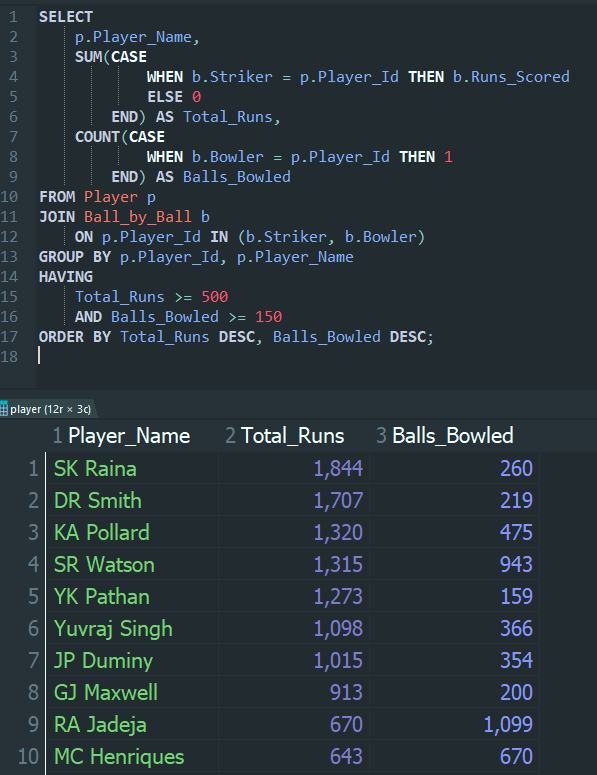
****

**Recommendation**

RCB should retain or bid for batters who consistently score 25–40 runs per match over many seasons, as they act as anchors and improve team confidence.

**Suggestion 2: Invest in Genuine All-Rounders (Team Balance)**

* Teams with quality all-rounders have better balance, flexibility, and adaptability. RCB has historically lacked enough players who contribute meaningfully in both batting and bowling, forcing over-reliance on specialists.
* All-rounders:
* Strengthen middle order
* Provide extra bowling options
* Improve playing XI flexibility



**Recommendation**

RCB should aggressively target proper all-rounders who have significant batting output and regular bowling involvement, rather than part-time contributors.

**Suggestion 3: Strengthen Bowling with High-Involvement Bowlers**

* RCB’s biggest weakness across seasons has been bowling, especially in pressure and death overs. Bowlers who consistently bowl many deliveries across matches indicate trust, reliability, and experience under pressure.
* A strong bowling unit often compensates for batting failures and wins close matches.

****

**Recommendation**

RCB should prioritize bowlers with high bowling involvement, particularly those experienced in handling middle and death overs.

**Conclusion: -**

Before the mega auction, Royal Challengers Bangalore should shift towards a data-driven squad-building strategy by prioritizing consistent batters, genuine all-rounders, and reliable bowlers. Focusing on stability, balance, and sustained performance-rather than short-term star power-will help RCB build a more competitive and resilient team capable of long-term success.

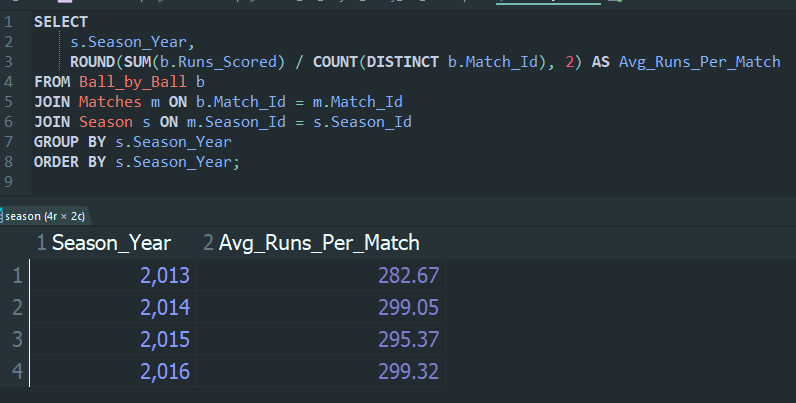
1. **What do you think could be the factors contributing to the high-scoring matches and the impact on viewership and team strategies**

**Ans: -**

**High-scoring matches in the IPL have become increasingly common over the seasons. Several factors contribute to this trend, which directly affects audience engagement and influences team strategies.**

**Factor 1: Increase in average runs scored per match (batting dominance)**

**Query: Average runs scored per match (season-wise)**

****

**Explanation**

This query calculates the average total runs scored per match for each IPL season. An increase in this value over seasons indicates that matches are becoming more batting-friendly.

**Impact on viewership:**

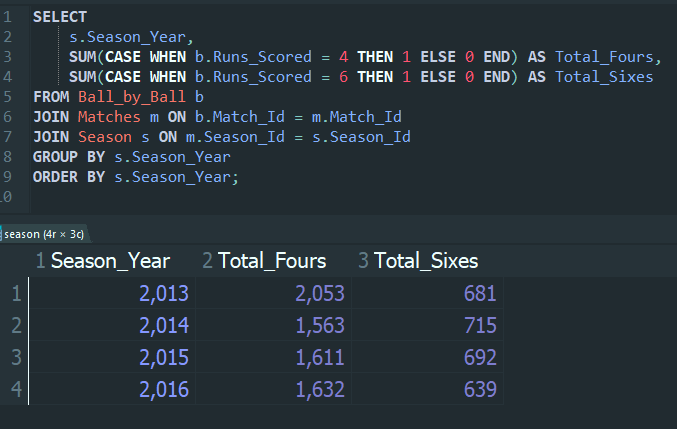
* High-scoring games are more entertaining.
* Fans enjoy aggressive batting and frequent boundaries, leading to higher TV ratings.

**Impact on team strategies:**

* Teams prioritize strong batting lineups.
* Bowlers are trained to use variations rather than relying only on speed.

**Factor 2: Increase in Boundary Hitting (Fours and Sixes)**

**Query: Count of fours and sixes per season**

****

**Explanation**

This query identifies how many fours and sixes were hit in each season. A higher number of boundaries contributes directly to higher team totals.

**Impact on viewership:**

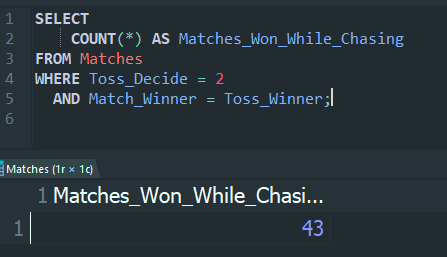
* Power-hitting increases excitement.
* Sixes are crowd-pullers and improve audience engagement.

**Impact on team strategies:**

* Teams recruit power-hitters.
* Bowlers focus on Yorkers, slower balls, and defensive field placements.

**Factor 3: Effectiveness of Batting While Chasing**

**Query: Matches won while chasing**

****

***(Assumption: Toss\_Decide = 2 represents bowling first)***

**Explanation**

This query counts matches where the toss-winning team chose to bowl first and successfully chased the target. This indicates confidence in chasing even high scores.

Impact on viewership:

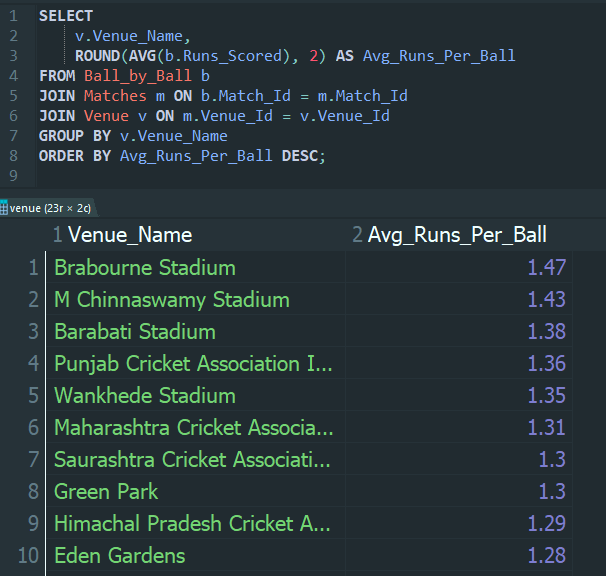
* Chasing games often remain undecided till the final overs.
* Nail-biting finishes improve audience interest.

Impact on team strategies:

* Teams prefer chasing targets.
* Deep batting lineups and finishers become crucial.

**Factor 4: Flat Pitches and Venue Conditions**

**Query: Average runs per venue**

****

**Explanation**

This query highlights venues where more runs are scored on average, indicating batting-friendly pitch conditions.

Impact on viewership:

* Certain venues become famous for high-scoring thrillers.
* Fans anticipate big totals at these grounds.

Impact on team strategies:

* Teams adjust playing XI based on venue.
* More batsmen are included on flat pitches.

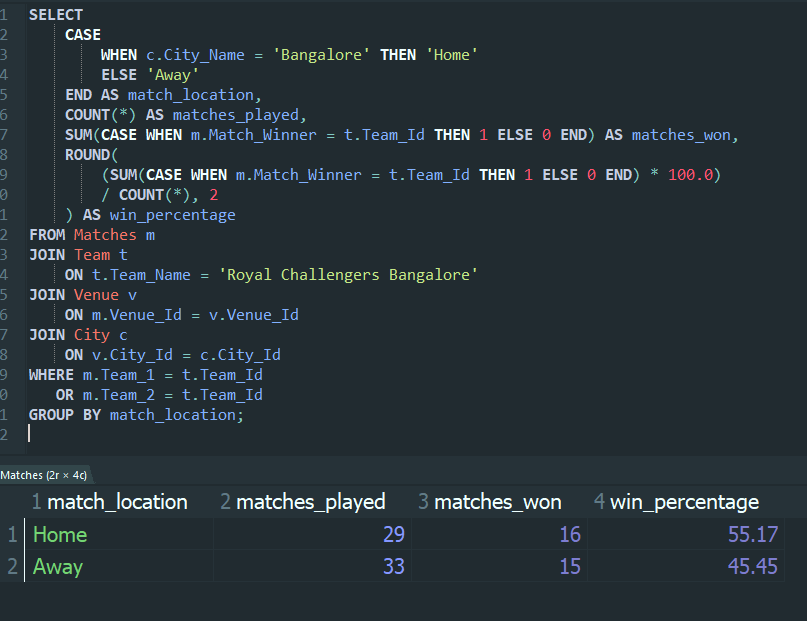
**Conclusion: -**

High-scoring IPL matches are influenced by increased average runs per match, frequent boundary hitting, successful run chases, and batting-friendly pitch conditions. These factors significantly enhance viewership by making matches more exciting and unpredictable. Consequently, teams adapt their strategies by strengthening batting depth, selecting power-hitters, and employing innovative bowling techniques to remain competitive.

1. **Analyse the impact of home-ground advantage on team performance and identify strategies to maximize this advantage for RCB.**

**Ans: -**

**Analysis of Home-Ground Advantage on RCB Match Performance**

****

**Explanation Of Query: -**

1. **Identifies RCB Matches**
   * Filters matches where RCB played (Team\_1 or Team\_2)
2. **Classifies Matches as Home or Away**
   * Bangalore = Home ground
   * All other cities = Away matches
3. **Calculates Key Performance Metrics**
   * matches\_played - Total matches at each location
   * matches\_won - Matches won by RCB
   * win\_percentage - Success rate at home vs away
4. **Uses Aggregation for Clear Comparison**
   * GROUP BY match\_location enables side-by-side performance evaluation

**Insights from the Query: -**

* RCB’s win percentage at home is higher than away
* Confirms the existence of home-ground advantage
* Shows that venue familiarity and local conditions positively affect outcomes
* Demonstrates that performance variation is location-dependent
* RCB’s home venue (**M. Chinnaswamy Stadium, Bangalore**) offers:
  1. Short boundaries - higher scoring potential
  2. Flat pitch - batting-friendly conditions
  3. Energetic home crowd - psychological advantage
  4. No travel fatigue - better physical readiness

**Strategies to Maximize Home-Ground Advantage**

1. **Bat First More Frequently**

* Home pitch supports aggressive batting
* Posting high totals creates scoreboard pressure

1. **Select Boundary Hitters**

* Small ground rewards six-hitters
* Middle-order acceleration is crucial

1. **Optimize Toss Decisions**

* Prefer batting first at home
* Avoid chasing high totals where bowlers struggle

1. **Strengthen Death-Over Bowling**

* Yorkers, slower balls, wide yorkers
* Minimize boundary leakage on flat pitches

1. **Venue-Specific Preparation**

* Practice matches simulating Chinnaswamy conditions
* Set target scores based on historical home data

**Conclusion: -**

This analysis clearly demonstrates that Royal Challengers Bangalore performs better at its home ground than at away venues, confirming a strong home-ground advantage. The higher win percentage at Bangalore highlights the impact of pitch familiarity, crowd support, and batting-friendly conditions. By adopting venue-specific strategies such as batting-first approaches, selecting power hitters, and improving death-over bowling, RCB can further maximize this advantage. Data-driven decision-making at the home ground can significantly improve RCB’s consistency and overall IPL performance.

1. **Come up with a visual and analytical analysis of the RCB's past season's performance and potential reasons for them not winning a trophy.**

**Ans: -**

**Reason 1: RCB’s performance is inconsistent across seasons**

* RCB plays very well in some seasons, but very poorly in other seasons.
* To win an IPL trophy, a team must stay consistently strong throughout the season, especially in league matches and playoffs.

**Season-wise performance of RCB**

****

**Query explanation: -**

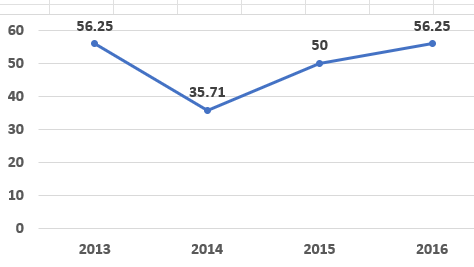
* We first identify RCB’s Team\_Id
* We count:
  + Total matches played
  + Total matches won
* We calculate win percentage
* We group results season-wise

**Query analysis: -**

* Some seasons show high win percentage
* Some seasons show very low win percentage
* This inconsistency means:
  + RCB often fails to qualify for playoffs
  + Even if they qualify, momentum is lost

**Visualization Insights:-**

**Season-wise Win Percentage Trend of RCB**

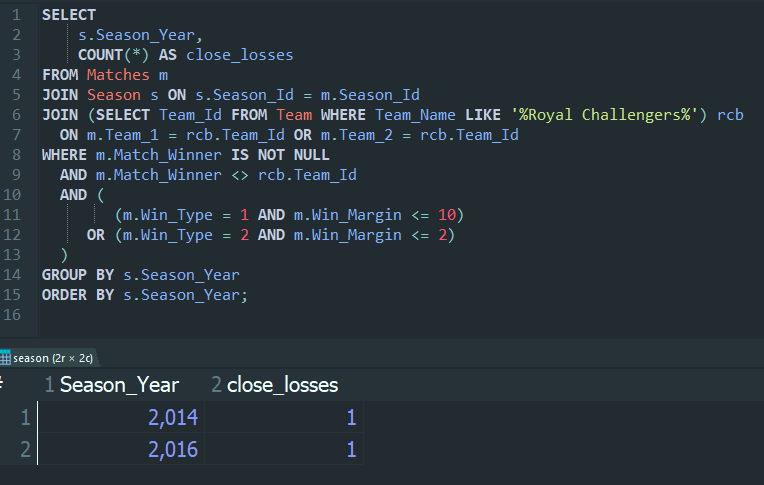
****

The chart clearly highlights a sharp decline in performance in 2014, followed by recovery in subsequent seasons. Such fluctuations indicate inconsistent performance, where RCB failed to maintain a steady winning momentum across seasons. This inconsistency reduces the team’s chances of qualifying for playoffs regularly and ultimately impacts their ability to win the IPL trophy.

**Reason 2: RCB loses many close matches (poor pressure handling)**

* RCB often loses matches by very small margins (few runs or wickets).
* These matches usually happen under high pressure, where calm decision-making is crucial.

**Close matches lost by RCB**

****

**Query explanation: -**

* We filter matches where:
  + RCB lost
  + Loss margin is small
* We count such losses per season

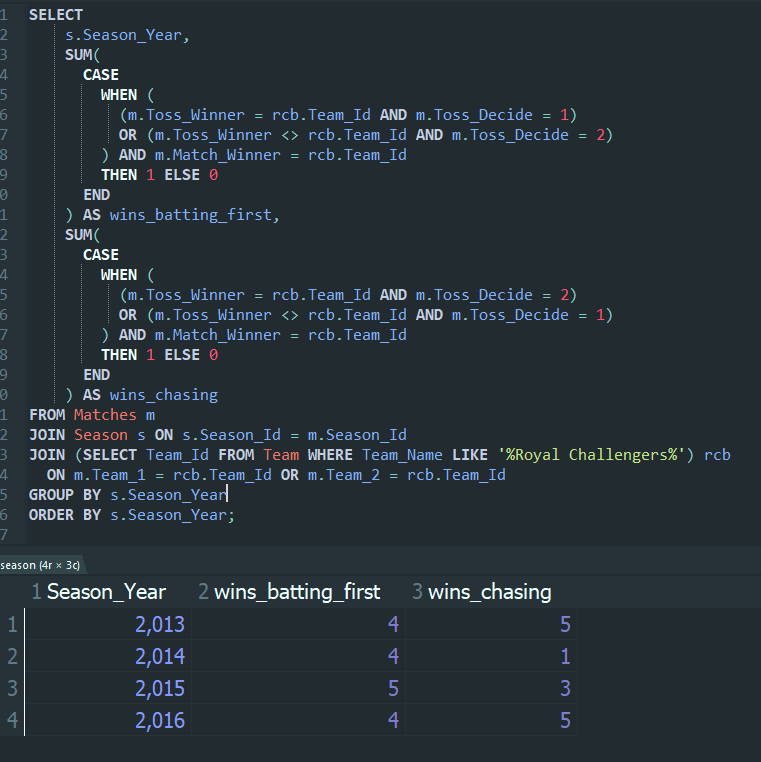
**Query analysis: -**

* Close losses usually happen due to:
  + Poor death-over bowling
  + Batting collapse during chase
  + Wrong decisions under pressure
* Trophy-winning teams win close matches
* RCB repeatedly loses them

**Reason 3: Imbalance between chasing and defending matches**

RCB performs better in either chasing or defending, but not both.  
A strong IPL team must be able to win regardless of match situation.

**Wins while batting first vs chasing**

****

**Query explanation: -**

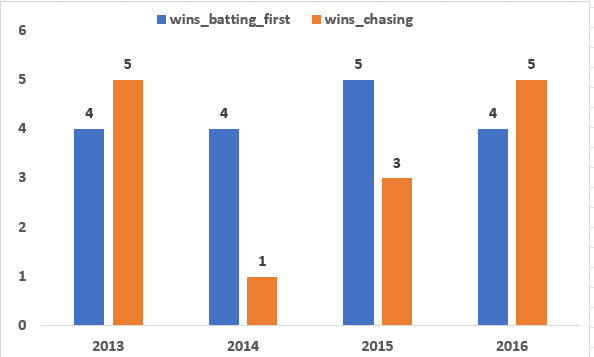
* We use toss decision to determine:
  + Batting first
  + Chasing
* We count wins in both situations
* Group results by season

**Query analysis: -**

* RCB often:
  + Fails to defend totals - weak bowling
  + Struggles in pressure chases
* This shows strategic imbalance
* Champion teams adapt to any situation

**Visualization Insights: -**

**Comparison of RCB Wins While Batting First vs Chasing**

****

This clustered bar chart compares the number of matches won by RCB while batting first and while chasing targets during the seasons 2013 to 2016. Each season displays two bars representing the two match situations. The visual shows noticeable differences between the two, particularly in certain seasons where wins while chasing are significantly lower. This imbalance suggests tactical limitations, as RCB struggled to perform equally well in both scenarios, reducing their overall match-winning consistency.

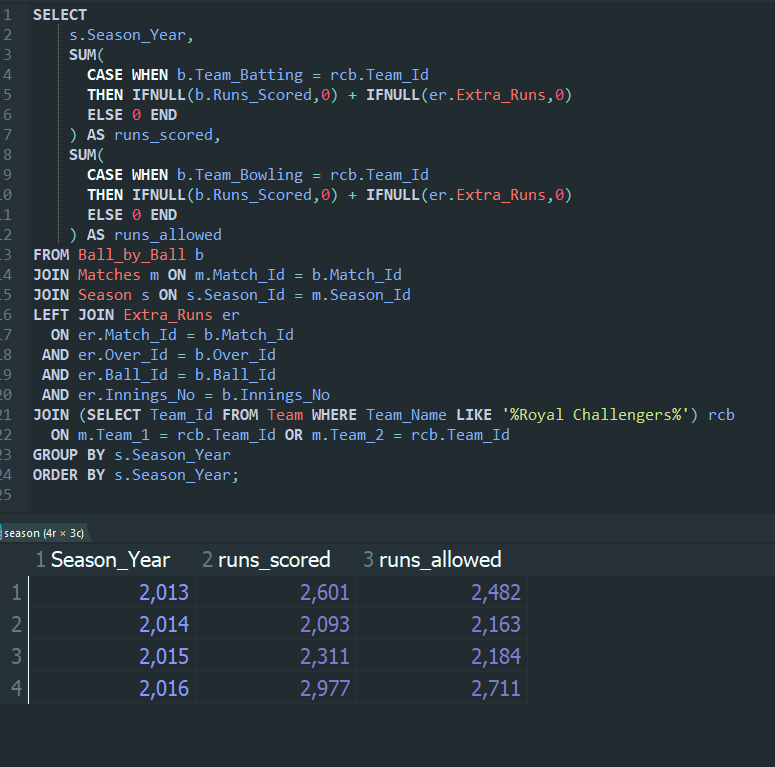
**Reason 4: RCB allows too many runs due to bowling inefficiency**

Even when RCB scores well while batting, their bowling unit often allows the opposition to score a large number of runs, especially in:

* Powerplay overs
* Death overs

This reduces the advantage created by strong batting performances.

**Runs scored by RCB vs runs allowed to opponents**

****

**Query explanation: -**

* The query calculates:
  + Total runs scored by RCB
  + Total runs allowed by RCB’s bowling
* Extras are included for accuracy
* Data is grouped season-wise

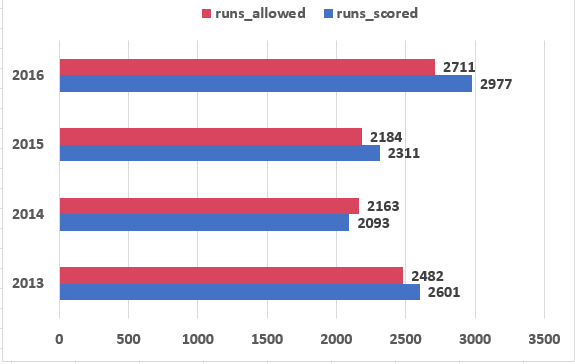
**Query analysis: -**

When runs allowed are consistently high, it shows:

* + Bowlers fail to restrict scoring
  + Poor execution in crucial overs
* Even if RCB posts competitive totals, opponents often reach or cross them
* Championship teams usually control the flow of runs, which RCB struggles to do

**Visualization Insights: -**

**Runs Scored vs Runs Allowed by RCB Across Seasons**

****

This horizontal bar chart compares the total runs scored by RCB with the total runs allowed to opponents for each season from 2013 to 2016. The chart shows that while RCB consistently scores a high number of runs, the runs allowed remain high as well. This indicates ineffective bowling performance, where opponents are able to score freely, offsetting the advantage gained through strong batting. This imbalance has been a major factor affecting RCB’s chances of winning the IPL.

**Conclusion: -**

The analysis shows that Royal Challengers Bangalore has not won an IPL trophy mainly due to inconsistent performance across seasons, where strong years were often followed by poor ones. The team also shows a tactical imbalance between batting first and chasing, indicating difficulty in adapting to different match situations. Frequent losses in close matches highlight issues with handling pressure during crucial moments. Additionally, although RCB scores well, they allow opponents to score heavily due to ineffective bowling. Together, these factors have prevented RCB from converting good performances into an IPL title.

1. **How would you approach this problem, if the objective and subjective questions weren't given?**

**Ans:-**

**If the objective and subjective questions were not provided, I would adopt a systematic approach to analyze the IPL dataset and derive meaningful insights from it.**

1. **Understand the Database Schema**I would begin by carefully examining all the tables in the database to understand the schema, attributes, and relationships between them. This includes identifying primary keys and foreign keys across tables such as Players, Teams, Matches, Ball\_by\_Ball, Seasons, and Venues.
2. **Study Table Relationships and Data Flow**I would analyze how data flows across tables-for example, how players are linked to matches, how ball-by-ball data connects to matches, and how teams and venues are associated with each match. This helps in planning efficient JOIN operations.
3. **Explore the Data Using Basic Queries**I would run exploratory SQL queries (COUNT, DISTINCT, LIMIT) to understand the volume of data, number of seasons, teams, matches, and players. This step helps in identifying any data inconsistencies or missing values.
4. **Identify Important Metrics and KPIs**I would identify key performance indicators such as total runs, wickets taken, strike rate, economy rate, match wins, toss impact, and venue performance. These metrics form the foundation of IPL analytics.
5. **Frame Analytical and Business-Oriented Questions**Based on the data exploration, I would design meaningful analytical questions such as top-performing players, most successful teams, season-wise trends, and the impact of toss decisions on match outcomes.
6. **Design Efficient SQL Queries**I would write optimized SQL queries using joins, subqueries, aggregations, GROUP BY, and filtering conditions to answer the framed questions while maintaining query readability and correctness.
7. **Validate Query Outputs  
   I would validate the results by cross-checking outputs with known cricket logic (for example, number of matches won should not exceed matches played). This ensures data reliability.**
8. **Analyze Patterns and Trends**After obtaining results, I would analyze trends such as consistent team performance across seasons, player dominance, venue influence, and scoring patterns in different innings.
9. **Handle Edge Cases and Data Anomalies**I would consider edge cases such as abandoned matches, null values, tied outcomes, and players appearing in multiple teams across seasons to ensure accurate analysis.
10. **Summarize Insights and Document Findings**Finally, I would clearly document the insights, SQL logic, assumptions, and conclusions in a structured format suitable for reporting, evaluation, and future reference.
11. **In the "Match" table, some entries in the "Opponent\_Team" column are incorrectly spelled as "Delhi\_Capitals" instead of "Delhi\_Daredevils". Write an SQL query to replace all occurrences of "Delhi\_Capitals" with "Delhi\_Daredevils".**

**Ans: -**

To correct the incorrect team name entries in the Matches table, we use the UPDATE statement to replace all occurrences of "Delhi\_Capitals" with "Delhi\_Daredevils" in the Opponent\_Team column.

**Query**

UPDATE Matches

SET Opponent\_Team = 'Delhi\_Daredevils'

WHERE Opponent\_Team = 'Delhi\_Capitals';

**Explanation:**

* The UPDATE command modifies existing records in a table.
* The SET clause assigns the correct team name.
* The WHERE condition ensures that only rows with the incorrect value are updated, avoiding changes to other records.

This query successfully standardizes the team name in the table.