**Objective Questions**

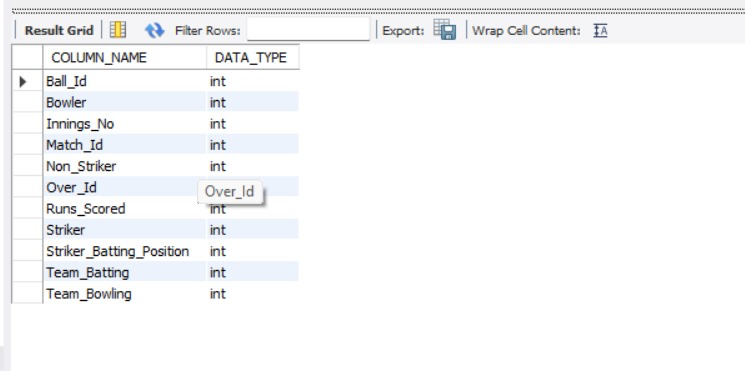
1. **List the different dtypes of columns in table “ball\_by\_ball” (using information\_schema)**

* Code:

**SELECT COLUMN\_NAME, DATA\_TYPE FROM information\_schema.columns**

**WHERE table\_name = 'Ball\_by\_Ball' AND table\_schema = "ipl";**

* This query fetches the column names and their data types for the table Ball\_by\_Ball in the IPL database.
* The query uses the information\_schema.columns table, which stores metadata about all the columns in the database.
* The WHERE clause ensures that only the columns of the Ball\_by\_Ball table are returned.
* The table “ball\_by\_ball” contains the following data types:



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)**

* Code:

**WITH extra\_run\_data AS (**

**SELECT**

**Team\_Batting AS team\_Id,**

**SUM(e.Extra\_Runs) as total\_extra**

**FROM ball\_by\_ball b**

**JOIN extra\_runs e ON e.Match\_Id = b.Match\_Id AND e.Innings\_No = b.Innings\_No AND e.Over\_Id = b.Over\_Id AND e.Ball\_Id = b.Ball\_Id**

**WHERE Team\_Batting = 2**

**AND b.Match\_Id IN(**

**SELECT distinct Match\_Id FROM matches WHERE Season\_Id = ( SELECT MIN(Season\_Id) as first\_season FROM Matches WHERE Team\_1 = 2 OR Team\_2 = 2))**

**),**

**run\_scored\_data AS (**

**SELECT**

**Team\_Batting AS team\_Id,**

**SUM(b.Runs\_Scored) AS total\_score**

**FROM ball\_by\_ball b**

**JOIN matches m ON m.Match\_Id = b.Match\_Id**

**WHERE Team\_Batting = 2 AND (Team\_1 = 2 OR Team\_2 = 2) AND m.Season\_Id = ( SELECT MIN(Season\_Id) as first\_season FROM Matches WHERE Team\_1 = 2 OR Team\_2 = 2)**

**)**

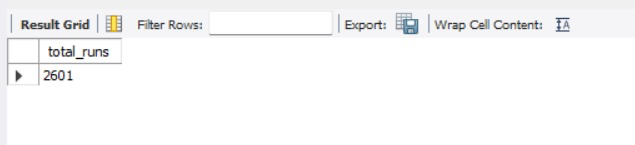
**SELECT**

**(total\_score + total\_extra) AS total\_runs**

**FROM run\_scored\_data s**

**JOIN extra\_run\_data e ON e.team\_Id = s.team\_Id;**

* This query calculates the total number of runs scored by RCB in the first season they played.
* The query uses two subqueries:
* extra\_run\_data: It calculates the total extra runs (like wides, no-balls) from the extra\_runs table.
* run\_scored\_data: It sums up the runs scored by RCB in the first season from the ball\_by\_ball table.
* The final SELECT sums up both the runs scored and the extra runs to give the total.



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

* Code:

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

**FROM Player p**

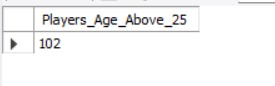
**JOIN Player\_Match pm ON p.Player\_Id = pm.Player\_Id**

**JOIN Matches m ON pm.Match\_Id = m.Match\_Id**

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

**WHERE s.Season\_Year = 2014**

**AND TIMESTAMPDIFF(YEAR, p.DOB, '2014-12-31') > 25;**



* This query counts the number of distinct players whose age was greater than 25 during the 2014 season.
* The query joins the players table (where player information like birth date is stored) and the matches table (to filter by the 2014 season).
* The age calculation is done by subtracting the year of birth from the current year, and checking if the result is greater than 25.

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

* Code:

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

**FROM Matches m**

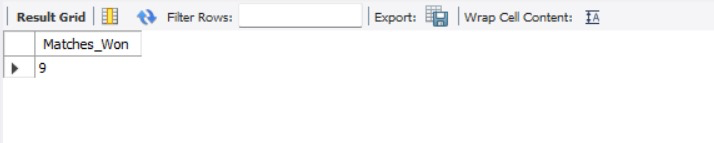
**JOIN Season s ON m.Season\_Id = s.Season\_Id**

**JOIN Team t ON m.Match\_Winner = t.Team\_Id**

**WHERE s.Season\_Year = 2013**

**AND t.Team\_Name = 'Royal Challengers Bangalore'**

**AND m.Match\_Winner IS NOT NULL;**

* The query counts the number of matches won by RCB in the 2013 season by filtering matches where RCB was the winner.

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

* Code:

**WITH Last\_4\_Seasons AS (**

**SELECT Season\_Year FROM Season ORDER BY Season\_Year DESC LIMIT 4**

**),**

**Striker\_Rate AS (**

**SELECT**

**B.Striker,**

**ROUND((SUM(B.Runs\_Scored) / NULLIF(COUNT(B.Ball\_Id), 0)) \* 100, 2) AS Strike\_Rate**

**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 Last\_4\_Seasons L4S ON S.Season\_Year = L4S.Season\_Year**

**GROUP BY B.Striker**

**HAVING COUNT(B.Ball\_Id) > 100**

**)**

**SELECT**

**RANK() OVER (ORDER BY SR.Strike\_Rate DESC) AS Ranking,**

**P.Player\_Name,**

**SR.Strike\_Rate**

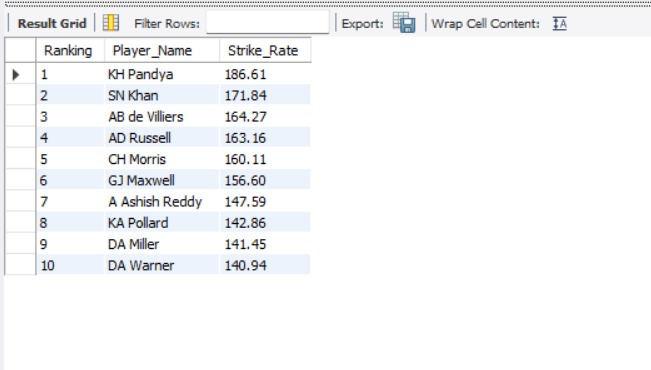
**FROM Striker\_Rate SR**

**JOIN Player P ON SR.Striker = P.Player\_Id**

**ORDER BY SR.Strike\_Rate DESC**

**LIMIT 10;**

* CTE Last\_4\_Seasons: Fetches the last 4 seasons based on the most recent seasons.
* CTE Striker\_Rate:
* Calculates the strike rate for each striker (batsman) in the last 4 seasons by dividing the total runs scored by the total balls faced, multiplied by 100.
* Filters to include only players who faced more than 100 balls.
* Main Query:
* RANK() is used to assign ranks to the players based on their strike rate in descending order.
* It joins the Player table to get player names and selects the top 10 players with the highest strike rate**s.**



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

* Code:

**SELECT**

**p.Player\_Name,**

**SUM(COALESCE(b.Runs\_Scored, 0)) AS Total\_Runs,**

**COUNT(DISTINCT CONCAT(b.Match\_Id, '-', b.Innings\_No)) AS Innings\_Played,**

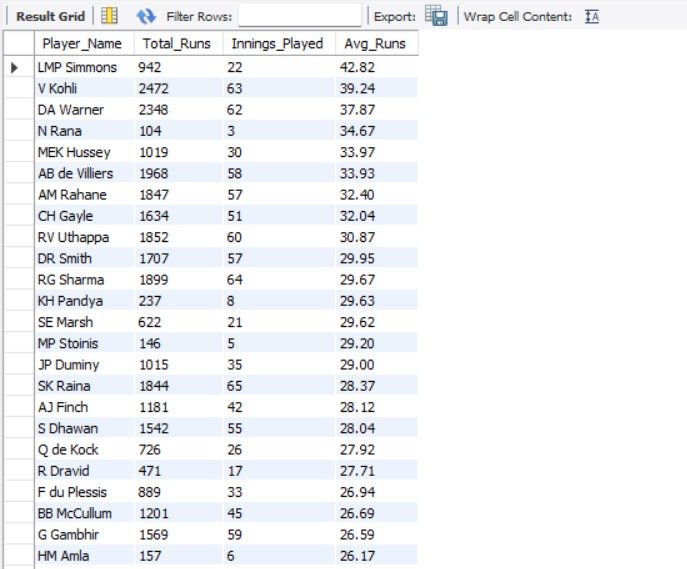
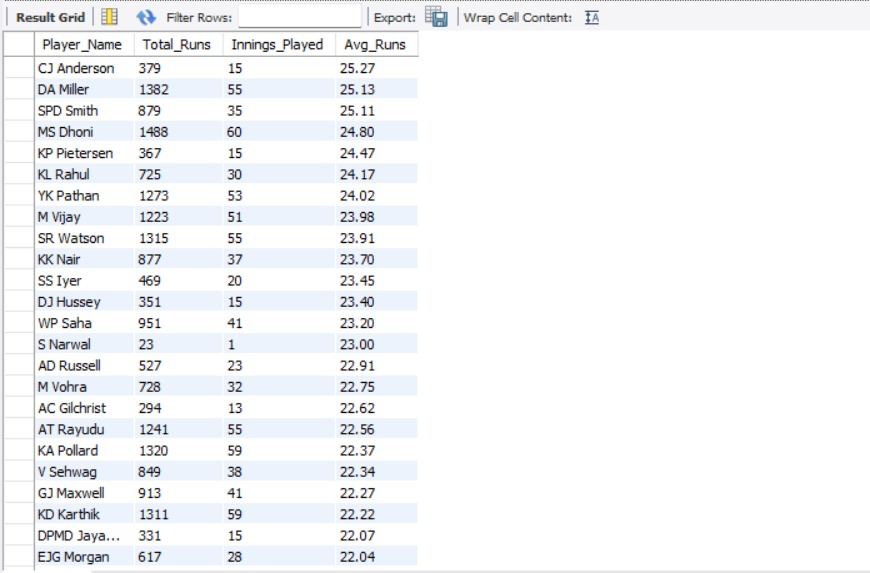
**ROUND(SUM(COALESCE(b.Runs\_Scored, 0)) / NULLIF(COUNT(DISTINCT CONCAT(b.Match\_Id, '-', b.Innings\_No)), 0), 2) AS Avg\_Runs**

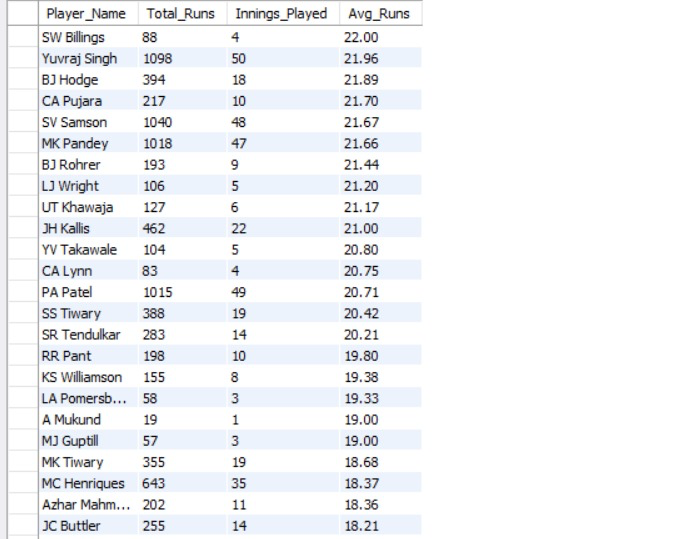
**FROM Ball\_by\_Ball b**

**JOIN Player p ON b.Striker = p.Player\_Id**

**GROUP BY p.Player\_Name**

**ORDER BY Avg\_Runs DESC;**

* The query is designed to calculate the average runs scored by each batsman across all seasons. It first sums up the total runs scored by each player, ensuring that any missing or NULL values in the runs are treated as zeros.
* It then calculates how many innings each player has played by counting the unique combinations of match ID and innings number.
* With these two pieces of information, the query computes the average runs per innings by dividing the total runs by the innings played.
* The final result is a list of players, ranked by their average runs, showing the most consistent batsman at the top.



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

* Code:

**WITH wickets\_count\_per\_player\_per\_season AS**

**(SELECT b.Bowler, m.Season\_Id, COUNT(w.Player\_Out) AS wickets\_taken**

**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 m.Match\_Id = w.Match\_Id**

**GROUP BY 1,2**

**ORDER BY b.Bowler ASC, m.Season\_Id ASC),**

**avg\_per\_season AS (**

**SELECT \*, AVG(wickets\_taken) OVER(PARTITION BY Bowler) AS avg\_wicket\_per\_bowler**

**FROM wickets\_count\_per\_player\_per\_season)**

**SELECT DISTINCT p.Player\_Name,ROUND(a.avg\_wicket\_per\_bowler,2) AS Avg\_wicket**

**FROM avg\_per\_season a**

**JOIN Player p**

**ON p.Player\_Id = a.Bowler**

**WHERE a.avg\_wicket\_per\_bowler > 0**

**ORDER BY Avg\_wicket DESC;**

* This query calculates the average wickets taken by each bowler over all seasons. First, it counts the number of wickets taken by each bowler in every match and season.
* The data is grouped by bowler and season, ensuring we know how many wickets each bowler took in each season. Then, using a window function (AVG), it calculates the average number of wickets each bowler took across all seasons they played.
* Finally, the query joins the results with the Player table to display the bowler's name and their average wickets taken, sorting the players by their average wickets in descending order.



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.**

* Code:

**WITH Player\_Avg\_Runs AS (**

**SELECT**

**bb.Striker AS Player\_Id,**

**p.Player\_Name,**

**COUNT(bb.Ball\_Id) AS Balls\_Faced,**

**SUM(bb.Runs\_Scored) AS Total\_Runs,**

**SUM(bb.Runs\_Scored) \* 1.0 / NULLIF(COUNT(bb.Ball\_Id), 0) AS Avg\_Runs**

**FROM Ball\_by\_Ball bb**

**JOIN Player p ON bb.Striker = p.Player\_Id**

**GROUP BY bb.Striker, p.Player\_Name**

**), Overall\_Avg AS (**

**SELECT SUM(Runs\_Scored) \* 1.0 / NULLIF(COUNT(Ball\_Id), 0) AS Overall\_Avg\_Runs**

**FROM Ball\_by\_Ball**

**)**

**SELECT p.Player\_Id, p.Player\_Name, p.Total\_Runs, p.Avg\_Runs**

**FROM Player\_Avg\_Runs p**

**JOIN Overall\_Avg oa**

**ON p.Avg\_Runs > oa.Overall\_Avg\_Runs**

**ORDER BY p.Avg\_Runs DESC;**

**WITH Player\_Wickets AS (**

**SELECT**

**wt.Player\_Out AS Player\_Id,**

**p.Player\_Name,**

**COUNT(\*) AS Total\_Wickets**

**FROM Wicket\_Taken wt**

**JOIN Player p ON wt.Player\_Out = p.Player\_Id**

**GROUP BY wt.Player\_Out, p.Player\_Name**

**), Overall\_Wickets AS (**

**SELECT AVG(Total\_Wickets) AS Overall\_Avg\_Wickets**

**FROM (**

**SELECT COUNT(\*) AS Total\_Wickets FROM Wicket\_Taken GROUP BY Player\_Out**

**) AS Wicket\_Data**

**)**

**SELECT pw.Player\_Id, pw.Player\_Name, pw.Total\_Wickets**

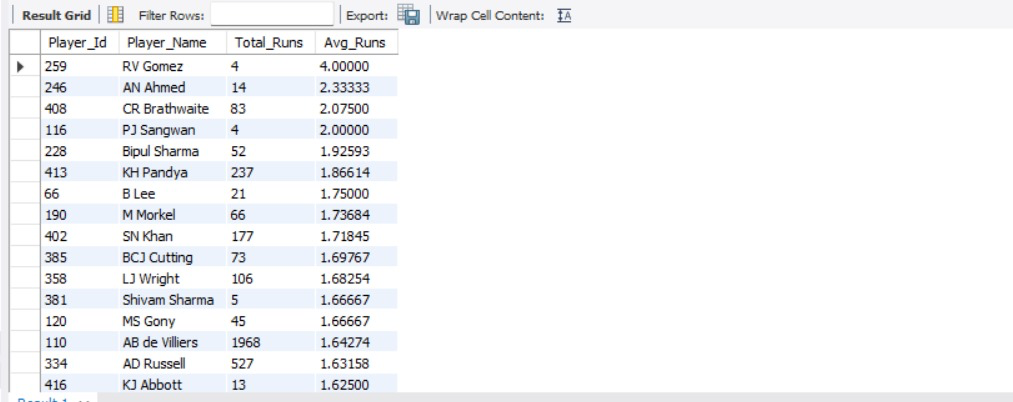
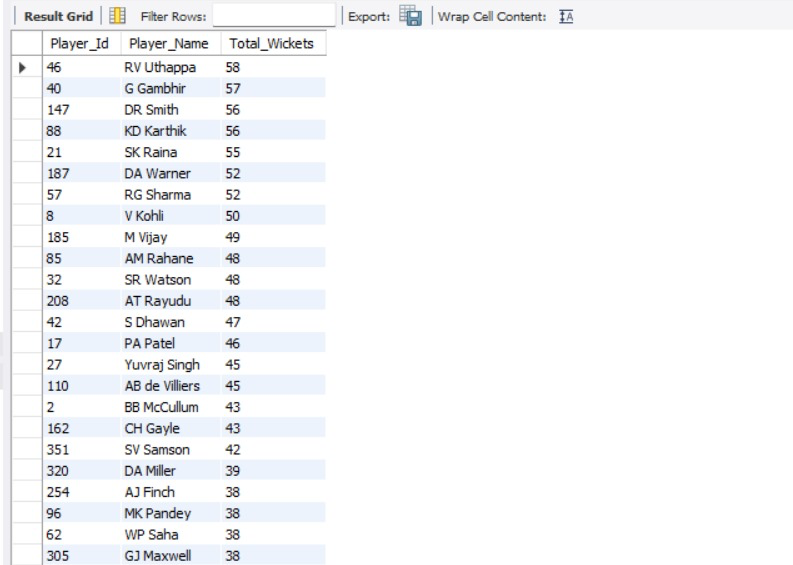
**FROM Player\_Wickets pw**

**JOIN Overall\_Wickets ow**

**ON pw.Total\_Wickets > ow.Overall\_Avg\_Wickets**

**ORDER BY pw.Total\_Wickets DESC;**

* This dataset represents the total wickets taken by players over the seasons.
* It helps to identify the key bowlers based on their total wicket count.



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

* Code:

**DROP TABLE IF EXISTS rcb\_record\_table;**

**CREATE TABLE IF NOT EXISTS rcb\_record\_table AS**

**WITH rcb\_record AS**

**(SELECT m.Venue\_Id, v.Venue\_Name,**

**SUM(CASE WHEN Match\_Winner = 2 THEN 1 ELSE 0 END) AS Win\_record,**

**SUM(CASE WHEN Match\_Winner != 2 THEN 1 ELSE 0 END) AS Loss\_record**

**FROM matches m**

**JOIN venue v**

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

**WHERE (Team\_1 = 2 OR Team\_2 = 2) AND m.Outcome\_type != 2**

**GROUP BY 1,2)**

**SELECT \*, Win\_record + Loss\_record AS Total\_Played,**

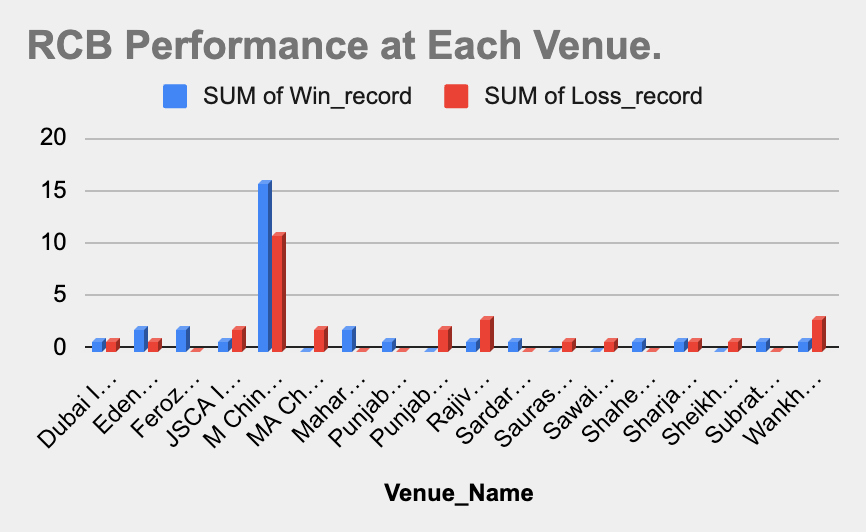
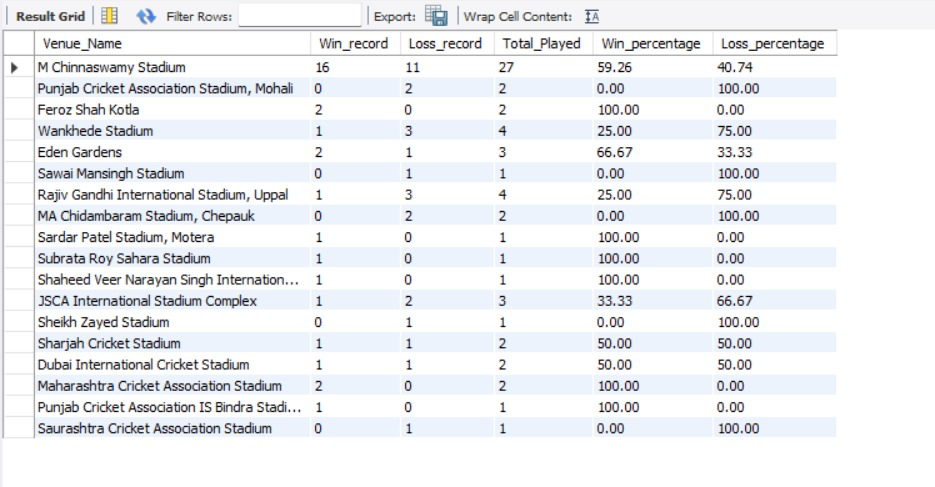
**ROUND((Win\_record/(Win\_record + Loss\_record))\*100,2) AS Win\_percentage, ROUND((Loss\_record/(Win\_record + Loss\_record))\*100,2) AS Loss\_percentage**

**FROM rcb\_record**

**ORDER BY Venue\_Id;**

**SELECT Venue\_Name,Win\_record,Loss\_record,Total\_Played,Win\_percentage,Loss\_percentage FROM rcb\_record\_table;**

* The query first removes any existing table to avoid duplication and then creates a new table to track RCB’s performance at different venues. It calculates the number of matches RCB won and lost at each venue and ensures that ties are excluded.
* Once the total matches are counted, the query computes win percentage and loss percentage for each venue. The final table displays venue name, wins, losses, total matches played, and win/loss percentages, ordered by venue.



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

* Code:

**WITH no\_of\_wicket\_per\_bowler AS (**

**SELECT bb.bowler, COUNT(w.Player\_Out) AS no\_of\_wickets**

**FROM wicket\_taken w**

**JOIN ball\_by\_ball bb**

**ON w.Match\_Id = bb.Match\_Id**

**AND w.Over\_Id = bb.Over\_Id**

**AND w.Ball\_Id = bb.Ball\_Id**

**AND w.Innings\_No = bb.Innings\_No**

**GROUP BY bb.Bowler),**

**bowler\_skill\_wicket AS**

**(SELECT n.bowler, st.Bowling\_skill, no\_of\_wickets**

**FROM no\_of\_wicket\_per\_bowler n**

**JOIN player p**

**ON n.bowler = p.Player\_Id**

**JOIN bowling\_style st**

**ON st.Bowling\_Id = p.Bowling\_skill**

**ORDER BY no\_of\_wickets DESC)**

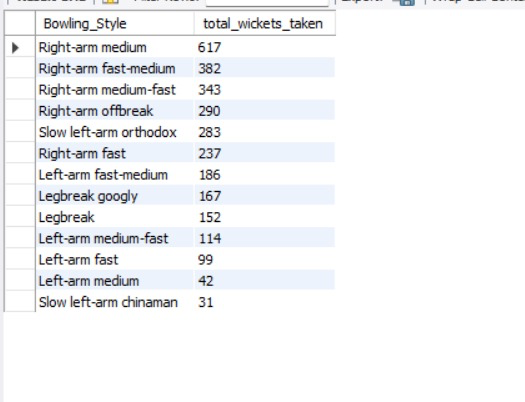
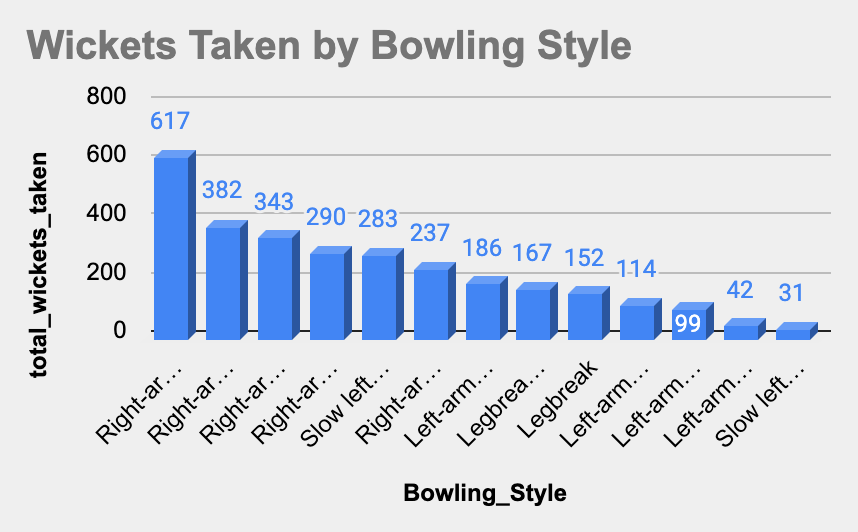
**SELECT Bowling\_skill AS Bowling\_Style, SUM(no\_of\_wickets) AS total\_wickets\_taken**

**FROM bowler\_skill\_wicket**

**GROUP BY Bowling\_skill**

**ORDER BY total\_wickets\_taken DESC;**

* The query starts by calculating the total number of wickets taken by each bowler from the wicket\_taken and ball\_by\_ball tables. The no\_of\_wicket\_per\_bowler CTE counts how many wickets each bowler took in each match.
* Next, the bowling style is linked to each bowler by joining the player and bowling\_style tables. The bowler\_skill\_wicket CTE brings together the number of wickets taken by each bowler and their bowling style.
* Finally, the query groups the data by bowling style and calculates the total wickets taken for each bowling style, ordering the results by the number of wickets in descending order.



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.**

* Code:

**WITH total\_run\_match\_id AS (**

**-- Total runs per innings per match**

**SELECT**

**Match\_Id,**

**Innings\_No,**

**SUM(Runs\_Scored) AS total\_runs**

**FROM Ball\_by\_Ball**

**GROUP BY Match\_Id, Innings\_No**

**),**

**total\_runs\_per\_season AS (**

**-- Total runs per season for Team\_Id = 2**

**SELECT**

**m.Season\_Id,**

**SUM(CASE**

**WHEN m.Toss\_Decide = 1 AND m.Toss\_Winner = 2 AND t.Innings\_No = 2 THEN t.total\_runs**

**WHEN m.Toss\_Decide = 2 AND m.Toss\_Winner = 2 AND t.Innings\_No = 1 THEN t.total\_runs**

**WHEN m.Toss\_Decide = 1 AND m.Toss\_Winner != 2 AND t.Innings\_No = 1 THEN t.total\_runs**

**WHEN m.Toss\_Decide = 2 AND m.Toss\_Winner != 2 AND t.Innings\_No = 2 THEN t.total\_runs**

**ELSE 0**

**END) AS total\_runs**

**FROM total\_run\_match\_id t**

**JOIN Matches m ON t.Match\_Id = m.Match\_Id**

**WHERE m.Team\_1 = 2 OR m.Team\_2 = 2**

**GROUP BY m.Season\_Id**

**),**

**total\_wickets\_per\_match\_innings AS (**

**-- Total wickets per match per innings**

**SELECT**

**w.Match\_Id,**

**w.Innings\_No,**

**COUNT(w.Player\_Out) AS total\_wickets**

**FROM Wicket\_Taken w**

**JOIN Matches m ON m.Match\_Id = w.Match\_Id**

**WHERE m.Team\_1 = 2 OR m.Team\_2 = 2**

**GROUP BY w.Match\_Id, w.Innings\_No**

**),**

**total\_wickets\_per\_season AS (**

**-- Total wickets per season for Team\_Id = 2**

**SELECT**

**m.Season\_Id,**

**SUM(CASE**

**WHEN m.Toss\_Decide = 1 AND m.Toss\_Winner = 2 AND w.Innings\_No = 1 THEN w.total\_wickets**

**WHEN m.Toss\_Decide = 2 AND m.Toss\_Winner = 2 AND w.Innings\_No = 2 THEN w.total\_wickets**

**WHEN m.Toss\_Decide = 1 AND m.Toss\_Winner != 2 AND w.Innings\_No = 2 THEN w.total\_wickets**

**WHEN m.Toss\_Decide = 2 AND m.Toss\_Winner != 2 AND w.Innings\_No = 1 THEN w.total\_wickets**

**ELSE 0**

**END) AS total\_wickets**

**FROM total\_wickets\_per\_match\_innings w**

**JOIN Matches m ON m.Match\_Id = w.Match\_Id**

**GROUP BY m.Season\_Id**

**)**

**SELECT**

**s.Season\_Id,**

**COALESCE(r.total\_runs, 0) AS total\_runs,**

**COALESCE(w.total\_wickets, 0) AS total\_wickets**

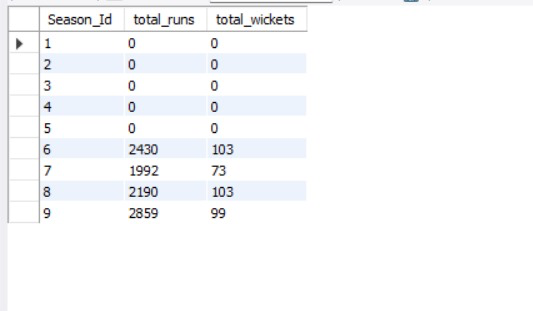
**FROM Season s**

**LEFT JOIN total\_runs\_per\_season r ON s.Season\_Id = r.Season\_Id**

**LEFT JOIN total\_wickets\_per\_season w ON s.Season\_Id = w.Season\_Id**

**ORDER BY s.Season\_Id;**

* The query starts by calculating the total number of wickets taken by each bowler for each match. First gathered the number of wickets taken per bowler for each match by combining data from the wicket\_taken and ball\_by\_ball tables.
* Next, the query associates each bowler with their bowling style by joining the player and bowling\_style tables. Then arrange the data by the total number of wickets taken by each bowler, making sure that to get a clear ranking of bowlers.
* Finally, the query groups the data by bowling style and sums up the total wickets taken for each bowling style. The result is a list showing which bowling styles have contributed the most to taking wickets, with styles that yield more wickets ranked higher.



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

* Code:

**-- KPI #1 Boundary %**

**SELECT pm.Player\_Id, p.Player\_Name,**

**ROUND((SUM(CASE WHEN b.Runs\_Scored = 4 THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 2) AS Four\_Percentage,**

**ROUND((SUM(CASE WHEN b.Runs\_Scored = 6 THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 2) AS Six\_Percentage**

**FROM Ball\_by\_Ball b**

**JOIN Matches m**

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

**JOIN Player\_Match pm**

**ON m.Match\_Id = pm.Match\_Id AND pm.Player\_Id = b.Striker**

**JOIN Player p**

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

**WHERE m.Season\_Id IN (SELECT DISTINCT Season\_Id FROM Matches WHERE Team\_1 = 2 OR Team\_2 = 2)**

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

**ORDER BY Six\_Percentage DESC, Four\_Percentage DESC**

**LIMIT 15;**

**-- KPI #2 Bowling strike rate (Lower is better)**

**SELECT bb.Bowler, p.Player\_Name,**

**ROUND(COUNT(bb.Ball\_Id) / COUNT(w.Player\_Out),2) AS Strike\_Rate**

**FROM ball\_by\_ball bb**

**LEFT JOIN wicket\_taken w**

**ON bb.Match\_Id = w.Match\_Id**

**AND bb.Over\_Id = w.Over\_Id**

**AND bb.Ball\_Id = w.Ball\_Id**

**JOIN player p**

**ON p.Player\_Id = bb.Bowler**

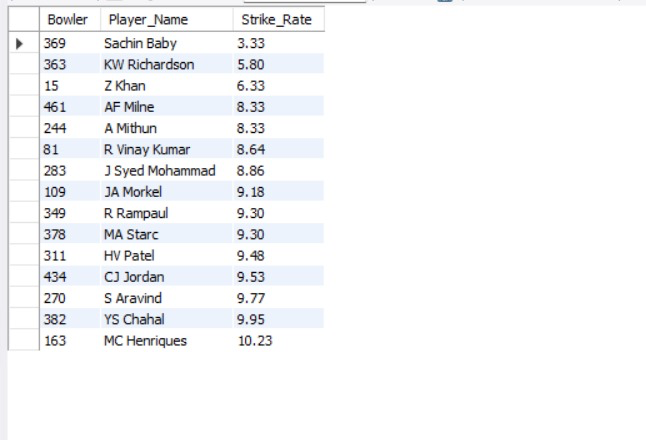
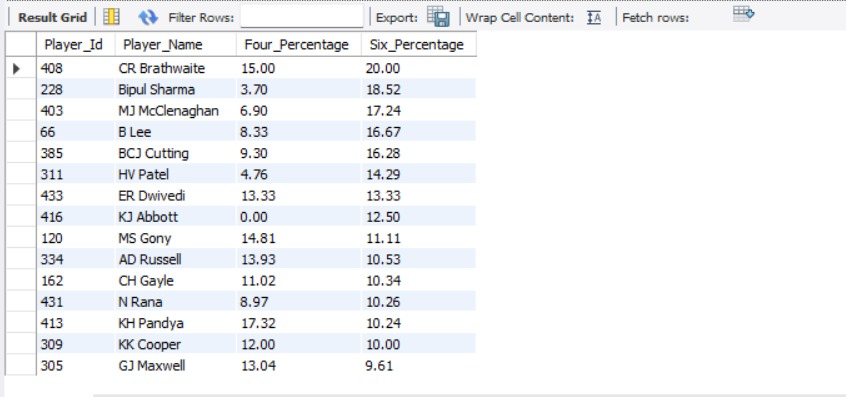
**WHERE bb.Team\_Bowling = 2**

**GROUP BY bb.Bowler**

**HAVING Strike\_Rate IS NOT NULL**

**ORDER BY Strike\_Rate ASC**

**LIMIT 15;**

* In this query, we focus on deriving two key performance indicators (KPIs) that can provide valuable insights into RCB’s (Team 2) batting and bowling performance to guide their strategy.
* **KPI #1: Boundary Percentage:** The first part of the query calculates how often RCB’s players hit boundaries (4s and 6s) during a match. This is done by checking each ball faced by the players and counting the number of times a 4 or 6 was scored. The results are then presented as percentages of total balls faced to determine the boundary percentage. The query ranks the players based on the percentage of 6s hit, followed by 4s, helping us understand which players are aggressive boundary hitters.
* **KPI #2: Bowling Strike Rate:**  
   The second KPI focuses on the strike rate of bowlers, which is an important measure of how often a bowler takes a wicket. The query calculates the strike rate for each bowler, defined as the number of balls bowled per wicket taken. A lower strike rate is preferable, as it means a bowler takes wickets more frequently. The results show the top bowlers based on the best (lowest) strike rate, helping the team assess which bowlers are most effective at taking wickets.

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.**

* Code:

**WITH player\_wickets AS (**

**SELECT v.Venue\_Id, v.Venue\_Name,**

**p.Player\_Name,**

**COUNT(w.Player\_Out) AS total\_wickets,**

**COUNT(DISTINCT m.Match\_Id) AS matches\_played -- Distinct matches where the player bowled**

**FROM Wicket\_Taken w**

**JOIN Ball\_by\_Ball b**

**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 -- Ensuring correct innings mapping**

**JOIN Matches m**

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

**JOIN Player\_Match pm**

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

**AND pm.Player\_Id = b.Bowler -- Ensuring only actual bowlers are counted**

**JOIN Player p**

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

**JOIN Venue v**

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

**GROUP BY v.Venue\_Id, v.Venue\_Name, p.Player\_Name**

**),**

**unranked\_table AS (**

**SELECT Venue\_Id, Venue\_Name, Player\_Name,**

**total\_wickets,**

**matches\_played,**

**ROUND(total\_wickets / matches\_played, 2) AS avg\_wickets**

**FROM player\_wickets**

**)**

**SELECT \*, DENSE\_RANK() OVER(ORDER BY avg\_wickets DESC) AS Ranking**

**FROM unranked\_table**

**WHERE matches\_played > 10;**

* This query aims to determine which bowlers perform best at different venues based on their average wickets per match.

1. Counting Wickets at Each Venue:

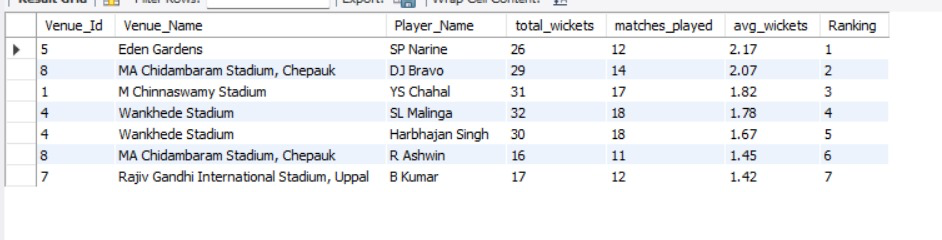
* The query starts by counting the total wickets taken by each bowler at every venue.
* It ensures that only actual bowlers who played in the match are included.
* The number of distinct matches where the player bowled is also counted to calculate averages later.

2. Calculating Average Wickets per Match:

* It then calculates the average wickets per match for each bowler at a specific venue by dividing total wickets by matches played.
* This helps identify which bowlers are the most effective wicket-takers at different venues

3. Ranking Bowlers by Average Wickets:

* The query assigns a rank to each bowler based on their average wickets per match, using DENSE\_RANK() so that bowlers with the same average wickets share the same rank.
* Only bowlers who have bowled in more than 10 matches are considered to ensure a fair ranking



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

* Code:

**#Bowling performance**

**WITH Player\_Season\_Performance AS (**

**SELECT**

**p.Player\_Name,**

**s.Season\_Year,**

**SUM(bbb.Runs\_Scored) AS Total\_Runs,**

**COUNT(wt.Player\_Out) AS Total\_Wickets,**

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

**FROM Player p**

**INNER JOIN Ball\_by\_Ball bbb ON p.Player\_Id = bbb.Striker**

**LEFT JOIN Wicket\_Taken wt ON bbb.Match\_Id = wt.Match\_Id**

**AND bbb.Over\_Id = wt.Over\_Id**

**AND bbb.Ball\_Id = wt.Ball\_Id**

**AND bbb.Innings\_No = wt.Innings\_No**

**INNER JOIN Matches m ON bbb.Match\_Id = m.Match\_Id**

**INNER JOIN Season s ON m.Season\_Id = s.Season\_Id**

**WHERE p.Player\_Id = bbb.Bowler OR p.Player\_Id = bbb.Striker**

**GROUP BY p.Player\_Name, s.Season\_Year**

**)**

**SELECT**

**Player\_Name,**

**AVG(Total\_Runs) AS Avg\_Runs\_Per\_Season,**

**AVG(Total\_Wickets) AS Avg\_Wickets\_Per\_Season,**

**COUNT(Season\_Year) AS Seasons\_Played**

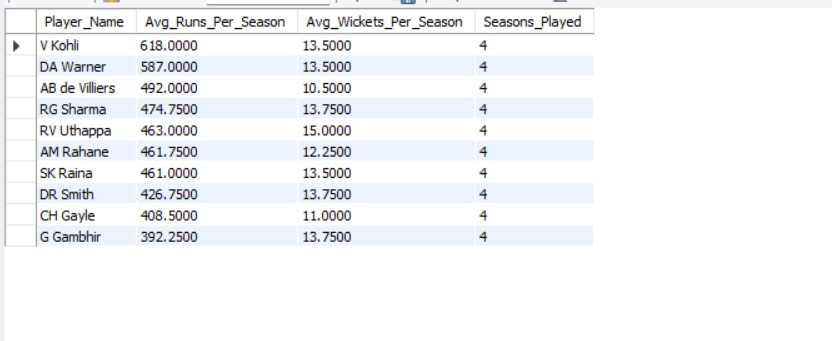
**FROM Player\_Season\_Performance**

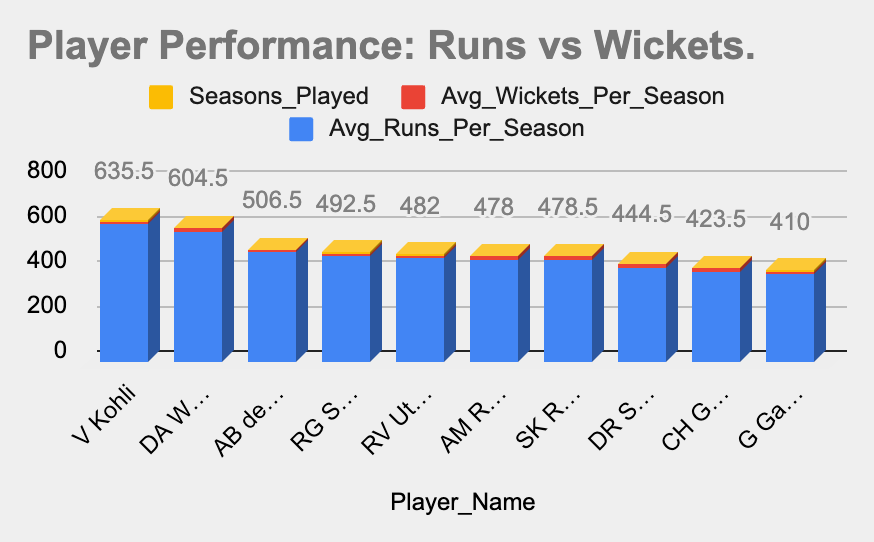
**GROUP BY Player\_Name**

**HAVING Seasons\_Played > 3**

**ORDER BY Avg\_Runs\_Per\_Season DESC, Avg\_Wickets\_Per\_Season DESC**

**LIMIT 10;**

* The query aims to identify players who have shown consistent performance over multiple seasons, both in terms of batting and bowling. It first calculates each player’s total runs and total wickets for each season, while also counting how many matches they played. After calculating these statistics for each season, it finds the average runs and average wickets across the seasons.
* The focus is on players who have played more than 3 seasons, ensuring the consistency of their performance. Finally, the query sorts these players based on their average performance and returns the top 10 players who have consistently excelled in both runs and wickets.
* These players stand out as key performers who have been reliable over time, helping the team in both batting and bowling.



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

* Code:

**#Batting performance**

**SELECT p.Player\_Name, v.Venue\_Name,**

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

**COUNT(b.Ball\_Id) AS Balls\_Faced,**

**ROUND(SUM(b.Runs\_Scored) / COUNT(b.Ball\_Id), 2) \* 100 AS Strike\_Rate**

**FROM Ball\_by\_Ball b**

**JOIN Matches m ON m.Match\_Id = b.Match\_Id**

**JOIN Player p ON p.Player\_Id = b.Striker**

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

**GROUP BY p.Player\_Name, v.Venue\_Name**

**HAVING Total\_Runs > 0 AND Balls\_Faced > 100**

**ORDER BY Total\_Runs DESC, p.Player\_Name;**

**#Bowling performance**

**SELECT p.Player\_Name, v.Venue\_Name,**

**COUNT(w.Player\_Out) AS Wickets\_Taken,**

**COUNT(b.Ball\_Id) AS Balls\_Bowled**

**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 m.Match\_Id = w.Match\_Id**

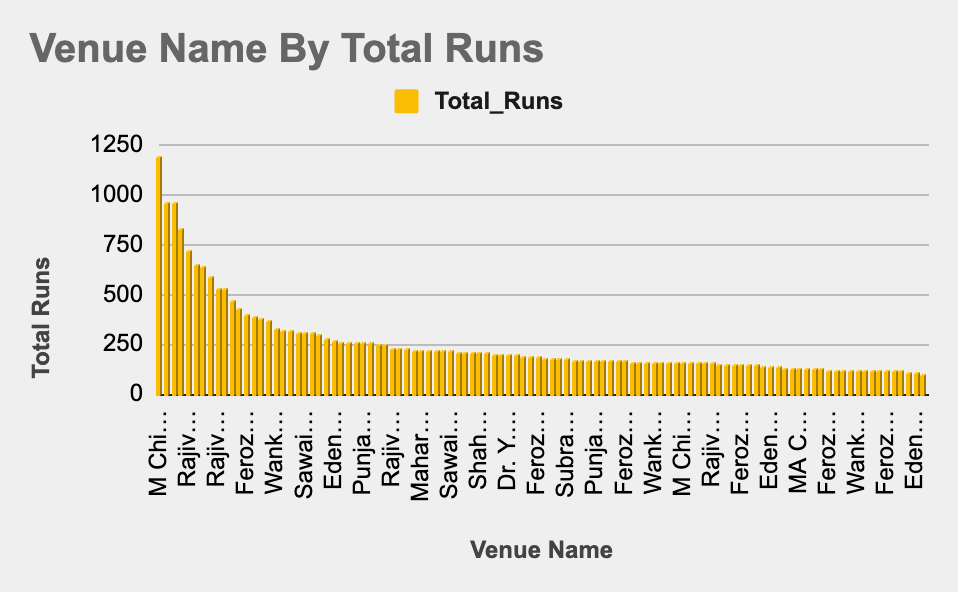
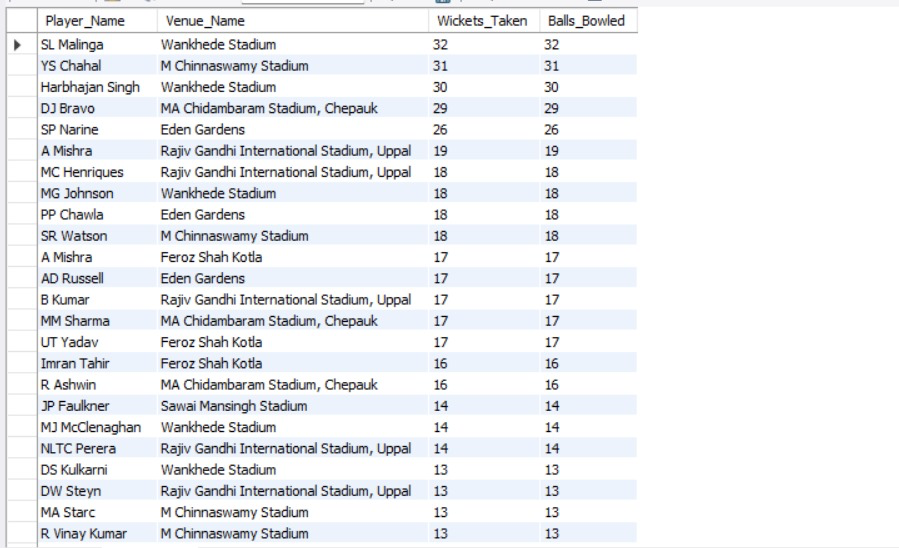
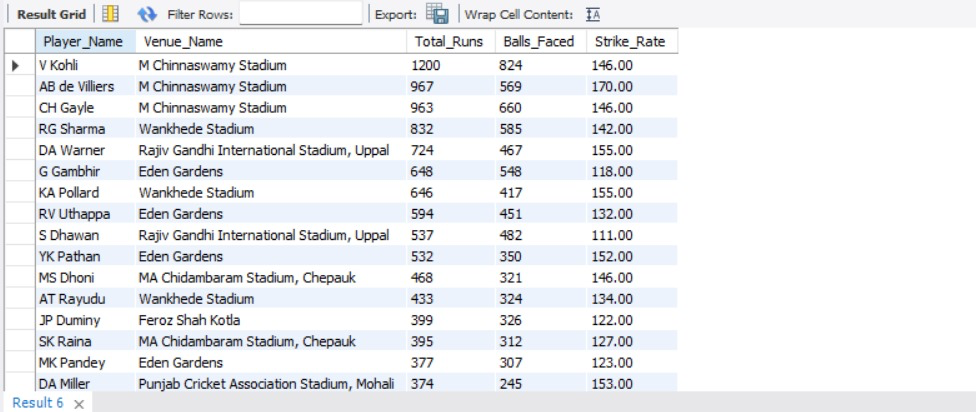
**JOIN player p ON p.Player\_Id = b.Bowler**

**JOIN venue v ON m.Venue\_Id = v.Venue\_Id**

**GROUP BY p.Player\_Name, v.Venue\_Name**

**HAVING Balls\_Bowled > 5**

**ORDER BY Wickets\_Taken DESC,p.Player\_Name;**



**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?**

* Code:

**SELECT v.Venue\_Id, v.Venue\_Name,**

**CASE WHEN m.Toss\_Decide = 1 THEN 'Field' ELSE 'Bat' END AS Toss\_Decide,**

**COUNT(\*) AS Total\_Matches,**

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

**SUM(CASE WHEN m.Toss\_Winner != m.Match\_Winner THEN 1 ELSE 0 END) AS Toss\_Winner\_Losses,**

**ROUND((SUM(CASE WHEN m.Toss\_Winner = m.Match\_Winner THEN 1 ELSE 0 END) \* 100.0) / COUNT(\*), 2) AS Win\_Percentage**

**FROM Matches m**

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

**GROUP BY v.Venue\_Id, v.Venue\_Name, m.Toss\_Decide**

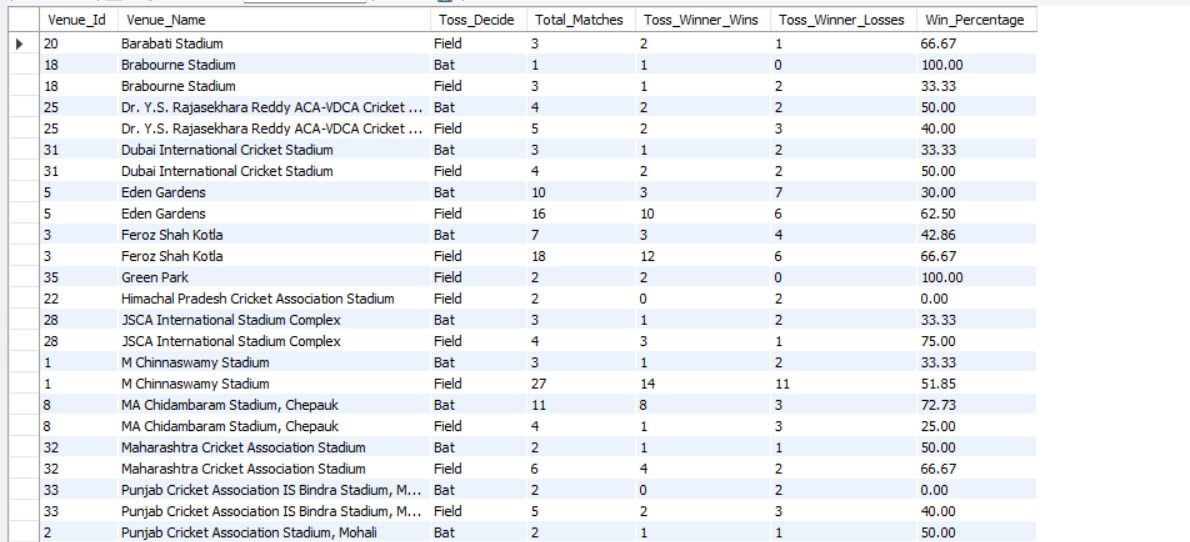
**ORDER BY v.Venue\_Name, Toss\_Decide;**

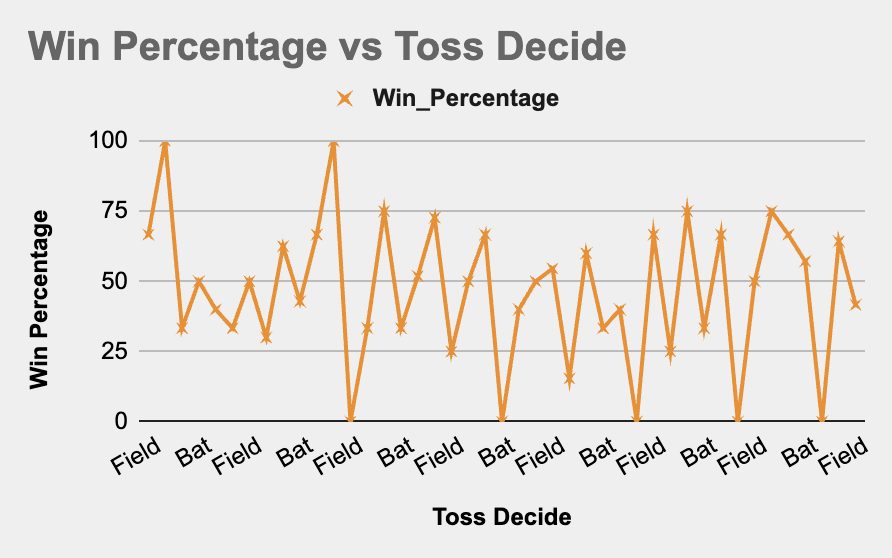
* This query examines how the toss decision (whether a team chooses to bat or field) impacts the outcome of the match. It looks at each venue and how the toss influences the match results. The query breaks down the data by venue and toss decision (whether the team decided to bat or field). It calculates the total number of matches, toss winner wins, and toss winner losses. The win percentage of the team that won the toss is also calculated, giving us an insight into how often the team winning the toss ends up winning the match.
* By analyzing this for different venues, the toss decision has a stronger impact at certain venues. This could indicate that certain conditions (like pitch conditions, weather, etc.) at particular venues make the toss decision more critical to the outcome.
* **Recommendations**:

1. **Strategic Focus on Toss Decision**:

* Teams should focus on making better decisions after winning the toss, especially at venues where the decision to bat or field has a high impact on the result (e.g., Brabourne Stadium).

1. **Monitor Toss Influence**:

* Analyze weather conditions or pitch characteristics at different venues to refine the impact of toss decisions.

****

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

* Code:

**#List of consistently performing batsmen**

**SELECT p.Player\_Name,**

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

**COUNT(b.Ball\_Id) AS Balls\_Faced,**

**ROUND((SUM(b.Runs\_Scored) / COUNT(b.Ball\_Id)) \* 100, 2) AS Strike\_Rate,**

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

**FROM Player p**

**JOIN Ball\_by\_Ball b ON p.Player\_Id = b.Striker**

**JOIN Matches m ON b.Match\_Id = m.Match\_Id**

**WHERE m.Season\_Id >= 4**

**GROUP BY p.Player\_Name**

**ORDER BY Total\_Runs DESC, Strike\_Rate DESC**

**LIMIT 10;**

**#List of consistent bowlers**

**SELECT p.Player\_Name,**

**COUNT(w.Player\_Out) AS Wickets\_Taken,**

**ROUND(SUM(bb.Ball\_Id) / COUNT(w.Player\_Out),2) AS Strike\_Rate,**

**ROUND(SUM(bb.Runs\_Scored) / (SUM(bb.Ball\_Id)/6),2) AS Economy\_Rate**

**FROM Player p**

**JOIN Ball\_by\_Ball bb ON p.Player\_Id = bb.Bowler**

**JOIN Matches m ON bb.Match\_Id = m.Match\_Id**

**JOIN Wicket\_Taken w**

**ON bb.Match\_Id = w.Match\_Id AND bb.Over\_Id = w.Over\_Id AND bb.Innings\_No = w.Innings\_No AND bb.Ball\_Id = w.Ball\_Id**

**WHERE m.Season\_Id >= 4**

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

**ORDER BY Wickets\_Taken DESC, Economy\_Rate ASC, Strike\_Rate ASC**

**LIMIT 10;**

* This query identifies the best-performing batsmen and bowlers based on their consistent performances over multiple seasons.

1. **Identifying the Best Batsmen:**

* The first query analyzes players who have scored the most runs while maintaining a high strike rate and strong batting average.
* It filters for matches from Season 4 onwards, ensuring that only recent and consistent performers are considered.
* The results list the top 10 batsmen who have scored the most runs, with priority given to those who have a better strike rate

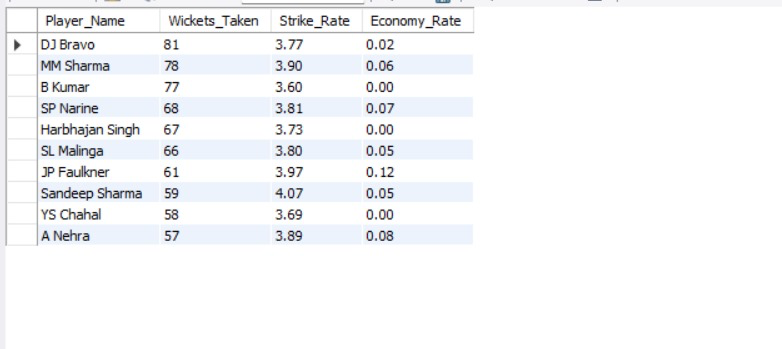
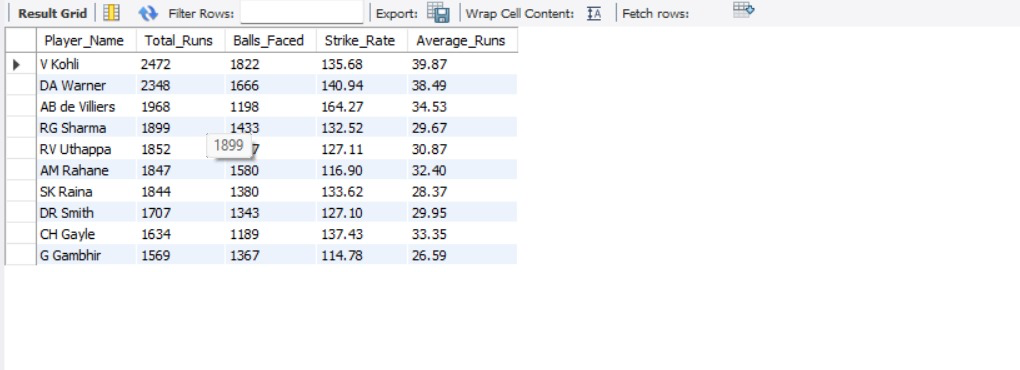
1. **Identifying the Best Bowlers:**

* The second query focuses on bowlers who have taken the most wickets while maintaining a good economy rate and low strike rate.
* A low strike rate means the bowler takes wickets frequently, and a low economy rate means they concede fewer runs.
* The top 10 bowlers are selected based on their wickets taken, economy rate, and strike rate to find those who can consistently deliver match-winning performances.
* **Recommendations**:

1. **Target Efficient Bowlers**:

* Players like DJ Bravo and MM Sharma should be prioritized for recruitment, as they offer a good balance of wickets taken, strike rate, and economy rate.
* Bowlers who excel in both wickets taken and economy rate (such as B Kumar and DJ Bravo) should be valued for their consistency.

1. **Bowling Strategy**:

* For better performance, the team should consider having bowlers with both low economy rates and quick wicket-taking abilities, ensuring they can contain the opposition while also picking up wickets.

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

* Code:

**#Key parameters for selecting players**

**# A. Death over bowling performance**

**SELECT p.Player\_Name,**

**SUM(CASE WHEN bb.Over\_Id >= 16 AND bb.Over\_Id <= 20 AND p.Player\_Id IN (SELECT Bowler FROM ball\_by\_ball) THEN bb.Runs\_Scored ELSE 0 END) AS Death\_Over\_Runs\_Conceded**

**FROM Player p**

**JOIN ball\_by\_ball bb ON p.Player\_Id = bb.Striker OR p.Player\_Id = bb.Bowler**

**JOIN Matches m ON bb.Match\_Id = m.Match\_Id**

**WHERE m.Season\_Id >= 4**

**GROUP BY p.Player\_Name**

**HAVING COUNT(bb.Ball\_Id) > 100 AND Death\_Over\_Runs\_Conceded != 0**

**ORDER BY Death\_Over\_Runs\_Conceded ASC**

**LIMIT 10;**

**# B. Batting performance across different venues**

**SELECT p.Player\_Name,**

**v.Venue\_Id, v.Venue\_Name,**

**SUM(bb.Runs\_Scored) AS Total\_Runs,**

**COUNT(bb.Ball\_Id) AS Balls\_Faced,**

**ROUND(SUM(bb.Runs\_Scored) / COUNT(bb.Ball\_Id), 2) \* 100 AS Strike\_Rate**

**FROM Player p**

**JOIN Ball\_by\_Ball bb ON p.Player\_Id = bb.Striker**

**JOIN Matches m ON bb.Match\_Id = m.Match\_Id**

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

**JOIN Ball\_by\_Ball bb2**

**ON bb.Match\_Id = bb2.Match\_Id**

**AND bb.Over\_Id = bb2.Over\_Id**

**AND bb.Ball\_Id = bb2.Ball\_Id**

**AND bb.Innings\_No = bb2.Innings\_No**

**GROUP BY p.Player\_Name, v.Venue\_Id, v.Venue\_Name**

**ORDER BY Total\_Runs DESC, Strike\_Rate DESC**

**LIMIT 10;**

* When selecting players for the team, there are key parameters to focus on that reflect both batting and bowling performances under different circumstances:

1. **Death Over Bowling Performance:**

* The first parameter focuses on death-over bowling, which is critical in the final overs of a match. This performance is important because a bowler's ability to restrict runs during the final overs (overs 16 to 20) can make or break a game.
* The query calculates the runs conceded by bowlers during death overs. The best bowlers are those who have a low number of runs conceded in these crucial overs, signaling their ability to restrict scoring in high-pressure situations.
* It filters out players who have bowled less than 100 balls and only includes those who have conceded runs in death overs, providing a focus on specialist death-over bowlers.

1. **Batting Performance Across Different Venues:**

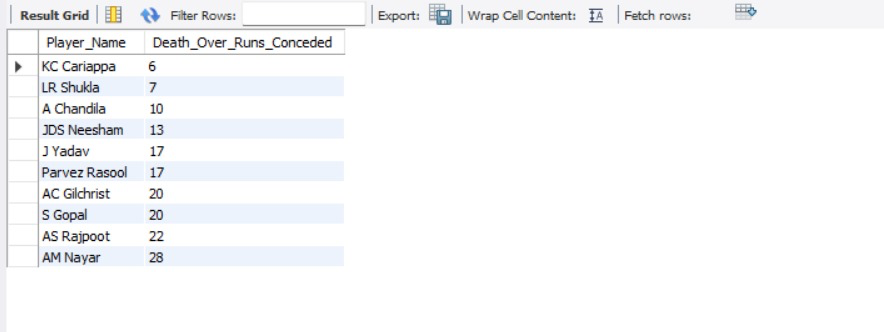
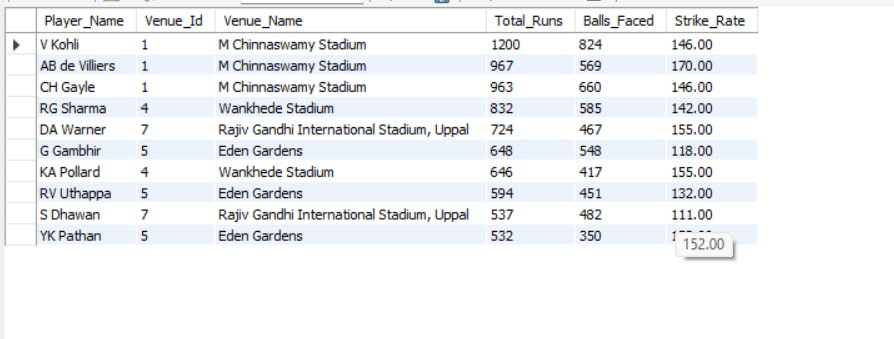
* The second key parameter focuses on a player's batting consistency across different venues. A player's strike rate and total runs at different venues offer insights into how adaptable they are in varying conditions (e.g., different pitch types and weather conditions).
* The query calculates total runs and strike rate for each player at each venue and ranks players who have performed the best. A high strike rate coupled with a high total runs shows a player's ability to dominate in different conditions, making them versatile and valuable in various match scenarios.
* **Recommendations**:

1. **Prioritize High Strike Rate Players**:

* Select players with high strike rates (like AB de Villiers and DA Warner), as they provide the ability to score quickly and change the course of the game.

1. **Consistency in Runs**:

* Focus on players who consistently score high total runs, like V Kohli, as their consistent performance over multiple seasons adds stability to the batting lineup.



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)**

* Code:

**#We can find all-rounder performance for all players**

**WITH batting\_performance AS (**

**SELECT p.Player\_Id, p.Player\_Name,**

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

**COUNT(bb.Ball\_Id) AS Balls\_Faced,**

**ROUND((SUM(b.Runs\_Scored) / COUNT(bb.Ball\_Id)) \* 100, 2) AS Batting\_Strike\_Rate**

**FROM Player p**

**JOIN Ball\_by\_Ball bb ON p.Player\_Id = bb.Striker**

**JOIN Ball\_by\_Ball b**

**ON bb.Match\_Id = b.Match\_Id**

**AND bb.Over\_Id = b.Over\_Id**

**AND bb.Ball\_Id = b.Ball\_Id**

**AND bb.Innings\_No = b.Innings\_No**

**WHERE b.Runs\_Scored IS NOT NULL -- Ensuring only valid scoring deliveries are considered**

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

**),**

**bowling\_performance AS (**

**SELECT p.Player\_Id, p.Player\_Name,**

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

**ROUND(SUM(bb.Team\_Batting) / COUNT(bb.Ball\_Id),2) AS Economy\_Rate**

**FROM player p**

**JOIN ball\_by\_ball bb ON p.Player\_Id = bb.Bowler**

**JOIN wicket\_taken w ON bb.Match\_Id = w.Match\_Id**

**AND bb.Over\_Id = w.Over\_Id**

**AND bb.Ball\_Id = w.Ball\_Id**

**AND bb.Innings\_No = w.Innings\_No**

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

**)**

**SELECT bp.Player\_Id, bp.Player\_Name,**

**bp.Total\_Runs, bp.Batting\_Strike\_Rate, bp.Balls\_Faced,**

**bw.Total\_Wickets, bw.Economy\_Rate**

**FROM batting\_performance bp**

**JOIN bowling\_performance bw ON bp.Player\_Id = bw.Player\_Id**

**ORDER BY bp.Batting\_Strike\_Rate DESC, bw.Economy\_Rate ASC**

**LIMIT 10;**

* This query aims to identify the all-rounders who perform well both with the bat and the ball, contributing to the team in multiple ways. The focus is on players who can dominate with the bat while also restricting runs and taking wickets with the ball.

1. **Batting Performance**:

* The first part of the query calculates batting performance by summarizing each player's total runs and strike rate (how quickly they score runs). It also includes the number of balls faced to give context to the strike rate.
* Players with high strike rates and total runs are generally more aggressive batsmen, helping the team build or accelerate the innings.

1. **Bowling Performance**:

* The second part calculates bowling performance by counting total wickets taken by the player and calculating their economy rate (how many runs they concede per over). A low economy rate is desirable, as it shows the bowler is effective in restricting runs while taking wickets.
* Bowlers who consistently take wickets and maintain a low economy rate are vital for restricting the opposition's scoring.

1. **Combining Both Performances**:

* The final query joins the batting performance and bowling performance to identify all-rounders. The players are ranked first by batting strike rate (to highlight aggressive batsmen) and then by economy rate (to prioritize efficient bowlers).
* **Recommendations**:

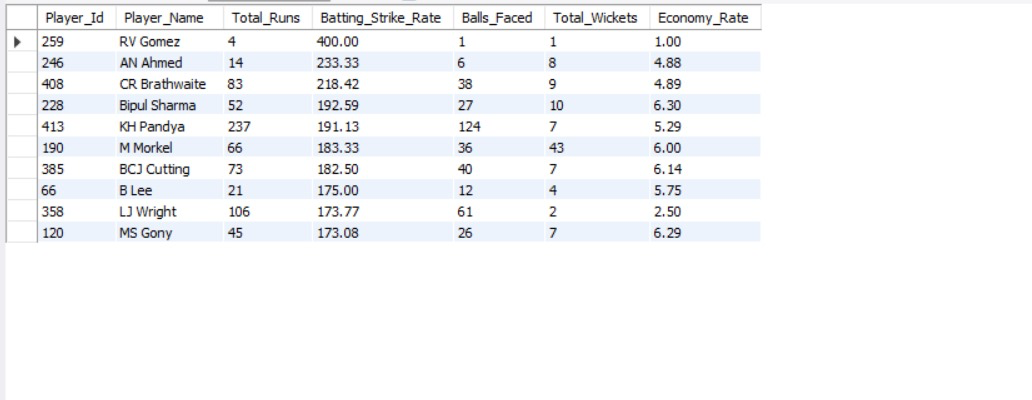
1. **Prioritize All-Rounders**:

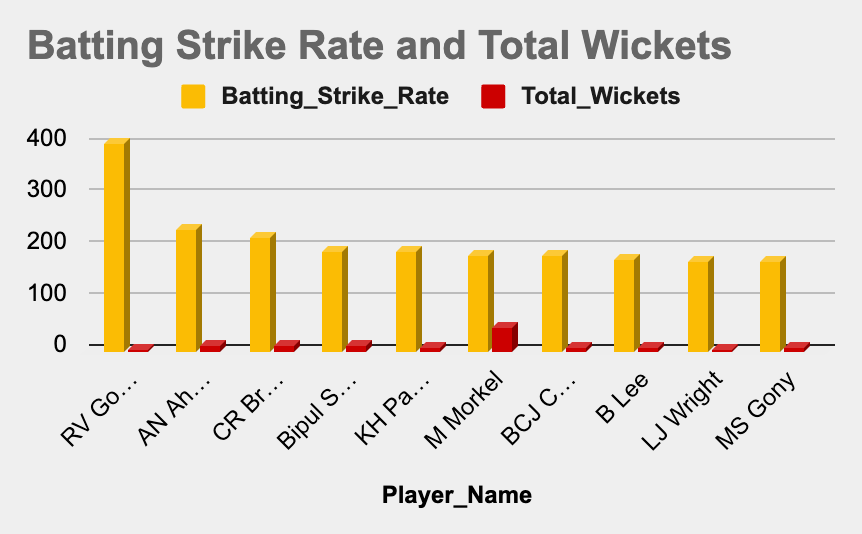
* KH Pandya, CR Brathwaite, and AN Ahmed should be prioritized as they provide strong batting and bowling contributions, making them versatile assets for the team.

1. **Balance Batting and Bowling**:

* Players like Bipul Sharma and BCJ Cutting offer balance, contributing significantly in both batting and bowling departments. These players should be considered for a well-rounded team.

1. **Consider Bowler Efficiency**:

* Ensure that players with good economy rates (like CR Brathwaite and KH Pandya) are selected, as they restrict runs while taking wickets.

****

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

* Code:

**WITH cte AS (**

**-- Extract relevant match details for the 2015 and 2016 seasons**

**SELECT bbb.Striker, m.Season\_Id, s.Season\_Year,**

**bbb.Match\_Id, bbb.Over\_Id, bbb.Ball\_Id,**

**bbb.Innings\_No, bbb.Runs\_Scored**

**FROM ball\_by\_ball bbb**

**JOIN matches m**

**ON bbb.Match\_Id = m.Match\_Id**

**JOIN season s**

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

**WHERE s.Season\_Year IN (2015, 2016)**

**),**

**cte2 AS (**

**-- Calculate total runs per player**

**SELECT Striker, SUM(Runs\_Scored) AS Total\_Runs**

**FROM cte**

**GROUP BY Striker**

**),**

**cte3 AS (**

**-- Calculate runs from boundaries (4s and 6s) per player**

**SELECT Striker, SUM(Runs\_Scored) AS Runs\_In\_Boundaries**

**FROM cte**

**WHERE Runs\_Scored IN (4, 6)**

**GROUP BY Striker**

**)**

**-- Final output with boundary percentage calculation**

**SELECT c2.Striker AS Player\_Id, p.Player\_Name,**

**c2.Total\_Runs, c3.Runs\_In\_Boundaries,**

**ROUND((c3.Runs\_In\_Boundaries \* 100.0 / c2.Total\_Runs), 2) AS Boundary\_Percentage**

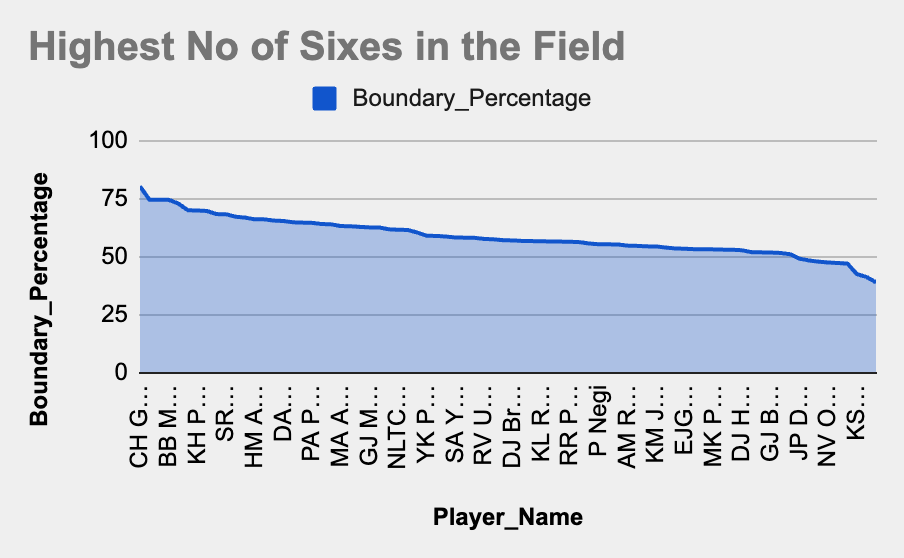
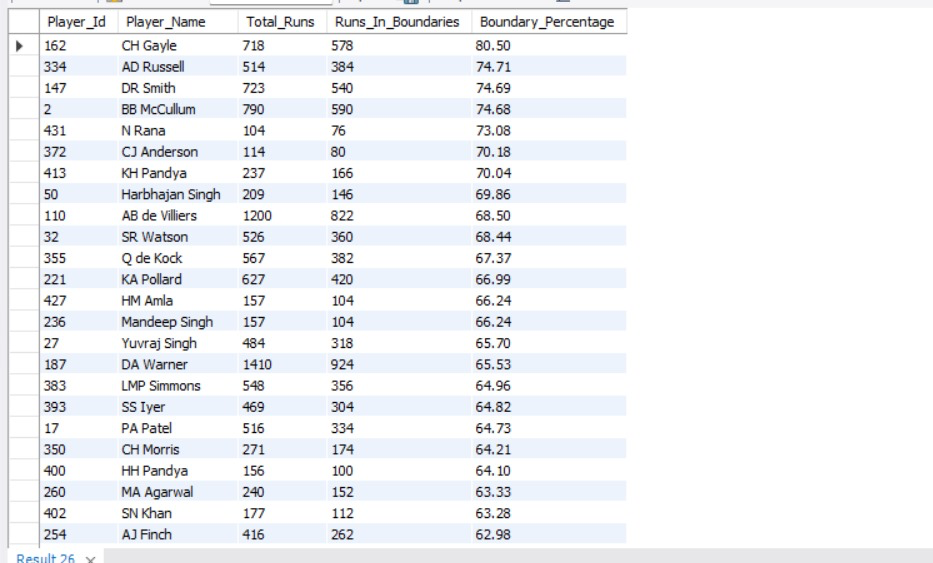
**FROM cte2 c2**

**JOIN cte3 c3 ON c2.Striker = c3.Striker**

**JOIN player p ON c2.Striker = p.Player\_Id**

**WHERE c2.Total\_Runs >= 100**

**ORDER BY Boundary\_Percentage DESC;**

* This code calculates the percentage of runs scored from boundaries (fours and sixes) for players in the 2015 and 2016 IPL seasons, and filters players with a total run count of 100 or more.
* It analyzes the performance of batsmen in the 2015 and 2016 IPL seasons, focusing on the percentage of their runs scored from boundaries (fours and sixes).
* Then it calculates the total runs and boundary runs for each player, filters for those who have scored 100 or more total runs, and orders the results by the percentage of runs from boundaries, in descending order.
* The output includes the player ID, player name, total runs, runs from boundaries, and the boundary percentage

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

* Code:

**# Identify good all-rounders for better team combinations.**

**WITH batting\_performance AS (**

**SELECT p.Player\_Id, p.Player\_Name,**

**SUM(bb.Runs\_Scored) AS Total\_Runs,**

**COUNT(bb.Ball\_Id) AS Balls\_Faced,**

**ROUND((SUM(bb.Runs\_Scored) / COUNT(bb.Ball\_Id)) \* 100, 2) AS Batting\_Strike\_Rate**

**FROM player p**

**JOIN ball\_by\_ball bb**

**ON p.Player\_Id = bb.Striker**

**JOIN matches m**

**ON bb.Match\_Id = m.Match\_Id**

**JOIN ball\_by\_ball b**

**ON bb.Match\_Id = b.Match\_Id**

**AND bb.Over\_Id = b.Over\_Id**

**AND bb.Ball\_Id = b.Ball\_Id**

**AND bb.Innings\_No = b.Innings\_No**

**WHERE bb.Runs\_Scored IS NOT NULL**

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

**),**

**bowling\_performance AS (**

**SELECT p.Player\_Id, p.Player\_Name,**

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

**ROUND(SUM(bb.Runs\_Scored) / (COUNT(bb.Ball\_Id) / 6.0), 2) AS Economy\_Rate**

**FROM player p**

**JOIN ball\_by\_ball bb ON p.Player\_Id = bb.Bowler**

**LEFT JOIN wicket\_taken w**

**ON bb.Match\_Id = w.Match\_Id**

**AND bb.Over\_Id = w.Over\_Id**

**AND bb.Ball\_Id = w.Ball\_Id**

**AND bb.Innings\_No = w.Innings\_No**

**JOIN ball\_by\_ball bs**

**ON bs.Match\_Id = bb.Match\_Id**

**AND bs.Over\_Id = bb.Over\_Id**

**AND bs.Ball\_Id = bb.Ball\_Id**

**AND bs.Innings\_No = bb.Innings\_No**

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

**HAVING COUNT(bb.Ball\_Id) > 100**

**)**

**SELECT DISTINCT bp.Player\_Id, bp.Player\_Name,**

**bp.Total\_Runs, bp.Batting\_Strike\_Rate, bp.Balls\_Faced,**

**bw.Total\_Wickets, bw.Economy\_Rate**

**FROM batting\_performance bp**

**JOIN bowling\_performance bw ON bp.Player\_Id = bw.Player\_Id**

**JOIN player\_match pm ON bp.Player\_Id = pm.Player\_Id**

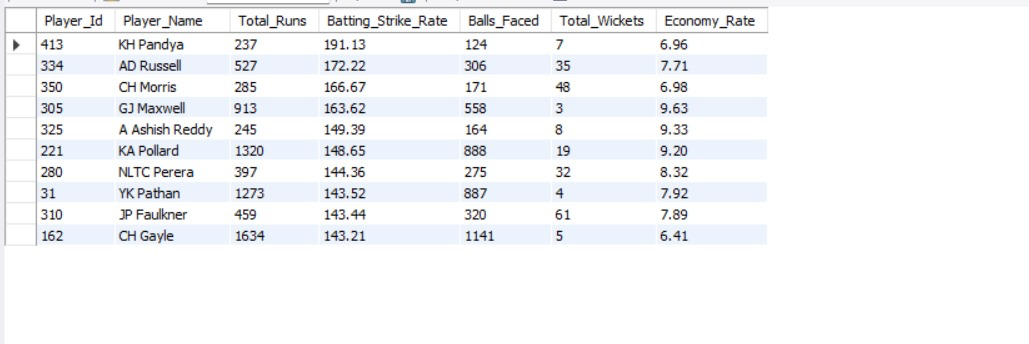
**WHERE pm.Role\_Id NOT IN (SELECT Role\_Id FROM rolee WHERE Role\_Desc IN ("Keeper","CaptainKeeper"))**

**AND bp.Balls\_Faced > 100**

**ORDER BY bp.Batting\_Strike\_Rate DESC, bw.Economy\_Rate ASC**

**LIMIT 10;**

* Before heading into the mega auction, RCB needs to focus on building a well-balanced squad that covers both batting and bowling strengths. This query helps in identifying top-performing all-rounders who can contribute in both departments.
* First, the query examines batting performance, filtering out players who have faced at least 100 balls and calculating their total runs and batting strike rate. This ensures only consistent batsmen are considered.
* Then, it analyzes bowling performance, calculating the total wickets taken and the economy rate of each bowler. Only bowlers who have bowled more than 100 balls are included to avoid small sample sizes.
* After identifying strong batters and bowlers, the query finds players who excel in both aspects, ensuring RCB targets genuine all-rounders in the auction. It also excludes wicket keepers and captains, as the focus is on players who contribute as both batsmen and bowlers.
* The final list prioritizes players with a high batting strike rate and a low economy rate, ensuring they can score quickly while also being effective with the ball.
* **Recommendation:**
* Invest in all-rounders who consistently perform in both batting and bowling.
* Prioritize power-hitters with a high strike rate and bowlers who keep the economy rate low.
* Avoid relying too much on specialist batsmen or bowlers; instead, target multi-dimensional players who provide flexibility in team selection.



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

* Code:

**/\* Powerplay and Death Over Utilization: In high-scoring matches, teams aim to maximize the power play (overs 1-6) and death overs (Overs 16-20) by scoring aggressively. \*/**

**SELECT t.Team\_Name,**

**SUM(CASE WHEN bb.Over\_Id BETWEEN 1 AND 6 THEN bb.Runs\_Scored ELSE 0 END) AS Powerplay\_Runs,**

**SUM(CASE WHEN bb.Over\_Id BETWEEN 16 AND 20 THEN bb.Runs\_Scored ELSE 0 END) AS Death\_Over\_Runs**

**FROM team t**

**JOIN matches m ON t.Team\_Id = m.Team\_1 OR t.Team\_Id = m.Team\_2**

**JOIN ball\_by\_ball bb ON m.Match\_Id = bb.Match\_Id**

**GROUP BY t.Team\_Name**

**ORDER BY Powerplay\_Runs DESC, Death\_Over\_Runs DESC;**

**/\* High Scoring Venues: Some venues favour the batsmen more then others, venues play a significant role in a high-scoring match \*/**

**SELECT v.Venue\_Name,**

**AVG(match\_runs.Total\_Runs) AS Avg\_Runs\_Per\_Match,**

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

**FROM venue v**

**JOIN matches m ON v.Venue\_Id = m.Venue\_Id**

**JOIN (**

**SELECT bb.Match\_Id, SUM(bb.Runs\_Scored) AS Total\_Runs**

**FROM ball\_by\_ball bb**

**GROUP BY bb.Match\_Id**

**) AS match\_runs ON m.Match\_Id = match\_runs.Match\_Id**

**GROUP BY v.Venue\_Name**

**ORDER BY Total\_Matches DESC, Avg\_Runs\_Per\_Match DESC**

**LIMIT 10;**

* In high-scoring matches, there are several critical factors that contribute to a higher run rate. These factors impact not only the match outcome but also the viewership and team strategies.

1. **Powerplay and Death Over Utilization:**

* The first factor contributing to high-scoring matches is how well teams utilize the power play (overs 1-6) and death overs (overs 16-20). Teams that maximize scoring in these overs by taking risks and being aggressive can post higher totals.
* The query calculates how many runs each team scores during the powerplay and death overs, helping us understand the aggressive approach teams adopt in these periods. Teams that score heavily in powerplay and death overs often push the total to higher levels.

1. **High-Scoring Venues:**

* Some venues tend to favour batsmen more than others due to factors like pitch conditions, weather, and altitude. These venues naturally lead to high-scoring matches.
* The second query identifies high-scoring venues by calculating the average runs per match at each venue. This allows us to see which venues tend to produce higher scores, which could impact team strategy (e.g., choosing to bowl first or setting aggressive targets).
* Understanding these venues helps teams tailor their strategies to the conditions and even influences viewership, as fans love watching high-scoring, exciting matches.

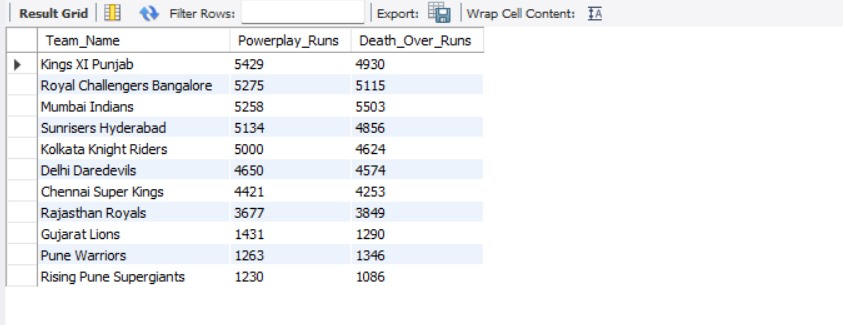
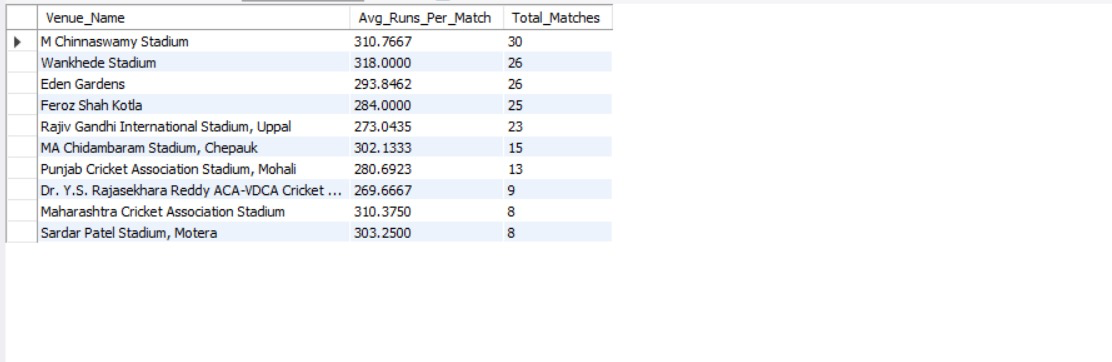
#### **Impact on Viewership and Team Strategies:**

1. **Viewership:**

* High-scoring matches are more entertaining and exciting to watch, which attracts more fans. These matches, especially in the powerplay and death overs, are full of big shots and intense moments, which is exactly what viewers enjoy. This contributes to higher ratings and fan engagement.

1. **Team Strategies**:

* In high-scoring matches, teams may focus on aggressive batting strategies, maximizing runs in the powerplay and death overs.
* Teams may opt for fast bowlers who can bowl at the death, or spin bowlers who can restrict runs in the middle overs.
* Venue-specific strategies come into play, where teams choose to bowl or bat first depending on whether the venue is historically batsman-friendly or bowler-friendly.



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

* Code:

**# Home vs Away Win/Loss record**

**WITH win\_loss\_record AS (**

**SELECT m.Match\_Id, v.Venue\_Name,**

**CASE WHEN m.Match\_Winner = 2 THEN 'Win' ELSE 'Loss'**

**END AS Result,**

**CASE WHEN v.Venue\_Id = 1 THEN 'Home' ELSE 'Away'**

**END AS Venue\_Type**

**FROM matches m**

**JOIN venue v ON m.Venue\_Id = v.Venue\_Id**

**WHERE (m.Team\_1 = 2 OR m.Team\_2 = 2) AND Outcome\_type != 2**

**)**

**SELECT**

**Venue\_Type,**

**COUNT(CASE WHEN Result = 'Win' THEN 1 END) AS Wins,**

**COUNT(CASE WHEN Result = 'Loss' THEN 1 END) AS Losses,**

**COUNT(\*) AS Total\_Matches,**

**ROUND(COUNT(CASE WHEN Result = 'Win' THEN 1 END) / COUNT(\*) \* 100, 2) AS Win\_Percentage**

**FROM win\_loss\_record**

**GROUP BY Venue\_Type;**

**#Home away batting performance**

**WITH rcb\_run\_stats AS (**

**SELECT m.Match\_Id, v.Venue\_Name,**

**CASE WHEN v.Venue\_Id = 1 THEN 'Home' ELSE 'Away' END AS Venue\_Type,**

**SUM(CASE WHEN bb.Team\_Batting = 2 THEN bb.Runs\_Scored ELSE 0 END) AS Runs\_Scored,**

**SUM(CASE WHEN bb.Team\_Bowling = 2 THEN bb.Runs\_Scored ELSE 0 END) AS Runs\_Conceded**

**FROM matches m**

**JOIN venue v ON m.Venue\_Id = v.Venue\_Id**

**JOIN ball\_by\_ball bb ON m.Match\_Id = bb.Match\_Id**

**WHERE (m.Team\_1 = 2 OR m.Team\_2 = 2) -- 2 is the team ID for RCB**

**GROUP BY m.Match\_Id, v.Venue\_Name**

**)**

**SELECT Venue\_Type,**

**ROUND(AVG(Runs\_Scored), 2) AS Avg\_Runs\_Scored,**

**ROUND(SUM(Runs\_Scored), 2) AS Total\_Runs\_Scored**

**FROM rcb\_run\_stats**

**GROUP BY Venue\_Type;**

**WITH bowling\_performance AS (**

**SELECT v.Venue\_Name,**

**CASE WHEN v.Venue\_Id = 1 THEN 'Home' ELSE 'Away' END AS Venue\_Type,**

**SUM(CASE WHEN bb.Team\_Bowling = 2 THEN bb.Runs\_Scored ELSE 0 END) AS Runs\_Conceded,**

**COUNT(CASE WHEN bb.Team\_Bowling = 2 AND w.Player\_Out IS NOT NULL THEN 1 ELSE NULL END) AS Wickets\_Taken,**

**COUNT(CASE WHEN bb.Team\_Bowling = 2 THEN bb.Ball\_Id ELSE NULL END) AS Balls\_Bowled**

**FROM matches m**

**JOIN venue v ON m.Venue\_Id = v.Venue\_Id**

**JOIN ball\_by\_ball bb ON m.Match\_Id = bb.Match\_Id**

**LEFT JOIN wicket\_taken w ON bb.Match\_Id = w.Match\_Id**

**AND bb.Over\_Id = w.Over\_Id**

**AND bb.Ball\_Id = w.Ball\_Id**

**AND bb.Innings\_No = w.Innings\_No -- Ensuring correct innings mapping**

**WHERE (m.Team\_1 = 2 OR m.Team\_2 = 2) -- 2 is the team ID for RCB**

**GROUP BY v.Venue\_Name,Venue\_Type**

**)**

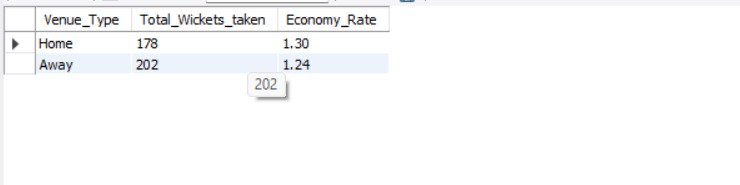
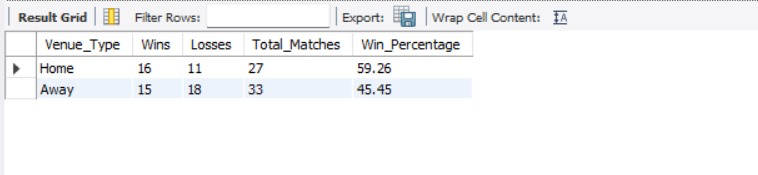
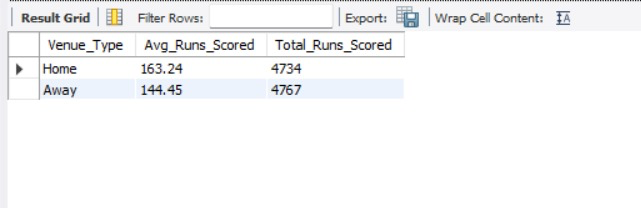
**SELECT Venue\_Type,**

**ROUND(SUM(Wickets\_taken),2) AS Total\_Wickets\_taken,**

**ROUND(SUM(Runs\_Conceded) / SUM(Balls\_Bowled), 2) AS Economy\_Rate**

**FROM bowling\_performance**

**GROUP BY Venue\_Type;**



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.**

* Code:

**# A. Win-Loss Performance Over Seasons**

**WITH win\_loss\_record AS (**

**SELECT m.Season\_Id,CASE WHEN m.Match\_Winner = 2 THEN 'Win' ELSE 'Loss' END AS Result**

**FROM matches m**

**WHERE (m.Team\_1 = 2 OR m.Team\_2 = 2) AND Outcome\_type != 2**

**)**

**SELECT Season\_Id,COUNT(CASE WHEN Result = 'Win' THEN 1 END) AS Wins,**

**COUNT(CASE WHEN Result = 'Loss' THEN 1 END) AS Losses,**

**COUNT(\*) AS Total\_Matches,**

**ROUND(COUNT(CASE WHEN Result = 'Win' THEN 1 END) / COUNT(\*) \* 100, 2) AS Win\_Percentage**

**FROM win\_loss\_record**

**GROUP BY Season\_Id**

**ORDER BY Season\_Id;**

**# B. Batting performance each season**

**WITH rcb\_batting\_in\_powerplay AS (**

**SELECT bb.Match\_Id, bb.Innings\_No, bb.Striker AS Batsman\_Id, p.Player\_Name,**

**SUM(bb.Runs\_Scored) AS total\_runs\_in\_power\_play,**

**COUNT(bb.Ball\_Id) AS balls\_faced\_in\_power\_play**

**FROM Ball\_by\_Ball bb**

**JOIN Matches m ON bb.Match\_Id = m.Match\_Id**

**JOIN Player p ON bb.Striker = p.Player\_Id**

**WHERE (m.Team\_1 = 2 OR m.Team\_2 = 2)**

**AND bb.Over\_Id BETWEEN 1 AND 6**

**GROUP BY bb.Match\_Id, bb.Innings\_No, bb.Striker, p.Player\_Name**

**)**

**SELECT rcb.Player\_Name,**

**SUM(rcb.total\_runs\_in\_power\_play) AS total\_runs\_in\_power\_play,**

**SUM(rcb.balls\_faced\_in\_power\_play) AS total\_balls\_faced\_in\_powerplay,**

**ROUND((SUM(rcb.total\_runs\_in\_power\_play) / NULLIF(SUM(rcb.balls\_faced\_in\_power\_play), 0)) \* 100, 2) AS strike\_rate\_in\_power\_play**

**FROM rcb\_batting\_in\_powerplay rcb**

**GROUP BY rcb.Player\_Name**

**HAVING total\_balls\_faced\_in\_powerplay > 100**

**ORDER BY strike\_rate\_in\_power\_play DESC;**

**# C. Bowling performance each season**

**WITH death\_overs\_bowling AS (**

**SELECT bb.Match\_Id, bb.Innings\_No, bb.Bowler AS Bowler\_Id, p.Player\_Name,**

**SUM(bb.Runs\_Scored) AS runs\_conceded,**

**COUNT(bb.Ball\_Id) AS balls\_bowled,**

**COUNT(w.Player\_Out) AS wickets\_taken**

**FROM Ball\_by\_Ball bb**

**LEFT JOIN Wicket\_Taken w ON bb.Match\_Id = w.Match\_Id**

**AND bb.Over\_Id = w.Over\_Id**

**AND bb.Ball\_Id = w.Ball\_Id**

**AND bb.Innings\_No = w.Innings\_No**

**JOIN Player p ON bb.Bowler = p.Player\_Id**

**JOIN Matches m ON bb.Match\_Id = m.Match\_Id**

**WHERE (m.Team\_1 = 2 OR m.Team\_2 = 2)**

**AND bb.Over\_Id BETWEEN 16 AND 20**

**GROUP BY bb.Match\_Id, bb.Innings\_No, bb.Bowler, p.Player\_Name**

**)**

**SELECT d.Player\_Name,**

**SUM(d.runs\_conceded) AS runs\_conceded\_in\_death,**

**SUM(d.balls\_bowled) AS total\_balls\_bowled\_in\_death,**

**SUM(d.wickets\_taken) AS total\_wickets\_in\_death,**

**ROUND((SUM(d.runs\_conceded) / NULLIF((SUM(d.balls\_bowled) / 6), 0)), 2) AS economy\_rate\_in\_death**

**FROM death\_overs\_bowling d**

**GROUP BY d.Player\_Name**

**HAVING total\_balls\_bowled\_in\_death > 100**

**ORDER BY economy\_rate\_in\_death ASC;**

* The query focuses on analyzing RCB's performance over the past seasons, breaking down the performance in key areas: Win-Loss record, batting performance in powerplay, and bowling performance in death overs. Here's a breakdown:

1. **Win-Loss Performance Over Seasons**

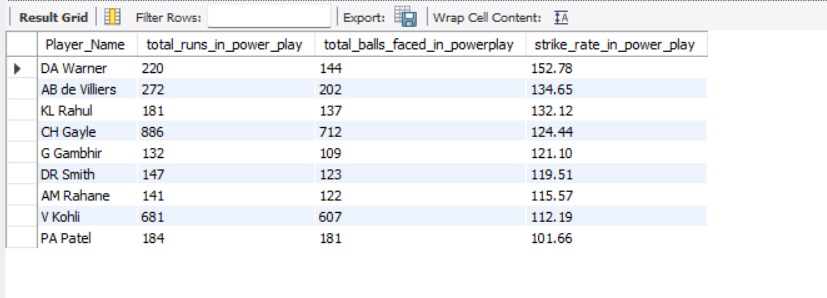
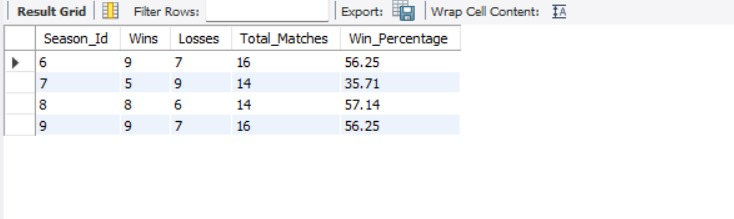
* This query calculates the win percentage of RCB in each season. It looks at the number of wins and losses for RCB, specifically analyzing seasons where RCB participated (Team 1 or Team 2).
* By calculating the win percentage, we can understand how often RCB won in each season. A low win percentage may indicate inconsistencies or poor match strategies, contributing to the team's failure to win a trophy.

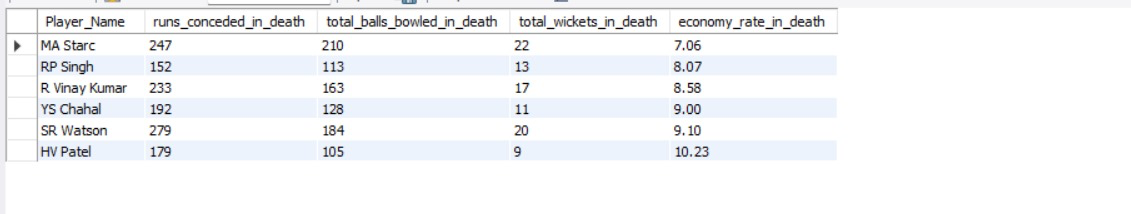
1. **Batting Performance in Powerplay**

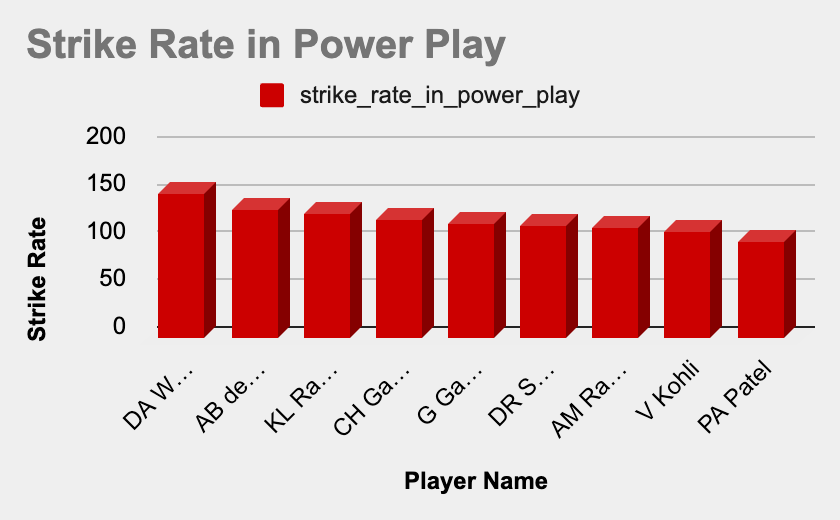
* This query focuses on RCB's batting performance in the powerplay (overs 1-6). The strike rate during powerplay is calculated for each batsman who faced over 100 balls during the season.
* A high strike rate in powerplay is essential to get the team off to a strong start. If RCB’s batsmen have low strike rates, it might suggest that the team struggled to build momentum in the early overs, reducing the chances of a high total.

1. **Bowling Performance in Death Overs:**

* This query analyzes RCB's bowling performance in death overs (overs 16-20). It calculates the economy rate and wickets taken by each bowler during the death overs.
* A high economy rate (too many runs conceded) or low wickets taken during death overs could suggest that RCB’s bowlers struggled to contain the opposition or take crucial wickets in the final overs, contributing to poor match outcomes.







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

* If the objective and subjective questions weren't provided, I would approach the problem by first focusing on RCB's performance data and then analyzing key factors that could help improve their chances of winning in the future. Here’s a step-by-step approach::

1. **Data Collection and Preparation:**

* Understand the available data by exploring the IPL match data for RCB, including details about team performance, individual player statistics, and match outcomes. This would involve reviewing tables like:
* Matches (Match outcomes, toss decisions, team performance).
* Player stats (Batting, bowling, strike rates, runs, wickets).
* Venue-specific data (to understand performance at different locations).
* Innings performance (Powerplay and death overs).

1. **Data Cleaning and Structuring:**

* Clean and structure the data to ensure that only relevant information is retained. For example:
* Remove invalid or incomplete records.
* Filter out inconsistent data for players who didn't play enough matches.
* Ensure correct mapping between match IDs, player IDs, and performance statistics.

1. **Comparative Analysis:**

* Comparing RCB's performance with other teams: By comparing RCB's batting and bowling performance to other top-performing teams to identify gaps or areas for improvement.
* Venue Impact: Analyze whether RCB performs better or worse at specific venues. For example, do they have a better win percentage in high-scoring venues or are they struggling in bowler-friendly venues?

1. **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".**

* Code:

**Select \* from Team;**

**select Team\_Id, replace(Team\_name,"Delhi Daredevils","Delhi Capitals") as Team\_name from team;**