**IPL ANALYSIS**

Objective Questions

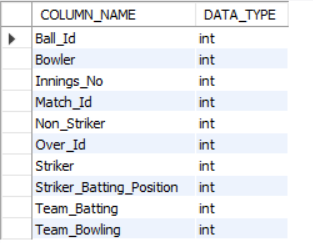
1. List the different dtypes of columns in table “ball\_by\_ball” (using information schema)

**SQL query**

SELECT COLUMN\_NAME, DATA\_TYPE

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'ball\_by\_ball'



1. What is the total number of run scored in 1st season by RCB (bonus : also include the extra runs using the extra runs table)  
     
   **Approach:** For resolving this question first we need to join ball\_by\_ball, batsman\_scored, extra\_runs,team and matches table to get proper output. Then taking sum of runs\_scored and extra runs to get total scores by RCB in season 1.

**SQL query:**

select sum(bs.Runs\_Scored+e.Extra\_Runs) as Total\_runs\_by\_RCB from ball\_by\_ball b left join batsman\_scored bs on b.Match\_Id=bs.Match\_Id and b.Over\_Id=bs.Over\_Id and b.Ball\_Id=bs.Ball\_Id and b.Innings\_No=bs.Innings\_No

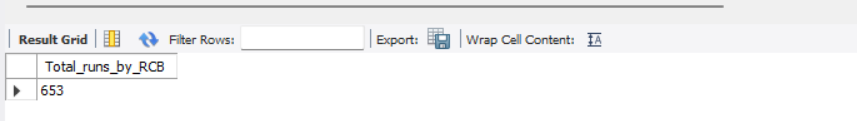
left join extra\_runs e on b.Match\_Id=e.Match\_Id and b.Over\_Id=e.Over\_Id and b.Ball\_Id=e.Ball\_Id and b.Innings\_No=e.Innings\_No

join team t on b.Team\_Batting=t.Team\_Id

join matches m on b.Match\_Id = m.Match\_Id

where m.Season\_Id=1;

**Output:**

****

**Visualization:**

**Conclusion:** From above query, output and charts we can conclude that in season 1 RCB scored only 653 runs.

1. How many players were more than age of 25 during season 2 ?

**Approach:** For calculating count of players were more than age of 25 during season 2, first need to join required tables then absolute difference between date of birth and season 2 year to get desired output.

**SQL query**

SELECT COUNT(\*) AS players\_over\_25

FROM Player p

JOIN Season s ON s.Season\_Id = 2

WHERE DATE\_FORMAT(CURDATE(), '%Y') - YEAR(p.DOB) > 25

AND s.Season\_Year = 2009;

**Output:**



1. How many matches did RCB win in season 1?  
     
   **Approach:** first need to join matches, team and matches\_won table then filter for season 1 and team RCB to get desired output.

**SQL query**

SELECT COUNT(\*) AS matches\_won

FROM matches m

JOIN team t ON (m.Match\_Winner = t.Team\_Id)

WHERE m.Season\_Id = 1 AND t.Team\_Name = 'Royal Challengers Bangalore';

**Output:**



1. List top 10 players according to their strike rate in last 4 seasons  
     
   **Approach:** first need to join ball\_by\_ball, batsman\_scored, player\_match, matches and player then calculate strike rate arrange in desc , filter it using season id from last 4 seasons select top 10 players using limit.

**SQL query:**

SELECT

p.Player\_id,

p.Player\_Name,

(SUM(bs.Runs\_Scored) \* 100.0 / COUNT(b.Ball\_Id)) AS Strike\_rate

FROM

ball\_by\_ball b

JOIN

batsman\_scored bs ON b.Match\_Id = bs.Match\_Id

AND b.Over\_Id = bs.Over\_Id

AND b.Ball\_Id = bs.Ball\_Id

JOIN

player\_match pm ON pm.Match\_Id = b.Match\_Id

AND pm.Player\_Id = b.Striker

join matches m on m.Match\_Id=b.Match\_Id

JOIN

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

where m.Season\_Id in ('6','7','8','9')

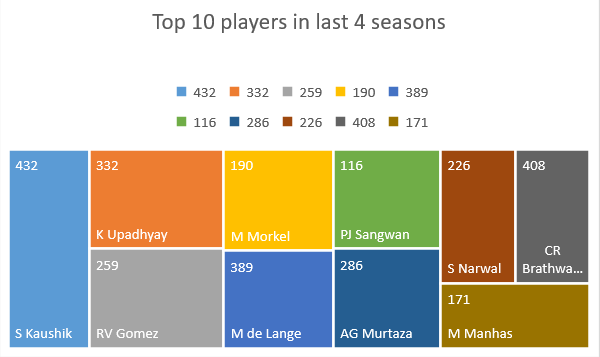
GROUP BY p.Player\_id, p.Player\_Name

order by Strike\_rate desc limit 10;

**Output:**

****

**Visualization:**

****

**Conclusion:** From above Output and chart we can conclude that S Kaushik, K Upadhyay and RV Gomez are the top most players in last 4 seasons.

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

**Approach:** For calculating average runs scored by each batsman considering all seasons first need to join batsman\_scored, ball\_by\_ball, matches and player table then create cte for players total runs across all season and then take average of total runs in other cte for all players across all seasons to get proper output.

**SQL query:**

WITH per\_player\_per\_season AS (

SELECT p.Player\_Name AS batsman,   
 m.Season\_Id, SUM(bs.Runs\_Scored) AS total\_runs  
 FROM batsman\_scored bs   
 JOIN ball\_by\_ball b ON bs.Match\_Id = b.Match\_Id   
 AND bs.Over\_Id = b.Over\_Id   
 AND bs.Ball\_Id = b.Ball\_Id   
 AND bs.Innings\_No = b.Innings\_No  
 JOIN matches m ON m.Match\_Id = bs.Match\_Id  
 JOIN player p ON p.Player\_Id = b.Striker  
 GROUP BY   
 p.Player\_Name, m.Season\_Id  
),

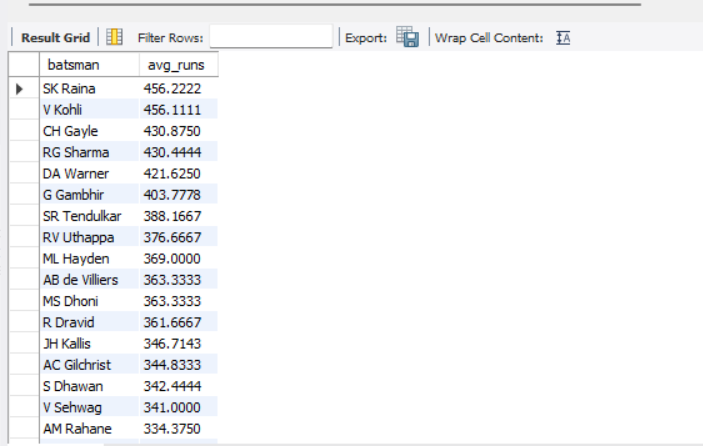
avg\_runs1 AS (

SELECT batsman, AVG(total\_runs) AS avg\_runs  
 FROM   
 per\_player\_per\_season  
 WHERE Season\_Id <= 9  
 GROUP BY   
 batsman

)

SELECT batsman, avg\_runs  
FROM avg\_runs1  
ORDER BY  
 avg\_runs DESC;

**Output:**

****

**Conclusion**: From above query and output we can calculate average runs for each batsman considering all season. SK Raina, V Kohli and CH Gayle are top 3 players for all seasons.

1. What are the average wickets taken by each bowler considering all the seasons?  
   **Approach:** for resolving question joins required table from schema and first calculate total wickets and then take an average of total wickets with respective player names and seasons.

**SQL query**

Select

p.Player\_Name,

avg(wt.Player\_Out) as AvgWickets

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

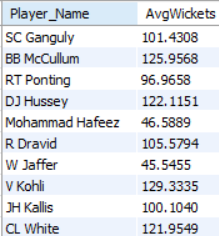
join

Wicket\_Taken wt on wt.Match\_Id=m.Match\_ID

group by

p.Player\_Name

**Output**

  
**Conclusion:** from above query and output we can conclude that average wickets considering all seasons. From output we can conclude that Sc Ganguly,BBMcCullum,RT Ponting are top 3 bowlers till season 9.

1. List all the players who have average runs scored greater than overall average and who have taken wickets greater than overall average  
     
   **Approach:** first find total runs and matches played for batsman then calculate total wicket taken and matches played for bowlers then calculate overall avg runs and overall avg wickets by using cross join for above two cte’s. After that player table join with bowler\_stat and batsman\_stat.

**SQL query**

with AvgRuns as (

Select

p.Player\_Name,

avg(bs.Runs\_Scored) as AvgRuns

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

join Batsman\_Scored bs on bs.Match\_Id = m.Match\_ID

group by p.Player\_Name

),

AvgWickets as (

Select

p.Player\_Name,

avg(wt.Player\_Out) as AvgWickets

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

join Wicket\_Taken wt on wt.Match\_Id = m.Match\_ID

group by p.Player\_Name

),

OverallAverages as (

Select

(Select avg(bs.Runs\_Scored) from Batsman\_Scored bs) as OverallAvgRuns,

(Select avg(wt.Player\_Out) from Wicket\_Taken wt) as OverallAvgWickets

)

Select

r.Player\_Name

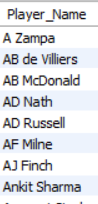
from AvgRuns r

join AvgWickets w on r.Player\_Name = w.Player\_Name

cross join OverallAverages oa

where r.AvgRuns > oa.OverallAvgRuns

and w.AvgWickets > oa.OverallAvgWickets;  
**Output**

  
  
**Conclusion:** From above query we can conclude thatthe query is designed to identify players who are exceptional in both batting and bowling, and provides a way to compare their performance to the overall averages of all batsmen and bowlers.

1. Create a table rcb\_record table that shows wins and losses of RCB in an individual venue.  
   **Approach:** The query creates a table rcb\_record that stores the **win** and **loss** counts for RCB across different **venues** and **seasons**. CASE statement to count the wins and losses based on whether RCB was one of the teams (Team\_1 or Team\_2) and if they won the match.

**SQL query:**

CREATE TABLE rcb\_record AS

SELECT

v.Venue\_Name,

s.Season\_Id,

SUM(CASE

WHEN (m.Team\_1 = 2 OR m.Team\_2 = 2) AND m.Match\_Winner = 2 THEN 1

ELSE 0

END) AS WIN,

SUM(CASE

WHEN (m.Team\_1 = 2 OR m.Team\_2 = 2) AND m.Match\_Winner != 2 THEN 1

ELSE 0

END) AS LOSS

FROM Matches m

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

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

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

GROUP BY v.Venue\_Name, s.Season\_Id;

select \* from rcb\_record  
**Output**

  
  
**Conclusion:** This query efficiently tracks RCB's performance by tallying wins and losses for each venue and season. The data will help analyse RCB's success patterns based on venue and season over time.

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

**Approach**: The query is creating output for the impact of bowling style on wickets taken using joins and perform some mathematical tasks to calculate wicket\_taken, matches\_played, balling skill impact on wickets.

**SQL Query:**

with bowler\_wickets as(select p.Player\_Id,bs.Bowling\_Skill, count(w.Player\_Out)as wickets\_taken,

count(b.Match\_Id) as matches\_played

from ball\_by\_ball b

join bowling\_style bs on b.Team\_Bowling=bs.Bowling\_Id

join wicket\_taken w on b.Match\_Id=w.Match\_Id

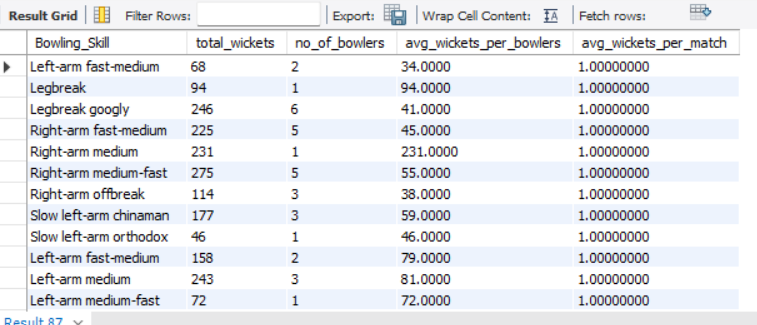
and b.Over\_Id=w.Over\_Id and b.Ball\_Id=w.Ball\_Id and b.Innings\_No=w.Innings\_No

join player\_match p on b.Match\_Id=p.Match\_Id group by 1,2)

select bw.Bowling\_Skill,sum(bw.wickets\_taken) as total\_wickets, count(Player\_Id) as no\_of\_bowlers, avg(wickets\_taken) as avg\_wickets\_per\_bowlers,avg(wickets\_taken/matches\_played) as avg\_wickets\_per\_match

from bowler\_wickets bw group by bw.Bowling\_Skill, bw.wickets\_taken order by avg\_wickets\_per\_match;

**Output:**



**Conclusion:** The query calculates bowling performance metrics grouped by **bowling skill type**. It computes the **total wickets taken**, the **number of bowlers** per skill type, the **average wickets per bowler,** and the **average wickets per match** for each bowling skill. This helps analyse how different bowling styles perform in terms of wicket-taking efficiency across matches and bowlers.

1. Write the sql query to provide a status of whether the performance of the team better than the previous year performance on the basis of number of runs scored by the team in the season and number of wickets taken   
   **Approach:** First calculate team performance and then calculate previous year performance using lag function. By doing comparisons using case statement to decide performance is mixed, better or worsed.

**SQL query**

WITH team\_performance AS (

SELECT

m.Season\_Id,

m.Team\_1,

SUM(bs.Runs\_Scored) AS total\_runs,

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

FROM

Matches m

JOIN

ball\_by\_ball b ON m.Match\_Id = b.Match\_Id

LEFT JOIN

Batsman\_Scored bs ON b.Match\_Id = bs.Match\_Id AND b.Over\_Id = bs.Over\_Id AND b.Ball\_Id = bs.Ball\_Id

LEFT 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

GROUP BY

m.Season\_Id, m.Team\_1

),

performance\_comparison AS (

SELECT

tp.Season\_Id,

tp.Team\_1,

tp.total\_runs,

tp.total\_wickets,

LAG(tp.total\_runs, 1) OVER (PARTITION BY tp.Team\_1 ORDER BY tp.Season\_Id) AS prev\_season\_runs,

LAG(tp.total\_wickets, 1) OVER (PARTITION BY tp.Team\_1 ORDER BY tp.Season\_Id) AS prev\_season\_wickets

FROM

team\_performance tp

)

SELECT

Season\_Id,

Team\_1,

total\_runs,

total\_wickets,

prev\_season\_runs,

prev\_season\_wickets,

CASE

WHEN total\_runs > prev\_season\_runs AND total\_wickets > prev\_season\_wickets THEN 'Better'

WHEN total\_runs = prev\_season\_runs AND total\_wickets = prev\_season\_wickets THEN 'Same'

ELSE 'Worse'

END AS performance\_status

FROM

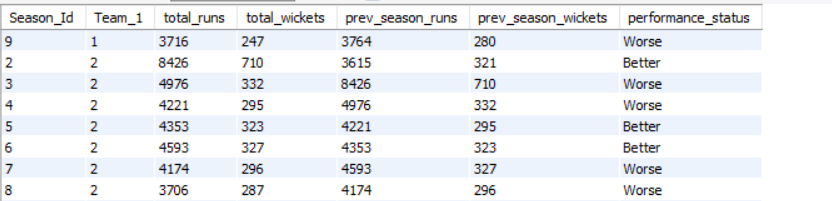
performance\_comparison

WHERE

prev\_season\_runs IS NOT NULL -- To exclude the first season

ORDER BY

Team\_1, Season\_Id;  
**Output:**



**Conclusion:** The query provides comparison of current performance is better than previous year or not. By using this, then can analyse improvement points and they will work on it to improve their performance if their performance was worse. This analysis will increase overall performance of team.

1. Can you derive more KPIs for the team strategy if possible?  
   **Approach**: for deriving more KPI’s for team strategy first we can calculate team id, team name, matches\_played, total innings, total runs, total\_wickets, fours, sixes, dot balls , balls faced, matches won ,strike rate, avg runs per match, avg wickets per match, use case statement to decide performance of the as compared to previous year.

**SQL query:**

WITH team\_performance AS (

SELECT

m.Season\_Id,

t.Team\_Id,

t.Team\_Name,

COUNT(DISTINCT m.Match\_Id) AS matches\_played,

COUNT(b.Innings\_No) AS total\_innings,

SUM(bs.Runs\_Scored) AS total\_runs,

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

SUM(CASE WHEN bs.Runs\_Scored = 4 THEN 1 ELSE 0 END) AS fours,

SUM(CASE WHEN bs.Runs\_Scored = 6 THEN 1 ELSE 0 END) AS sixes,

SUM(CASE WHEN bs.Runs\_Scored = 0 THEN 1 ELSE 0 END) AS dot\_balls,

COUNT(b.Ball\_Id) AS balls\_faced,

COUNT(DISTINCT CASE WHEN m.Win\_Type = t.Team\_Id THEN m.Match\_Id END) AS matches\_won

FROM team t

JOIN matches m ON t.Team\_Id IN (m.Team\_1, m.Team\_2)

LEFT JOIN ball\_by\_ball b ON b.Match\_Id = m.Match\_Id

LEFT JOIN batsman\_scored bs ON b.Match\_Id = bs.Match\_Id AND b.Over\_Id = bs.Over\_Id AND b.Ball\_Id = bs.Ball\_Id AND b.Innings\_No = bs.Innings\_No

LEFT 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 1, 2, 3

),

performance\_comparison AS (

SELECT

Season\_Id,

Team\_Id,

Team\_Name,

matches\_played,

total\_innings,

total\_runs,

total\_wickets,

fours,

sixes,

dot\_balls,

balls\_faced,

matches\_won,

ROUND(total\_runs/balls\_faced \* 100, 2) AS strike\_rate,

ROUND(total\_runs/matches\_played, 2) AS avg\_runs\_per\_match,

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

ROUND(dot\_balls/matches\_played \* 100, 2) AS dot\_balls\_percentage,

LAG(total\_runs) OVER (PARTITION BY Team\_Id ORDER BY Season\_Id) AS prev\_year\_runs,

LAG(total\_wickets) OVER (PARTITION BY Team\_Id ORDER BY Season\_Id) AS prev\_year\_wickets,

LAG(ROUND(total\_runs/balls\_faced, 2)) OVER (PARTITION BYTeam\_Id ORDER BY Season\_Id) AS prev\_year\_strike\_rate

FROM team\_performance

)

SELECT

Season\_Id,

Team\_Name,

matches\_played,

total\_runs,

total\_wickets,

fours,

sixes,

strike\_rate,

avg\_runs\_per\_match,

avg\_wickets\_per\_match,

prev\_year\_runs,

prev\_year\_wickets,

prev\_year\_strike\_rate,

CASE

WHEN total\_runs > prev\_year\_runs AND total\_wickets > prev\_year\_wickets AND strike\_rate > prev\_year\_strike\_rate THEN 'Significantly Better'

WHEN total\_runs > prev\_year\_runs AND total\_wickets > prev\_year\_wickets THEN 'Better'

WHEN total\_runs < prev\_year\_runs AND total\_wickets < prev\_year\_wickets AND strike\_rate < prev\_year\_strike\_rate THEN 'Significantly Worse'

WHEN total\_runs < prev\_year\_runs AND total\_wickets < prev\_year\_wickets THEN 'Worse'

WHEN strike\_rate > prev\_year\_strike\_rate THEN 'Improved Scoring Rate'

ELSE 'Mixed'

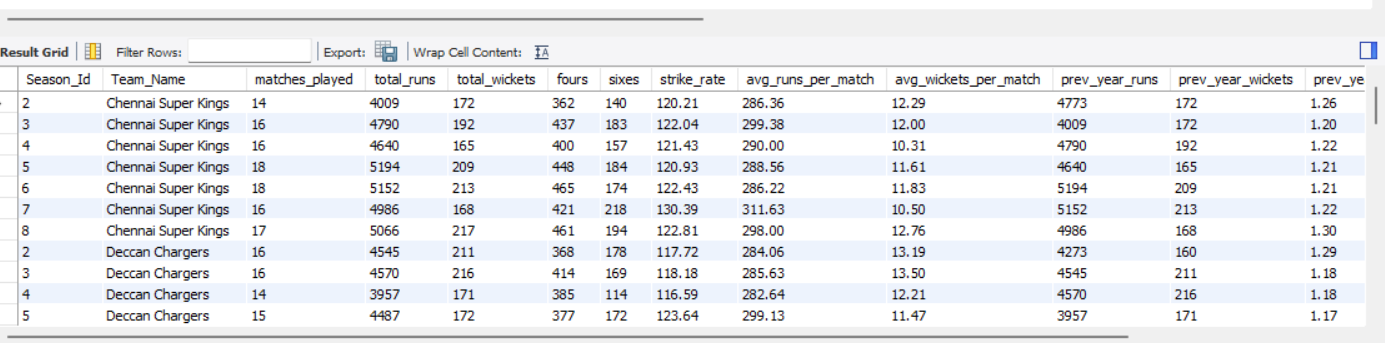
END AS performance\_status

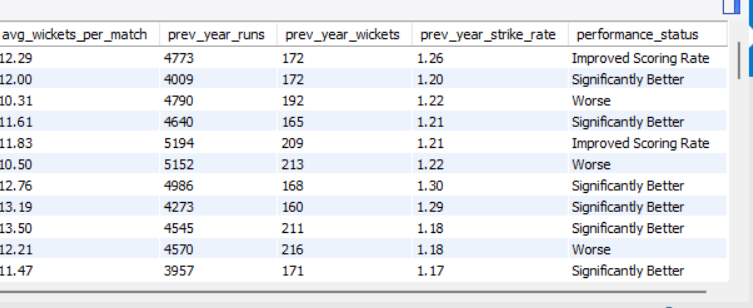
FROM performance\_comparison

WHERE prev\_year\_runs IS NOT NULL AND prev\_year\_wickets IS NOT NULL

ORDER BY Team\_Name, Season\_Id;

**Output:**

****

****

**Conclusion:** From above output we can conclude that this analysis will help us to build strategy for increasing team performance in matches.

1. Using SQL, write a query to find out average wickets taken by each bowler in each venue. Also rank the gender according to the average value.  
   **Approach:** For resolving this query first need to calculate bowler’s performance on different venues like total wickets, avg\_wickets and we need to give rank to the bowlers using dense rank function to get proper output.

**SQL query:**

WITH bowler\_performance AS (

SELECT

b.Bowler,

v.Venue\_Name,

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

FROM

matches m

JOIN

ball\_by\_ball b ON m.Match\_Id = b.Match\_Id

JOIN

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

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

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

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

JOIN

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

GROUP BY

1, 2

),

avg\_wickets AS (

SELECT

Bowler,

Venue\_Name,

total\_wickets,

AVG(total\_wickets) OVER(PARTITION BY Bowler) AS avg\_wickets

FROM

bowler\_performance

),

ranked\_bowler AS (

SELECT

Bowler,

Venue\_Name,

total\_wickets,

avg\_wickets,

DENSE\_RANK() OVER(ORDER BY avg\_wickets DESC) AS bowler\_rank

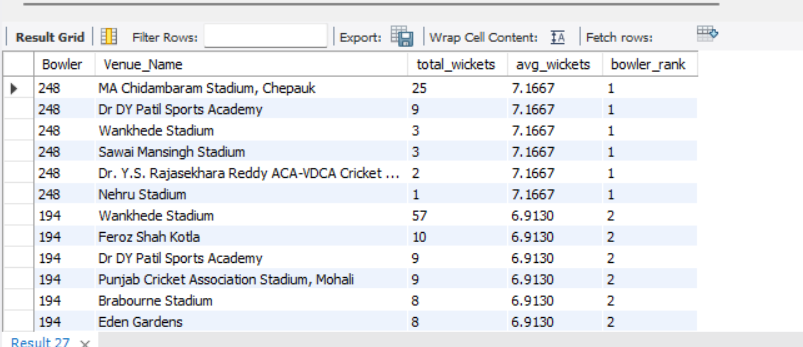
FROM

avg\_wickets

)

SELECT \* FROM ranked\_bowler ORDER BY avg\_wickets DESC, total\_wickets DESC;

**Output:**



**Conclusion:** From the above query and output we can conclude that it ranked bowlersbased on their **average wickets taken per venue.** This helps identify top-performing bowlers by comparing their wicket-taking consistency across different venues.

1. Which of the given players have consistently performed well in past seasons? (will you use any visualisation to solve the problem) **Approach:** For calculating consistency of given players we need to some calculations like total runs, matches played, sixes, fours, average runs across seasons, average wickets across season, season above 30 runs and season above 2 wickets. With the help of this parameter we can decide players who have consistently performed well in past.

**SQL query:**

WITH player\_performance AS (

SELECT

m.Season\_Id,

p.Player\_Id,

p.Player\_Name,

SUM(bs.Runs\_Scored) AS total\_runs,

COUNT(DISTINCT m.Match\_Id) AS matches\_played,

-- Count the number of wickets where the player was the bowler

SUM(CASE WHEN w.Player\_Out IS NOT NULL OR p.Player\_Id = b.Bowler THEN 1 ELSE 0 END) AS wickets\_taken,

COUNT(b.Ball\_Id) AS balls\_faced,

SUM(CASE WHEN bs.Runs\_Scored = 4 THEN 1 ELSE 0 END) AS fours,

SUM(CASE WHEN bs.Runs\_Scored = 6 THEN 1 ELSE 0 END) AS sixes,

-- Calculate averages for each season

ROUND(SUM(bs.Runs\_Scored) / COUNT(DISTINCT m.Match\_Id), 2) AS avg\_runs\_per\_match,

ROUND(SUM(CASE WHEN w.Player\_Out IS NOT NULL OR p.Player\_Id = b.Bowler THEN 1 ELSE 0 END)

/ COUNT(DISTINCT m.Match\_Id), 2) AS avg\_wickets\_per\_match

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

ball\_by\_ball b ON m.Match\_Id = b.Match\_Id AND p.Player\_Id = b.Striker

JOIN

batsman\_scored bs ON b.Match\_Id = bs.Match\_Id AND b.Over\_Id = bs.Over\_Id AND b.Ball\_Id = bs.Ball\_Id

LEFT 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

GROUP BY

m.Season\_Id, p.Player\_Id, p.Player\_Name

)

SELECT

Player\_Id,

Player\_Name,

COUNT(DISTINCT Season\_Id) AS seasons\_played,

AVG(avg\_runs\_per\_match) AS avg\_runs\_across\_seasons,

AVG(avg\_wickets\_per\_match) AS avg\_wickets\_across\_seasons,

COUNT(CASE WHEN avg\_runs\_per\_match > 50 THEN 1 END) AS seasons\_above\_30\_runs,

COUNT(CASE WHEN avg\_wickets\_per\_match > 2 THEN 1 END) AS seasons\_above\_2\_wickets

FROM

player\_performance

GROUP BY

Player\_Id, Player\_Name

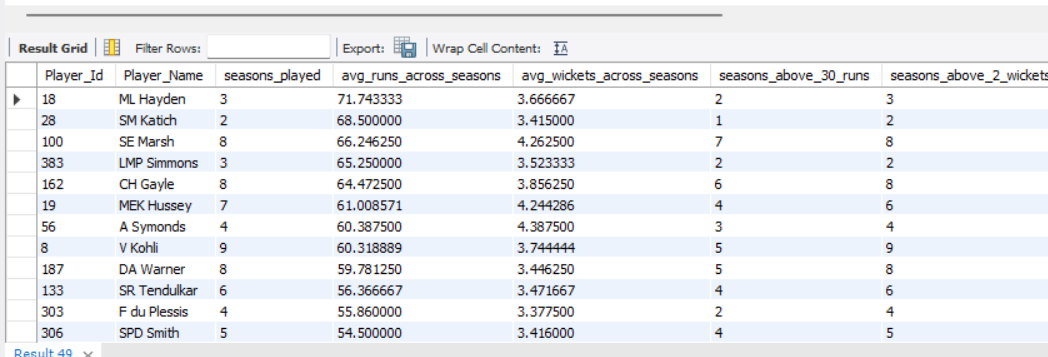
HAVING

seasons\_above\_30\_runs >= 2 OR seasons\_above\_2\_wickets >= 2

ORDER BY

avg\_runs\_across\_seasons DESC, avg\_wickets\_across\_seasons DESC limit 15;

**Output:**

****

**Visualization:**

**Conclusion:** The query evaluates player performance across different seasons by calculating **total runs, wickets taken**, and other metrics like **fours, sixes**, and **balls faced**. It computes the **average runs** and **wickets per match** for each season, then filters and ranks players who have excelled by having at least two seasons with an average of more than 30 runs or 2 wickets per match. The final output ranks players based on their performance consistency over the seasons.

1. Are there players whose performance is more suited to specific venues or conditions? (how would you present this using charts?)   
   **Approach:** For resolving this we need to calculate average runs scored by each player, average wickets taken, seasons above 50 runs, seasons above 2 wickets etc.

**SQL query**

WITH player\_performance AS (

SELECT

m.Season\_Id,

v.Venue\_Name, -- Include the venue name

p.Player\_Id,

p.Player\_Name,

SUM(bs.Runs\_Scored) AS total\_runs,

COUNT(DISTINCT m.Match\_Id) AS matches\_played,

-- Count the number of wickets where the player was the bowler

SUM(CASE WHEN w.Player\_Out IS NOT NULL OR p.Player\_Id = b.Bowler THEN 1 ELSE 0 END) AS wickets\_taken,

COUNT(b.Ball\_Id) AS balls\_faced,

SUM(CASE WHEN bs.Runs\_Scored = 4 THEN 1 ELSE 0 END) AS fours,

SUM(CASE WHEN bs.Runs\_Scored = 6 THEN 1 ELSE 0 END) AS sixes,

-- Calculate averages for each season

ROUND(SUM(bs.Runs\_Scored) / COUNT(DISTINCT m.Match\_Id), 2) AS avg\_runs\_per\_match,

ROUND(SUM(CASE WHEN w.Player\_Out IS NOT NULL or p.Player\_Id = b.Bowler THEN 1 ELSE 0 END)

/ COUNT(DISTINCT m.Match\_Id), 2) AS avg\_wickets\_per\_match

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

venue v ON m.Venue\_Id = v.Venue\_Id -- Join to get venue information

JOIN

ball\_by\_ball b ON m.Match\_Id = b.Match\_Id AND p.Player\_Id = b.Striker

JOIN

batsman\_scored bs ON b.Match\_Id = bs.Match\_Id AND b.Over\_Id = bs.Over\_Id AND b.Ball\_Id = bs.Ball\_Id

LEFT 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

GROUP BY

m.Season\_Id, v.Venue\_Name, p.Player\_Id, p.Player\_Name

)

SELECT

Player\_Id,

Player\_Name,

Venue\_Name, -- Include the venue name in the final output

COUNT(DISTINCT Season\_Id) AS seasons\_played,

AVG(avg\_runs\_per\_match) AS avg\_runs\_across\_seasons,

AVG(avg\_wickets\_per\_match) AS avg\_wickets\_across\_seasons,

COUNT(CASE WHEN avg\_runs\_per\_match > 50 THEN 1 END) AS seasons\_above\_50\_runs,

COUNT(CASE WHEN avg\_wickets\_per\_match > 2 THEN 1 END) AS seasons\_above\_2\_wickets

FROM

player\_performance

GROUP BY

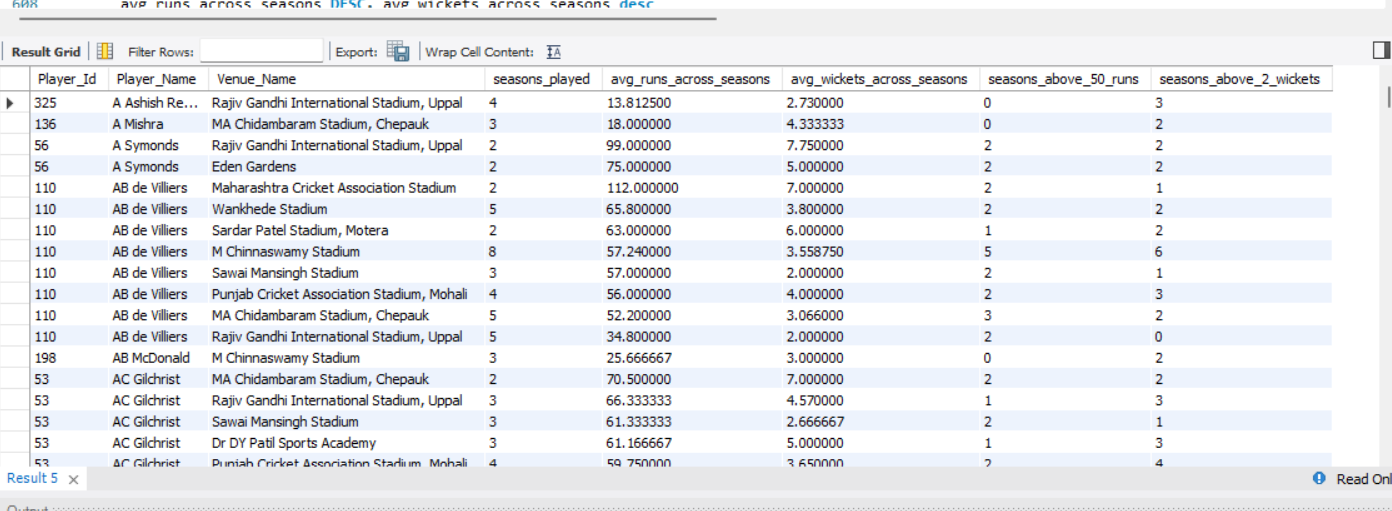
Player\_Id, Player\_Name, Venue\_Name -- Group by Venue\_Name as well

HAVING

seasons\_above\_50\_runs >= 2 OR seasons\_above\_2\_wickets >= 2

ORDER BY

avg\_runs\_across\_seasons DESC, avg\_wickets\_across\_seasons DESC;  
  
**Output:**

  
  
**Visualization:**

Subjective Questions

1. How does toss decision have affected the result of the match ? (which visualisations could be used to better present your answer)

And is the impact limited to only specific venues?

**Approach:** For calculating toss decision have affected the result of match we need to use case statement for deciding toss decision either field or bat. Then calculate toss winner wins or losses, total matches, win\_percentage to decide affect.

**SQL query:**

select v.Venue\_Name,

case when m.Toss\_Decide=1 then 'Field'else 'Bat' end as toss\_decide,

sum(case when m.Toss\_Winner=m.Match\_Winner then 1 else 0 end ) as toss\_winner\_wins,

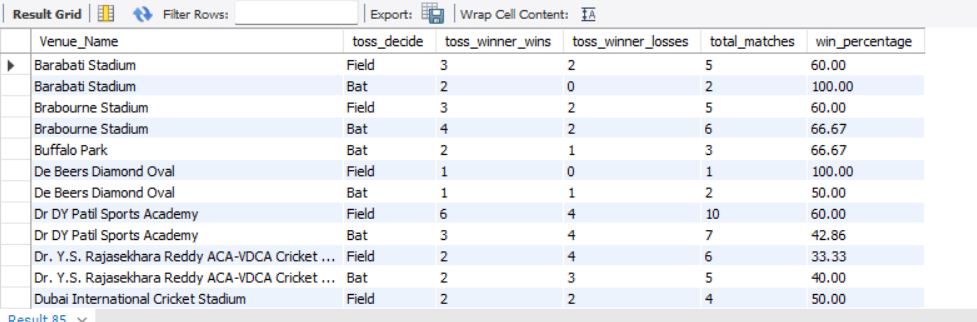
sum(case when m.Toss\_Winner!=m.Match\_Winner then 1 else 0 end ) as toss\_winner\_losses,

count(m.Match\_Id) as total\_matches,

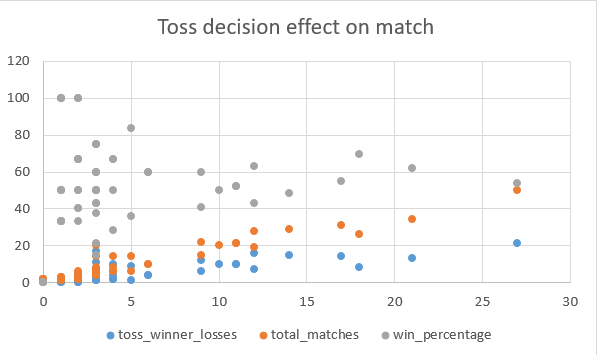
round(sum(case when m.Toss\_Winner=m.Match\_Winner then 1 else 0 end)/count(m.Match\_Id) \*100 , 2) as win\_percentage from matches m join venue v on m.Venue\_Id=v.Venue\_Id

group by v.Venue\_Name,m.Toss\_Decide order by v.Venue\_Name,m.Toss\_Decide;

**Output:**

****

**Visualization:**

****

**Conclusion:** From the above chart we can conclude that total matches (orange dots) generally increase as we move right on the x-axis, suggesting more data for later teams or matches. Toss winner losses (blue dots) are consistently lower than total matches, which is expected. Win percentage (grey dots) shows high variability, ranging from very low (near 0%) to very high (near 100%). This suggests that winning the toss doesn't guarantee winning the match. Estimating from the spread of grey dots, it appears that the overall winning percentage when winning the toss is roughly in the 50-60% range. This suggests that winning the toss provides a slight advantage, but it's not a decisive factor in determining the match outcome.

1. Suggest some of the players who would be best fit for the team?  
   **Approach:** To analyse the performance of batsmen and bowlers, the query takes the following approach: joins tables, filter matches, group by player, calculation metric, sort and limit.

**SQL query:**

#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 b.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 batsman\_scored 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

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 accross different venues

SELECT p.Player\_Name,

v.Venue\_Id,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 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

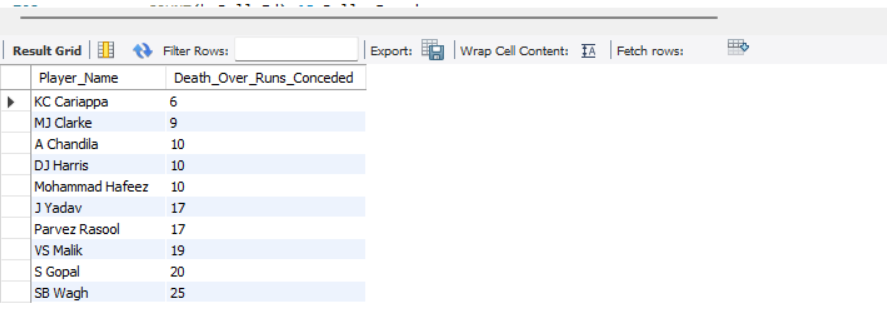
JOIN batsman\_scored b

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

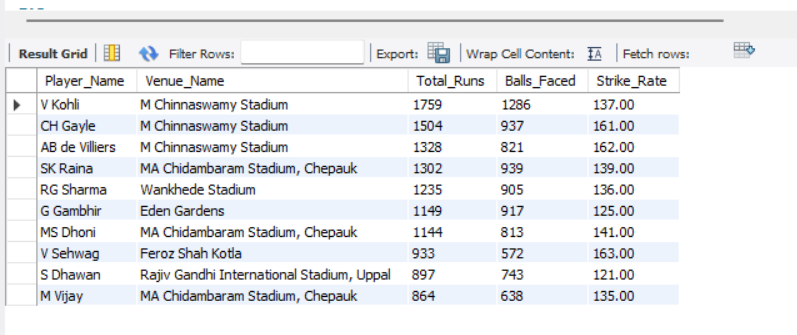
GROUP BY p.Player\_Name, v.Venue\_Name

ORDER BY Total\_Runs DESC, Strike\_Rate DESC LIMIT 10;

**Output:**

For bowlers:

For batsman:



**Visualization:**

**Conclusion:** These two queries provide valuable insights into the performance of bowlers and batsmen in a cricket tournament. By analysing death over bowling performance and batting performance across different venues, coaches and selectors can:

Identify bowlers who are most effective in death overs and can be relied upon to restrict the opposition's score.

Identify batsmen who perform well at specific venues and can be selected accordingly.

Develop strategies to counter the strengths and weaknesses of opposing teams.

Make informed decisions about team selection and player development.

By combining these insights with other performance metrics, coaches and selectors can build a well-rounded team with a strong balance of batting, bowling, and all-rounding abilities.

Suggestions

|  |  |
| --- | --- |
| Captain | Shreyas iyer |
| batting | Shreyas iyer |
|  | KOHLI |
| Bowlers: | jasprit bumrah |
|  | yuzvendra chahal |
| coach | Mahela jayawardhane |

1. What are some of parameters that should be focused while selecting the players?  
   **Approach:** To analyse the performance of batsmen and bowlers, the query takes the following approach: joins tables, filter matches, group by player, calculation metric, sort and limit.

**SQL query:**

#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 b.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 batsman\_scored 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

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 accross different venues

SELECT p.Player\_Name,

v.Venue\_Id,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 ASStrike\_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

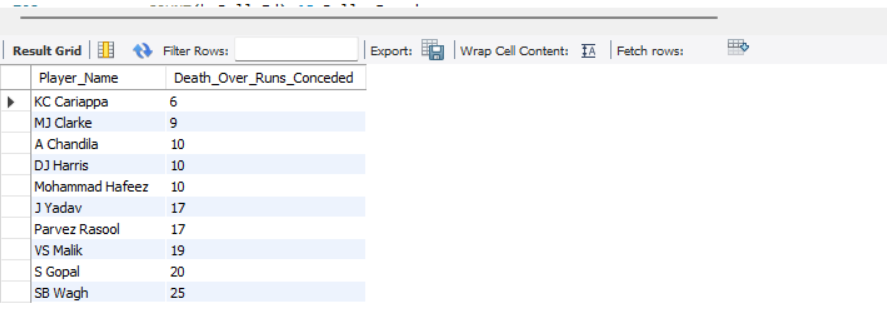
JOIN batsman\_scored b

ON b.Match\_Id = bb.Match\_Id AND b.Over\_Id = bb.Over\_Id ANDb.Ball\_Id = bb.Ball\_Id AND b.Innings\_No = bb.Innings\_No

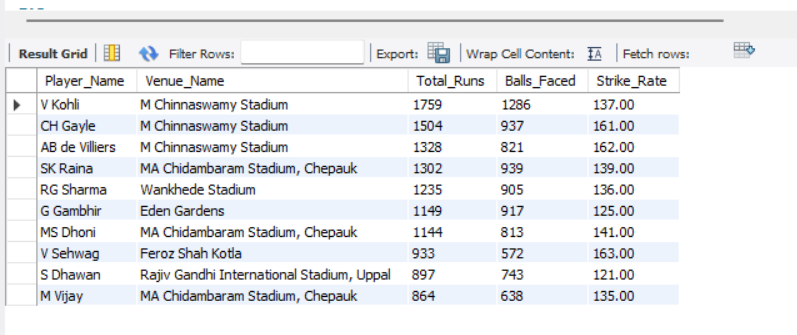
GROUP BY p.Player\_Name, v.Venue\_Name

ORDER BY Total\_Runs DESC, Strike\_Rate DESC LIMIT 10;

**Output:**

For bowlers:

For batsman:



**Visualization:**

**Conclusion:** These two queries provide valuable insights into the performance of bowlers and batsmen in a cricket tournament. By analysing death over bowling performance and batting performance across different venues, coaches and selectors can:

Identify bowlers who are most effective in death overs and can be relied upon to restrict the opposition's score.

Identify batsmen who perform well at specific venues and can be selected accordingly.

Develop strategies to counter the strengths and weaknesses of opposing teams.

Make informed decisions about team selection and player development.

By combining these insights with other performance metrics, coaches and selectors can build a well-rounded team with a strong balance of batting, bowling, and all-rounding abilities.

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)  
   **Approach:** first finding batting performance for players, then bowling performance for players and joining them on player id to get common players who offers versitality

**SQL query:**

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 batsman\_scored 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

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

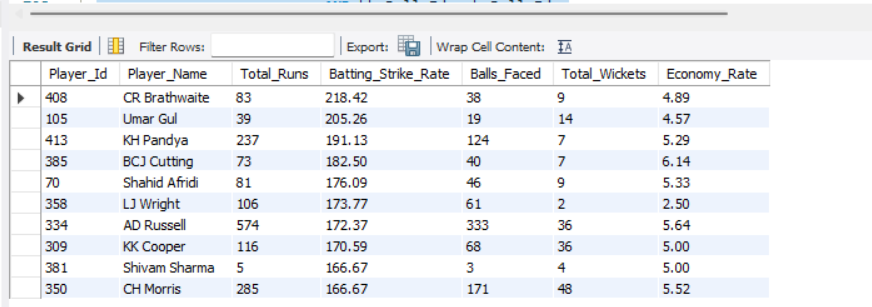
FROM batting\_performance bp

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

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

LIMIT 10;

**Output:**

****

**Visualization:**

**Conclusion:** This query analyses the performance of cricket players, combining their batting and bowling statistics. It calculates:

Batting performance: total runs, balls faced, and batting strike rate

Bowling performance: total wickets taken and economy rate

The results are sorted by batting strike rate in descending order and economy rate in ascending order, highlighting the top all-round performers. The query provides a comprehensive view of each player's strengths and weaknesses, enabling coaches and selectors to make informed decisions about team selection and strategy development.

1. Are there players whose presence positively influences the morale and performance of the team? (justify your answer using visualisation)  
     
   **Approach:** To find players presence positively influences the morale and performance of the team by calculating wins with player, matches with player, win rate with player, win without player, total matches, win rate without player.

**SQL query:**

WITH player\_influence AS (

-- Check team's win rate when player is in the playing 11

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

SUM(CASE WHEN m.Match\_Winner = pm.Team\_Id THEN 1 ELSE 0 END) AS Wins\_With\_Player,

COUNT(pm.Match\_Id) AS Matches\_With\_Player

FROM player\_match pm

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

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

WHERE pm.Team\_Id = 2

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

),

team\_win\_rate AS (

-- Calculate overall team win rate

SELECT m.Team\_1 AS Team\_Id,

SUM(CASE WHEN m.Match\_Winner = 2 THEN 1 ELSE 0 END) AS Wins\_Without\_Player,

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

FROM matches m

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

GROUP BY m.Team\_1

)

SELECT pi.Player\_Name, pi.Wins\_With\_Player, pi.Matches\_With\_Player,

ROUND((pi.Wins\_With\_Player / pi.Matches\_With\_Player) \* 100, 2) AS Win\_Rate\_With\_Player,

tw.Wins\_Without\_Player, tw.Total\_Matches,

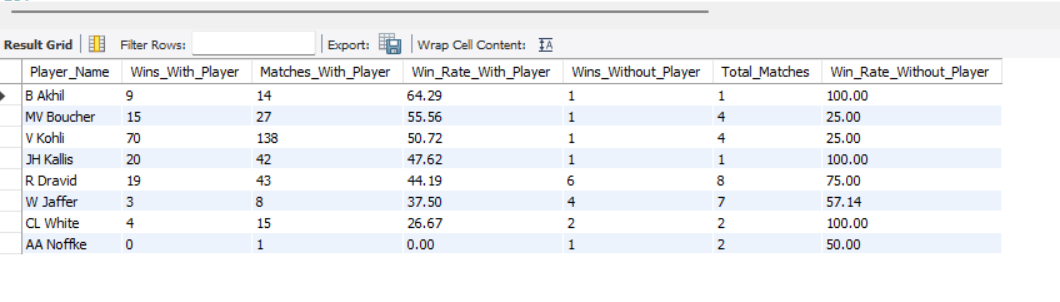
ROUND((tw.Wins\_Without\_Player / tw.Total\_Matches) \* 100, 2) ASWin\_Rate\_Without\_Player

FROM player\_influence pi

JOIN team\_win\_rate tw ON pi.Player\_Id = tw.Team\_Id

ORDER BY Win\_Rate\_With\_Player DESC;

**Output:**

**Visualization:**

**Conclusion:** Sometime, a team’s performance is affected by a player’s presence on the field.

In our data, we can observe that in Virat Kohli’s presence, RCB has won almost 51% of their matches, which indicates how strongly him being on the field affects the team’s moral and performance

We can also see that the team has won only 25% percent of the matches without Kohli’s presence.

Making sure that players who have such positive impact on field arealways given a chance is key to the team’s success and player’s moral.

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

* Rcb management lacks positivity.
* Players are giving everything for the team-ownership lacks positivity.
* Think tank should be changed and key batters like virat ,will jacks and maxwell must be retained.
* Bowling should be strengthened

Batting Performance:

* KH Pandya has the highest batting strike rate, followed closely by AD Russell.
* S Jayasuriya has the lowest batting strike rate among the players shown.
* Most players have batting strike rates between 150 and 200, indicating aggressive batting styles typical in T20 cricket.

Bowling Economy:

* GJ Maxwell has the highest economy rate, suggesting he's the most expensive bowler in terms of runs conceded per over.
* S Jayasuriya has the lowest economy rate, making him the most economical bowler of the group.
* There's significant variation in economy rates, ranging from about 6 to 10 runs per over.

All-round Performance:

* Some players like KH Pandya and AD Russell show high batting strike rates but also higher economy rates, suggesting they're aggressive all-rounders.
* Players like S Jayasuriya and MF Maharoof have lower batting strike rates but also lower economy rates, indicating they might be more defensive or specialist bowlers.

The data suggests different player roles - some are clearly batting specialists with high strike rates, others are bowling specialists with low economy rates, and some are all-rounders balancing both aspects.

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

* Chinnaswamy stadium is one of the smallest stadiums in india.
* So it always brings fun to the match...batters usually smash the bowlers all around the ground and some rival games agaisnt csk and mi were so iconic in this stadium...
* Although it is small, rcb management always fails to find right talent to play at chinnaswamy stadium. this is a major drawback and a substantial reason for their trophy drought.
* calculating the factors which are contributing to the high – scoring matches and the impact on viewership and team strategies using two factors 1) power play and death over scores ,2)average runs on each venue.

**SQL query:**

SELECT t.Team\_Name,

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

SUM(CASE WHEN bb.Over\_Id BETWEEN 16 AND 20 THEN b.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

JOIN batsman\_scored b ON bb.Match\_Id = b.Match\_Id AND bb.Over\_Id = b.Over\_Id AND bb.Ball\_Id = b.Ball\_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(b.Runs\_Scored) AS Total\_Runs

FROM ball\_by\_ball bb

JOIN batsman\_scored b ON bb.Match\_Id = b.Match\_Id

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

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

GROUP BY bb.Match\_Id

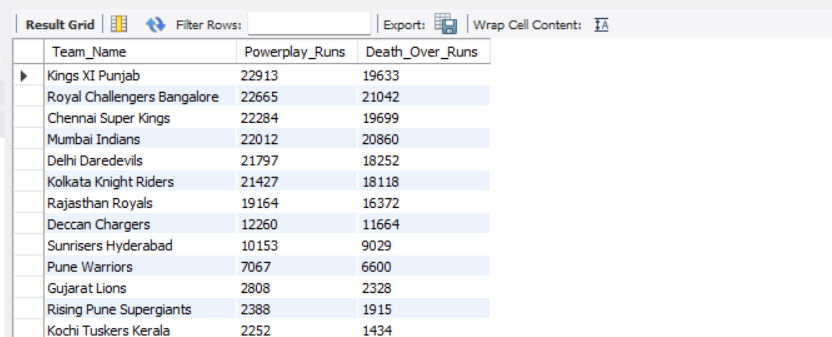
) AS match\_runs ON m.Match\_Id = match\_runs.Match\_Id

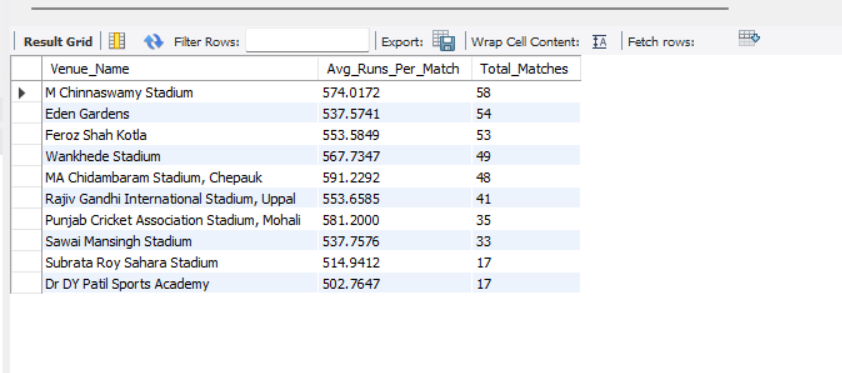
GROUP BY v.Venue\_Name

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

LIMIT 10;

**Output:**





**Visualization:**

**Conclusion:** 1) This graph compares the performance of different cricket teams in power play and death overs. Kings XI Punjab leads in both categories, scoring the most runs in power play (22,913) and death overs (15,833). There's a clear trend showing that teams with higher power play scores generally also have higher death over scores, indicating consistent aggressive batting throughout the innings. However, the gap between power play and death over runs narrows for teams lower on the list, suggesting these teams might struggle more in the final overs compared to their early innings performance.

* 2) The graph illustrates the average runs scored per match at different cricket venues, alongside the total number of matches played at each. M Chinnaswamy Stadium in Bangalore stands out as the highest-scoring venue with an average of 574 runs per match, while also hosting a significant number of games (58). There's a general trend showing that venues with more matches played tend to have higher average run scores, suggesting that familiarity with the ground or its batting-friendly nature may contribute to higher scoring games.

1. Analyze the impact of home ground advantage on team performance and identify strategies to maximize this advantage for RCB.  
   **Approach:** for resolving this issue, we can calculate home vs away performances like wins, losses, total matches, win\_percentage, batting performance, bowling performance

**SQL query:**

# 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 b.Runs\_Scored ELSE 0 END) AS Runs\_Scored,

SUM(CASE WHEN bb.Team\_Bowling = 2 THEN b.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

JOIN batsman\_scored b ON bb.Match\_Id = b.Match\_Id

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

AND bb.Ball\_Id = b.Ball\_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;

#Bowling performance at home

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 b.Runs\_Scored ELSE 0 END) AS Runs\_Conceded,

COUNT(CASE WHEN bb.Team\_Bowling = 2 THEN w.Player\_Out ELSE NULL END) AS Wickets\_Taken,

COUNT(CASE WHEN bb.Team\_Bowling = 2 THEN bb.Ball\_Id ELSENULL 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

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 batsman\_scored b ON bb.Match\_Id = b.Match\_Id

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

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

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

GROUP BY v.Venue\_Name

)

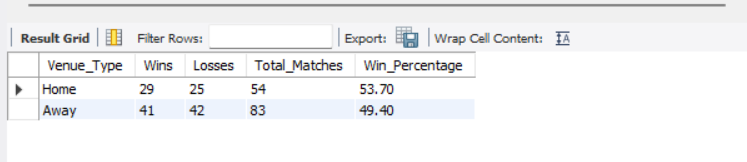
SELECT Venue\_Type,

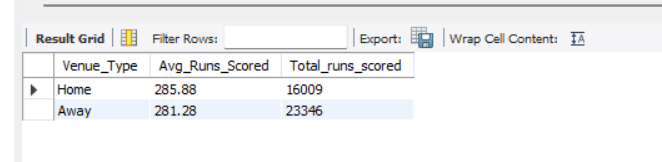
ROUND(SUM(Wickets\_taken),2) AS Total\_Wickets\_taken,

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

FROM bowling\_performance

GROUP BY Venue\_Type;

**OutPut:** ****





**Visualization:**

**Conclusion:** from above 3 graph we can conclude that when RCB is playing on home ground then performance is good as compared to away. Win percentage is more when RCB is playing at home ground. For batting performance, its good when RCB is playing away and for bowling performance like economy rate is better when they played at home ground as compared to total\_wickets\_taken.

1. Come up with a visual and analytical analysis with the RCB past seasons performance and potential reasons for them not winning a trophy.  
     
   **Approach:** For resolving this issue first we find win percentage over the seasons and then calculate batting performance over the seasons like total runs in power play, strike rate in power play etc. and bowling perforce like runs conceded in death, total balls in death, economy rate, total wickets in death etc.

**SQL query:**

# 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\_death\_overs AS (

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

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

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

FROM Batsman\_Scored bs

JOIN Ball\_by\_Ball bb 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

JOIN Matches m ON bs.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 bs.Over\_Id BETWEEN 1 AND 6

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

)

SELECT p.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\_death\_overs,

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

FROM rcb\_batting\_in\_death\_overs rcb

JOIN Player p ON rcb.Batsman\_Id = p.Player\_Id

GROUP BY p.Player\_Name

HAVING total\_balls\_faced\_in\_death\_overs >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, p.Player\_Name,

SUM(bs.Runs\_Scored) AS runs\_conceded,

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

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

FROM ball\_by\_ball bb

JOIN batsman\_scored bs ON bb.Match\_Id = bs.Match\_Id

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

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

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

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 p.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) / (SUM(d.balls\_bowled) / 6)), 2) AS economy\_rate\_in\_death

FROM death\_overs\_bowling d

JOIN player p ON d.Bowler = p.Player\_Id

JOIN matches m ON d.Match\_Id = m.Match\_Id

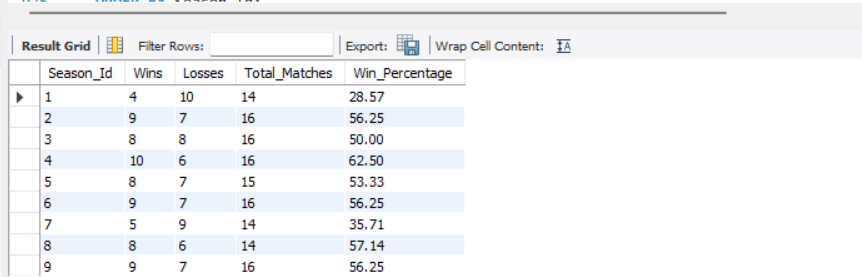
WHERE (m.Team\_1 = 2 OR m.Team\_2 = 2)

GROUP BY p.Player\_Name

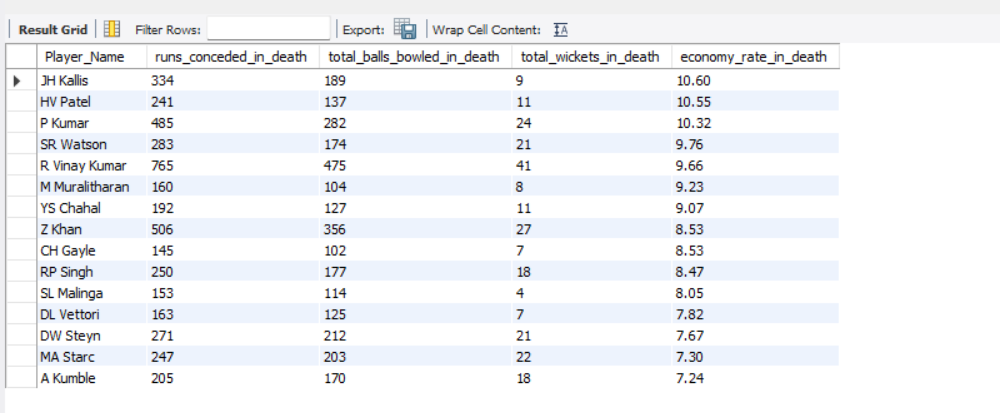
HAVING total\_balls\_bowled\_in\_death > 100

ORDER BY economy\_rate\_in\_death DESC;

**Output:**

****

****

****

**Visualization:**

**Conclusion:** From the first graph,it shows the win percentage over 9 seasons. There's a general upward trend in win percentage, starting at 28.57% in season 1 and ending at 56.25% in season 9. The win percentage fluctuates between seasons, with notable peaks in seasons 3 and 5, reaching 62.5% at its highest point.

From second graph, it shows batting performance across multiple seasons, tracking total runs in power play, total balls faced in death overs, and strike rate in power play. There's significant variation between players, with a few standout performances in runs and balls faced. The strike rate generally follows a declining trend across the seasons, with some fluctuations.

From third graph, it depicts bowling performance across seasons for different players, showing runs conceded, balls bowled, wickets taken, and economy rate in death overs. There's significant variation among players, with some standout performances in wickets taken (e.g., P Kumar, Y Chahal). Economy rates remain relatively consistent across players, while runs conceded and balls bowled vary more widely.

To overcome their trophy drought, RCB needs to address the following:

* **Management Stability**: Building a consistent management and leadership structure is crucial for long-term success.
* **Player Backing**: Creating a culture where players are backed despite a few failures would improve morale and team cohesion.
* **Mental Strength**: Instilling a winning mentality by embracing pressure and focusing on performing in key moments could help the team break their final hurdles.
* **Balanced Team**: Relying on a few stars has not worked for them. RCB needs to focus on building a well-rounded team where both batting and bowling departments are strong.

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

**1. Define Objectives**

Primary Goal

Identify top-performing and reliable players who offer the best value for money.

**Secondary Goals:**

Ensure a balanced team composition, consider future potential

**2. Data Collection**

Player Performance Data: Gather historical performance data for players, including batting averages, strike rates, bowling economy, and fielding statistics.

Player Profiles: Consider age, fitness levels, and versatility.

**3. Data Analysis**

Performance Metrics: Calculate key performance indicators (KPIs) such as runs scored, wickets taken, and catches/stumpings.

Consistency: Evaluate consistency by looking at performance over multiple seasons.

**4. Player Segmentation**

Core Players

Identify players who consistently perform well and can form the backbone of the team.

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"

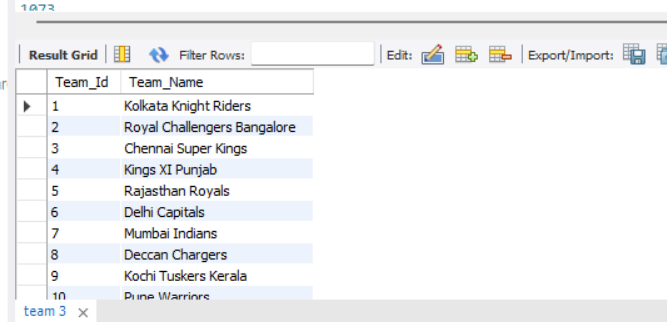
SQL Query

UPDATE Team

SET Team\_Name = 'Delhi\_Capitals'

WHERE Team\_Name = Delhi\_Daredevils '  
And Team\_Id=6.

Output

****