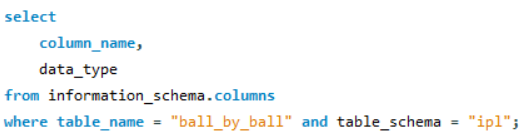
**Objective Questions**

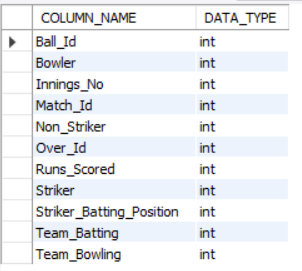
1. **List the different types of columns in table “ball\_by\_ball” (using information schema)**

**Ans:**

There are 11 columns present in the table “ball\_by\_ball”, which are fetched by the below query.



**Output:**



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

**Ans:**

with Score as (select

b.Match\_id,

b.Over\_Id,

b.Ball\_Id,

b.Innings\_No,

m.Season\_Id,

b.Team\_Batting,

t1.Team\_Name as Batting\_Team\_Name,

b.Team\_Bowling,

t2.Team\_Name as Bowling\_Team\_Name,

b.Runs\_Scored,

coalesce(e.Extra\_Runs,0) as Extra\_Runs

from ball\_by\_ball b join team t1

on t1.Team\_Id = b.Team\_Batting

join team t2 on t2.Team\_Id = b.Team\_Bowling

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

left join extra\_runs e on e.Match\_Id = b.Match\_Id

and e.Over\_Id = b.Over\_Id

and e.Ball\_Id = b.Ball\_Id

and e.Innings\_No = b.Innings\_No)

select

sum(Runs\_Scored + Extra\_Runs) as Total\_Runs

from Score

where Season\_Id = (select min(Season\_Id) from Score)

and Batting\_Team\_Name = "Royal Challengers Bangalore";

**Output:**



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

**Ans:**

with playercount as (select

count(distinct p.Player\_Id) as PlayerCountAbove25,

s.Season\_Id,

s.Season\_Year

from player p join player\_match pm

on p.Player\_Id = pm.Player\_Id

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

join season s on s.Season\_Id = m.Season\_Id

where s.Season\_Year = 2014

and timestampdiff(year, p.DOB, m.Match\_Date) > 25

group by s.Season\_Id, s.Season\_Year)

select

PlayerCountAbove25

from playercount;

**Output:**



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

**Ans:**

select

t.team\_Id,

t.Team\_Name,

count(m.Match\_Winner) as win\_count

from matches m join season s

on s.Season\_Id = m.Season\_Id

join team t

on m.Match\_Winner = t.Team\_Id

where t.Team\_Name = "Royal Challengers Bangalore"

and Season\_Year = 2013

group by t.team\_Id;

**Output:**



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

**Ans:**

with seasonYear as (select

Season\_Year

from season

order by Season\_Year desc

limit 4)

select

p.Player\_Name,

sum(b.Runs\_Scored) as totalRuns,

nullif(count(b.Ball\_Id),0) as balls\_faced,

round((sum(b.Runs\_Scored) \* 100 )/ nullif(count(b.Ball\_Id),0),2) as strike\_rate

from ball\_by\_ball b join matches m on b.Match\_Id = m.Match\_Id

join player p on p.Player\_Id = b.Striker

join season s on s.Season\_Id = m.Season\_Id

join seasonYear sy on sy.Season\_Year = s.Season\_Year

left join extra\_runs e on e.Match\_Id = b.Match\_Id

and e.Over\_Id = b.Over\_Id

and e.Ball\_Id = b.Ball\_Id

and e.Innings\_No = b.Innings\_No

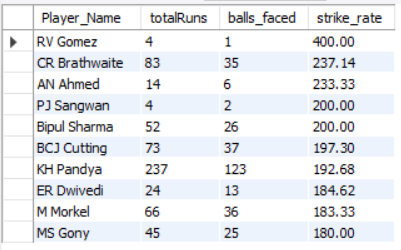
where e.Extra\_Runs is null

group by p.Player\_Name, b.Striker

order by strike\_rate desc

limit 10;

**Output:**



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

**Ans:**

select

p.player\_name,

round(sum(b.runs\_scored)/ count(distinct Match\_Id), 2) as average\_runs

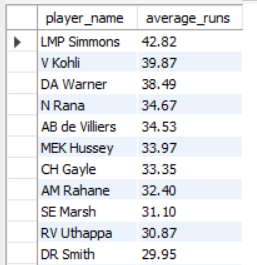
from ball\_by\_ball b

join player p on p.player\_id = b.striker

group by p.player\_name

order by average\_runs desc;

**Output:**



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

**Ans:**

select

p.Player\_Name,

count(wc.Player\_Out) as wickets,

count(distinct b.Match\_Id) as Matches\_played,

round(count(wc.Player\_Out)/count(distinct b.Match\_Id),2) as avg\_wickets\_taken

from wicket\_taken wc left join ball\_by\_ball b

on wc.Match\_Id = b.Match\_Id

and wc.Over\_Id = b.Over\_Id

and wc.Ball\_Id = b.Ball\_Id

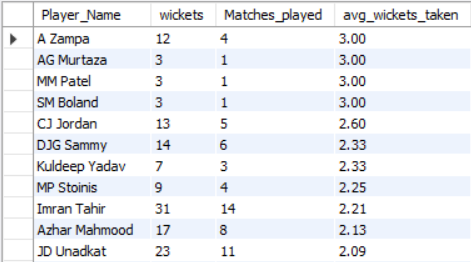
and wc.Innings\_No = b.Innings\_No

join player p on p.Player\_Id = b.Bowler

group by p.Player\_Name

order by avg\_wickets\_taken desc;

**Output:**



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

**Ans:**

with avg\_runs as (select

p.player\_name,

round(sum(b.runs\_scored)\*1.0 / count(distinct Match\_Id), 2) as average\_runs

from ball\_by\_ball b

join player p on p.player\_id = b.striker

group by p.player\_name),

overall\_avg\_runs as (select

avg(average\_runs) as overall\_ar

from avg\_runs),

avg\_wickets as (select

p.Player\_Name,

round(count(wc.Player\_Out)\* 1.0/count(distinct b.Match\_Id),2) as avg\_wickets\_taken

from wicket\_taken wc left join ball\_by\_ball b

on wc.Match\_Id = b.Match\_Id

and wc.Over\_Id = b.Over\_Id

and wc.Ball\_Id = b.Ball\_Id

and wc.Innings\_No = b.Innings\_No

join player p on p.Player\_Id = b.Bowler

group by p.Player\_Name),

overall\_avg\_wickets as (select

avg(avg\_wickets\_taken) as overall\_aw

from avg\_wickets)

select

ar.Player\_Name

from avg\_runs ar join avg\_wickets aw

on ar.Player\_Name = aw.Player\_Name

join overall\_avg\_runs oar on ar.average\_runs > oar.overall\_ar

join overall\_avg\_wickets oaw on aw.avg\_wickets\_taken > oaw.overall\_aw;

**Output:**



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

**Ans:**

create table rcb\_record as (select

v.Venue\_Name,

count(\*) as total\_matches,

sum(case when m.Match\_Winner = t.Team\_Id then 1 else 0 end) as wins,

sum(case when m.Match\_Winner <> Team\_Id and (m.Team\_1 = Team\_Id or m.Team\_2 = Team\_Id) then 1 else 0 end) as losses,

sum(case when m.Match\_Winner is null then 1 else 0 end) as no\_result

from venue v join matches m on v.Venue\_Id = m.Venue\_Id

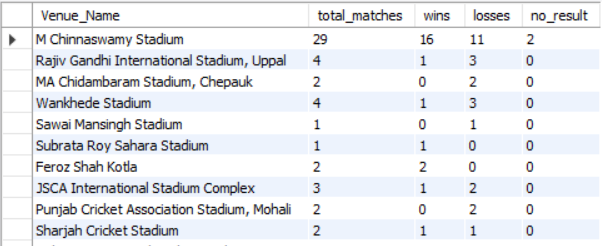
join team t on t.Team\_Name = "Royal Challengers Bangalore"

where (t.Team\_Id = m.Team\_1 or t.Team\_Id = m.Team\_2)

group by v.Venue\_Name);

select \* from rcb\_record;

**Output:**



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

**Ans:**

select

bs.Bowling\_skill as bowling\_style,

count(w.Player\_Out) as total\_wickets

from ball\_by\_ball b 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

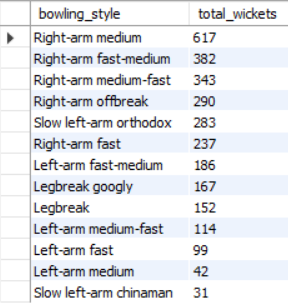
join player p on p.Player\_Id = b.Bowler

join bowling\_style bs on bs.Bowling\_Id = p.Bowling\_skill

group by bs.Bowling\_skill

order by total\_wickets desc;

**Output:**



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

**Ans:**

with team\_performance as (select

t.Team\_Name,

s.Season\_Year,

sum(b.Runs\_Scored) as Runs\_scored,

count(w.Player\_Out) as Wickets\_taken

from team t join player\_match p on p.Team\_Id = t.Team\_Id

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

join season s on s.Season\_Id = m.Season\_Id

left join ball\_by\_ball b on b.Match\_Id = p.Match\_Id and b.Striker = p.Player\_Id

left join ball\_by\_ball b1 on b1.Match\_Id = p.Match\_Id and b1.Bowler = p.Player\_Id

left join wicket\_taken w on w.Match\_Id = b1.Match\_Id and w.Over\_Id = b1.Over\_Id

and w.Ball\_Id = b1.Ball\_Id and w.Innings\_No = b1.Innings\_No

group by t.Team\_Name, s.Season\_Year)

select

t1.Team\_Name,

t1.Season\_Year as Previous\_year,

t2.Season\_Year as Current\_Year,

t1.Runs\_scored as Previous\_year\_runs,

t2.Runs\_scored as Current\_year\_runs,

t1.Wickets\_taken as Previous\_year\_wickets,

t2.Wickets\_taken as Current\_year\_Wickets,

case

when t2.Runs\_scored > t1.Runs\_scored and t2.Wickets\_taken > t1.Wickets\_taken then "Overall Improved"

when t2.Runs\_scored = t1.Runs\_scored and t2.Wickets\_taken = t1.Wickets\_taken then "Same"

when t2.Runs\_scored > t1.Runs\_scored and t2.Wickets\_taken <=t1.Wickets\_taken then "Batting Improved"

when t2.Runs\_scored < t1.Runs\_scored and t2.Wickets\_taken >= t1.Wickets\_taken then "Wickets Improved"

else "Worse"

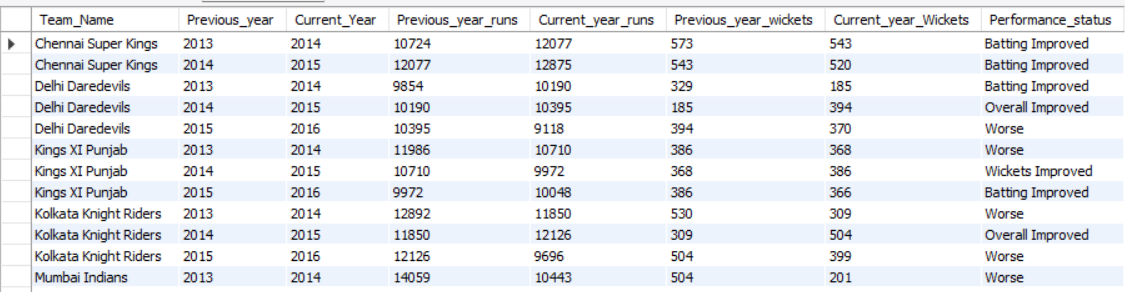
end as Performance\_status

from team\_performance t1 join team\_performance t2

on t1.Team\_Name = t2.Team\_Name and t1.Season\_Year = t2.Season\_Year - 1

order by t1.Team\_Name, t1.Season\_Year;

**Output:**



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

**Ans:**

1. **Boundary Frequency Percentage**

select

p.Player\_Name,

(sum(case when b.Runs\_Scored in (4,6) then 1 else 0 end)\* 100 / count(\*)) as Boundary\_frequency

from player p join ball\_by\_ball b

on p.Player\_Id = b.Striker

group by p.Player\_Name

order by Boundary\_frequency desc;

1. **Death Over Performance**

select

P.Player\_Name as Batsman\_Name,

p1.Player\_Name as Bowler\_Name,

t.Team\_Name as Batting\_team,

t1.Team\_Name as Bowler\_team,

sum(b.Runs\_Scored) as Death\_over\_runs,

count(w.Player\_Out) as Death\_over\_wickets

from ball\_by\_ball b 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

and b.Innings\_No = w.Innings\_No

join team t on t.Team\_Id = b.Team\_Batting join team t1 on t1.Team\_Id = b.Team\_Bowling

join player p on p.Player\_Id = b.Striker join player p1 on p1.Player\_Id = b.Bowler

where b.Over\_Id between 17 and 20

group by p.Player\_Name, t.Team\_Name, p1.Player\_Name, t1.Team\_Name, b.Team\_Batting, b.Team\_Bowling

order by Death\_over\_runs desc, Death\_over\_wickets desc;

1. **Powerplay performance**

select

P.Player\_Name as Batsman\_Name,

p1.Player\_Name as Bowler\_Name,

t.Team\_Name as Batting\_team,

t1.Team\_Name as Bowler\_team,

sum(b.Runs\_Scored) as Powerplay\_runs,

count(w.Player\_Out) as Powerplay\_wickets

from ball\_by\_ball b 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

and b.Innings\_No = w.Innings\_No

join team t on t.Team\_Id = b.Team\_Batting join team t1 on t1.Team\_Id = b.Team\_Bowling

join player p on p.Player\_Id = b.Striker join player p1 on p1.Player\_Id = b.Bowler

where b.Over\_Id between 1 and 6

group by p.Player\_Name, t.Team\_Name, p1.Player\_Name, t1.Team\_Name, b.Team\_Batting, b.Team\_Bowling

order by Powerplay\_runs desc, Powerplay\_wickets desc;

1. **Average Contribution by top order**

with first\_three\_batters as (select

distinct Match\_Id,

Team\_Batting,

Striker

from (select

Match\_Id,

Team\_Batting,

Striker,

row\_number() over(partition by Match\_Id, Team\_Batting order by Over\_Id, Ball\_Id) as entry\_order

from ball\_by\_ball) rs where entry\_order <=3),

top\_order\_stats as (select

m.Match\_Id,

t.Team\_Name,

sum(case when f.Striker is not null then b.Runs\_Scored else 0 end) as top\_order\_runs,

sum(b.Runs\_Scored) as match\_total\_runs

from matches m join ball\_by\_ball b on m.Match\_Id = b.Match\_Id

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

left join first\_three\_batters f on f.Match\_Id = b.Match\_Id

and f.Team\_Batting = b.Team\_Batting and f.Striker = b.Striker

group by m.Match\_Id, t.Team\_Name)

select

Team\_Name,

round(avg((top\_order\_runs \* 1.0 / match\_total\_runs)\* 100),2) as Avg\_Top\_Order\_Contribution

from top\_order\_stats

group by Team\_Name

order by Avg\_Top\_Order\_Contribution desc;

1. **Powerplay Strike Rate**

select

p.Player\_Name,

round(sum(b.Runs\_Scored)\*100/count(\*),2) as Strike\_Rate

from ball\_by\_ball b join player p

on p.Player\_Id = b.Striker

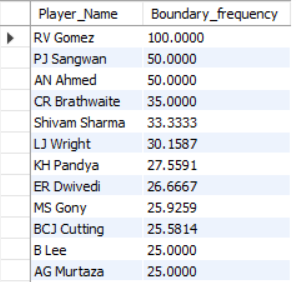
where b.Over\_Id between 1 and 6

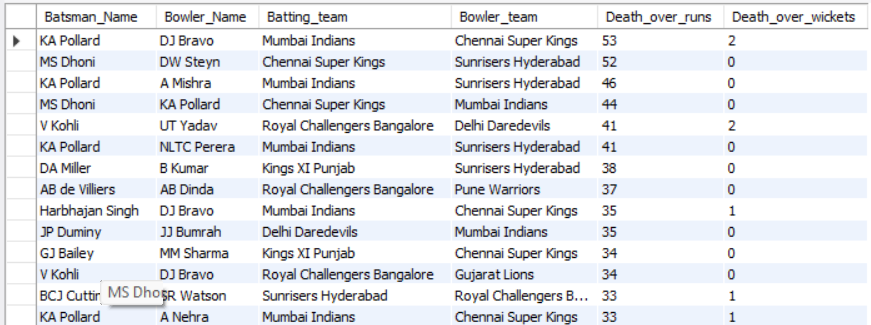
group by p.Player\_Name

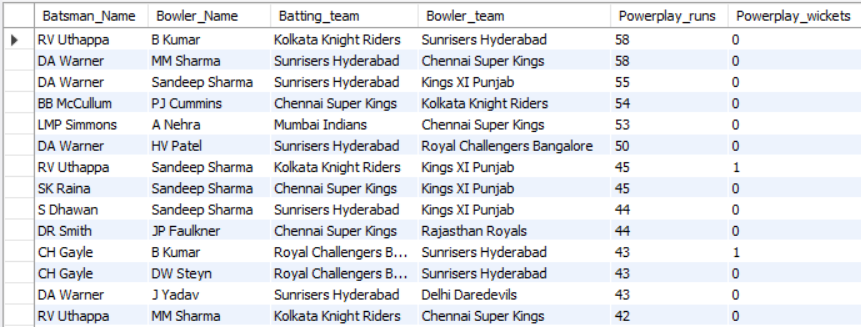
having count(\*) > 20

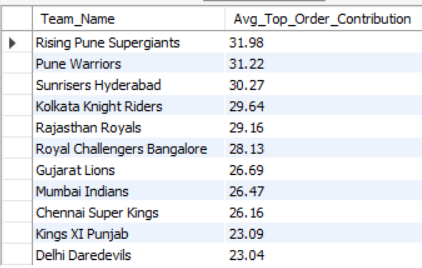
order by Strike\_Rate desc;

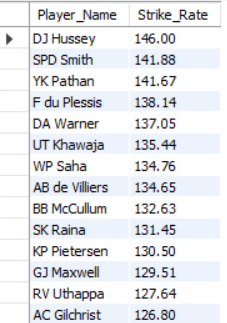
**Output:**











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

**Ans:**

with wickets\_per\_venue as (

select

p.player\_id,

p.player\_name,

v.venue\_name,

count(wt.player\_out) as total\_wickets,

count(distinct m.match\_id) as total\_matches,

round(count(wt.player\_out) \* 1.0 / count(distinct m.match\_id), 2) as avg\_wickets

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 wt on bb.match\_id = wt.match\_id

and bb.over\_id = wt.over\_id

and bb.ball\_id = wt.ball\_id

join venue v on m.venue\_id = v.venue\_id

group by p.player\_id, p.player\_name, v.venue\_name

)

select

player\_id,

player\_name,

venue\_name,

total\_wickets,

total\_matches,

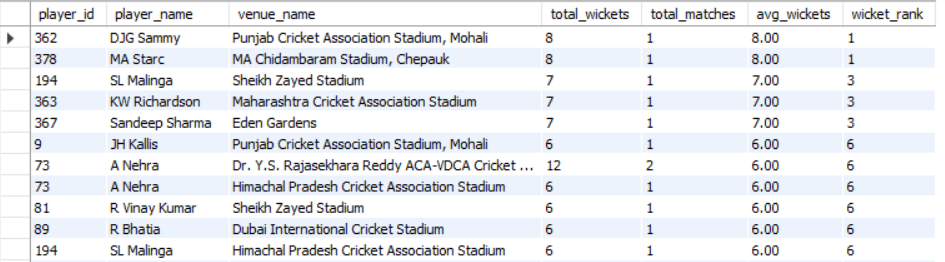
avg\_wickets,

rank() over (order by avg\_wickets desc) as wicket\_rank

from wickets\_per\_venue

order by wicket\_rank;

**Output:**



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

**Ans:**

with player\_season\_performance as (select

p.Player\_Name,

s.Season\_Year,

sum(case when b.Striker = p.Player\_Id then b.Runs\_Scored else 0 end) as total\_runs,

count(distinct case when w.Player\_Out = p.Player\_Id then b.Match\_Id else 0 end) as total\_wickets

from player p left join ball\_by\_ball b

on p.Player\_Id = b.Striker or p.Player\_Id = b.Bowler

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

left join season s on s.Season\_Id = m.Season\_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

and b.Innings\_No = w.Innings\_No

group by p.Player\_Name, s.Season\_Year)

select

Player\_Name,

count(distinct Season\_Year) as Seasons\_Played,

round(avg(total\_runs),2) as avg\_runs\_per\_season,

round(avg(total\_wickets),2) as avg\_wickets\_per\_season

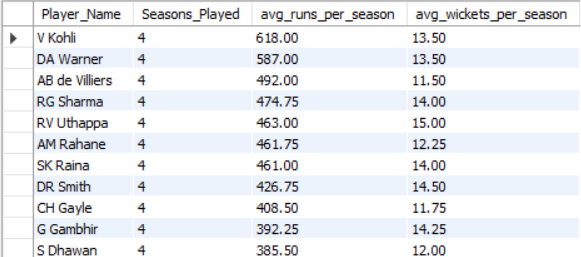
from player\_season\_performance

group by Player\_Name

having count(distinct Season\_Year) > 2

order by avg\_runs\_per\_season desc, avg\_wickets\_per\_season desc;

**Output:**



**Chart:**

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

**Ans:**

with player\_venue\_performance as (select

p.Player\_Name,

v.Venue\_Name,

sum(case when b.Striker = p.Player\_Id then b.Runs\_Scored else 0 end) as total\_runs,

count(distinct case when w.Player\_Out = p.Player\_Id then b.Match\_Id else 0 end) as total\_wickets

from player p left join ball\_by\_ball b

on p.Player\_Id = b.Striker or p.Player\_Id = b.Bowler

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

left join venue v on v.Venue\_Id = m.Venue\_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

and b.Innings\_No = w.Innings\_No

group by p.Player\_Name, v.Venue\_Name)

select

Player\_Name,

Venue\_Name,

round(avg(total\_runs),2) as avg\_runs,

round(avg(total\_wickets),2) as avg\_wickets

from player\_venue\_performance

group by Player\_Name, Venue\_Name

having round(avg(total\_runs),2) > 30

and round(avg(total\_wickets),2) > 5

order by avg\_runs desc, avg\_wickets desc;

**Output:**



**Chart:**

**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?**

**Tables Used:**

Venue, matches, team, toss\_decision

**Approach:**

* + Check if toss winner = match winner
  + Group by venue and toss decision
  + Calculate total matches, matches won after toss, and win percentage

**Insights**

* + Toss impacts match results, but the effect varies by venue
  + Some venues heavily favor batting/fielding first
  + Toss may give a significant edge in specific conditions (e.g., dew at night)

**Recommendations**

* + Use historical data to guide toss decisions
  + Tailor strategies based on venue-specific toss trends
  + Further analyze by match format or season for deeper insights

**Query:**

select

v.Venue\_Name,

t.Toss\_Name as toss\_decision,

count(\*) as Total\_matches,

sum(case when m.Toss\_Winner = m.Match\_Winner then 1 else 0 end) as matches\_won\_after\_toss,

round(sum(case when m.Toss\_Winner = m.Match\_Winner then 1 else 0 end) \*100 / count(\*),2) as win\_percentage

from venue v join matches m on v.Venue\_Id = m.Venue\_Id

join team tm on tm.Team\_Id = m.Toss\_Winner

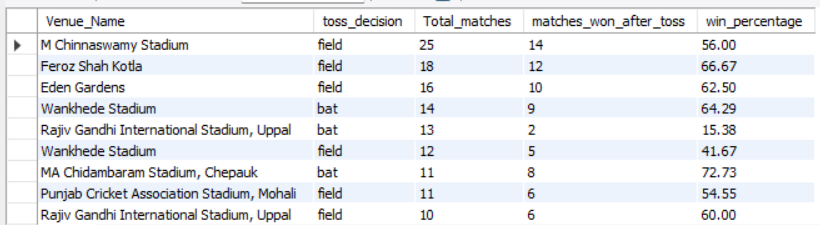
join toss\_decision t on t.Toss\_Id = m.Toss\_Decide

where m.Match\_Winner is not null

group by v.Venue\_Name, t.Toss\_Name

order by Total\_matches desc, win\_percentage desc;

**Output:**



**Chart:**

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

**Tables Used:**

Player, ball\_by\_ball, wicket\_taken, matches, season

**Approach:**

* Create a CTE to calculate:
  + Total runs scored per player per season
  + Total wickets taken per player per season
* Group by Player\_Name and Season\_Year
* Calculate:
  + Number of seasons played
  + Average runs per season
  + Average wickets per season
* Filter players who played more than 2 seasons
* Order results by average runs and then average wickets

**Insights:**

* Identifies **consistent top-performing players** across multiple seasons
* Highlights **batting all-rounders** and dual contributors
* Helps filter out one-season wonders and prioritize long-term performance

**Recommendations:**

* Select **players with high average runs and decent wickets** for formats like T20/ODI
* Use this analysis to build a **balanced team of specialists and all-rounders**
* Add more metrics like strike rate, economy rate, and recent season form for finer selection
* Customize based on **format-specific needs** (e.g., explosive batsmen for T20)

**Query:**

with player\_season\_performance as (select

p.Player\_Name,

s.Season\_Year,

sum(case when b.Striker = p.Player\_Id then b.Runs\_Scored end) as total\_runs,

count(distinct case when w.Player\_Out = p.Player\_Id then b.Match\_Id else 0 end) as total\_wickets

from player p left join ball\_by\_ball b

on p.Player\_Id = b.Striker or p.Player\_Id = b.Bowler

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

left join season s on s.Season\_Id = m.Season\_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

and b.Innings\_No = w.Innings\_No

group by p.Player\_Name, s.Season\_Year)

select

Player\_Name,

count(distinct Season\_Year) as Seasons\_Played,

round(avg(total\_runs),2) as avg\_runs\_per\_season,

round(avg(total\_wickets),2) as avg\_wickets\_per\_season

from player\_season\_performance

group by Player\_Name

having count(distinct Season\_Year) > 2

order by avg\_runs\_per\_season desc, avg\_wickets\_per\_season desc;

**Output:**



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

**Tables Used:**

player, player\_match, matches, ball\_by\_ball, wicket\_taken

**Approach:**

* Use a **CTE (player\_stats)** to gather comprehensive player statistics
* For each player:
  + Count **matches played**
  + Sum **total runs** scored and **balls faced** (batting)
  + Count **balls bowled**, **runs conceded**, and **wickets taken** (bowling)
* Calculate:
  + **Strike rate** = (Total Runs × 100) / Balls Faced
  + **Economy rate** = (Runs Conceded × 100) / Balls Bowled
* Order results to highlight **top-performing players**

**Insights:**

* Players with **high strike rates** and **low economy rates** are ideal for shorter formats (T20, ODI)
* High **run scorers** and **wicket-takers** consistently contribute to match outcomes
* **All-rounders** (good in both batting & bowling) can be easily identified from this data

**Recommendations:**

* + Pick batters with high **strike rate** and **total runs**.
  + Choose bowlers with low **economy rate** and high **wicket count**.
  + Prefer players with **consistent performance** over multiple matches.
  + Include **all-rounders** for team balance.
  + Prioritize players with good **match experience**.

**Query:**

with player\_stats as (select

p.Player\_Name,

count(distinct m.Match\_Id) as matches\_played,

sum(case when b.Striker = p.Player\_Id then b.Runs\_Scored else 0 end) as total\_runs,

count(case when b.Striker = p.Player\_Id then 1 end) as balls\_faced,

count(case when b.Bowler = p.Player\_Id then 1 end) as balls\_bowled,

sum(case when b.Bowler = p.Player\_Id then b.Runs\_Scored else 0 end) as runs\_conceded,

count(case when wt.Player\_Out = p.Player\_Id then 1 end) as wickets\_taken

from player p join player\_match pm

on p.Player\_Id = pm.Player\_Id

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

left join ball\_by\_ball b on b.Match\_Id = m.Match\_Id

and (b.Striker = p.Player\_Id or b.Bowler = p.Player\_Id)

left join wicket\_taken wt on wt.Match\_Id = b.Match\_Id

and wt.Over\_Id = b.Over\_Id

and wt.Ball\_Id = b.Ball\_Id

and wt.Innings\_No = b.Innings\_No

group by p.Player\_Name)

select

Player\_Name,

matches\_played,

total\_runs,

balls\_faced,

round(case when balls\_faced > 0 then (total\_runs \* 100) / balls\_faced else 0 end, 2) as strike\_rate,

wickets\_taken,

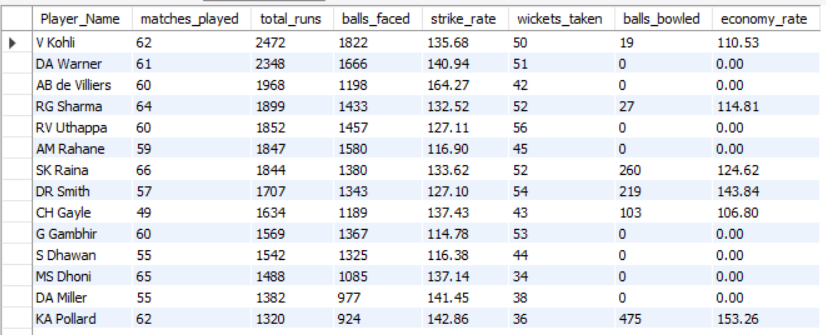
balls\_bowled,

round(case when balls\_bowled > 0 then (runs\_conceded \* 100) / balls\_bowled else 0 end, 2) as economy\_rate

from player\_stats

order by total\_runs desc, wickets\_taken desc;

**Output:**

****

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)**

**Tables used:**

player, player\_match, matches, ball\_by\_ball, wicket\_taken

**Approach:**

* Join all relevant tables to gather batting and bowling actions by player.
* Calculate per player:
  + total\_runs, balls\_faced, strike\_rate
  + wickets\_taken, balls\_bowled, runs\_conceded, economy\_rate
* Filter for:
  + total\_runs > 500
  + wickets\_taken > 20
* Sort by highest runs and wickets to get top all-rounders.

**Insights:**

* Identifies **true all-rounders** who contribute significantly with both bat and ball.
* Players with high **strike rate + low economy rate** are key assets in T20/ODI.
* Only a few players meet both thresholds, highlighting rarity and value.

**Recommendations:**

* Focus on players with **balanced performance**: >500 runs & >20 wickets.
* Select all-rounders to **increase squad depth** and adapt to match scenarios.
* Use these metrics for **auction picks, team drafts, or match strategy**.
* Visualize using a **scatter plot** of Runs vs Wickets to spot top contributors easily.

**Query:**

with player\_stats as (select

p.Player\_Name,

count(distinct m.Match\_Id) as matches\_played,

sum(case when b.Striker = p.Player\_Id then b.Runs\_Scored else 0 end) as total\_runs,

count(case when b.Striker = p.Player\_Id then 1 end) as balls\_faced,

count(case when b.Bowler = p.Player\_Id then 1 end) as balls\_bowled,

sum(case when b.Bowler = p.Player\_Id then b.Runs\_Scored else 0 end) as runs\_conceded,

count(case when wt.Player\_Out = p.Player\_Id then 1 end) as wickets\_taken

from player p join player\_match pm

on p.Player\_Id = pm.Player\_Id

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

left join ball\_by\_ball b on b.Match\_Id = m.Match\_Id

and (b.Striker = p.Player\_Id or b.Bowler = p.Player\_Id)

left join wicket\_taken wt on wt.Match\_Id = b.Match\_Id

and wt.Over\_Id = b.Over\_Id

and wt.Ball\_Id = b.Ball\_Id

and wt.Innings\_No = b.Innings\_No

group by p.Player\_Name)

select

Player\_Name,

matches\_played,

total\_runs,

balls\_faced,

round(case when balls\_faced > 0 then (total\_runs \* 100) / balls\_faced else 0 end, 2) as strike\_rate,

wickets\_taken,

balls\_bowled,

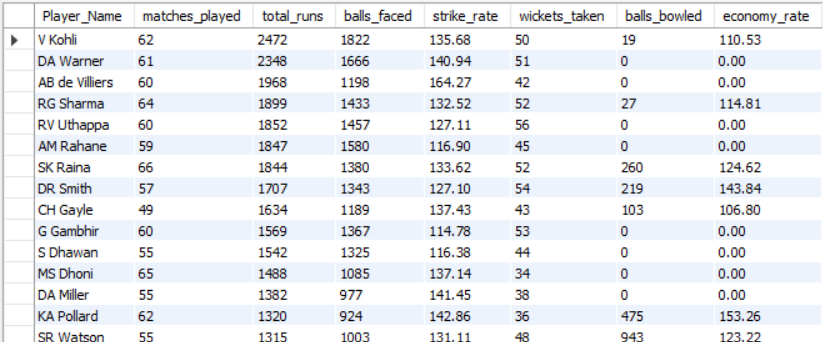
round(case when balls\_bowled > 0 then (runs\_conceded \* 100) / balls\_bowled else 0 end, 2) as economy\_rate

from player\_stats

where total\_runs > 500 and wickets\_taken > 20

order by total\_runs desc, wickets\_taken desc;

**Output:**



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

**Tables Used:**

player\_match, matches, team, player

**Approach:**

* Join **match** and **player participation** data.
* Count **total matches** each player has played for a **team**.
* Count **wins** where the player's team was the **match winner**.
* Calculate **win percentage** when the player was in the **playing XI**.
* Filter players with **more than 10 matches**.
* Order by **highest win percentage**.

**Insights:**

* Players with **high match win percentages** likely **boost team morale and performance**.
* A consistently **high win rate** suggests a **positive impact** on outcomes.
* Such players may act as **match-winners** or **key motivators**.

**Recommendations:**

* Retain players with **high win percentages** in the **core squad**.
* Use them in **crucial matches** to leverage their **influence**.
* Analyze their **roles, leadership, or field positions** contributing to wins.
* Consider for **captaincy** or **mentorship roles**.

**Query:**

select

p.Player\_Name,

t.Team\_Name,

count(distinct pm.Match\_Id) as matches\_played,

sum(case when m.Match\_Winner = pm.Team\_Id then 1 else 0 end) as matches\_won,

round(100 \* sum(case when m.Match\_Winner = pm.Team\_Id then 1 else 0 end) / count(distinct pm.Match\_Id), 2) as matches\_won\_percentage

from player\_match pm join matches m

on pm.Match\_Id = m.Match\_Id

join team t on t.Team\_Id = pm.Team\_Id

join player p on p.Player\_Id = pm.Player\_Id

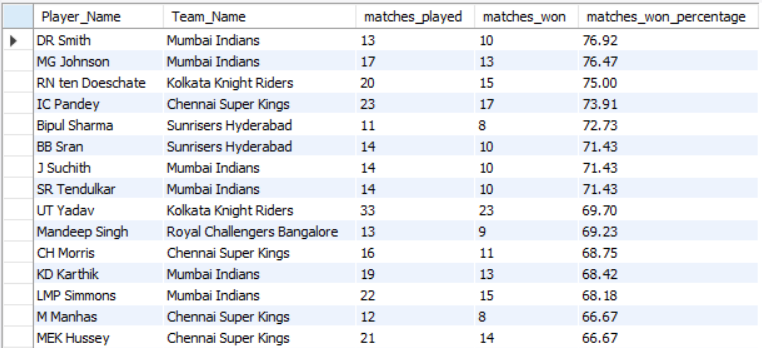
where m.Match\_Winner is not null

group by p.Player\_Name, t.Team\_Name

having count(distinct pm.Match\_Id) > 10

order by matches\_won\_percentage desc;

**Output:**



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

**Invest in Versatile Players**

* Target players who have **500+ runs and 20+ wickets** over seasons.
* All-rounders improve team balance and flexibility.

 **Retain Players with High Impact**

* Look at players with **70%+ win percentage** when in the team.
* Such players positively influence match outcomes and team morale.

 **Strengthen Middle Order & Death Bowling**

* Consistent failure in these areas has cost matches.
* Invest in **finishers** and **death-over specialists**.

 **Optimize Toss-Based Strategy**

* Analyze past results by **venue and toss decision**.
* Pick players who can adapt to **batting/fielding first conditions**.

 **Focus on Youth + Experience Mix**

* Blend of young Indian talent and seasoned international players ensures **depth and adaptability**.

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

**Factors Contributing to High-Scoring Matches:**

**1. Home Ground Influence**

* **Approach:** Analyze average team scores at each venue, especially for home teams.
* **Insight:** Smaller grounds and flat pitches (e.g., Chinnaswamy, Wankhede) consistently produce high scores; home teams perform better due to pitch familiarity.
* **Recommendation:** Build squads tailored to home conditions and leverage pitch behavior when planning strategies and team selection.

**2. Toss Decision Impact**

* **Approach:** Compare outcomes based on toss decisions (batting vs. fielding first), considering time of day and dew factor.
* **Insight:** Teams prefer chasing in night matches due to dew aiding batting; batting first works better on dry or turning pitches.
* **Recommendation:** Make toss decisions based on pitch reports and weather conditions; prepare for both scenarios with adaptable strategies.

**3. Powerplay Performance (Overs 1–6)**

* **Approach:** Evaluate run rate and wicket loss during Powerplay across teams.
* **Insight:** A quick start (50+ runs) often leads to high totals and sets the tone for the innings.
* **Recommendation:** Use aggressive openers to exploit fielding restrictions and gain early momentum.

**4. Death Over Performance (Overs 16–20)**

* **Approach:** Measure runs scored and economy rates in the final 5 overs.
* **Insight:** The death overs contribute significantly (25–30%) to the final score; a strong finish can swing the match.
* **Recommendation:** Train explosive finishers and use experienced bowlers with death-over skills (yorkers, slower balls, etc.).

**Query:**

**1. Home-ground influence**

with player\_venue\_performance as (select

p.Player\_Name,

v.Venue\_Name,

sum(case when b.Striker = p.Player\_Id then b.Runs\_Scored else 0 end) as total\_runs,

count(distinct case when w.Player\_Out = p.Player\_Id then b.Match\_Id else 0 end) as total\_wickets

from player p left join ball\_by\_ball b

on p.Player\_Id = b.Striker or p.Player\_Id = b.Bowler

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

left join venue v on v.Venue\_Id = m.Venue\_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

and b.Innings\_No = w.Innings\_No

group by p.Player\_Name, v.Venue\_Name)

select

Player\_Name,

Venue\_Name,

round(avg(total\_runs),2) as avg\_runs,

round(avg(total\_wickets),2) as avg\_wickets

from player\_venue\_performance

group by Player\_Name, Venue\_Name

having round(avg(total\_runs),2) > 30

and round(avg(total\_wickets),2) > 5

order by avg\_runs desc, avg\_wickets desc;

**2. Toss-decision impact**

select

v.Venue\_Name,

t.Toss\_Name as toss\_decision,

count(\*) as Total\_matches,

sum(case when m.Toss\_Winner = m.Match\_Winner then 1 else 0 end) as matches\_won\_after\_toss,

round(sum(case when m.Toss\_Winner = m.Match\_Winner then 1 else 0 end) \*100 / count(\*),2) as win\_percentage

from venue v join matches m on v.Venue\_Id = m.Venue\_Id

join team tm on tm.Team\_Id = m.Toss\_Winner

join toss\_decision t on t.Toss\_Id = m.Toss\_Decide

where m.Match\_Winner is not null

group by v.Venue\_Name, t.Toss\_Name

order by Total\_matches desc, win\_percentage desc;

**3. Death Over Performance**

select

P.Player\_Name as Batsman\_Name,

p1.Player\_Name as Bowler\_Name,

t.Team\_Name as Batting\_team,

t1.Team\_Name as Bowler\_team,

sum(b.Runs\_Scored) as Death\_over\_runs,

count(w.Player\_Out) as Death\_over\_wickets

from ball\_by\_ball b 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

and b.Innings\_No = w.Innings\_No

join team t on t.Team\_Id = b.Team\_Batting join team t1 on t1.Team\_Id = b.Team\_Bowling

join player p on p.Player\_Id = b.Striker join player p1 on p1.Player\_Id = b.Bowler

where b.Over\_Id between 17 and 20

group by p.Player\_Name, t.Team\_Name, p1.Player\_Name, t1.Team\_Name, b.Team\_Batting, b.Team\_Bowling

order by Death\_over\_runs desc, Death\_over\_wickets desc;

**4. Powerplay performance**

select

P.Player\_Name as Batsman\_Name,

p1.Player\_Name as Bowler\_Name,

t.Team\_Name as Batting\_team,

t1.Team\_Name as Bowler\_team,

sum(b.Runs\_Scored) as Powerplay\_runs,

count(w.Player\_Out) as Powerplay\_wickets

from ball\_by\_ball b 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

and b.Innings\_No = w.Innings\_No

join team t on t.Team\_Id = b.Team\_Batting join team t1 on t1.Team\_Id = b.Team\_Bowling

join player p on p.Player\_Id = b.Striker join player p1 on p1.Player\_Id = b.Bowler

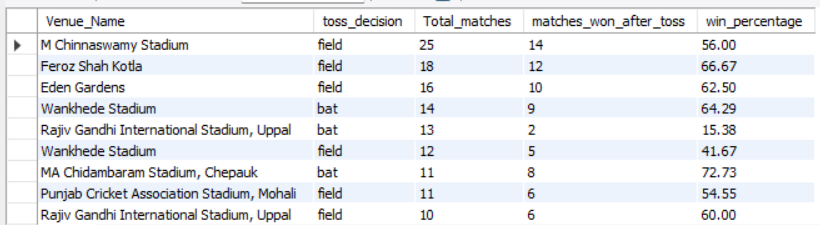
where b.Over\_Id between 1 and 6

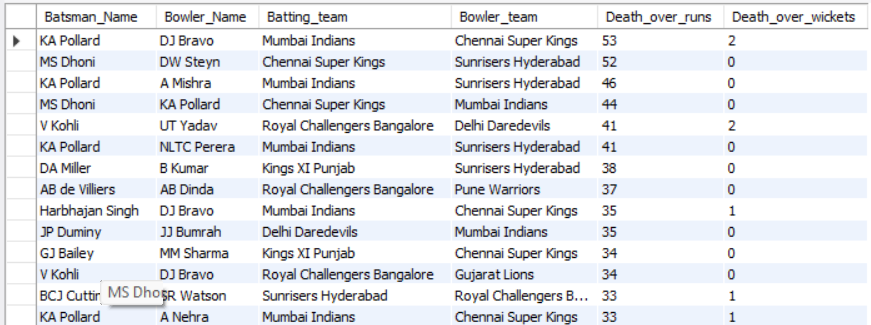
group by p.Player\_Name, t.Team\_Name, p1.Player\_Name, t1.Team\_Name, b.Team\_Batting, b.Team\_Bowling

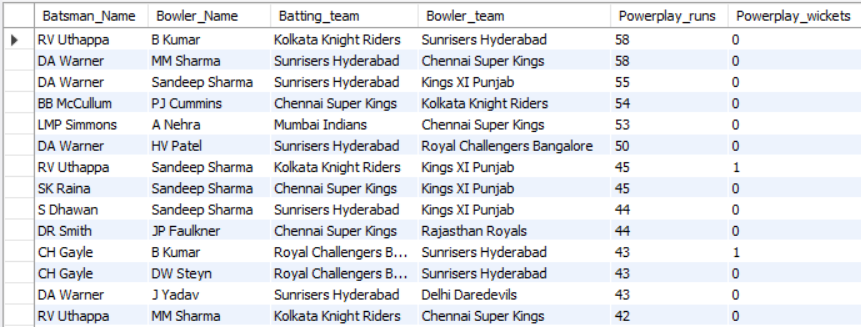
order by Powerplay\_runs desc, Powerplay\_wickets desc;

**Output:**









**Visuals:**

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

**Tables used:**

matches, team, venue

**Approach:**

1. A **Common Table Expression (CTE)** home\_matches is created to extract all matches and associate them with the team playing at that venue.
2. It **joins matches, team, and venue** to get each match’s venue and both teams.
3. Filters records to count each team’s **total matches played at each venue**.
4. Then, calculates:
   * total\_home\_matches – total matches a team played at a venue.
   * home\_matches\_won – matches where the playing team was also the winner.
   * home\_win\_percentage – success rate at the venue.

**Insights:**

* Some teams (like **CSK**) perform better at home due to:
  + Familiar pitch and weather conditions.
  + Supportive home crowd.
  + Tactical confidence in home settings.
* The **home\_win\_percentage** metric clearly shows how dominant or underwhelming a team is at their home ground.

**Recommendations for RCB:**

1. **Build a bowling attack** that suits flat pitches like Chinnaswamy — focus on slower deliveries and yorkers.
2. **Maximize batting firepower**: Prioritize big hitters and strong finishers in team selection.
3. **Win the toss, chase more often** at home where conditions favor high-scoring chases.
4. **Invest in data analytics** for home-game-specific strategies.
5. **Boost fan engagement** — use the crowd as a 12th man to create psychological pressure on opponents.

**Query:**

with home\_matches AS (select

m.match\_id,

case

when m.team\_1 = t.team\_id then m.team\_1

when m.team\_2 = t.team\_id then m.team\_2

end as team\_id, team\_name, m.match\_winner, v.venue\_id

from matches m join team t on (m.team\_1 = t.team\_id or m.team\_2 = t.team\_id)

join venue v on m.venue\_id = v.venue\_id

where (m.team\_1 = t.team\_id or m.team\_2 = t.team\_id)

and m.venue\_id = v.venue\_id)

select

team\_id,

team\_name,

count(\*) as total\_home\_matches,

sum(case when match\_winner = team\_id then 1 else 0 end) as home\_matches\_won,

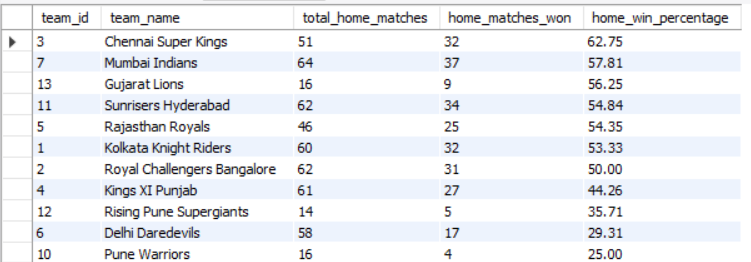
round(sum(case when match\_winner = team\_id then 1 else 0 end) \* 100.0 / COUNT(\*), 2) as home\_win\_percentage

from home\_matches

group by team\_id, team\_name

order by home\_win\_percentage desc;

**Output:**



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.**

**Tables used:**

Matches, team, season

**Approach:**

* Filter matches involving Royal Challengers Bangalore (RCB) by joining the team and matches tables.
* Aggregate match data season-wise to calculate total matches played, matches won, and matches lost by RCB.
* Calculate the win percentage for each season.
* Join with the season table to get readable season year details.
* Order results by season year to analyze performance trends over time.

**Insights:**

* **Inconsistent Win Percentage:** RCB’s win percentage varies significantly season-to-season, indicating inconsistency.
* **Matches Won vs Lost Gap:** The difference between matches won and lost does not show a steady upward trend.
* **No Trophies Despite High Talent:** Even in seasons with decent win percentages, RCB has not won the trophy.
* **Possible Clutch Performance Issues:** Lack of converting good performances into tournament wins could indicate problems in knockout stages or pressure situations.

**Potential Reasons for Not Winning a Trophy:**

1. **Inconsistent Team Performance:**
   * Fluctuating win percentages show that team performance isn't stable.
2. **Key Player Injuries:**
   * Star players might have missed critical matches reducing overall team strength.
3. **Pressure in Knockout Matches:**
   * Possible struggles in high-pressure knockout games impacting chances of winning the trophy.
4. **Lack of Balanced Team Composition:**
   * May lack all-rounders or depth in either batting or bowling during crucial moments.
5. **Strategic Decisions:**
   * Toss decisions, team selection, or game strategies possibly not optimized for match conditions.
6. **Home Ground Advantage:**
   * Underutilization of home ground advantage could impact match wins.

**Recommendations for RCB:**

* **Focus on Consistency:** Build a core team that performs steadily across all matches.
* **Strengthen Middle Order & Bowling Depth:** To improve balance and handle pressure situations better.
* **Analyze Knockout Matches:** Identify key weaknesses in high-stake games and strategize accordingly.
* **Use Data-Driven Strategies:** Leverage analytics for player fitness, pitch conditions, and opposition strengths.
* **Capitalize on Home Advantage:** Maximize performance in home venue matches.

**Query:**

with rcb\_performance as (

select

m.season\_id as season\_id,

count(m.match\_id) as matches\_played,

sum(case when m.match\_winner = t.team\_id then 1 else 0 end) as matches\_won,

sum(case when m.match\_winner != t.team\_id then 1 else 0 end) as matches\_lost,

(sum(case when m.match\_winner = t.team\_id then 1 else 0 end) \* 100.0) / count(m.match\_id) as win\_percentage

from matches m

inner join team t on t.team\_id = m.team\_1 or t.team\_id = m.team\_2

where t.team\_name = 'royal challengers bangalore'

group by m.season\_id

)

select

s.season\_year,

rp.matches\_played,

rp.matches\_won,

rp.matches\_lost,

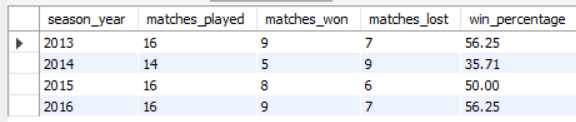
round(rp.win\_percentage, 2) as win\_percentage

from rcb\_performance rp

inner join season s on rp.season\_id = s.season\_id

order by s.season\_year;

**Output:**



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

**If no predefined questions were provided, I would approach the IPL dataset as follows:**

1. **Understand the Dataset:**
   * Review available tables and columns (players, matches, teams, venues, performance stats).
   * Identify key metrics such as wins, losses, player statistics (batting and bowling), and venue details.
2. **Set Analysis Goals:**
   * Explore team performance trends across seasons (matches played, wins, losses).
   * Identify top-performing players and consistent contributors (batsmen, bowlers, all-rounders).
   * Analyze player performance in high-pressure situations (playoffs, death overs).
   * Assess team strengths and weaknesses (e.g., batting order, bowling at the death).
   * Examine match patterns like impact of toss, batting first vs chasing.
3. **Perform Statistical Analysis:**
   * Match-level: Analyze factors influencing outcomes (toss decision, batting order).
   * Player-level: Evaluate averages, strike rates, economy rates, and all-round contributions.
   * Venue-level: Investigate home-ground advantage effects.
4. **Visualize Insights:**
   * Create charts for season-wise performance, top players, home vs away stats, and toss influence.
   * Use visualization tools like Power BI, Tableau, or Excel.
5. **Provide Recommendations:**
   * Suggest areas for team improvement (e.g., bowling in death overs).
   * Identify players to retain or replace.
   * Advise preparation strategies for pressure matches.

This approach ensures a comprehensive, data-driven understanding of the IPL team’s performance and areas for growth.

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".**

**Tables used:**

team

**Identify the Issue:**

* There is no column named “Opponent\_Team” in the matches table. Query the distinct team names in the team table to find incorrect or inconsistent entries.
* Verify spelling mistakes or naming inconsistencies (e.g., "Delhi\_Capitals" vs "Delhi\_Daredevils").

**Correct the Data:**

* Use an UPDATE SQL statement to replace all occurrences of "Delhi\_Capitals" with "Delhi\_Daredevils".
* This correction ensures data consistency and accuracy for future analyses involving team names.

**Query:**

update team

set Team\_Name = 'delhi\_daredevils'

where Team\_Name = 'delhi\_capitals';

select \* from team;

**Output:**

