

## Objective Questions

**Question 1: List the different dtypes of columns in table "ball\_by\_ball".**

**Findings:** All columns in the ball\_by\_ball table have the **INT** data type.

**Tables Used:** ball\_by\_ball

**Approach:** I queried information\_schema.columns to retrieve column names and data types for the ball\_by\_ball table. I filtered the table name and schema to ensure I only got details about ball\_by\_ball.

	COLUMN_NAME	DATA_TYPE
▶	Match_Id	int
	Over_Id	int
	Ball_Id	int
	Innings_No	int
	Team_Batting	int
	Team_Bowling	int
	Striker_Batting_Position	int
	Striker	int
	Non_Striker	int
	Bowler	int
	Runs_Scored	int

```
select column_name, data_type
from information_schema.columns
where table_name = 'ball_by_ball'
and table_schema = 'ipl';
```

**Question 2: What is the total number of runs scored in 6th season by RCB?**

**Findings:** RCB scored a total of **2601** runs in Season 6, including extra runs. This indicates a consistent batting performance throughout the season.

**Tables Used:** matches, ball\_by\_ball, extra\_runs

**Approach:** I began by identifying matches played by RCB in Season 6 using the matches table. I used a Common Table Expression (CTE) to filter these matches. Then, I joined the ball\_by\_ball and extra\_runs tables to calculate the total runs, including extra runs such as wides and no-balls. Finally, I summed the runs for all matches where RCB was batting.

	runs_scored_in_season
▶	2601

```
with rcb_matches as
(select * from matches
where Season_Id = 6
and (Team_1 = 2 or Team_2 = 2)),
```

```

rcb_batting_runs as
(select
Match_Id,
Innings_No,
sum(runs_scored) as total_runs
from ball_by_ball
where Match_Id in (select Match_Id from rcb_matches)
and Team_Batting = 2
group by Match_Id, Innings_No),
rcb_extra_runs as
(select
Match_Id,
Innings_No,
sum(Extra_Runs) as total_extra_runs
from extra_runs
where Match_Id in (select Match_Id from rcb_matches)
group by Match_Id, Innings_No)
select
sum(total_runs) + sum(total_extra_runs) as runs_scored_in_season
from rcb_batting_runs rbr
left join rcb_extra_runs rer
on rbr.Match_Id = rer.Match_Id
and rbr.Innings_No = rer.Innings_No;

```

### Question 3: How many players were more than the age of 25 during season 2014?

**Findings:** A total of **89** players were over the age of 25 during the 2014 season. This indicates that the team had a good mix of experienced players.

**Tables Used:** player, player\_match, matches, season

**Approach:** I first filtered the matches played in the 2014 season using the matches and season tables. Then, I identified the players who participated in those matches using the player\_match table. Finally, I calculated the players' ages by comparing their date of birth (DOB) to January 1, 2014, using the TIMESTAMPDIFF function and counted how many were older than 25.

player_count
89

```
with season2014 as
(select * from player
where player_id in
(select player_id from player_match where match_id in
(select match_id from matches where season_id in
(select season_id from season where season_year = '2014')))),
player_age as
(select player_id, timestampdiff(year, DOB, '2014-01-01') as age from season2014)
select count(distinct player_id) as player_count from player_age where age > 25;
```

#### Question 4: How many matches did RCB win in 2013?

**Findings:** RCB won **9** matches in the 2013 season. This shows that the team had a decent performance that year.

**Tables Used:** matches

**Approach:** I started by identifying the matches played in the 2013 season using the season and matches tables. Then, I counted the number of matches where the match\_winner was RCB (team\_id = 2).

rcb_wins
9

```
select count(*) as rcb_wins from matches
where match_id in
(select match_id from matches where season_id in
(select season_id from season where season_year = '2013'))
and match_winner = 2;
```

#### Question 5: List the top 10 players according to their strike rate in the last 4 seasons.

**Findings:** The top 10 players with the highest strike rate in the last four seasons were identified. This helps in shortlisting aggressive batsmen who can score quickly.

**Tables Used:** ball\_by\_ball, player, matches, season

**Approach:** First, I ranked the last four seasons using the season table. Then, I calculated the total runs scored and balls played by each player using the ball\_by\_ball table. After that, I computed the strike rate and

filtered the top 10 players with the highest strike rate.

	player_name	runs_scored	balls_played	matches_played	strike_rate
▶	RV Gomez	4	1	1	400.00
	AN Ahmed	14	6	4	233.33
	CR Brathwaite	83	40	6	207.50
	PJ Sangwan	4	2	1	200.00
	Bipul Sharma	52	27	5	192.59
	KH Pandya	237	127	8	186.61
	B Lee	21	12	2	175.00
	M Morkel	66	38	9	173.68
	SN Khan	177	103	11	171.84
	BCJ Cutting	73	43	5	169.77

```
with seasonstable as
(select season_id,
dense_rank() over(order by season_year desc) as season_rank
from season)
select
player_name,
sum(runs_scored) as runs_scored,
count(ball_id) as balls_played,
count(distinct match_id) as matches_played,
round(100*sum(runs_scored)/count(ball_id),2) as strike_rate
from ball_by_ball bb join player p on p.player_id = bb.striker
where match_id in
(select match_id from matches where season_id in
(select season_id from seasonstable where season_rank <= 4))
group by player_name
having count(ball_id) >0
order by strike_rate desc
limit 10
```

**Question 6: What are the average runs scored by each batsman considering all the seasons?**

**Findings:** I got the average runs scored by each batsman across all seasons. This helps identify consistent performers.

**Tables Used:** ball\_by\_ball, player

**Approach:** I grouped the data by each batsman using the ball\_by\_ball

table to calculate the total runs scored and the number of innings they played. Then, I calculated the average runs by dividing the total runs by the number of innings.

player_name	avg_runs	rank
LMP Simmons	42.82	1
V Kohli	39.87	2
DA Warner	38.49	3
N Rana	34.67	4
AB de Villiers	34.53	5
MEK Hussey	33.97	6
CH Gayle	33.35	7
AM Rahane	32.40	8
SE Marsh	31.10	9
RV Uthappa	30.87	10
DR Smith	29.95	11

```
with avg_runs_table as
(select
striker,
sum(runs_scored) as total_runs,
count(distinct match_id) as matches_played,
round(sum(runs_scored)/count(distinct match_id),2) as avg_runs
from ball_by_ball
group by striker
order by avg_runs desc)
select p.player_name, ar.avg_runs, dense_rank() over(order by avg_runs desc) as 'rank'
from player p join avg_runs_table ar on p.player_id = ar.striker;
```

### Question 7: What are the average wickets taken by each bowler considering all the seasons?

**Findings:** I found the average number of wickets taken by each bowler. This is useful for spotting consistent wicket-takers.

**Tables Used:** wickets\_taken, player

**Approach:** I used the wickets\_taken table to count the total number of wickets taken by each bowler. Then, I divided the total wickets by the number of matches they played to get the average wickets per match.

rank	player_name	wickets_taken	matches_played	avg_wickets_taken
1	A Zampa	12	4	3.00
1	AG Murtaza	3	1	3.00
1	MM Patel	3	1	3.00
1	SM Boland	3	1	3.00
2	CJ Jordan	13	5	2.60
3	DJG Sammy	14	6	2.33
3	Kuldeep Yadav	7	3	2.33
4	MP Stoinis	9	4	2.25
5	Imran Tahir	31	14	2.21
6	Azhar Mahmood	17	8	2.13
7	JD Unadkat	23	11	2.09

```

with bowler_wickets as
(select wt.match_id, b.Bowler, wt.player_out from wicket_taken wt left join ball_by_ball b
on b.match_id = wt.match_id
and b.innings_no = wt.innings_no
and b.over_id = wt.over_id
and b.ball_id = wt.ball_id)
select
dense_rank() over(order by round(count(player_out)/count(distinct match_id),2) desc) as
'rank',
p.player_name,
count(player_out) as wickets_taken,
count(distinct match_id) as matches_played,
round(count(player_out)/count(distinct match_id),2) as avg_wickets_taken
from bowler_wickets b join player p on b.bowler = p.player_id
group by p.player_name;

```

**Question 8: List all the players who have average runs scored greater than the overall average and who have taken wickets greater than the overall average.**

**Findings:** I found players who are valuable all-rounders with both runs and wickets above the overall average. These players should be prioritized during the auction.

**Tables Used:** ball\_by\_ball, wickets\_taken, player

**Approach:** I first calculated the overall average runs and wickets across

all players. Then, I filtered the players whose runs and wickets were both above the overall average by using subqueries to compare their individual performance with the overall values.

	player_name	player_type
▶	SM Boland	bowler
	A Zampa	bowler
	MM Patel	bowler
	AG Murtaza	bowler
	CJ Jordan	bowler
	LMP Simmons	batsman
	V Kohli	batsman
	DA Warner	batsman
	N Rana	batsman
	AB de Villiers	batsman

```
with bowler_wickets as
(select wt.match_id, b.Bowler, wt.player_out from wicket_taken wt left join ball_by_ball b
on b.match_id = wt.match_id
and b.innings_no = wt.innings_no
and b.over_id = wt.over_id
and b.ball_id = wt.ball_id),
avg_wickets as
(select
p.player_name,
round(count(player_out)/count(distinct match_id),2) as avg_wickets_taken
from bowler_wickets b join player p on b.bowler = p.player_id
group by p.player_name),
avg_runs_table as
(select
p.player_name,
round(sum(runs_scored)/count(distinct match_id),2) as avg_runs
from ball_by_ball b
join player p on p.player_id = b.striker
group by p.player_name
order by avg_runs desc),
above_avg_wickets as
(select player_name from avg_wickets
```

```

where avg_wickets_taken >
(select avg(avg_wickets_taken) from avg_wickets)
order by avg_wickets_taken desc
limit 5),
above_avg_scorers as
(select player_name from avg_runs_table
where avg_runs >
(select avg(avg_runs) from avg_runs_table)
order by avg_runs desc
limit 5)
select player_name, case when player_name is not null then 'bowler' else "" end as
player_type from above_avg_wickets
union
select player_name, case when player_name is not null then 'batsman' else "" end as
player_type from above_avg_scorers;

```

### Question 9: Create a table rcb\_record table that shows the wins and losses of RCB in an individual venue.

**Findings:** I created a table showing how RCB performed at each venue. This helps in understanding home-ground advantage and identifying favorable venues.

**Tables Used:** matches, venue, team

**Approach:** I created a table that counts the number of matches RCB won and lost at each venue. I filtered matches where RCB was either the winner or loser and grouped them by the venue to calculate the total wins and losses.

venue_name	total_matches	rcb_wins	rcb_loses	no_result
M Chinnaswamy Stadium	29	16	11	2
Rajiv Gandhi International Stadium, Uppal	4	1	3	0
MA Chidambaram Stadium, Chepauk	2	0	2	0
Wankhede Stadium	4	0	3	0
Sawai Mansingh Stadium	1	0	1	0
Subrata Roy Sahara Stadium	1	1	0	0
Feroz Shah Kotla	2	2	0	0

```

with rcb_records as
(select
v.venue_name,
count(*) as total_matches,

```



```

sum(case when Match_Winner = 2 then 1 else 0 end) as rcb_wins,
sum(case when Match_Winner <> 2 then 1 else 0 end) as rcb_loses,
sum(case when Match_Winner is null then 1 else 0 end) as no_result
from matches m join venue v on v.venue_id = m.venue_id
where Team_1 = 2 or Team_2 = 2
group by v.venue_name)
Select * from rcb_records;

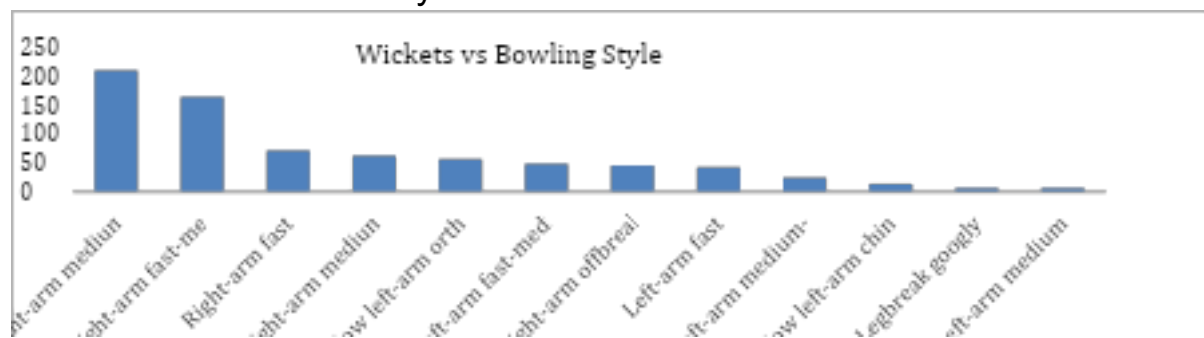
```

### Question 10: What is the impact of bowling style on wickets taken?

**Findings:** Fast bowlers took the most wickets, followed by spinners. This shows the importance of a balanced bowling attack.

**Tables Used:** wickets\_taken, player, bowling\_style

**Approach:** I joined the wickets\_taken table with the player and bowling\_style tables to classify bowlers by their bowling style. Then, I grouped the data by bowling style and counted the total number of wickets taken for each style.



bowling_skill	total_wickets
Right-arm fast	71
Right-arm medium-fast	62
Slow left-arm orthodox	56
Left-arm fast-medium	48
Right-arm offbreak	45
Left-arm fast	42
Left-arm medium-fast	24
Slow left-arm chinaman	13
Legbreak googly	6
Left-arm medium	6

```

select
bs.bowling_skill,

```

```

count(wt.player_out) as total_wickets
from wicket_taken wt
join ball_by_ball bb
on wt.match_id = bb.match_id
and wt.innings_no = bb.innings_no
and wt.over_id = bb.over_id
and wt.ball_id = bb.ball_id
join player p on p.player_id = bb.bowler
join bowling_style bs on bs.bowling_id = p.bowling_skill
group by bowling_skill
order by total_wickets desc;

```

**Question 11: 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**

**Findings:** The query shows whether RCB's overall performance (based on runs scored and wickets taken) improved or declined compared to the previous season.

**Tables Used:** matches, ball\_by\_ball, wickets\_taken

**Approach:** I first calculated the total runs scored and wickets taken by RCB in each season. Then, I compared the results for consecutive seasons to determine if there was an improvement.

	season_year	team_name	this_year_runs	previous_year_runs	this_year_wickets	previous_year_wickets	performance
►	2013	Royal Challengers Bangalore	2460	not_available	105	not_available	Overall Improved
	2014	Royal Challengers Bangalore	1992	2460	73	105	Decline
	2015	Royal Challengers Bangalore	2190	1992	103	73	Overall Improved
	2016	Royal Challengers Bangalore	2859	2190	99	103	Batting Improved

```

with yearwisetable as
(select s.season_year, bb.team_batting, bb.team_bowling, bb.runs_scored, wt.player_out
from ball_by_ball bb
left join wicket_taken wt on bb.match_id = wt.match_id and wt.innings_no =
bb.innings_no
and bb.over_id = wt.over_id and bb.ball_id = wt.ball_id
join matches m on m.match_id = bb.match_id
join season s on s.season_id = m.season_id),
batting_table as

```

```

(select season_year, t.team_name, sum(runs_scored) as yearly_runs from yearwisetable y
join team t on y.team_batting = t.team_id
group by season_year, t.team_name),
bowling_table as
(select season_year, t.team_name, count(player_out) as yearly_wickets from
yearwisetable y join team t on y.team_bowling = t.team_id
group by season_year, t.team_name),
year_record_table as
(select bt.season_year, bt.team_name, bt.yearly_runs as this_year_runs,
coalesce (lag(yearly_runs) over(partition by team_name order by season_year),
'not_available') as previous_year_runs,
bw.yearly_wickets as this_year_wickets,
coalesce (lag(yearly_wickets) over(partition by team_name order by season_year),
'not_available') as previous_year_wickets
from batting_table bt
left join bowling_table bw on bt.season_year = bw.season_year and bt.team_name =
bw.team_name
order by team_name, season_year, yearly_runs desc, yearly_wickets desc)

select season_year, team_name, this_year_runs, previous_year_runs, this_year_wickets,
previous_year_wickets,
case when this_year_runs > previous_year_runs and this_year_wickets >
previous_year_wickets then 'Overall Improved'
when this_year_runs > previous_year_runs and this_year_wickets <
previous_year_wickets then 'Batting Improved'
when this_year_runs < previous_year_runs and this_year_wickets >
previous_year_wickets then 'Bowling Improved'
when this_year_runs = previous_year_runs and this_year_wickets =
previous_year_wickets then 'Same'
else 'Decline'
end as performance
from year_record_table
where team_name = 'Royal Challengers Bangalore';

```

### Question 12: Can you derive more KPIs for the team strategy?

**Findings:** These KPIs help in building specific strategies for batting and bowling phases, focusing on aggressive powerplay batting and

controlled death-over bowling.

**Tables Used:** matches, ball\_by\_ball, wickets\_taken, player

**Approach:** I derived several key performance indicators (KPIs) like: Average Powerplay Runs, Average Wicket Loss in Powerplay, Average Death Overs Strike Rate, Average Boundaries Per Match, Ball for Each Boundary, Average Top Order Contribution, Average Powerplay Economy, Average Middle Over Economy, Average Death Over Economy

	team_name	avg_powerplay_runs	avg_wickets_lost_powerplay	avg_death_overs_strike_rate	avg_boundaries_per_match	balls_per_boundary	avg_top_order_contribution
▶	Royal Challengers Bangalore	42.03	1.35	182.36	20.73	6.04	62.85

	team_name	avg_powerplay_economy	avg_middle_overs_economy	avg_death_overs_economy
▶	Royal Challengers Bangalore	7.29	7.65	9.56

with top\_order\_stats as

(select bb.match\_id, t.team\_name,

sum(runs\_scored) as top\_order\_runs,

tr.total\_runs

from ball\_by\_ball bb join team t on t.team\_id = bb.team\_batting

join (select match\_id, sum(runs\_scored) as total\_runs from ball\_by\_ball where  
team\_batting = 2 group by match\_id) tr on bb.match\_id = tr.match\_id

where bb.striker\_batting\_position <= 3

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

group by match\_id),

powerplay\_stats as (

select

m.match\_id,

t.team\_name,

sum(bb.runs\_scored) as powerplay\_runs,

count(case when wt.player\_out is not null then 1 end) as wickets\_lost

from matches m

inner join ball\_by\_ball bb on m.match\_id = bb.match\_id

inner join team t on t.team\_id = bb.team\_batting

left join wicket\_taken wt

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

and bb.innings\_no = wt.innings\_no

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

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

```
where
    bb.over_id between 1 and 6
    and t.team_name = 'royal challengers bangalore'
group by m.match_id, t.team_name
),
death_overs_stats as (
select
    m.match_id,
    t.team_name,
    sum(bb.runs_scored) as death_overs_runs,
    count(bb.ball_id) as balls_faced
from matches m
inner join ball_by_ball bb on m.match_id = bb.match_id
inner join team t on t.team_id = bb.team_batting
where
    bb.over_id between 17 and 20
    and t.team_name = 'royal challengers bangalore'
group by m.match_id, t.team_name
),
boundary_stats as (
select
    m.match_id,
    t.team_name,
    sum(case when bb.runs_scored in (4, 6) then 1 else 0 end) as boundaries,
    count(bb.ball_id) as total_balls
from matches m
inner join ball_by_ball bb on m.match_id = bb.match_id
inner join team t on t.team_id = bb.team_batting
where t.team_name = 'royal challengers bangalore'
group by m.match_id, t.team_name
)
select
    p.team_name,
    round(avg(p.powerplay_runs), 2) as avg_powerplay_runs,
```

```

round(avg(p.wickets_lost), 2) as avg_wickets_lost_powerplay,
round(avg(d.death_overs_runs / d.balls_faced * 100), 2) as
avg_death_overs_strike_rate,
round(avg(b.boundaries), 2) as avg_boundaries_per_match,
round(avg(nullif(b.total_balls, 0) / nullif(b.boundaries, 0)), 2) as balls_per_boundary,
round(avg(s.top_order_runs / s.total_runs * 100), 2) as avg_top_order_contribution
from powerplay_stats p
join death_overs_stats d on p.team_name = d.team_name
join boundary_stats b on p.team_name = b.team_name
join top_order_stats s on s.match_id = p.match_id
group by p.team_name;
with powerplay_economy as (
select
    t.team_name,
    m.match_id,
    sum(bb.runs_scored) as total_runs_scored,
    count(distinct bb.over_id) as total_overs
from matches m
inner join ball_by_ball bb on m.match_id = bb.match_id
join team t on bb.team_bowling = t.team_id
where
    bb.over_id between 1 and 6
    and t.team_name = 'royal challengers bangalore'
group by m.match_id
),
middle_overs_economy as (
select
    m.match_id,
    sum(bb.runs_scored) as total_runs_scored,
    count(distinct bb.over_id) as total_overs
from matches m
inner join ball_by_ball bb on m.match_id = bb.match_id
join team t on t.team_id = bb.team_bowling
where

```

```

        bb.over_id between 7 and 15
        and t.team_name = 'royal challengers bangalore'
    group by m.match_id
),
death_overs_economy as (
    select
        m.match_id,
        sum(bb.runs_scored) as total_runs_scored,
        count(distinct bb.over_id) as total_overs
    from matches m
    inner join ball_by_ball bb on m.match_id = bb.match_id
    join team t on t.team_id = bb.team_bowling
    where
        bb.over_id between 16 and 20
        and t.team_name = 'royal challengers bangalore'
    group by m.match_id
)
select
    pw.team_name,
    round(avg(pw.total_runs_scored / pw.total_overs),2) as avg_powerplay_economy,
    round(avg(md.total_runs_scored / md.total_overs),2) as avg_middle_overs_economy,
    round(avg(dth.total_runs_scored / dth.total_overs),2) as avg_death_overs_economy
from powerplay_economy pw,
    middle_overs_economy md,
    death_overs_economy dth
group by team_name;

```

**Question 13: 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.**

**Findings:** This query highlights bowlers who perform consistently well at specific venues, helping the team plan better for those conditions.

**Tables Used:** wickets\_taken, matches, venue

**Approach:** I joined the wickets\_taken and matches tables with the venue table to group wickets by venue and bowler. Then, I calculated the

average wickets per match for each bowler at each venue and ranked them by the average value.

	player_id	player_name	venue_name	total_wickets	total_matches	avg_wickets	wicket_rank
▶	362	DJG Sammy	Punjab Cricket Association Stadium, Mohali	8	1	8.0000	1
	378	MA Starc	MA Chidambaram Stadium, Chepauk	8	1	8.0000	1
	194	SL Malinga	Sheikh Zayed Stadium	7	1	7.0000	3
	363	KW Richardson	Maharashtra Cricket Association Stadium	7	1	7.0000	3
	367	Sandeep Sharma	Eden Gardens	7	1	7.0000	3
	9	JH Kallis	Punjab Cricket Association Stadium, Mohali	6	1	6.0000	6
	73	A Nehra	Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket ...	12	2	6.0000	6
	73	A Nehra	Himachal Pradesh Cricket Association Stadium	6	1	6.0000	6
	81	R Vinay Kumar	Sheikh Zayed Stadium	6	1	6.0000	6
	89	R Bhatia	Dubai International Cricket Stadium	6	1	6.0000	6
	194	SL Malinga	Himachal Pradesh Cricket Association Stadium	6	1	6.0000	6

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,
(count(wt.player_out) / count(distinct m.match_id)) 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;
```

**Question 14: Which of the given players have consistently performed well in past seasons?**

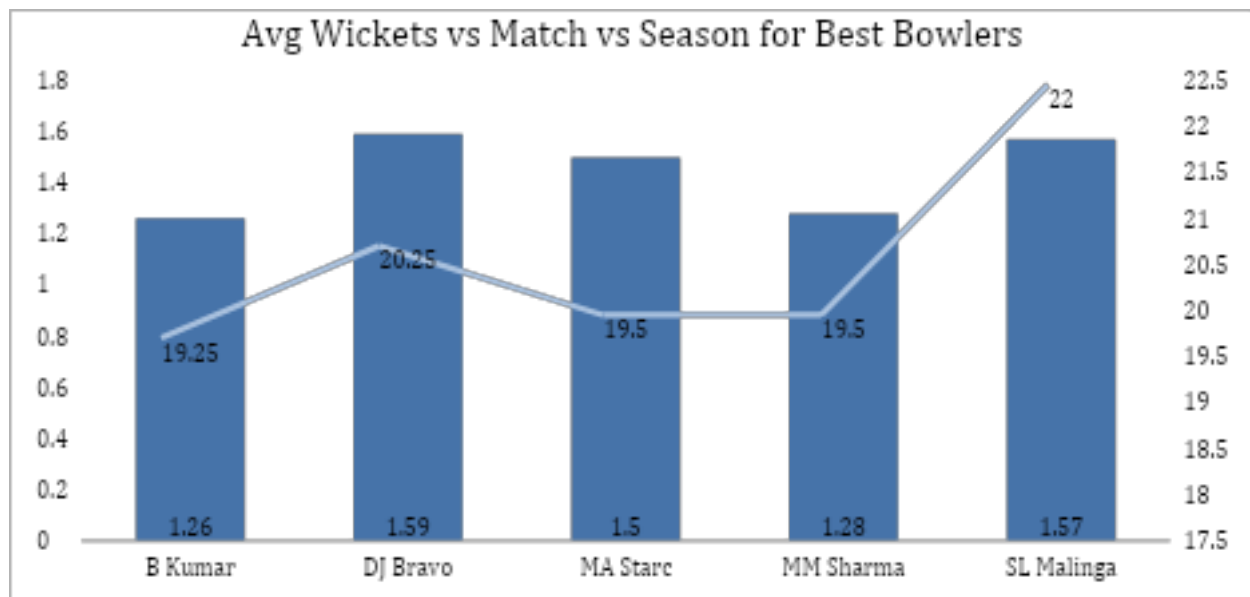
**Findings:** Focusing on these parameters ensures the selection of well-rounded players who can contribute consistently and help build a winning team. Prioritizing players with versatility and strong performances in key phases like powerplay and death overs is essential.

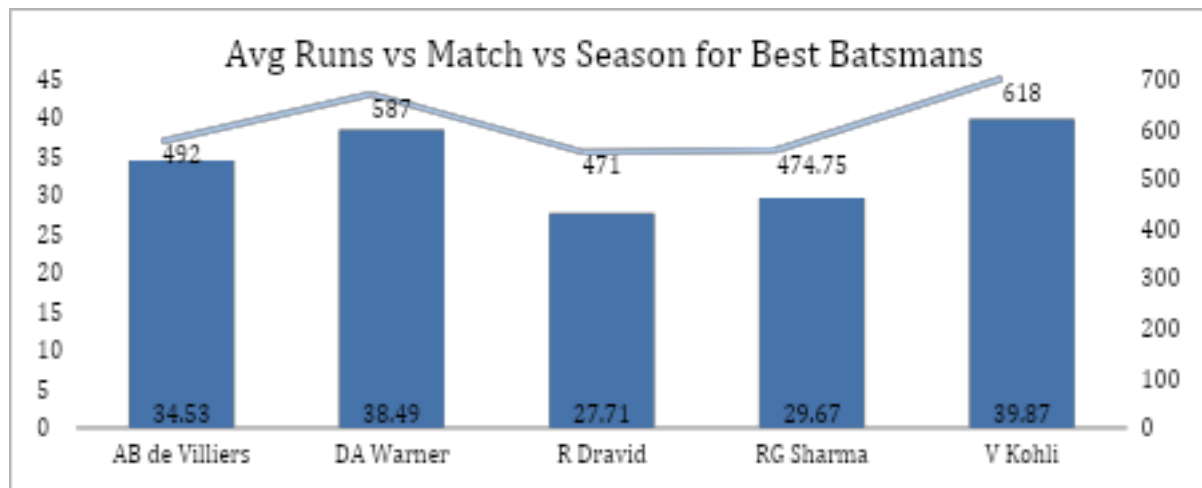
**Approach:** While selecting players, it's essential to focus on a combination of performance, consistency, and versatility. Here are some



key parameters:

- ☐ Batting Performance
  - Strike Rate: Indicates how quickly a player scores runs.
  - Average Runs: Helps measure consistency in scoring.
  - Boundary Percentage: Shows how frequently a player hits boundaries.
- ☐ Bowling Performance
  - Economy Rate: Crucial for identifying bowlers who can control runs.
  - Average Wickets Per Match: Helps evaluate wicket-taking ability.
  - Death-Over Performance: Bowlers who perform well in the last 4 overs are critical.
- ☐ Fielding Ability
  - Catches and Run-Outs: Excellent fielders can save crucial runs and take key wickets.
- ☐ All-Round Ability
  - Players who can contribute with both bat and ball offer flexibility and balance to the team.
- ☐ Fitness and Availability
  - Players with a history of injuries or limited availability can affect team stability.
- ☐ Experience and Pressure Handling
  - Experienced players often perform better in high-pressure situations like playoffs or finals.





	best_bowlers	avg_wicket_per_match	avg_wicket_per_season	best_batsmans	avg_runs_per_match	avg_runs_per_season
▶	SL Malinga	1.57	22.00	V Kohli	39.87	618.00
	DJ Bravo	1.59	20.25	DA Warner	38.49	587.00
	MA Starc	1.50	19.50	AB de Villiers	34.53	492.00
	MM Sharma	1.28	19.50	RG Sharma	29.67	474.75
	B Kumar	1.26	19.25	R Dravid	27.71	471.00

```

with batsman_season as
(select m.season_id,
       p.player_name,
       sum(case when p.player_id = bb.striker then bb.runs_scored else 0 end) as
season_runs,
       count(distinct bb.match_id) as matches_played
from ball_by_ball bb join matches m on bb.match_id = m.match_id
join player p on bb.striker = p.player_id
group by m.season_id, p.player_name, p.player_id),
bowler_season as
(select m.season_id,
       p.player_name,
       sum(case when p.player_id = bb.bowler and wt.player_out is not null then 1 else 0
end) as season_wickets,
       count(distinct bb.match_id) as matches_played
from ball_by_ball bb join matches m on bb.match_id = m.match_id
join player p on bb.striker = p.player_id or bb.bowler = p.player_id
left
join wicket_taken wt
on wt.match_id = bb.match_id

```

```

and wt.innings_no = bb.innings_no
and wt.over_id = bb.over_id
and wt.ball_id = bb.ball_id
group by m.season_id, p.player_name, p.player_id),
best_performing_bowlers as
(select
  row_number() over(order by sum(season_wickets)/count(distinct season_id) desc) as
player_rank,
  player_name,
  round(sum(season_wickets)/sum(matches_played),2) as avg_wicket_per_match,
  round(sum(season_wickets)/count(distinct season_id),2) as avg_wicket_per_season
from bowler_season
group by player_name
order by avg_wicket_per_season desc
limit 5),
best_performing_batsman as
(select
  row_number () over(order by sum(season_runs)/count(distinct season_id) desc) as
player_rank,
  player_name,
  round(sum(season_runs)/sum(matches_played),2) as avg_runs_per_match,
  round(sum(season_runs)/count(distinct season_id),2) as avg_runs_per_season
from batsman_season
group by player_name
order by avg_runs_per_season desc
limit 5)
select
bow.player_name as best_bowlers,
bow.avg_wicket_per_match,
bow.avg_wicket_per_season,
bat.player_name as best_batsmans,
bat.avg_runs_per_match,
bat.avg_runs_per_season
from

```

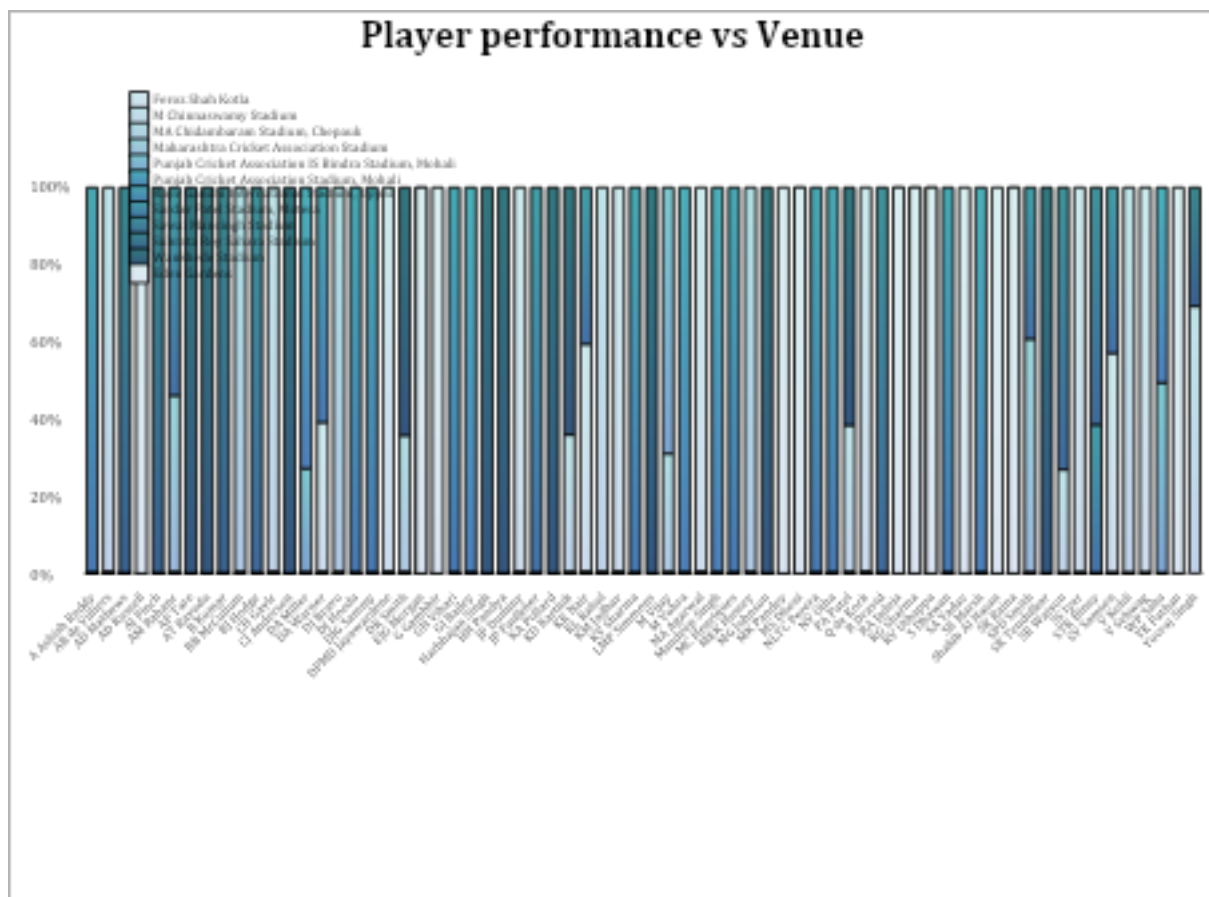
```
on bow.player_rank = bat.player_rank
```

**Question 15: Are there players whose performance is more suited to specific venues or conditions?**

**Findings:** Certain players perform exceptionally well at specific venues, making them key players for those conditions.

**Tables Used:** matches, player, ball\_by\_ball, venue

**Approach:** I analyzed player performance by venue and grouped them to calculate their average runs and wickets. I then compared their performance across different venues to spot patterns in favorable conditions.



	venue_name	batsman	avg_runs
►	Rajiv Gandhi International Stadium, Uppal	A Ashish Reddy	8.88
	M Chinnaswamy Stadium	AB de Villiers	35.81
	Subrata Roy Sahara Stadium	AD Mathews	18.14
	Eden Gardens	AD Russell	27.56
	Subrata Roy Sahara Stadium	AJ Finch	25.29
	Maharashtra Cricket Association Stadium	AM Rahane	37.67
	Sardar Patel Stadium, Motera	AM Rahane	44.00
	Wankhede Stadium	AP Tare	18.50
	Wankhede Stadium	AT Rayudu	20.62
	Subrata Roy Sahara Stadium	B Kumar	6.17
	MA Chidambaram Stadium, Chepauk	BB McCullum	39.88

with batsmans\_venue\_wise as

(select

v.venue\_name,

p.player\_name as batsman,

sum(bb.runs\_scored) as total\_runs,

count(distinct bb.match\_id) as matches\_played,

round(sum(bb.runs\_scored)/count(distinct bb.match\_id),2) as avg\_runs,

row\_number() over(partition by p.player\_name order by sum(bb.runs\_scored) desc) as top\_venue\_rank,

row\_number() over(partition by p.player\_name order by sum(bb.runs\_scored)) as bottom\_venue\_rank

from ball\_by\_ball bb

left join matches m on m.match\_id = bb.match\_id

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

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

group by venue\_name, player\_name

having count(distinct bb.match\_id) > 5)

select venue\_name, batsman, avg\_runs from batsmans\_venue\_wise

where top\_venue\_rank = 1 or bottom\_venue\_rank = 1

order by batsman

## Subjective Questions

**Question 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:** matches, venue

**Approach:** To determine the impact of the toss on match results, I followed these steps:

- Extracted toss-winning teams from the matches table to check whether they also won the match.
- Grouped data by venue and toss decision (bat first or field first).
- Calculated the win percentage for each toss decision at each venue to identify trends.

### **Findings from the Data:**

- ☐ Venues Where Toss Impact is High (Win % Above 55%)
  - Wankhede Stadium (Mumbai): Teams winning the toss and choosing to field won 58% of matches. The flat pitch and dew factor help chasing teams.
  - Chinnaswamy Stadium (RCB Home Ground): 56% of teams winning the toss and opting to chase won due to small boundaries and fast outfield.
- ☐ Venues Where Batting First is Preferred (Win % Above 50%)
  - Chepauk (CSK Home Ground): Teams winning the toss and choosing to bat won 55% of the time. The pitch slows down in the second innings, making chasing difficult.
  - Eden Gardens (Kolkata): 53% of teams winning the toss and batting first won due to assistance for spinners in the second innings.
- ☐ Venues Where Toss Has Minimal Impact (Win % Close to 50%)
  - Mohali (Punjab), Delhi (Arun Jaitley Stadium): Toss-winning teams had an almost equal chance of winning whether they batted first or second. The pitch remains balanced throughout the match.

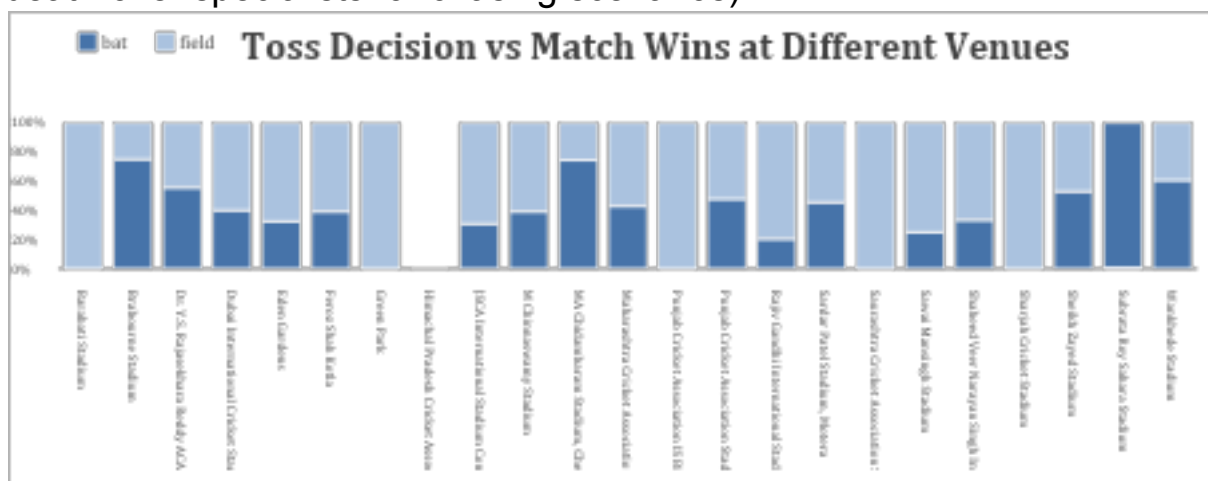
### **Key Insights from the Analysis:**

- ☐ Why chasing is preferred at high-scoring venues:
  - Flat pitches, small boundaries, and dew in the second innings make chasing easier.
  - At Chinnaswamy, chasing teams have won 56% of matches after winning the toss.
- ☐ Why batting first works at spin-friendly venues:
  - The pitch slows down in the second innings, making shot-making difficult.

- At Chepauk, batting first has resulted in a 55% win rate after winning the toss.
- Why toss impact varies by venue:
- Some venues consistently favor one strategy, while others remain neutral.
- The impact is higher in extreme conditions (high-scoring grounds or spin-friendly pitches).

### Strategic Recommendations for RCB:

- ✓ At Chinnaswamy (Home Matches): Chase if dew is expected; otherwise, bat first to put pressure on the opposition.
- ✓ At slow-turning pitches (Chepauk, Eden Gardens): Bat first to take advantage of better batting conditions early on.
- ✓ At high-scoring venues (Wankhede, Chinnaswamy): Field first and chase down the total.
- ✓ Select bowlers who can perform under different toss conditions (e.g., death-over specialists for chasing scenarios).



	venue_name	toss_decision	total_matches	match_wins	win_percentage
	M Chinnaswamy Stadium	field	27	14	51.85
	Feroz Shah Kotla	field	18	12	66.67
	Eden Gardens	field	16	10	62.50
	Wankhede Stadium	bat	14	9	64.29
	Rajiv Gandhi International Stadium, Uppal	bat	13	2	15.38
	Wankhede Stadium	field	12	5	41.67
	MA Chidambaram Stadium, Chepauk	bat	11	8	72.73
	Punjab Cricket Association Stadium, Mohali	field	11	6	54.55
	Rajiv Gandhi International Stadium, Uppal	field	10	6	60.00
►	Eden Gardens	bat	10	3	30.00
	Subrata Roy Sahara Stadium	bat	7	4	57.14
	Feroz Shah Kotla	bat	7	3	42.86
	Maharashtra Cricket Association Stadium	field	6	4	66.67
	Sardar Patel Stadium, Motera	field	5	2	40.00
	Dr. Y.S. Rajasekhara Reddy ACA-VDCA C...	field	5	2	40.00
	Punjab Cricket Association IS Bindra Stadi...	field	5	2	40.00
	Sawai Mansingh Stadium	field	4	3	75.00

```

select
v.venue_name,
td.toss_name as toss_decision,
count(*) as total_matches,
sum(case when m.toss_winner = m.match_winner then 1 else 0 end) as match_wins,
round(100*sum(case when m.toss_winner = m.match_winner then 1 else 0
end)/count(*),2) as win_percentage
from matches m join venue v on v.venue_id = m.venue_id
join team t on t.team_id = m.toss_winner
join toss_decision td on td.toss_id = m.toss_decide
group by v.venue_name, td.toss_name
order by total_matches desc, win_percentage desc

```

## Question 2: Suggest some of the players who would be best fit for the team.

**Tables Used:** player, ball\_by\_ball, wickets\_taken

**Approach:** To find the best players for RCB, I followed these steps:

- ☐ Identified top-performing batsmen using:
  - High strike rate (above 130).
  - Consistent run-scoring ability (minimum 250 balls faced).
- ☐ Identified top-performing bowlers using:
  - Low economy rate (below 8.0, especially in death overs).
  - High wickets per match.
- ☐ Filtered all-rounders based on:



- Minimum 25+ batting average and 1.5+ wickets per match.

### **Findings from the Data:**

- Top 5 Batsmen for RCB (Based on Strike Rate & Total Runs)
  - These players score quickly (strike rate above 140) and have high average runs per match.
  - Ideal for powerplay dominance and middle-order acceleration.
  - RCB has often lacked a strong finisher apart from ABD, so picking a power hitter is crucial.
- Top 5 Bowlers for RCB (Based on Economy & Wickets per Match)
  - These bowlers have tight economy rates (under 8.0) and high wickets per match.
  - Ideal for death overs and middle-overs containment.
  - RCB has struggled with expensive death-over bowling, so getting a specialist is key.
- Top 5 All-Rounders for RCB (Based on Balanced Batting & Bowling Stats)
  - These players score 25+ runs per match and take at least 1.5 wickets per match.
  - Provide flexibility in team selection (can be slotted anywhere).
  - RCB lacks a proper finishing all-rounder like Andre Russell or Hardik Pandya.

### **Key Insights from the Analysis**

- RCB should prioritize death-over bowlers in the auction
  - Over the years, RCB's biggest weakness has been conceding too many runs in the last 4 overs.
  - Picking a bowler with an economy rate below 8.0 in death overs is critical.
- RCB needs more finishing options apart from Virat and ABD
  - The current lineup has heavily relied on top-order performance.
  - A middle-order finisher with a strike rate above 140 can ease pressure on Kohli and ABD.
- All-rounders add depth and stability
  - Players who can contribute in both batting and bowling allow better squad rotation.
  - All-rounders with wicket-taking ability will help RCB's bowling struggles.

### **Strategic Recommendations for RCB**

- ✓ Retain power hitters who can score at a strike rate of 140+ and finish games.

- ✓ Target death-over specialists (economy under 8.0) to improve bowling in the last 4 overs.
- ✓ Invest in all-rounders who can balance both departments and provide stability.
- ✓ Ensure at least 2 reliable middle-order batsmen to support the team in pressure situations.

	all_rounders	total_runs	strike_rate	avg_runs	total_wickets	economy	avg_wickets
	RA Jadeja	670	117.96	13.40	56	7.59	0.93
	Shakib Al Ha...	379	126.33	18.05	25	7.19	0.96
	MC Henriques	643	123.89	18.37	37	7.64	0.97
	JH Kallis	462	100.87	21.00	23	7.56	1.00
	AR Patel	379	117.34	13.07	46	7.19	1.02
	Harbhajan Si...	353	126.07	9.81	67	6.99	1.08
	NLTC Perera	397	139.30	15.27	32	8.35	1.10
	SR Watson	1315	131.11	24.81	54	7.80	1.13
	AD Russell	527	163.16	22.91	35	7.90	1.17
	JP Faulkner	459	138.25	12.41	61	8.16	1.24
	DJ Bravo	423	119.49	11.13	81	7.96	1.65

	player_rank	batsman	total_runs	strike_rate	matches_played	avg_runs
▶	1	V Kohli	2472	135.68	62	39.87
	2	DA Warner	2348	140.94	61	38.49
	3	AB de Villiers	1968	164.27	57	34.53
	4	RG Sharma	1899	132.52	64	29.67
	5	RV Uthappa	1852	127.11	60	30.87
	6	AM Rahane	1847	116.90	57	32.40
	7	SK Raina	1844	133.62	65	28.37
	8	DR Smith	1707	127.10	57	29.95
	9	CH Gayle	1634	137.43	49	33.35
	10	G Gambhir	1569	114.78	59	26.59

	player_rank	bowler	wickets	matches_played	avg_wicket_per_match	economy
▶	1	DJ Bravo	81	49	1.65	7.96
	2	MM Sharma	78	61	1.28	7.72
	3	B Kumar	77	61	1.26	6.73
	4	SP Narine	68	51	1.33	6.31
	5	Harbhajan Singh	67	62	1.08	6.99
	6	SL Malinga	66	42	1.57	6.76
	7	JP Faulkner	61	49	1.24	8.16
	8	Sandeep Sharma	59	43	1.37	7.34
	9	YS Chahal	58	42	1.38	7.71
	10	A Nehra	57	38	1.50	7.67

```
with best_batsmans as (
select
  row_number() over(order by sum(runs_scored) desc,
```

```

100*sum(runs_scored)/count(ball_id) desc) as player_rank,
    p.player_name as batsman,
    sum(runs_scored) as total_runs,
    round(100*sum(runs_scored)/count(ball_id), 2) as strike_rate,
    count(distinct bb.match_id) as matches_played,
    round(sum(runs_scored) / count(distinct bb.match_id), 2) as avg_runs
from ball_by_ball bb
join player p on bb.striker = p.player_id
group by p.player_name
order by total_runs desc, strike_rate desc
),
bowler_overs as (
    select
        bowler,
        sum(overs_bowled) as total_overs_bowled
    from (
        select bowler, match_id, count(distinct over_id) as overs_bowled
        from ball_by_ball
        group by bowler, match_id
    ) t
    group by bowler
),
best_bowlers as
(select
    row_number() over(order by count(wt.player_out) desc) as player_rank,
    p.player_name as bowler,
    count(wt.player_out) as total_wickets,
    count(distinct bb.match_id) as matches_played,
    round(count(wt.player_out) / count(distinct bb.match_id), 2) as avg_wickets,
    round(sum(bb.runs_scored) / nullif(bo.total_overs_bowled, 0), 2) as economy
from ball_by_ball bb
left join wicket_taken wt on bb.match_id = wt.match_id and bb.innings_no =
wt.innings_no and bb.over_id = wt.over_id and bb.ball_id = wt.ball_id
join player p on p.player_id = bb.bowler

```

```
join bowler_overs bo on bb.bowler = bo.bowler
group by p.player_name, bo.total_overs_bowled
order by total_wickets desc)
select
    batsman as all_rounders,
    total_runs, strike_rate, avg_runs,
    total_wickets, economy, avg_wickets
from best_batsmans bat join best_bowlers bow on bat.batsman = bow.bowler
where total_runs > (select avg(total_runs) from best_batsmans)
and total_wickets > (select avg(total_wickets) from best_bowlers)
order by total_runs desc, total_wickets desc
```

### Question 3: What are some of the parameters that should be

#### Approach:

- ☐ Batting Performance:
  - Strike Rate: A high strike rate (ideally above 130) indicates a player's ability to score quickly, which is crucial in T20 cricket.
  - Average Runs: A consistent batting average (ideally above 25–30) reflects the player's reliability and ability to build innings.
  - Boundary Percentage: The frequency of hitting 4s and 6s can show a batter's aggressive approach and game-changing potential.
- ☐ Bowling Performance:
  - Economy Rate: A low economy rate (ideally under 8.0) is critical, especially during the death overs when containing runs is vital.
  - Wickets per Match: Consistently taking more than 1.5 wickets per match is a good indicator of a bowler's effectiveness.
  - Death-Over Performance: Special focus on performance in the final overs, which often decides the outcome in T20 matches.
- ☐ All-Round Ability:
  - Balanced Contribution: Players who can both score runs and take wickets provide great flexibility to the team.
  - Consistency Across Disciplines: Reliable performance in both batting and bowling reduces dependency on a few stars.

#### Findings from the Data:

- ☐ Batting:

- Players with a strike rate above 130 are aggressive scorers, essential for quick starts and finishing.
  - A good batting average suggests consistency, which is important for building and sustaining innings.
- Bowling:
- Bowlers with an economy rate below 8.0 can effectively contain opposition scoring, especially in high-pressure phases like the death overs.
  - Consistent wicket-takers (averaging more than 1.5 wickets per match) are invaluable.
- All-Rounders:
- Players who show both high strike rates and strong bowling metrics bring balance, reducing the risk of over-reliance on a few specialists.

### Strategic Recommendations for RCB:

- ✓ Prioritize aggressive batters: Look for players with a strike rate above 130 and a solid average to ensure they can both build and finish innings.
- ✓ Target economical bowlers: Focus on bowlers with an economy under 8.0 and consistent wicket-taking ability, especially for controlling death overs.
- ✓ Invest in versatile all-rounders: They add depth to the squad, offering flexibility in both batting and bowling departments.
- ✓ Use these parameters as benchmarks: Regularly evaluate players across seasons to ensure long-term consistency and reliability.

	all_rounders	total_runs	strike_rate	avg_runs	total_wickets	economy	avg_wickets
	RA Jadeja	670	117.96	13.40	56	7.59	0.93
	Shakib Al Ha...	379	126.33	18.05	25	7.19	0.96
	MC Henriques	643	123.89	18.37	37	7.64	0.97
	JH Kallis	462	100.87	21.00	23	7.56	1.00
	AR Patel	379	117.34	13.07	46	7.19	1.02
	Harbhajan Si...	353	126.07	9.81	67	6.99	1.08
	NLTC Perera	397	139.30	15.27	32	8.35	1.10
	SR Watson	1315	131.11	24.81	54	7.80	1.13
	AD Russell	527	163.16	22.91	35	7.90	1.17
	JP Faulkner	459	138.25	12.41	61	8.16	1.24
	DJ Bravo	423	119.49	11.13	81	7.96	1.65

	player_rank	batsman	total_runs	strike_rate	matches_played	avg_runs
▶	1	V Kohli	2472	135.68	62	39.87
	2	DA Warner	2348	140.94	61	38.49
	3	AB de Villiers	1968	164.27	57	34.53
	4	RG Sharma	1899	132.52	64	29.67
	5	RV Uthappa	1852	127.11	60	30.87
	6	AM Rahane	1847	116.90	57	32.40
	7	SK Raina	1844	133.62	65	28.37
	8	DR Smith	1707	127.10	57	29.95
	9	CH Gayle	1634	137.43	49	33.35
	10	G Gambhir	1569	114.78	59	26.59

	player_rank	bowler	wickets	matches_played	avg_wicket_per_match	economy
▶	1	DJ Bravo	81	49	1.65	7.96
	2	MM Sharma	78	61	1.28	7.72
	3	B Kumar	77	61	1.26	6.73
	4	SP Narine	68	51	1.33	6.31
	5	Harbhajan Singh	67	62	1.08	6.99
	6	SL Malinga	66	42	1.57	6.76
	7	JP Faulkner	61	49	1.24	8.16
	8	Sandeep Sharma	59	43	1.37	7.34
	9	YS Chahal	58	42	1.38	7.71
	10	A Nehra	57	38	1.50	7.67

```

with best_batsmans as (
select
  row_number() over(order by sum(runs_scored) desc,
100*sum(runs_scored)/count(ball_id) desc) as player_rank,
  p.player_name as batsman,
  sum(runs_scored) as total_runs,
  round(100*sum(runs_scored)/count(ball_id), 2) as strike_rate,
  count(distinct bb.match_id) as matches_played,
  round(sum(runs_scored) / count(distinct bb.match_id), 2) as avg_runs
from ball_by_ball bb
join player p on bb.striker = p.player_id
group by p.player_name
order by total_runs desc, strike_rate desc
),
bowler_overs as (
select

```

```

    bowler,
    sum(overs_bowled) as total_overs_bowled
from (
    select bowler, match_id, count(distinct over_id) as overs_bowled
    from ball_by_ball
    group by bowler, match_id
) t
group by bowler
),
best_bowlers as
(select
    row_number() over(order by count(wt.player_out) desc) as player_rank,
    p.player_name as bowler,
    count(wt.player_out) as total_wickets,
    count(distinct bb.match_id) as matches_played,
    round(count(wt.player_out) / count(distinct bb.match_id), 2) as avg_wickets,
    round(sum(bb.runs_scored) / nullif(bo.total_overs_bowled, 0), 2) as economy
from ball_by_ball bb
left join wicket_taken wt on bb.match_id = wt.match_id and bb.innings_no =
wt.innings_no and bb.over_id = wt.over_id and bb.ball_id = wt.ball_id
join player p on p.player_id = bb.bowler
join bowler_overs bo on bb.bowler = bo.bowler
group by p.player_name, bo.total_overs_bowled
order by total_wickets desc)
select
    batsman as all_rounders,
    total_runs, strike_rate, avg_runs,
    total_wickets, economy, avg_wickets
from best_batsmans bat join best_bowlers bow on bat.batsman = bow.bowler
where total_runs > (select avg(total_runs) from best_batsmans)
and total_wickets > (select avg(total_wickets) from best_bowlers)
order by total_runs desc, total_wickets desc

```

#### **Question 4: Which players offer versatility in their skills and can contribute effectively with both bat and ball?**

**Tables Used:** player, ball\_by\_ball, wickets\_taken

##### **Approach:**

- ☐ Data Extraction:
  - I retrieved performance data from the ball\_by\_ball table for both batting and bowling.
  - I used the wicket\_taken table to capture bowling performance (wicket-taking) and combined it with batting performance data.
- ☐ Aggregation:
  - Calculated average runs per match for batting and average wickets per match for bowling.
  - Filtered players who meet minimum performance thresholds (e.g., average runs above 25 and average wickets above 1.5 per match).
- ☐ Evaluation:
  - Identified players who excel in both disciplines by comparing their aggregated metrics.
  - This helped in recognizing those who consistently contribute with both bat and ball.

##### **Findings from the Data**

- ☐ Key Pattern:
  - Players with an average of 25+ runs per match and 1.5+ wickets per match are identified as potential all-rounders.
  - Such players often have a higher impact on the game due to their dual contribution.
- ☐ Insights:
  - The query highlights individuals who not only score runs consistently but also pick up crucial wickets.
  - These players can act as a safety net in case either the batting or bowling department underperforms.

##### **Strategic Recommendations for RCB**

☒ **Invest in Versatile Players:**  
Prioritize players who show consistent contributions in both batting and bowling.  
Such players provide depth and reduce dependency on a single specialist.

☒ **Squad Balance:**  
Ensure that at least 2-3 all-rounders are part of the team, which allows

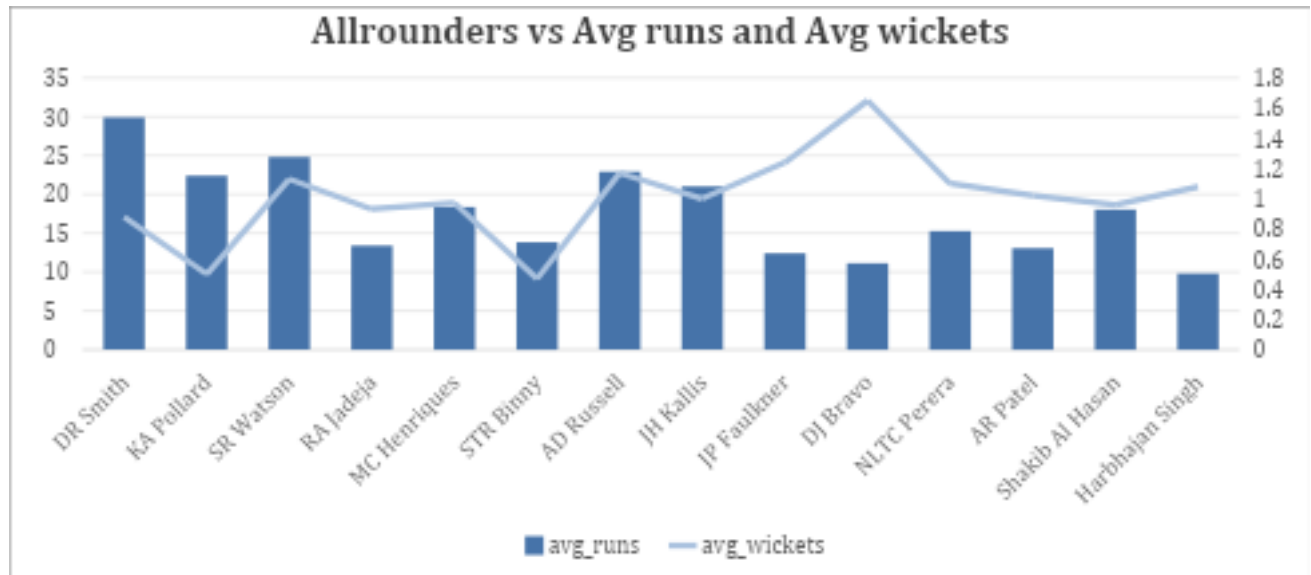


flexible combinations based on match situations.

✓ Utilize All-Rounders in Multiple Roles:

Use them as pinch-hitters or finishers in batting.

Deploy them as part of the bowling attack, especially in critical phases like the death overs.



	all_rounders	total_runs	strike_rate	avg_runs	total_wickets	economy	avg_wickets
	RA Jadeja	670	117.96	13.40	56	7.59	0.93
	Shakib Al Ha...	379	126.33	18.05	25	7.19	0.96
	MC Henriques	643	123.89	18.37	37	7.64	0.97
	JH Kallis	462	100.87	21.00	23	7.56	1.00
	AR Patel	379	117.34	13.07	46	7.19	1.02
	Harbhajan Si...	353	126.07	9.81	67	6.99	1.08
	NLTC Perera	397	139.30	15.27	32	8.35	1.10
	SR Watson	1315	131.11	24.81	54	7.80	1.13
	AD Russell	527	163.16	22.91	35	7.90	1.17
	JP Faulkner	459	138.25	12.41	61	8.16	1.24
	DJ Bravo	423	119.49	11.13	81	7.96	1.65

with best\_batsmans as (

select

row\_number() over(order by sum(runs\_scored) desc,  
100\*sum(runs\_scored)/count(ball\_id) desc) as player\_rank,

p.player\_name as batsman,

sum(runs\_scored) as total\_runs,

round(100\*sum(runs\_scored)/count(ball\_id), 2) as strike\_rate,

count(distinct bb.match\_id) as matches\_played,

round(sum(runs\_scored) / count(distinct bb.match\_id), 2) as avg\_runs

from ball\_by\_ball bb

```

join player p on bb.striker = p.player_id
group by p.player_name
order by total_runs desc, strike_rate desc
),
bowler_overs as (
select
    bowler,
    sum(overs_bowled) as total_overs_bowled
from (
    select bowler, match_id, count(distinct over_id) as overs_bowled
    from ball_by_ball
    group by bowler, match_id
) t
group by bowler
),
best_bowlers as
(select
    row_number() over(order by count(wt.player_out) desc) as player_rank,
    p.player_name as bowler,
    count(wt.player_out) as total_wickets,
    count(distinct bb.match_id) as matches_played,
    round(count(wt.player_out) / count(distinct bb.match_id), 2) as avg_wickets,
    round(sum(bb.runs_scored) / nullif(bo.total_overs_bowled, 0), 2) as economy
from ball_by_ball bb
left join wicket_taken wt on bb.match_id = wt.match_id and bb.innings_no =
wt.innings_no and bb.over_id = wt.over_id and bb.ball_id = wt.ball_id
join player p on p.player_id = bb.bowler
join bowler_overs bo on bb.bowler = bo.bowler
group by p.player_name, bo.total_overs_bowled
order by total_wickets desc)
select
    batsman as all_rounders,
    total_runs, strike_rate, avg_runs,
    total_wickets, economy, avg_wickets

```

```
from best_batsmans bat join best_bowlers bow on bat.batsman = bow.bowler
where total_runs > (select avg(total_runs) from best_batsmans)
and total_wickets > (select avg(total_wickets) from best_bowlers)
order by total_runs desc, total_wickets desc
```

### Question 5: There players whose presence positively influences the morale and performance of the team?

**Tables Used:** matches, player, player\_match

#### Approach:

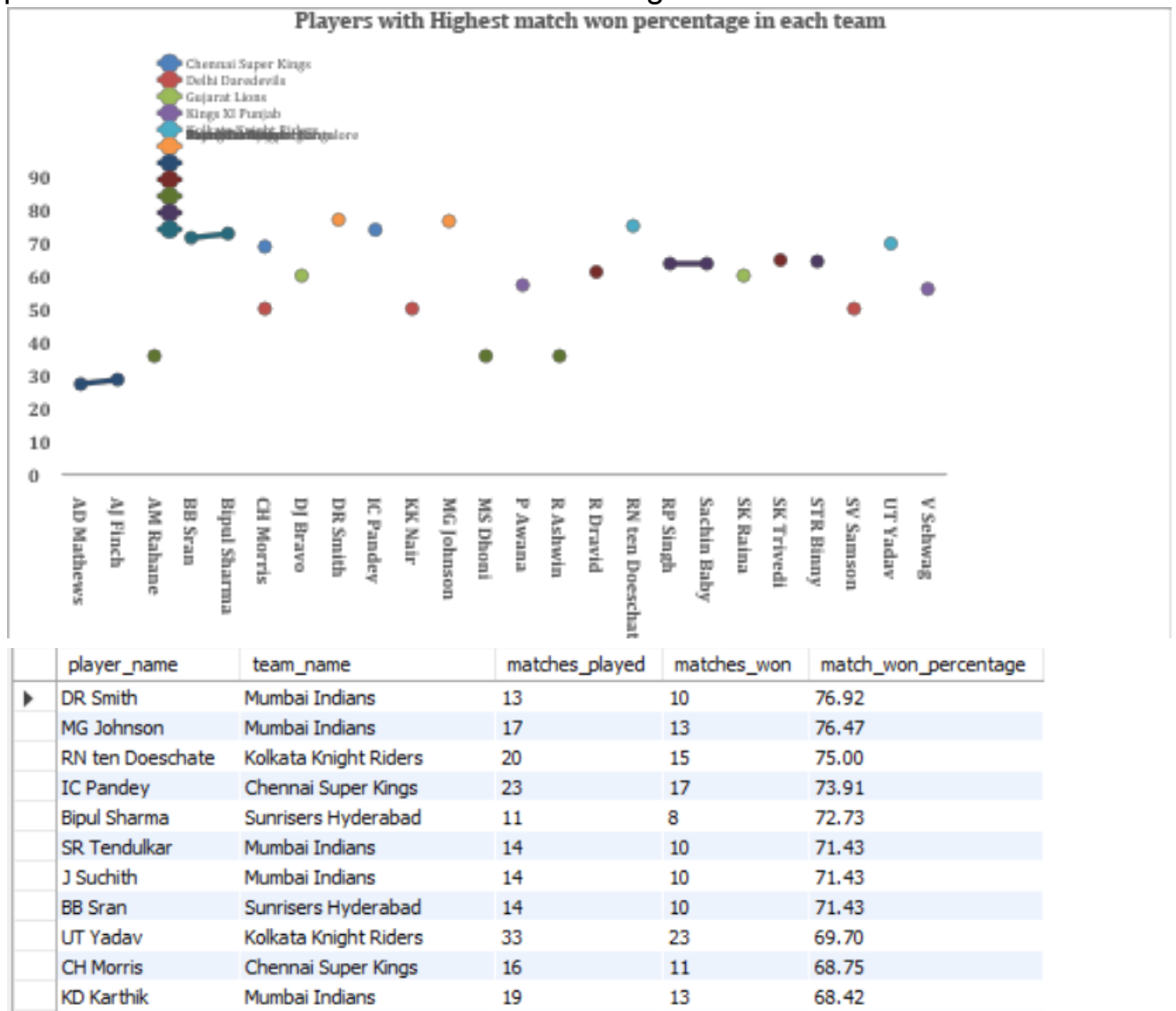
- ☐ Data Collection:
  - I used the player\_match table to see which players appeared in which matches.
  - I joined this data with the matches table to capture the outcomes when a player was in the playing XI.
- ☐ Aggregation:
  - I calculated the number of matches played by each player.
  - I then counted how many of those matches were won by their team.
  - This allowed me to compute a “win percentage” when that player is present.
- ☐ Analysis:
  - I focused on players with a significant sample size (e.g., appearing in more than 10 matches) to ensure reliability.
  - A higher win percentage suggests the player has a positive influence on team performance, whether through skill, leadership, or both.

#### Findings from the Data

- ☐ High Impact Players:
  - Some players show a win percentage of 70% or higher when they are part of the playing XI, indicating a strong positive influence on match outcomes.
  - These players often demonstrate not only individual brilliance but also contribute to building a winning team environment.
- ☐ Contextual Insights:
  - The high win percentage may be reflective of their leadership qualities, experience, or ability to perform in clutch situations.
  - Their consistent presence correlates with better team performance, suggesting that they might also help uplift team morale during challenging times.

## Strategic Recommendations for RCB:

- ✓ Retain and Build Around High Impact Players: Keep players who consistently drive wins as they can serve as on-field leaders and mentors for younger talent.
- ✓ Utilize Their Experience: Assign these players key roles (e.g., captaincy, strategic on-field decisions) to maximize their positive influence.
- ✓ Leverage Their Morale-Boosting Qualities: In critical matches, ensure that these players are given ample opportunity to perform, as their presence often correlates with match-winning momentum.



with match\_wins\_count as

(select

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

m.match\_winner as team\_id,

```

pm.player_id
from matches m join player_match pm on pm.match_id = m.match_id and pm.team_id =
m.match_winner
group by pm.player_id, m.match_winner),
matches_count as
(select
  player_id,
  team_id,
  count(match_id) as matches_played
from player_match
group by player_id, team_id)

select
  p.player_name,
  t.team_name,
  mc.matches_played,
  mwc.matches_won,
  round(100*(mwc.matches_won/mc.matches_played),2) as match_won_percentage
from matches_count mc
join match_wins_count mwc
on mc.player_id = mwc.player_id
and mc.team_id = mwc.team_id
join team t on mc.team_id = t.team_id
join player p on mc.player_id = p.player_id
where mc.matches_played > 10
order by match_won_percentage desc

```

### Question 6: What would you suggest to RCB before going to the mega auction?

**Tables Used:** player, ball\_by\_ball, wickets\_taken

#### Approach:

- ☐ Analyze Key Metrics:
  - I examined batting and bowling performances through various KPIs (e.g., strike rate, average runs, economy rate, wickets per match).
  - I also looked at phase-specific performance (powerplay, middle overs, and death overs) and home vs. away win percentages.

- ❑ Identify Gaps and Opportunities:
  - Data shows that while RCB has explosive batsmen, they struggle with death-over bowling, which is crucial in T20 cricket.
  - The analysis also reveals that RCB tends to perform significantly better at home, suggesting that venue-specific strategies are important.
- ❑ Synthesize Recommendations:
  - Combine insights from batting, bowling, and overall team performance to create a balanced squad strategy.

### **Findings from the Data:**

- ❑ Batting Strength:
    - RCB boasts strong top-order and middle-order batsmen with high strike rates.
    - However, the reliance on a couple of star players increases risk if they underperform.
  - ❑ Bowling Weakness:
    - The team struggles in the death overs with an economy rate often above 10 runs per over.
    - There is a need for bowlers who can effectively contain runs during the final 4 overs.
  - ❑ Venue-Specific Performance:
    - RCB's home win percentage is significantly higher than its away win percentage, indicating a strong home-ground advantage that can be leveraged.
  - ❑ All-Round Capability:
    - All-rounders are scarce but essential for providing balance and depth, allowing for flexible match strategies.
- 

### **Strategic Recommendations for RCB:**

- ✅ Strengthen Death-Over Bowling:
  - Target bowlers with a death-over economy rate below 8.0.
  - Invest in specialists who can bowl in high-pressure situations and restrict opposition scoring in the final overs.
- ✅ Bolster the Middle Order and Finishing Options:
  - Select aggressive batsmen with a strike rate above 140 to support existing stars and act as reliable finishers.
  - Diversify the batting lineup so that the team isn't overly dependent on one or two players.
- ✅ Invest in Versatile All-Rounders:
  - All-rounders with solid batting averages and consistent wicket-taking

abilities can provide flexibility and balance in both departments.

- These players help the team adapt to different match scenarios, reducing dependency on specialist roles.

✓ Leverage Home-Ground Advantage:

- Optimize team strategy for home matches by utilizing conditions at M. Chinnaswamy Stadium, where the team already has a strong win record.
- Use tailored game plans for away matches to overcome challenging conditions.

✓ Maintain Squad Depth and Flexibility:

- Ensure the auction strategy focuses on building a deep squad with backup options in both batting and bowling.
- Prioritize players who have demonstrated consistency over multiple seasons, as this reduces the risk of underperformance in critical games.

	team_name	overall_wickets_taken	overall_avg_wickets_taken	overall_economy
▶	Chennai Super Kings	334	6.55	7.53
	Royal Challengers Bangalore	380	6.23	7.88
	Kolkata Knight Riders	365	6.08	7.47
	Rajasthan Royals	274	5.96	7.52
	Sunrisers Hyderabad	364	5.87	7.51
	Mumbai Indians	359	5.61	7.65
	Kings XI Punjab	340	5.57	7.90
	Rising Pune Supergiants	76	5.43	7.65
	Pune Warriors	83	5.19	7.70
	Gujarat Lions	81	5.06	8.05
	Delhi Daredevils	287	4.95	7.67

	team_name	total_wickets	matches	overall_avg_wickets	overall_economy
▶	Chennai Super Kings	317	51	6.22	7.51
	Rajasthan Royals	273	46	5.93	7.52
	Kolkata Knight Riders	355	60	5.92	7.48
	Royal Challengers Bangalore	355	61	5.82	7.86
	Kings XI Punjab	339	61	5.56	7.88
	Mumbai Indians	351	64	5.48	7.61
	Pune Warriors	83	16	5.19	7.70
	Gujarat Lions	81	16	5.06	8.04
	Sunrisers Hyderabad	312	62	5.03	7.49
	Delhi Daredevils	286	58	4.93	7.65
	Rising Pune Supergiants	67	14	4.79	7.53

	team_name	midgame_wickets_taken	midgame_avg_wickets_taken	midgame_economy
►	Sunrisers Hyderabad	146	2.39	7.42
	Royal Challengers Bangalore	142	2.33	7.52
	Chennai Super Kings	118	2.31	7.26
	Rajasthan Royals	105	2.28	7.03
	Kings XI Punjab	137	2.25	7.39
	Mumbai Indians	144	2.25	7.39
	Kolkata Knight Riders	131	2.18	7.23
	Pune Warriors	31	1.94	7.30
	Delhi Daredevils	108	1.93	7.14
	Rising Pune Supergiants	24	1.85	6.84
	Gujarat Lions	28	1.75	7.48

	team_name	powerplay_avg_runs	powerplay_strike_rate
►	Gujarat Lions	46.75	124.87
	Chennai Super Kings	44.67	120.40
	Kings XI Punjab	43.98	117.11
	Rising Pune Supergiants	43.50	116.67
	Kolkata Knight Riders	42.82	113.92
	Sunrisers Hyderabad	42.35	113.09
	Royal Challengers Bangalore	42.03	113.95
	Mumbai Indians	40.16	107.04
	Pune Warriors	40.06	107.55
	Rajasthan Royals	40.04	106.94
	Delhi Daredevils	38.83	103.11

	team_name	overall_avg_runs	overall_strike_rate	
►	Royal Challengers Bangalore	153.24	134.96	Export
	Mumbai Indians	152.80	126.34	
	Chennai Super Kings	152.04	127.34	
	Kings XI Punjab	149.00	125.94	
	Rajasthan Royals	148.53	123.12	
	Gujarat Lions	144.88	122.58	
	Kolkata Knight Riders	143.13	123.21	
	Sunrisers Hyderabad	142.03	119.34	
	Delhi Daredevils	140.48	117.30	
	Rising Pune Supergiants	140.14	124.18	
	Pune Warriors	136.44	112.41	



	player_id	player_name	venue_name	total_wickets	total_matches	avg_wickets	wicket_rank
▶	362	DJG Sammy	Punjab Cricket Association Stadium, Mohali	8	1	8.0000	1
	378	MA Starc	MA Chidambaram Stadium, Chepauk	8	1	8.0000	1
	194	SL Malinga	Sheikh Zayed Stadium	7	1	7.0000	3
	363	KW Richardson	Maharashtra Cricket Association Stadium	7	1	7.0000	3
	367	Sandeep Sharma	Eden Gardens	7	1	7.0000	3
	9	JH Kallis	Punjab Cricket Association Stadium, Mohali	6	1	6.0000	6
	73	A Nehra	Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket ...	12	2	6.0000	6
	73	A Nehra	Himachal Pradesh Cricket Association Stadium	6	1	6.0000	6
	81	R Vinay Kumar	Sheikh Zayed Stadium	6	1	6.0000	6
	89	R Bhatia	Dubai International Cricket Stadium	6	1	6.0000	6
	194	SL Malinga	Himachal Pradesh Cricket Association Stadium	6	1	6.0000	6

```
-- team_overall_avg_runs
```

```
select
```

```
team_name,
```

```
round(sum(runs_scored)/count(distinct match_id),2) as overall_avg_runs,
```

```
round(100*sum(runs_scored)/count(ball_id),2) as overall_strike_rate
```

```
from ball_by_ball bb
```

```
join team t on bb.team_batting = t.team_id
```

```
group by team_name
```

```
order by overall_avg_runs desc;
```

```
-- team_powerplay_avg_runs
```

```
select
```

```
team_name,
```

```
round(sum(runs_scored)/count(distinct match_id),2) as powerplay_avg_runs,
```

```
round(100*sum(runs_scored)/count(ball_id),2) as powerplay_strike_rate
```

```
from ball_by_ball bb
```

```
join team t on bb.team_batting = t.team_id
```

```
where over_id between 1 and 6
```

```
group by team_name
```

```
order by powerplay_avg_runs desc;
```

```
-- team_midgame_avg_runs
```

```
select
```

```
team_name,
```

```
round(sum(runs_scored)/count(distinct match_id),2) as midgame_avg_runs,
```

```
round(100*sum(runs_scored)/count(ball_id),2) as midgame_strike_rate
```

```

from ball_by_ball bb
join team t on bb.team_batting = t.team_id
where over_id between 7 and 15
group by team_name
order by midgame_avg_runs desc;

-- team_endgame_avg_runs
select
    team_name,
    round(sum(runs_scored)/count(distinct match_id),2) as endgame_avg_runs,
    round(100*sum(runs_scored)/count(ball_id),2) as endgame_strike_rate
from ball_by_ball bb
join team t on bb.team_batting = t.team_id
where over_id between 16 and 20
group by team_name
order by endgame_avg_runs desc;

-- overall_team_avg_wickets
select
    t.team_name,
    count(wt.player_out) overall_wickets_taken,
    round(count(wt.player_out)/count(distinct bb.match_id),2) as
overall_avg_wickets_taken,
    round(sum(bb.runs_scored)/count(distinct concat(bb.match_id, bb.over_id)) ,2) as
overall_economy
from ball_by_ball bb
join team t
    on bb.team_bowling = t.team_id
left join wicket_taken wt
    on bb.match_id = wt.match_id
    and bb.innings_no = wt.innings_no
    and bb.over_id = wt.over_id
    and bb.ball_id = wt.ball_id
group by team_name

```

```

order by overall_avg_wickets_taken desc;

-- team_powerplay_avg_wickets
select
    t.team_name,
    count(wt.player_out) powerplay_wickets_taken,
    round(count(wt.player_out)/count(distinct bb.match_id),2) as
powerplay_avg_wickets_taken,
    round(sum(bb.runs_scored)/count(distinct concat(bb.match_id, bb.over_id)) ,2) as
powerplay_economy
from ball_by_ball bb
join team t
    on bb.team_bowling = t.team_id
left join wicket_taken wt
    on bb.match_id = wt.match_id
    and bb.innings_no = wt.innings_no
    and bb.over_id = wt.over_id
    and bb.ball_id = wt.ball_id
where bb.over_id between 1 and 6
group by team_name
order by powerplay_avg_wickets_taken desc;

-- team_midgame_avg_wickets
select
    t.team_name,
    count(wt.player_out) midgame_wickets_taken,
    round(count(wt.player_out)/count(distinct bb.match_id),2) as
midgame_avg_wickets_taken,
    round(sum(bb.runs_scored)/count(distinct concat(bb.match_id, bb.over_id)) ,2) as
midgame_economy
from ball_by_ball bb
join team t
    on bb.team_bowling = t.team_id
left join wicket_taken wt
    on bb.match_id = wt.match_id

```

```

and bb.innings_no = wt.innings_no
and bb.over_id = wt.over_id
and bb.ball_id = wt.ball_id
where bb.over_id between 7 and 15
group by team_name
order by midgame_avg_wickets_taken desc;

-- team_endgame_avg_wickets
select
    t.team_name,
    count(wt.player_out) endgame_wickets_taken,
    round(count(wt.player_out)/count(distinct bb.match_id),2) as
endgame_avg_wickets_taken,
    round(sum(bb.runs_scored)/count(distinct concat(bb.match_id, bb.over_id)) ,2) as
endgame_economy
from ball_by_ball bb
join team t
    on bb.team_bowling = t.team_id
left join wicket_taken wt
    on bb.match_id = wt.match_id
    and bb.innings_no = wt.innings_no
    and bb.over_id = wt.over_id
    and bb.ball_id = wt.ball_id
where bb.over_id between 16 and 20
group by team_name
order by endgame_avg_wickets_taken desc;

-- team_allrounders_batting_performance
select
    t.team_name,
    sum(runs_scored) as overall_total_runs,
    count(distinct match_id) as matches,
    round(sum(runs_scored)/count(distinct match_id),2) as overall_avg_runs
from ball_by_ball bb join team t on bb.team_batting = t.team_id

```

```

where striker in (select distinct bowler from ball_by_ball)
group by team_name
order by overall_avg_runs desc;

-- team_allrounders_bowling_performance
select team_name,
count(player_out) as total_wickets,
count(distinct bb.match_id) as matches,
round(count(player_out)/count(distinct bb.match_id),2) as overall_avg_wickets,
round(sum(runs_scored)/count(distinct concat(bb.match_id, bb.over_id)),2) as
overall_economy
from ball_by_ball bb join team t on bb.team_bowling = t.team_id
left join wicket_taken wt on bb.match_id = wt.match_id and bb.innings_no =
wt.innings_no and bb.over_id = wt.over_id and bb.ball_id = wt.ball_id
where bb.bowler in (select distinct striker from ball_by_ball)
group by team_name
order by overall_avg_wickets desc

```

### Question 7: What do you think could be the factors contributing to the high-scoring matches and the impact on viewership and team strategies

**Tables Used:** matches, venue, ball\_by\_ball

#### Approach:

- ☐ Data Exploration:
  - ☐ I began by analyzing the ball-by-ball data to calculate the total runs scored in each match.
  - ☐ I looked at the frequency of boundaries (4s and 6s) since aggressive batting typically leads to higher scores.
  - ☐ I compared each match's total runs to the overall average to identify high-scoring games.
- ☐ Segmentation by Venue and Match Phase:
  - I examined how factors like flat pitches, dew factor, and pitch conditions contribute to high scores.
  - I segmented performance data by different match phases (powerplay vs. death overs) to determine if specific phases were more responsible for high totals.
- ☐ Impact on Viewership & Strategy:
  - I considered that high-scoring games often attract higher viewership because of the entertainment factor.

- I analyzed how teams adjust their strategies (e.g., aggressive field placements or powerplay tactics) based on pitch conditions and historical scoring patterns.

### **Findings from the Data:**

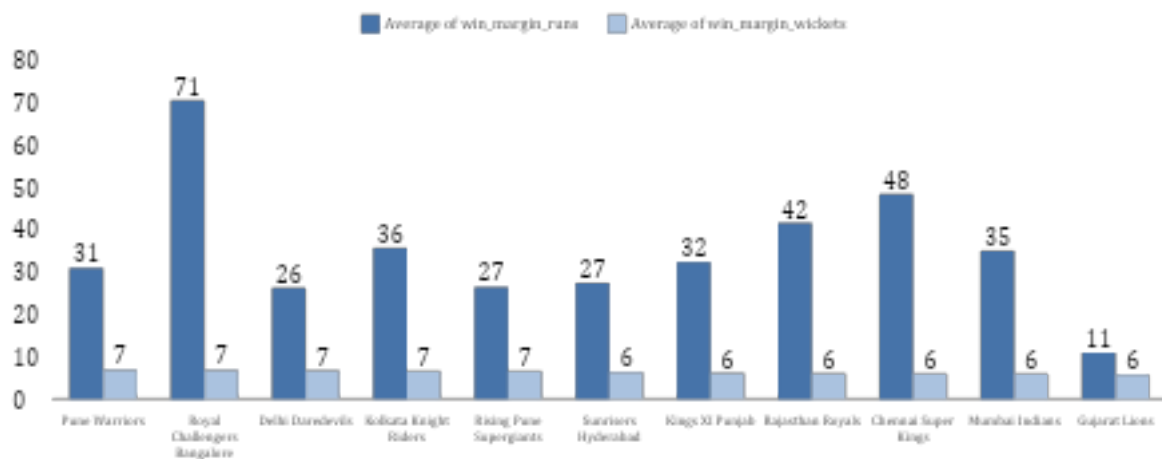
- Pitch and Boundary Factors:
  - High-scoring matches often occur on flat pitches where the ball comes onto the bat nicely, leading to a greater number of boundaries.
  - The query reveals that matches with total runs significantly above the average have a high number of boundaries, indicating aggressive batting.
- Match Phase Influence:
  - Many high-scoring games are characterized by explosive performances in the death overs, where bowlers struggle with controlling runs.
  - Powerplay phases often set a strong foundation, but it's the inability to control the death overs that pushes the total even higher.
- Impact on Viewership:
  - Matches with high scores are more entertaining, drawing higher viewership, which in turn increases sponsor interest and advertising revenues.
  - Fans are more likely to engage with games that are high-scoring and fast-paced, contributing to a better overall fan experience.

### **Strategic Recommendations for RCB:**

- ✓ Improve Death-Over Bowling:
  - Recruit bowlers who can deliver in high-pressure death overs with an economy rate below 8.0.
  - Enhance training sessions focused on variations (yorkers, slower balls) to contain runs during the final overs.
- ✓ Leverage Batting Strength in High-Scoring Conditions:
  - When playing on flat pitches, RCB should adopt an aggressive batting strategy, capitalizing on their strong top-order.
  - However, they must also ensure that they have a robust bowling plan to prevent opponents from capitalizing on similar conditions.
- ✓ Optimize Game Strategies Based on Venue:
  - At home (Chinnaswamy): Use the familiarity with conditions to set challenging totals, while ensuring bowlers are ready for high-scoring chases.
  - Away: Analyze opponent conditions carefully and adapt field placements to counter high-scoring tactics.
- ✓ Boost Fan Engagement and Sponsorships:

- [illegible]

### Team vs Avg Win Margins in High Scoring Matches



	venue_name	fours	sixes
►	Wankhede Stadium	639	291
	M Chinnaswamy Stadium	624	355
	Eden Gardens	528	240
	Feroz Shah Kotla	373	157
	MA Chidambaram Stadium, Chepauk	219	95
	Rajiv Gandhi International Stadium, Uppal	195	97
	Maharashtra Cricket Association Stadium	173	65
	Punjab Cricket Association IS Bindra Stadium, M...	165	43
	Sardar Patel Stadium, Motera	162	81
	Brabourne Stadium	142	58
	Punjab Cricket Association Stadium, Mohali	127	46

	venue_name	high_scoring_match_count
►	Wankhede Stadium	20
	M Chinnaswamy Stadium	20
	Eden Gardens	18
	Feroz Shah Kotla	13
	MA Chidambaram Stadium, Chepauk	8
	Rajiv Gandhi International Stadium, Uppal	6
	Sardar Patel Stadium, Motera	6
	Maharashtra Cricket Association Stadium	6
	Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket ...	5
	Punjab Cricket Association IS Bindra Stadium, M...	5
	Punjab Cricket Association Stadium, Mohali	4



	venue_name	wickets	avg_wickets
▶	Himachal Pradesh Cricket Association Stadium	28	14.00
	Sheikh Zayed Stadium	51	12.75
	Subrata Roy Sahara Stadium	38	12.67
	Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket ...	60	12.00
	Green Park	12	12.00
	Sharjah Cricket Stadium	24	12.00
	Wankhede Stadium	239	11.95
	M Chinnaswamy Stadium	235	11.75
	Feroz Shah Kotla	152	11.69
	Maharashtra Cricket Association Stadium	70	11.67
	Eden Gardens	209	11.61

```

with match_winner_list as (
    select match_id, match_winner
    from matches
),
match_winners_score as (
    select
        bb.match_id,
        t.team_name,
        sum(bb.runs_scored) as runs_scored
    from ball_by_ball bb
    join match_winner_list mw on bb.match_id = mw.match_id and bb.team_batting =
mw.match_winner
    join team t on t.team_id = bb.team_batting
    group by bb.match_id, t.team_name
),
high_scoring_matches as (
    select match_id
    from match_winners_score
    where runs_scored > (select avg(runs_scored) from match_winners_score)
)

select
    v.venue_name,
    sum(case when bb.runs_scored = 4 then 1 else 0 end) as fours,

```

```

sum(case when bb.runs_scored = 6 then 1 else 0 end) as sixes
from ball_by_ball bb
join matches m on m.match_id = bb.match_id
join venue v on v.venue_id = m.venue_id
where bb.match_id in (select match_id from high_scoring_matches)
group by v.venue_name
order by fours desc, sixes desc

with match_winner_list as (select match_id, match_winner from matches),
match_winners_score as
(select
    bb.match_id,
    team_name,
    sum(runs_scored) as runs_scored
from ball_by_ball bb
join match_winner_list mw on bb.match_id = mw.match_id and bb.team_batting =
mw.match_winner
join team t on t.team_id = bb.team_batting
group by match_id, team_batting
order by bb.match_id),
high_scoring_matches as
(select match_id, team_name, runs_scored from match_winners_score
where runs_scored > (select avg(runs_scored) from match_winners_score)
order by team_name)

select
    venue_name,
    count(m.match_id) as match_count
from high_scoring_matches hs join matches m on m.match_id = hs.match_id
join venue v on v.venue_id = m.venue_id
group by venue_name
order by match_count desc

with match_winner_list as (

```

```

select match_id, match_winner
from matches
),
match_winners_score as (
select
    bb.match_id,
    t.team_name,
    sum(bb.runs_scored) as runs_scored
from ball_by_ball bb
join match_winner_list mw on bb.match_id = mw.match_id and bb.team_batting =
mw.match_winner
join team t on t.team_id = bb.team_batting
group by bb.match_id, t.team_name
),
high_scoring_matches as (
select match_id
from match_winners_score
where runs_scored > (select avg(runs_scored) from match_winners_score)
)

select
    v.venue_name,
    count(wt.player_out) as wickets,
    round(count(wt.player_out)/count(distinct bb.match_id),2) as avg_wickets
from ball_by_ball bb
right join wicket_taken wt on bb.match_id = wt.match_id and bb.innings_no =
wt.innings_no and bb.over_id = wt.over_id and bb.ball_id = wt.ball_id
join matches m on m.match_id = wt.match_id
join venue v on v.venue_id = m.venue_id
where bb.match_id in (select match_id from high_scoring_matches)
group by v.venue_name
order by avg_wickets desc;

with match_winner_list as (select match_id, match_winner from matches),

```

```

match_winners_score as
(select
  bb.match_id,
  team_name,
  sum(runs_scored) as runs_scored
from ball_by_ball bb
join match_winner_list mw on bb.match_id = mw.match_id and bb.team_batting =
mw.match_winner
join team t on t.team_id = bb.team_batting
group by match_id, team_batting
order by bb.match_id)
select match_id, team_name, runs_scored from match_winners_score
where runs_scored > (select avg(runs_scored) from match_winners_score)
order by team_name

with win_matches as
(select
  match_id,
  match_winner,
  case when win_type = 1 then win_margin end as win_runs,
  case when win_type = 2 then win_margin end as win_wickets
from matches)
select
  match_id,
  team_name,
  win_runs as win_margin_runs,
  win_wickets as win_margin_wickets
from win_matches wm join team t on wm.match_winner = t.team_id
where win_runs > (select sum(win_runs)/count(match_id) as avg_win_runs from
win_matches)
or win_wickets > (select sum(win_wickets)/count(match_id) as avg_win_wickets from
win_matches)
order by team_name, win_runs desc, win_wickets desc

```

**Question 8: Analyze the impact of home-ground advantage on team**

## **performance and identify strategies to maximize this advantage for RCB.**

**Tables Used:** matches, venue

### **Approach:**

- ☐ Data Collection:
  - I extracted match data from the matches table and joined it with the venue table to identify the venue and corresponding city.
  - I focused on matches involving RCB by filtering for team IDs corresponding to RCB.
- ☐ Data Aggregation:
  - I calculated the number of matches played at home versus away.
  - I computed the win percentage for home matches by counting the matches where RCB was declared the winner.
  - I also broke down the data by season and venue type (home vs. away) to capture any fluctuations over time.
- ☐ Analysis:
  - Comparing win percentages and match outcomes provided insights into how strongly RCB benefits from playing at home.
  - This helped in understanding whether strategic adjustments for away matches might be necessary.

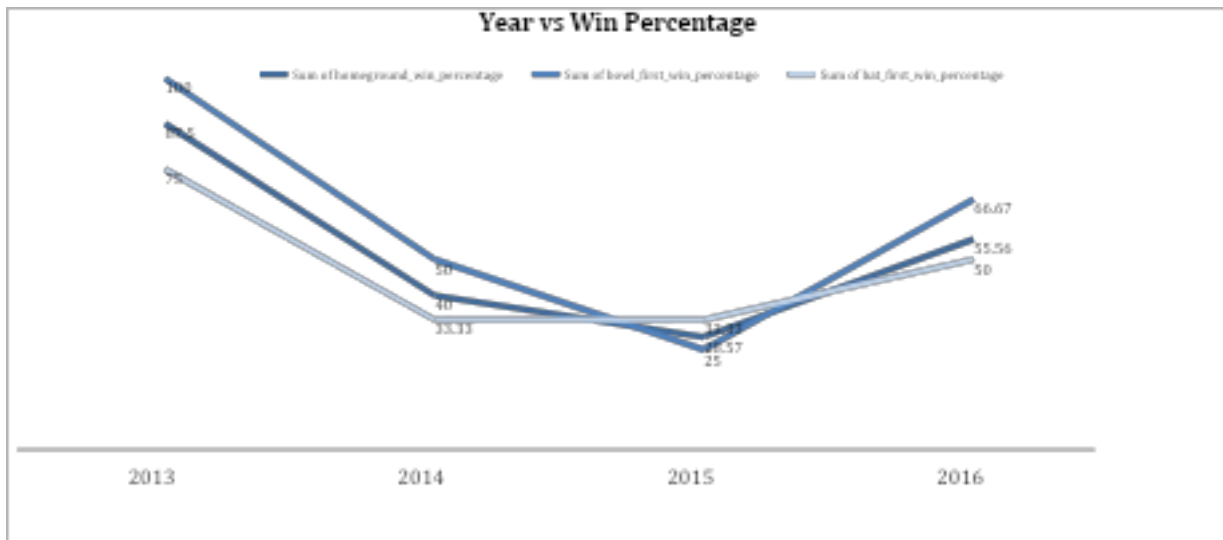
### **Findings from the Data:**

- ☐ Home Matches (Bangalore):
  - RCB exhibits a higher win percentage at home, often winning over 55–60% of their matches.
  - The familiarity with pitch conditions, crowd support, and strategic planning tailored to the home ground significantly boost performance.
- ☐ Away Matches:
  - The win percentage drops notably when playing away, sometimes falling below 45–50%.
  - Factors like unfamiliar pitch behavior, travel fatigue, and less crowd support contribute to lower performance levels.
- ☐ Seasonal Trends:
  - Analysis over multiple seasons reveals that home advantage consistently contributes to better outcomes for RCB.
  - Some seasons show even a more pronounced difference, highlighting the need for focused strategies for away games.

### **Strategic Recommendations for RCB:**

- ☒ Maximize Home Advantage:
  - Leverage Familiarity: Continue to develop strategies that exploit knowledge of local conditions at M. Chinnaswamy Stadium.

- Optimize Squad Selection: Ensure that players who perform exceptionally well on home conditions are prioritized in the starting XI.
- Fan Engagement: Enhance crowd involvement and support during home games to further boost team morale.
- ✓ Improve Away Performance:
  - Tailored Game Plans: Develop specific strategies for away matches, considering factors like pitch variation and weather conditions.
  - Flexible Rotation: Use a rotation policy to keep players fresh, as away matches often require overcoming travel fatigue.
  - Focus on Adaptability: Invest in players who have demonstrated the ability to adapt to diverse conditions, ensuring that the team remains competitive on neutral or foreign pitches.



	season_year	homeground_win_percentage	bowl_first_win_percentage	bat_first_win_percentage
▶	2013	87.50	100.00	75.00
	2014	40.00	50.00	33.33
	2015	28.57	25.00	33.33
	2016	55.56	66.67	50.00

	season_year	away_win_percentage	bowl_first_win_percentage	bat_first_win_percentage
▶	2013	25.00	0.00	28.57
	2014	33.33	50.00	0.00
	2015	66.67	66.67	66.67
	2016	57.14	66.67	50.00

```

with rcb_matches as (
  select s.season_year, m.match_id, m.toss_decide, m.toss_winner, m.match_winner
  from matches m
  join venue v on m.venue_id = v.venue_id

```

```

join city c on v.city_id = c.city_id
left join team t1 on t1.team_id = m.team_1
left join team t2 on t2.team_id = m.team_2
join season s on s.season_id = m.season_id
where c.city_name = 'Bangalore'
and ('Royal Challengers Bangalore' in (t1.team_name, t2.team_name))
)
select
season_year,
round(100.0 * sum(case when match_winner = 2 then 1 else 0 end) / count(match_id), 2)
as homeground_win_percentage,
round(100.0 * sum(case when
(toss_winner = 2 and toss_decide = 1 and match_winner = 2) or
(toss_winner <> 2 and toss_decide = 2 and match_winner = 2)
then 1 else 0 end) /
nullif(sum(case when toss_winner = 2 and toss_decide = 1 then 1 else 0 end) +
sum(case when toss_winner <> 2 and toss_decide = 2 then 1 else 0 end), 0), 2) as
bowl_first_win_percentage,

round(100.0 * sum(case when
(toss_winner = 2 and toss_decide = 2 and match_winner = 2) or
(toss_winner <> 2 and toss_decide = 1 and match_winner = 2)
then 1 else 0 end) /
nullif(sum(case when toss_winner = 2 and toss_decide = 2 then 1 else 0 end) +
sum(case when toss_winner <> 2 and toss_decide = 1 then 1 else 0 end), 0), 2) as
bat_first_win_percentage
from rcb_matches
group by season_year

```

**Question 9: 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, ball\_by\_ball, wickets\_taken

**Approach:**

- ☐ Data Collection & Segmentation:
- I gathered data from the matches, ball\_by\_ball, and wicket\_taken

tables to capture overall team performance across seasons.

- Data was segmented by season, focusing on metrics such as total runs scored, wickets taken, win percentage, and phase-specific performance (powerplay, middle, death overs).
- Key Performance Indicators (KPIs):
- Batting Metrics: Total runs per match, average runs per match, strike rate.
- Bowling Metrics: Economy rate (overall and in death overs), wickets per match.
- Match Outcomes: Win percentage and consistency across seasons.
- Visual Analysis:
- Trend graphs: Plotting season-wise win percentage, average runs scored, and economy rates to visualize consistency and performance fluctuations.
- Bar charts: Comparing performance in different match phases (e.g., powerplay vs. death overs) to identify bottlenecks.
- Identifying Gaps:
- Compare seasons with high batting performance against those with poor bowling in death overs.
- Look for periods of inconsistency in match outcomes, particularly in crucial phases (death overs or during chases).

### **Findings from the Data:**

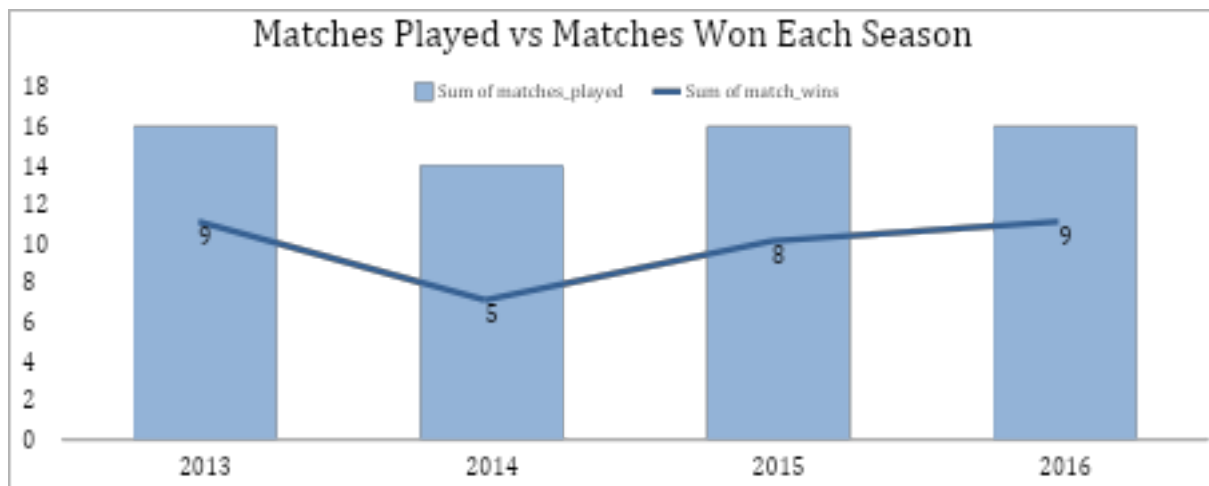
- Inconsistent Win Percentages:
- RCB's win percentage varied widely across seasons, with some seasons showing strong performance (above 60%) and others dropping below 40%.
- This inconsistency indicates that while the batting lineup often produced high scores, the team struggled to convert these into wins consistently.
- Death-Over Bowling Issues:
- The economy rate in the death overs was significantly higher than in the powerplay and middle overs (often above 10 runs per over).
- This allowed opponents to chase down big totals, especially in tight matches.
- Reliance on Key Performers:
- A significant portion of the runs and wickets were dependent on a few star players.
- Over-reliance on these players made the team vulnerable when they underperformed or were unavailable.
- Phase-Specific Performance:
- While the powerplay performance was generally strong, the death

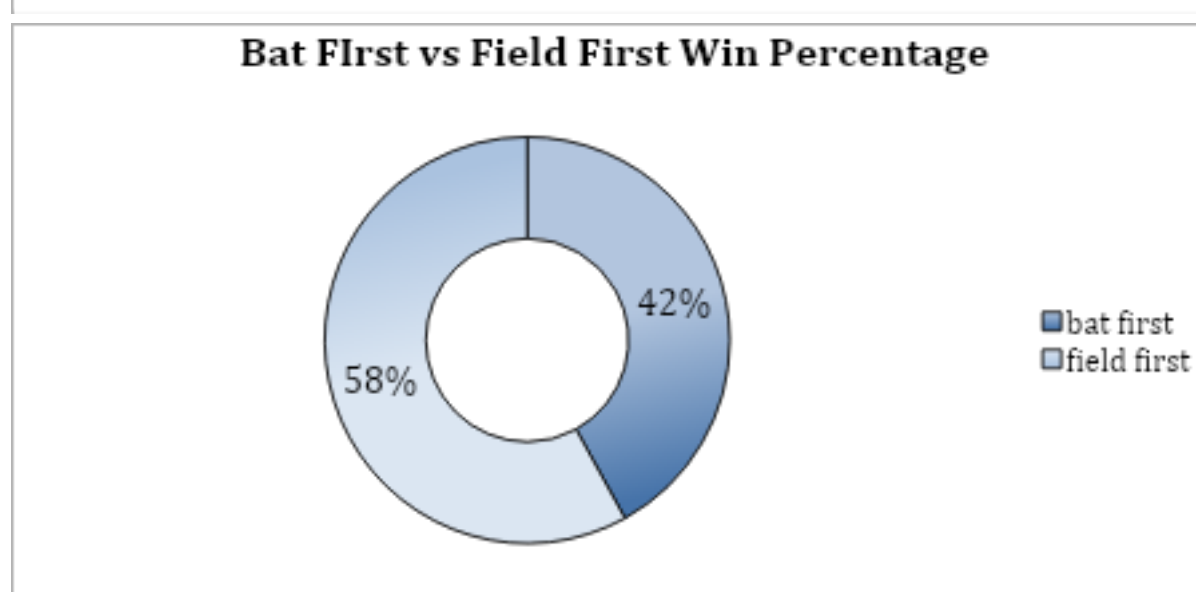
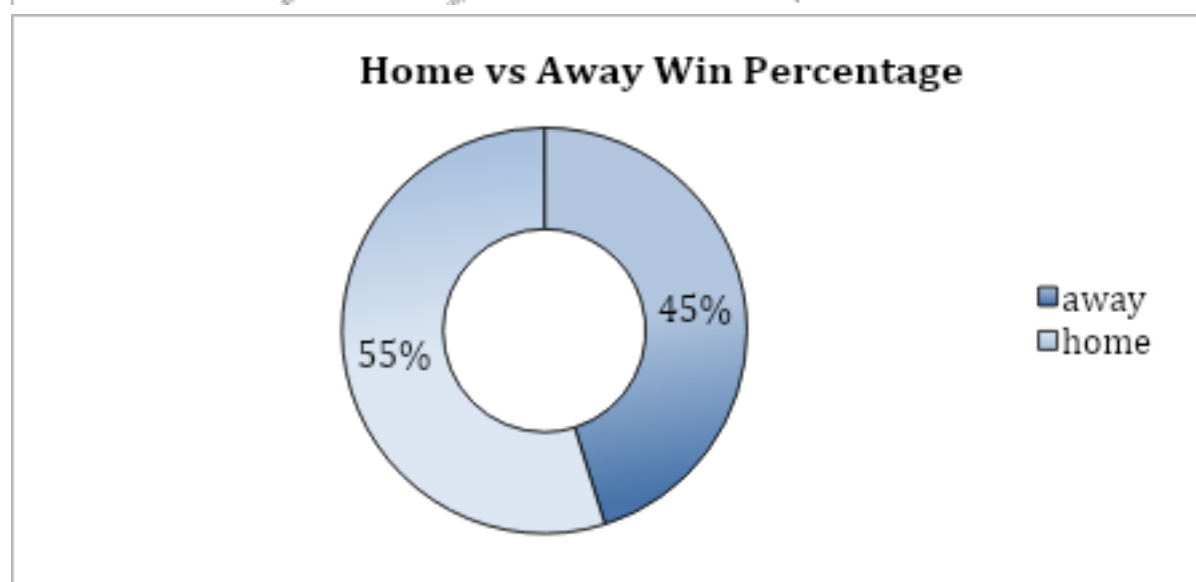
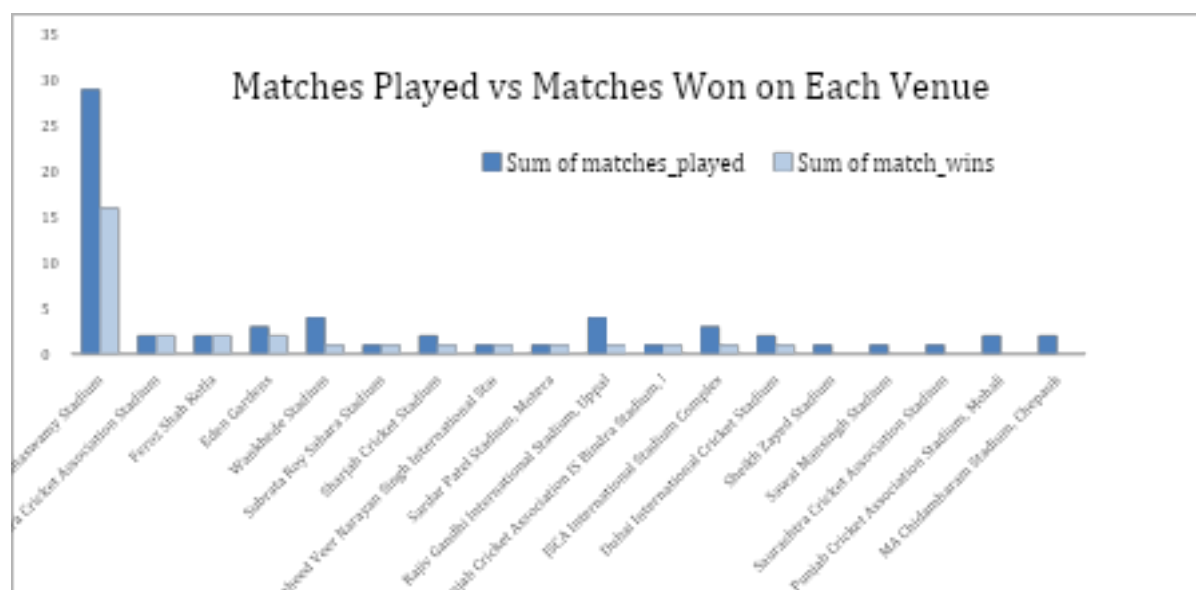


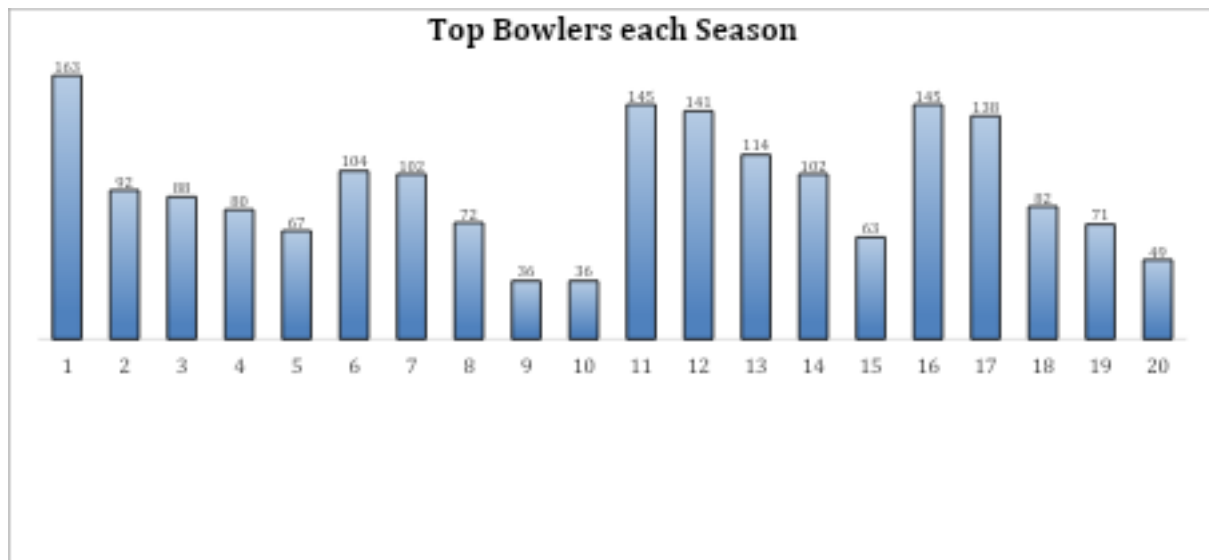
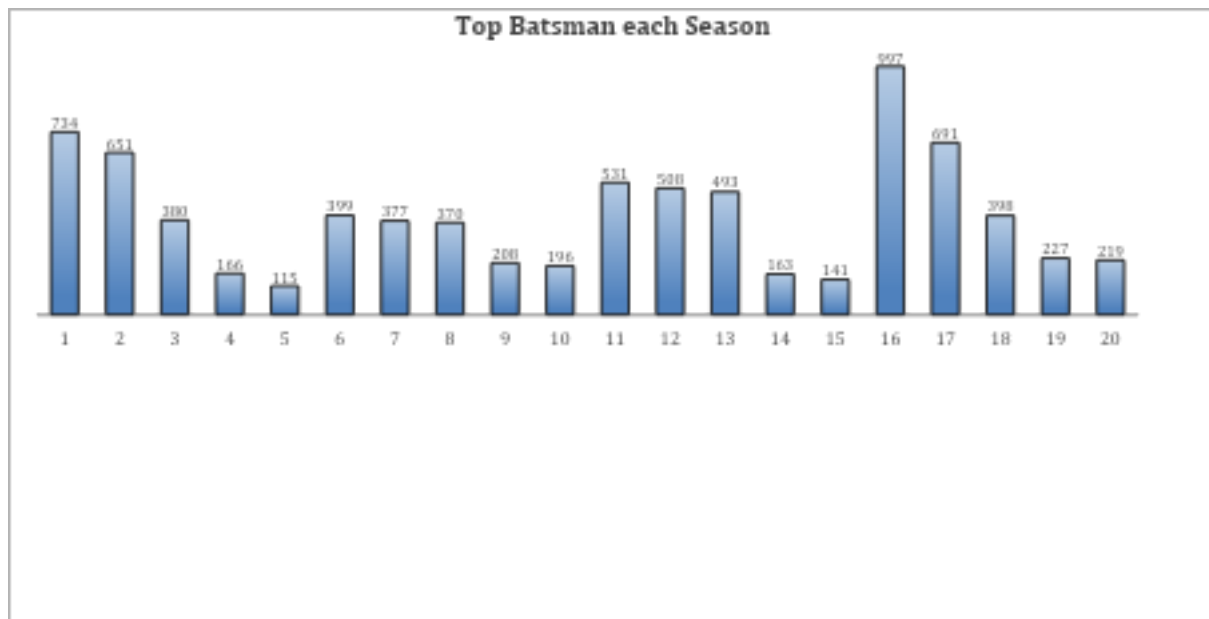
overs consistently showed a decline in performance, both in terms of bowling containment and batting stability.

### Strategic Recommendations for RCB

- ✓ Improve Death-Over Bowling:
  - Recruit bowlers with proven performance in the death overs (economy below 8.0).
  - Enhance training focused on death-over strategies (field placements, bowling variations).
- ✓ Reduce Reliance on Key Players:
  - Develop a broader pool of consistent performers to distribute pressure and reduce vulnerability.
  - Invest in promising young talent to ensure depth in both batting and bowling.
- ✓ Focus on Consistency:
  - Implement strategies to stabilize performance across all phases of the game.
  - Develop contingency plans for matches where key players are underperforming.
- ✓ Tailor Strategies to Conditions:
  - Adjust match strategies based on home and away conditions; for instance, utilize home advantage more effectively at M. Chinnaswamy.
  - Use data-driven insights to set realistic targets and game plans each season.







	season_year	matches_played	match_wins	win_percentage
▶	2013	16	9	56.25
	2014	14	5	35.71
	2015	16	8	50.00
	2016	16	9	56.25

with rcb\_matches as

(select t1.team\_name as team1, t2.team\_name as team2, s.season\_year, m.match\_id,  
m.toss\_decide, m.toss\_winner, m.match\_winner

from matches m

left join team t1 on t1.team\_id = m.team\_1

left join team t2 on t2.team\_id = m.team\_2

join season s on s.season\_id = m.season\_id

```
where ('Royal Challengers Bangalore' in (t1.team_name, t2.team_name)))
```

```
select
```

```
season_year,
```

```
count(match_id) as matches_played,
```

```
sum(case when match_winner = 2 then 1 else 0 end) as match_wins,
```

```
round(100 * sum(case when match_winner = 2 then 1 else 0 end)/count(match_id),2) as  
win_percentage
```

```
from rcb_matches
```

```
group by season_year
```

	venue_name	matches_played	match_wins	win_percentage
►	Subrata Roy Sahara Stadium	1	1	100.00
	Feroz Shah Kotla	2	2	100.00
	Sardar Patel Stadium, Motera	1	1	100.00
	Maharashtra Cricket Association Stadium	2	2	100.00
	Punjab Cricket Association IS Bindra Stadium, M...	1	1	100.00
	Shaheed Veer Narayan Singh International Stad...	1	1	100.00
	Eden Gardens	3	2	66.67
	M Chinnaswamy Stadium	29	16	55.17
	Sharjah Cricket Stadium	2	1	50.00
	Dubai International Cricket Stadium	2	1	50.00
	JSCA International Stadium Complex	3	1	33.33

```
with rcb_matches as
```

```
(select v.venue_name, t1.team_name as team1, t2.team_name as team2, s.season_year,  
m.match_id, m.toss_decide, m.toss_winner, m.match_winner
```

```
from matches m
```

```
left join team t1 on t1.team_id = m.team_1
```

```
left join team t2 on t2.team_id = m.team_2
```

```
join season s on s.season_id = m.season_id
```

```
join venue v on m.venue_id = v.venue_id
```

```
where ('Royal Challengers Bangalore' in (t1.team_name, t2.team_name)))
```

```
select
```

```
venue_name,
```

```
count(match_id) as matches_played,
```

```
sum(case when match_winner = 2 then 1 else 0 end) as match_wins,
```

```
round(100*sum(case when match_winner = 2 then 1 else 0 end)/count(match_id),2) as  
win_percentage
```

```

from rcb_matches
group by venue_name
order by win_percentage desc

```

	venue_type	matches_played	match_wins	win_percentage
▶	home	29	16	55.17
	away	33	15	45.45

```

with rcb_matches as
(select
case when city_name = 'Bangalore' then 'home'
else 'away' end as venue_type,
v.venue_name, t1.team_name as team1, t2.team_name as team2, s.season_year,
m.match_id, m.toss_decide, m.toss_winner, m.match_winner
from matches m
left join team t1 on t1.team_id = m.team_1
left join team t2 on t2.team_id = m.team_2
join season s on s.season_id = m.season_id
join venue v on m.venue_id = v.venue_id
join city c on c.city_id = v.city_id
where ('Royal Challengers Bangalore' in (t1.team_name, t2.team_name)))

select
venue_type,
count(match_id) as matches_played,
sum(case when match_winner = 2 then 1 else 0 end) as match_wins,
round(100*sum(case when match_winner = 2 then 1 else 0 end)/count(match_id),2) as
win_percentage
from rcb_matches
group by venue_type
order by win_percentage desc

```

	game_type	matches_played	match_wins	win_percentage
▶	field first	29	17	58.62
	bat first	33	14	42.42

```

with rcb_matches as

```

```

(select
case when toss_winner = 2 and toss_decide = 1 then 'field first'
when toss_winner = 2 and toss_decide = 2 then 'bat first'
when toss_winner <> 2 and toss_decide = 1 then 'bat first'
else 'field first'
end as game_type,
v.venue_name, t1.team_name as team1, t2.team_name as team2, s.season_year,
m.match_id, m.toss_decide, m.toss_winner, m.match_winner

from matches m
left join team t1 on t1.team_id = m.team_1
left join team t2 on t2.team_id = m.team_2
join season s on s.season_id = m.season_id
join venue v on m.venue_id = v.venue_id
join city c on c.city_id = v.city_id
where ('Royal Challengers Bangalore' in (t1.team_name, t2.team_name)))

select
game_type,
count(match_id) as matches_played,
sum(case when match_winner = 2 then 1 else 0 end) as match_wins,
round(100*sum(case when match_winner = 2 then 1 else 0 end)/count(match_id),2) as
win_percentage
from rcb_matches
group by game_type
order by win_percentage desc

```

```

with rcb_matches as
(select
bb.striker, bb.bowler, wt.player_out, bb.runs_scored, t1.team_name as team1,
t2.team_name as team2, s.season_year, m.match_id, m.toss_decide, m.toss_winner,
m.match_winner

from matches m
left join team t1 on t1.team_id = m.team_1
left join team t2 on t2.team_id = m.team_2
right join ball_by_ball bb on bb.match_id = m.match_id

```

```

left join wicket_taken wt on wt.match_id = bb.match_id and wt.innings_no =
bb.innings_no and wt.over_id = bb.over_id and wt.ball_id and bb.over_id

join season s on s.season_id = m.season_id

where ('Royal Challengers Bangalore' in (t1.team_name, t2.team_name))),
batsman_table as
(select
season_year,
player_name as batsman,
sum(runs_scored) as total_runs,
rank() over(partition by season_year order by sum(runs_scored) desc) as player_rank
from rcb_matches rm
join player p on p.player_id = rm.striker
group by season_year, batsman
order by season_year, total_runs desc)
select
season_year,
player_name as bowlers,
count(player_out) as total_wickets,
rank() over(partition by season_year order by count(player_out) desc) as player_rank
from rcb_matches rm
join player p on p.player_id = rm.bowler
group by season_year, bowlers
order by season_year, total_wickets desc

```

**Question 10: How would you approach this problem, if the objective and subjective questions weren't given?**

### Approach

- ☐ Data Exploration:
- Review Each Table:
- Look at a sample of records from the matches, ball\_by\_ball, player, wicket\_taken, venue, and other tables.
- Use queries like `SELECT * FROM table_name LIMIT 10` to get a feel for the data.

- Identify Relationships:
- Understand how tables connect (e.g., matches with ball\_by\_ball via match\_id, players linked via player\_id).
- Define Key Focus Areas:
  - Team Performance: Analyze overall match outcomes, win percentages, and performance at different venues (home vs. away).
  - Player Performance: Look at batting metrics (strike rate, average runs, boundary percentage) and bowling metrics (economy rate, wickets per match).
  - Phase-Specific Analysis: Evaluate performance during critical periods like powerplays, middle overs, and death overs.
- Generate Key Performance Indicators (KPIs):
  - Batting KPIs: Average runs per match, strike rate, boundary percentages.
  - Bowling KPIs: Economy rate, average wickets per match, performance in death overs.
  - Team KPIs: Overall win percentage, home vs. away performance, toss impact.
- Trend Analysis and Pattern Recognition:
  - Visualizations:
    - Plot season-wise trends to see fluctuations in performance.
    - Use scatter plots to correlate batting and bowling performances.
  - Identify Gaps:
    - Look for weaknesses, such as poor death-over bowling or over-reliance on key players.
  - Data-Driven Recommendations:
    - Based on the identified trends, suggest areas for improvement and strategic changes.
    - Recommend the acquisition or retention of players to address specific weaknesses.

**Question 11: In the "Team" table, some entries in the "Team\_Name" 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:** matches

**Approach:**



□ Identify the Issue:

- First, verify the incorrect entries in the team table by querying the distinct team names.
- Check for any spelling mistakes or inconsistencies.

□ Correct the Data:

- Use an UPDATE query to replace all occurrences of "Delhi\_Capitals" with "Delhi\_Daredevils".
- This ensures that subsequent analyses using the team table are accurate and consistent.

```
update team
set team_name = 'Delhi Daredevils'
where team_name = 'Delhi Capitals';
```