

IPL SQL Case Study

Introduction

Welcome to the IPL SQL Case Study! In this case study, we will be exploring the Indian Premier League (IPL) data for the years 2008, 2009, and 2017. The dataset consists of two tables: "deliveries" and "matches."

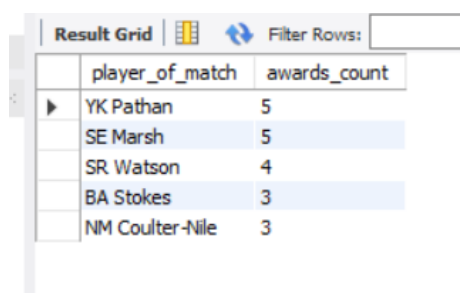
The "deliveries" table contains information about individual deliveries in each match, including details such as match ID, inning, batting team, bowling team, over, ball, batsman, non-striker, bowler, runs scored, extras, dismissal details, and more.

The "matches" table provides information about each match, including the match ID, season, city, date, teams involved, toss details, result, winning team, player of the match, venue, and umpires. By applying SQL queries to the IPL dataset, we can gain valuable insights into player performance, team statistics, and match outcomes. So, let's dive into the case study and explore the exciting world of IPL through data analysis!

Problem

1. WHAT ARE THE TOP 5 PLAYERS WITH THE MOST PLAYER OF THE MATCH AWARDS?

```
SELECT
    player_of_match,
    count(*) as awards_count
FROM matches
GROUP BY player_of_match
ORDER BY 2 desc
LIMIT 5;
```



	player_of_match	awards_count
▶	YK Pathan	5
	SE Marsh	5
	SR Watson	4
	BA Stokes	3
	NM Coulter-Nile	3

2. HOW MANY MATCHES WERE WON BY EACH TEAM IN EACH SEASON?

SELECT

season,

winner as team,

COUNT(*) as matches_won

FROM matches

GROUP BY 1,2;

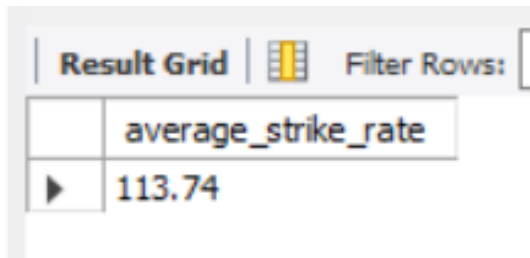
16

season	team	matches_won
2017	Sunrisers Hyderabad	8
2017	Rising Pune Supergiant	10
2017	Kolkata Knight Riders	9
2017	Kings XI Punjab	7
2017	Royal Challengers Bangalore	3
2017	Mumbai Indians	12
2017	Delhi Daredevils	6
2017	Gujarat Lions	4
2008	Kolkata Knight Riders	6
2008	Chennai Super Kings	9
2008	Delhi Daredevils	7
2008	Royal Challengers Bangalore	4
2008	Rajasthan Royals	13
2008	Kings XI Punjab	10
2008	Deccan Chargers	2
2008	Mumbai Indians	7
2009	Mumbai Indians	1
2009	Royal Challengers Bangalore	1
2009	Delhi Daredevils	2
2009	Deccan Chargers	3
2009	Chennai Super Kings	1
2009	Kolkata Knight Riders	1
2009	Rajasthan Royals	1
2009	Kings XI Punjab	1

Result 23 ×

3.WHAT IS THE AVERAGE STRIKE RATE OF BATSMEN IN THE IPL DATASET?

```
WITH result AS(  
  SELECT batsman,  
         (SUM(total_runs)/COUNT(ball))*100 as strike_rate  
  FROM deliveries  
  GROUP BY batsman)  
SELECT  
  ROUND(AVG(strike_rate),2) AS average_strike_rate  
FROM result ;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid has two columns: 'average_strike_rate' and a value '113.74'. There is a 'Filter Rows:' button and a play icon next to the value.

	average_strike_rate
▶	113.74

4.WHAT IS THE NUMBER OF MATCHES WON BY EACH TEAM BATTING FIRST VERSUS BATTING SECOND?

```
select  
  batting_first,  
  count(*) as matches_won  
from(  
  select  
    case when win_by_runs>0 then team1 else team2 end as batting_first  
  from matches  
  where winner!="Tie") as batting_first_teams  
group by 1  
order by 2 desc;
```

Result Grid		Filter Rows:	Export
	batting_first	matches_won	
▶	Mumbai Indians	20	
	Kings XI Punjab	18	
	Kolkata Knight Riders	17	
	Delhi Daredevils	16	
	Rajasthan Royals	13	
	Rising Pune Supergiant	10	
	Chennai Super Kings	9	
	Sunrisers Hyderabad	8	
	Royal Challengers Bangalore	8	
	Deccan Chargers	5	
	Gujarat Lions	4	

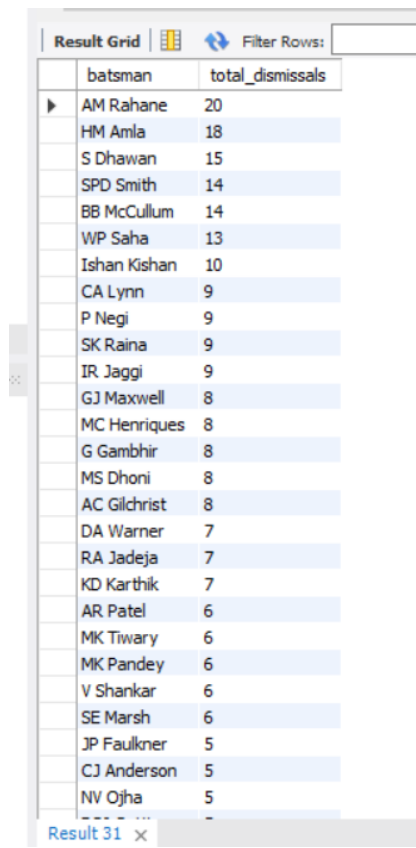
5.WHICH BATSMAN HAS THE HIGHEST STRIKE RATE (MINIMUM 200 RUNS SCORED)?

```
select
    batsman,
    round((sum(batsman_runs)*100/count(*)),2) as strike_rate
from
    deliveries
group by 1
having sum(batsman_runs)>=200
order by 2 desc
limit 1;
```

Result Grid		Filter Rows:
	batsman	strike_rate
▶	V Sehwag	175.78

6.HOW MANY TIMES HAS EACH BATSMAN BEEN DISMISSED BY THE BOWLER 'MALINGA'?

```
select
    batsman,
    count(*) as total_dismissals
from
    deliveries
where player_dismissed is not null
    and bowler='SL Malinga'
group by 1
order by 2 desc;
```



The screenshot shows a SQL query result grid with two columns: 'batsman' and 'total_dismissals'. The results are ordered by the number of dismissals in descending order. The batsmen listed are AM Rahane (20), HM Amla (18), S Dhawan (15), SPD Smith (14), BB McCullum (14), WP Saha (13), Ishan Kishan (10), CA Lynn (9), P Negi (9), SK Raina (9), IR Jaggi (9), GJ Maxwell (8), MC Henriques (8), G Gambhir (8), MS Dhoni (8), AC Gilchrist (8), DA Warner (7), RA Jadeja (7), KD Karthik (7), AR Patel (6), MK Tiwary (6), MK Pandey (6), V Shankar (6), SE Marsh (6), JP Faulkner (5), CJ Anderson (5), and NV Ojha (5). The grid also includes a 'Filter Rows' button and a 'Result 31' indicator at the bottom.

batsman	total_dismissals
AM Rahane	20
HM Amla	18
S Dhawan	15
SPD Smith	14
BB McCullum	14
WP Saha	13
Ishan Kishan	10
CA Lynn	9
P Negi	9
SK Raina	9
IR Jaggi	9
GJ Maxwell	8
MC Henriques	8
G Gambhir	8
MS Dhoni	8
AC Gilchrist	8
DA Warner	7
RA Jadeja	7
KD Karthik	7
AR Patel	6
MK Tiwary	6
MK Pandey	6
V Shankar	6
SE Marsh	6
JP Faulkner	5
CJ Anderson	5
NV Ojha	5

7.WHAT IS THE AVERAGE PERCENTAGE OF BOUNDARIES (FOURS AND SIXES COMBINED) HIT BY EACH BATSMAN?

select

batsman,

avg(case when batsman_runs=4 or batsman_runs=6 then 1 else 0 end)*100 as
average_boundaries

from deliveries

group by 1;

Result Grid

Filter Rows:

	batsman	average_boundaries
▶	DA Warner	19.2641
	S Dhawan	14.8611
	MC Henriques	15.0235
	Yuvraj Singh	20.8617
	DJ Hooda	16.9811
	BCJ Cutting	29.6296
	CH Gayle	17.8988
	Mandeep Singh	14.0449
	TM Head	13.7615
	KM Jadhav	18.0412
	SR Watson	18.1592
	Sachin Baby	11.7647
	STR Binny	11.6883
	S Aravind	5.0000
	YS Chahal	0.0000
	TS Mills	9.0909
	A Choudhary	10.0000
	PA Patel	16.0180
	JC Buttler	22.5806
	RG Sharma	17.3633
	N Rana	14.8014
	AT Rayudu	12.5000
	KH Pandya	16.6667
	KA Pollard	17.0068
	HH Pandya	18.0233
	TG Southee	33.3333
	AM Rahane	12.9683
	MA Agarwal	10.3448
	SPD Smith	12.5000
	BA Stokes	16.2996

Result 33

×

Output

8. WHAT IS THE AVERAGE NUMBER OF BOUNDARIES HIT BY EACH TEAM IN EACH SEASON?

```
select
    season,
    batting_team,
    avg(fours+sixes) as average_boundaries
from
    (select
        season,
        match_id,
        batting_team,
        sum(case when batsman_runs=4 then 1 else 0 end)as fours,
        sum(case when batsman_runs=6 then 1 else 0 end) as sixes
    from deliveries,matches
    where deliveries.match_id=matches.id
    group by 1,2,3) as team_bounsaries
group by 1,2;
```

Result Grid	Filter Rows:	Export:
season	batting_team	average_boundaries
2017	Sunrisers Hyderabad	20.7143
2017	Royal Challengers Bangalore	16.0000
2017	Mumbai Indians	20.1765
2017	Rising Pune Supergiant	17.8750
2017	Gujarat Lions	22.1429
2017	Kolkata Knight Riders	19.6875
2017	Kings XI Punjab	19.7857
2017	Delhi Daredevils	20.6429
2008	Kolkata Knight Riders	18.3077
2008	Royal Challengers Bangalore	17.5714
2008	Chennai Super Kings	20.6250
2008	Kings XI Punjab	21.7333
2008	Rajasthan Royals	21.5000
2008	Delhi Daredevils	19.4286
2008	Mumbai Indians	19.5000
2008	Deccan Chargers	21.2143
2009	Mumbai Indians	19.5000
2009	Chennai Super Kings	18.6667
2009	Royal Challengers Bangalore	17.2500
2009	Rajasthan Royals	11.0000
2009	Kings XI Punjab	16.0000
2009	Delhi Daredevils	16.5000
2009	Kolkata Knight Riders	13.0000
2009	Deccan Chargers	18.3333

9.WHAT IS THE HIGHEST PARTNERSHIP (RUNS) FOR EACH TEAM IN EACH SEASON?

```
select
    season,
    batting_team,
    max(total_runs) as highest_partnership
from(
    select
        season,
        batting_team,
        partnership,
        sum(total_runs) as total_runs
    from(
        select
            season,
            match_id,
            batting_team,
            d.over,
            sum(batsman_runs) as partnership,
            sum(batsman_runs)+sum(extra_runs) as total_runs
        from deliveries d,matches
        where d.match_id=matches.id
        group by 1,2,3,4) as team_scores
    group by 1,2,3) as highest_partnership
group by 1,2
order by 3 desc;
```


Result Grid |   Filter Rows: | Export: 

	season	batting_team	highest_partnership
▶	2017	Mumbai Indians	328
	2017	Gujarat Lions	315
	2008	Kings XI Punjab	288
	2008	Chennai Super Kings	281
	2017	Sunrisers Hyderabad	280
	2008	Rajasthan Royals	274
	2017	Rising Pune Supergiant	260
	2008	Deccan Chargers	253
	2017	Kolkata Knight Riders	246
	2017	Kings XI Punjab	240
	2008	Delhi Daredevils	222
	2008	Kolkata Knight Riders	218
	2017	Delhi Daredevils	216
	2017	Royal Challengers Ban...	210
	2008	Mumbai Indians	210
	2008	Royal Challengers Ban...	198
	2009	Kings XI Punjab	74
	2009	Royal Challengers Ban...	72
	2009	Chennai Super Kings	71
	2009	Deccan Chargers	62
	2009	Mumbai Indians	54
	2009	Kolkata Knight Riders	50

Result 49 ×

10.HOW MANY EXTRAS (WIDES & NO-BALLS) WERE BOWLED BY EACH TEAM IN EACH MATCH?

select

m.id as match_no,

d.bowling_team,

sum(d.extra_runs) as extras

from matches as m

join deliveries as d

on d.match_id=m.id

where extra_runs>0

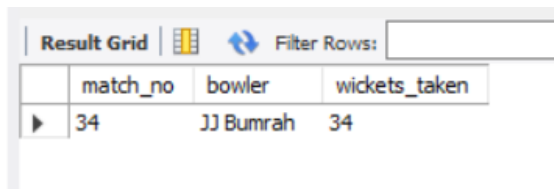
group by 1,2;

Result Grid	Filter Rows:	Export
match_no	bowling_team	extras
1	Royal Challengers Bangalore	7
1	Sunrisers Hyderabad	6
2	Rising Pune Supergiant	8
2	Mumbai Indians	4
3	Kolkata Knight Riders	4
3	Gujarat Lions	15
4	Kings XI Punjab	6
4	Rising Pune Supergiant	10
5	Delhi Daredevils	6
5	Royal Challengers Bangalore	5
6	Sunrisers Hyderabad	3
6	Gujarat Lions	3
7	Mumbai Indians	9
7	Kolkata Knight Riders	12
8	Kings XI Punjab	4
8	Royal Challengers Bangalore	6
9	Rising Pune Supergiant	8
9	Delhi Daredevils	5
10	Mumbai Indians	11
10	Sunrisers Hyderabad	4
11	Kolkata Knight Riders	16
11	Kings XI Punjab	11
12	Mumbai Indians	11
12	Royal Challengers Bangalore	13
13	Gujarat Lions	5
13	Rising Pune Supergiant	5
14	Sunrisers Hyderabad	11
14	Kolkata Knight Riders	7
15	Kings XI Punjab	10

Result 36

11. WHICH BOWLER HAS THE BEST BOWLING FIGURES (MOST WICKETS TAKEN) IN A SINGLE MATCH?

```
select
    m.id as match_no,
    d.bowler,
    count(*) as wickets_taken
from matches as m
join deliveries as d
on d.match_id=m.id
where d.player_dismissed is not null
group by 1,2
order by 3 desc
limit 1;
```



The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' button and a search input field. The grid displays a single row of results for match 34, where JJ Bumrah took 34 wickets.

	match_no	bowler	wickets_taken
▶	34	JJ Bumrah	34

12.HOW MANY MATCHES RESULTED IN A WIN FOR EACH TEAM IN EACH CITY?

```
select
    m.city,
    case
        when m.team1=m.winner then m.team1
        when m.team2=m.winner then m.team2 else 'draw' end as winning_team,
    count(*) as wins
from
    matches as m
join deliveries as d
on d.match_id=m.id
where m.result!='Tie'
group by 1,2;
```

Result Grid			
Filter Rows:			
	city	winning_team	wins
▶	Hyderabad	Sunrisers Hyderabad	1443
	Pune	Rising Pune Supergiant	1152
	Rajkot	Kolkata Knight Riders	218
	Indore	Kings XI Punjab	459
	Bangalore	Royal Challengers Bangalore	492
	Mumbai	Mumbai Indians	2038
	Pune	Delhi Daredevils	226
	Kolkata	Kolkata Knight Riders	1811
	Bangalore	Mumbai Indians	667
	Rajkot	Gujarat Lions	236
	Delhi	Delhi Daredevils	1927
	Bangalore	Rising Pune Supergiant	250
	Delhi	Kolkata Knight Riders	245
	Rajkot	Royal Challengers Bangalore	249
	Indore	Mumbai Indians	225
	Kolkata	Gujarat Lions	232
	Rajkot	Kings XI Punjab	245
	Mumbai	Rising Pune Supergiant	497
	Pune	Kolkata Knight Riders	239
	Bangalore	Gujarat Lions	212
	Chandigarh	Sunrisers Hyderabad	243
	Chandigarh	Kings XI Punjab	1859
	Kolkata	Rising Pune Supergiant	238
	Bangalore	Kings XI Punjab	478
	Hyderabad	Rising Pune Supergiant	246
	Delhi	Mumbai Indians	211
	Bangalore	Kolkata Knight Riders	598
	Chandigarh	Gujarat Lions	244
	Kanpur	Delhi Daredevils	243
	Mumbai	Kings XI Punjab	504
	Kanpur	Sunrisers Hyderabad	228

13.HOW MANY TIMES DID EACH TEAM WIN THE TOSS IN EACH SEASON?

select

season,

toss_winner,

count(*) as toss_wins

from matches

group by 1,2;

Result Grid	Filter Rows:	Expo
season	toss_winner	toss_wins
2017	Royal Challengers Bangalore	9
2017	Rising Pune Supergiant	6
2017	Kolkata Knight Riders	9
2017	Kings XI Punjab	4
2017	Sunrisers Hyderabad	5
2017	Mumbai Indians	11
2017	Gujarat Lions	7
2017	Delhi Daredevils	8
2008	Royal Challengers Bangalore	5
2008	Chennai Super Kings	5
2008	Rajasthan Royals	11
2008	Mumbai Indians	8
2008	Deccan Chargers	9
2008	Kings XI Punjab	8
2008	Kolkata Knight Riders	6
2008	Delhi Daredevils	6
2009	Chennai Super Kings	2
2009	Royal Challengers Bangalore	2
2009	Delhi Daredevils	2
2009	Kolkata Knight Riders	3
2009	Deccan Chargers	2

14.HOW MANY MATCHES DID EACH PLAYER WIN THE "PLAYER OF THE MATCH" AWARD?

select

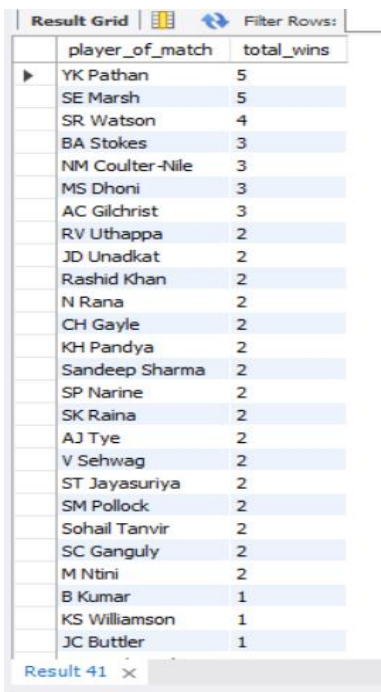
player_of_match,
count(*) as total_wins

from matches

where player_of_match is not null

group by 1

order by 2 desc;





The screenshot shows a SQL query result grid with two columns: 'player_of_match' and 'total_wins'. The results are ordered by 'total_wins' in descending order. The first two players, YK Pathan and SE Marsh, both have 5 wins. SR Watson has 4 wins, and 16 other players have 3 wins each. 16 players have 2 wins each, and 4 players have 1 win each.

player_of_match	total_wins
YK Pathan	5
SE Marsh	5
SR Watson	4
BA Stokes	3
NM Coulter-Nile	3
MS Dhoni	3
AC Gilchrist	3
RV Uthappa	2
JD Unadkat	2
Rashid Khan	2
N Rana	2
CH Gayle	2
KH Pandya	2
Sandeep Sharma	2
SP Narine	2
SK Raina	2
AJ Tye	2
V Sehwag	2
ST Jayasuriya	2
SM Pollock	2
Sohail Tanvir	2
SC Ganguly	2
M Ntini	2
B Kumar	1
KS Williamson	1
JC Buttler	1

15.WHAT IS THE AVERAGE NUMBER OF RUNS SCORED IN EACH OVER OF THE INNINGS IN EACH MATCH?

```
select
    m.id,
    d.inning,
    d.over,
    round(avg(d.total_runs),2) as average_runs_per_over
from matches as m
join deliveries as d
on d.match_id=m.id
group by 1,2,3;
```

Result Grid  Filter Rows: <input type="text"/> Exp				
	id	inning	over	average_runs_per_over
▶	1	1	1	1.00
	1	1	2	2.29
	1	1	3	1.00
	1	1	4	0.67
	1	1	5	1.50
	1	1	6	2.83
	1	1	7	0.83
	1	1	8	1.83
	1	1	9	1.50
	1	1	10	0.67
	1	1	11	1.67
	1	1	12	1.33
	1	1	13	2.57
	1	1	14	1.33
	1	1	15	3.17
	1	1	16	0.67
	1	1	17	1.17
	1	1	18	2.13
	1	1	19	2.00
	1	1	20	2.67
	1	2	1	1.00

Result 47 

16. WHICH TEAM HAS THE HIGHEST TOTAL SCORE IN A SINGLE MATCH?

SELECT

m.season,

m.id as match_no,

d.batting_team,

SUM(d.total_runs) as total_score

FROM matches as m

JOIN

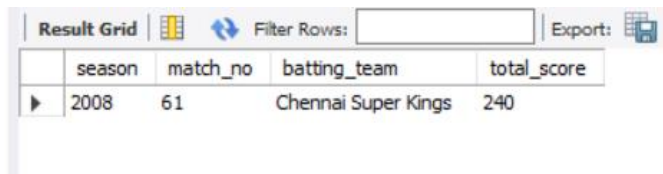
deliveries as d

ON d.match_id=m.id

GROUP BY 1,2,3

ORDER BY 4 DESC

LIMIT 1;



The screenshot shows a database interface with a 'Result Grid' tab. Above the grid, there is a 'Filter Rows:' input field and an 'Export:' button with a grid icon. The result grid contains a single row of data with the following columns: season, match_no, batting_team, and total_score.

	season	match_no	batting_team	total_score
▶	2008	61	Chennai Super Kings	240

17. WHICH BATSMAN HAS SCORED THE MOST RUNS IN A SINGLE MATCH?

SELECT

```
m.season,  
m.id as match_no,  
d.batsman,  
sum(d.batsman_runs) as total_runs
```

FROM

```
matches as m
```

JOIN

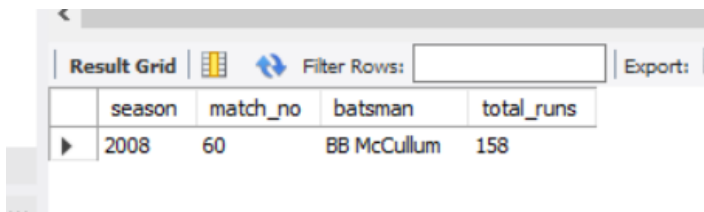
```
deliveries as d
```

```
ON d.match_id=m.id
```

```
GROUP BY 1,2,3
```

```
ORDER BY 4 desc
```

```
LIMIT 1;
```



The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' search bar and an 'Export' button. The table below displays the results of the SQL query, with columns for season, match_no, batsman, and total_runs. The first row shows BB McCullum with 158 runs in match 60 of the 2008 season.

	season	match_no	batsman	total_runs
▶	2008	60	BB McCullum	158