Fetching data using SQL Queries (Postgres)

Dataset   - IPL Data
Language   - SQL
DBMS Tool - Postgres

The following document contains several use cases and their possible solutions, created using SQL, for the IPL (Indian Premier League) dataset.

Use Case 1

Team batting second winning the match when the required run rate is above 12 runs an over.

Query:

SELECT matchid,batting_team,team1||' vs '||team2 AS
Match,season,SUM(total_runs) AS runs_needed,COUNT(player_dismissed) AS
wickets_taken
FROM matches as m
INNER JOIN deliveries as d
ON m.id=d.matchid
WHERE over>=17 AND inning=2 AND batting_team=winner
GROUP BY matchid,batting_team,team1,team2,season HAVING
SUM(total_runs)>48
ORDER BY season, SUM(total_runs) DESC;

Result:

	matchid integer	batsman character varying	team character varying	game text	season integer	runs bigint	balls bigint
1	146	DA Warner	Delhi Daredevils	Kolkata Knight Riders vs Delhi Daredevils	2009	36	23
2	154	DA Warner	Delhi Daredevils	Kolkata Knight Riders vs Delhi Daredevils	2009	36	22
3	197	DA Warner	Delhi Daredevils	Delhi Daredevils vs Royal Challengers Bangalore	2010	33	14
4	200	DA Warner	Delhi Daredevils	Delhi Daredevils vs Kolkata Knight Riders	2010	107	68
5	209	DA Warner	Delhi Daredevils	Delhi Daredevils vs Royal Challengers Bangalore	2010	33	21
6	250	DA Warner	Delhi Daredevils	Pune Warriors vs Delhi Daredevils	2011	46	28
7	259	DA Warner	Delhi Daredevils	Delhi Daredevils vs Kings XI Punjab	2011	77	48
8	331	DA Warner	Delhi Daredevils	Deccan Chargers vs Delhi Daredevils	2012	109	53
9	374	DA Warner	Delhi Daredevils	Kings XI Punjab vs Delhi Daredevils	2012	79	40

Total rows: 33 of 33   Query complete 00:00:00.062

Use Case 2
CSK Performance when CSK first won the IPL in 2010 – Creating a view

Query:

CREATE VIEW csk_2010 AS
SELECT date, CASE
        WHEN team1='Chennai Super Kings' THEN team2
        ELSE team1
        END
        AS opposition,
        CASE
        WHEN winner='Chennai Super Kings' THEN 'win'
        ELSE 'loss'
        END
        AS Result,
venue, win_by_runs, win_by_wickets,player_of_the_match FROM matches
WHERE (team1='Chennai Super Kings' OR team2='Chennai Super Kings') AND
season=2010

Result:

List of matches where a team won a match after losing 3 wickets during powerplay between 2008-2016.

Query :

SELECT matchid,team1||' vs '||team2 AS match, winner, season,
COUNT(player_dismissed) AS wickets_lost
FROM matches as m
INNER JOIN deliveries AS d
ON d.matchid=m.id
WHERE over<=6 AND batting_team=winner
GROUP BY matchid,inning,team1,team2,season, winner HAVING
COUNT(player_dismissed)>=3
ORDER BY season,wickets_lost DESC;

Result:

Data output	Messages	Notifications

	matchid integer	match text	winner character varying	season integer	wickets_lost bigint
1	75	Kolkata Knight Riders vs Mumbai Indians	Mumbai Indians	2008	3
2	100	Delhi Daredevils vs Kings XI Punjab	Kings XI Punjab	2008	3
3	65	Kings XI Punjab vs Rajasthan Royals	Rajasthan Royals	2008	3
4	64	Deccan Chargers vs Kolkata Knight Riders	Kolkata Knight Riders	2008	3
5	105	Kolkata Knight Riders vs Rajasthan Royals	Rajasthan Royals	2008	3
6	159	Chennai Super Kings vs Royal Challengers Bangalore	Royal Challengers Bangalo...	2009	3
7	140	Deccan Chargers vs Rajasthan Royals	Rajasthan Royals	2009	3
8	161	Delhi Daredevils vs Kings XI Punjab	Kings XI Punjab	2009	3
9	168	Rajasthan Royals vs Kolkata Knight Riders	Kolkata Knight Riders	2009	3

Total rows: 46 of 46   Query complete 00:00:00.080

Use Case 4
Updating incorrect data in Table

Query:

UPDATE win_after_losing_3_pp_wickets
SET winner = 'Rising Pune Supergiants'
WHERE winner LIKE '%giant';

Result:

```
Data output    Messages    Notifications

UPDATE 1

Query returned successfully in 48 msec.
```

Use Case 5
Number of matches when the team wins a Match when David Warner scores greater than 30 runs

Query:

SELECT d.matchid,d.batsman,d.batting_team AS Team,m.team1||' vs '||m.team2 AS
Game,season,SUM(d.batsman_runs) AS Runs,COUNT(d.batsman_runs)-
SUM(noball_runs)-SUM(wide_runs) AS Balls
FROM matches as m
INNER JOIN deliveries AS d
ON d.matchid=m.id
WHERE d.batsman LIKE '%Warner' AND m.winner=d.batting_team
GROUP BY
d.batsman,d.matchid,d.batting_team,m.team1,m.team2,season,d.batting_team
HAVING SUM(d.batsman_runs) >= 30
ORDER BY season;

Result:

	matchid integer	batsman character varying	team character varying	game text	season integer	runs bigint	balls bigint
1	146	DA Warner	Delhi Daredevils	Kolkata K...	2009	36	23
2	154	DA Warner	Delhi Daredevils	Kolkata K...	2009	36	22
3	197	DA Warner	Delhi Daredevils	Delhi Dar...	2010	33	14
4	200	DA Warner	Delhi Daredevils	Delhi Dar...	2010	107	68
5	209	DA Warner	Delhi Daredevils	Delhi Dar...	2010	33	21
6	250	DA Warner	Delhi Daredevils	Pune War...	2011	46	28
7	259	DA Warner	Delhi Daredevils	Delhi Dar...	2011	77	48
8	331	DA Warner	Delhi Daredevils	Deccan C...	2012	109	53
9	374	DA Warner	Delhi Daredevils	Kings XI ...	2012	79	40

Total rows: 33 of 33   Query complete 00:00:00.083

Use Case 6

All stats for Jasprit Bumrah in the year 2017 when Mumbai Indians won the IPL
Note: This code can be used to get the bowling stats for any player in any season.

Query:

SELECT bowler,season,
SUM(CASE
       WHEN dismissal_kind='run out' THEN 0
       WHEN dismissal_kind IS NULL THEN 0
       ELSE 1
       END) AS wickets,
       SUM(total_runs-legbye_runs-bye_runs) AS runs,

       SUM(total_runs-legbye_runs-bye_runs)/SUM(CASE
       WHEN dismissal_kind='run out' THEN 0

```sql
WHEN dismissal_kind IS NULL THEN 0
ELSE 1
END) AS average,

SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)
    AS balls,

  SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)/SUM(CASE
WHEN dismissal_kind='run out' THEN 0
WHEN dismissal_kind IS NULL THEN 0
ELSE 1
END) AS strike_rate,

  CAST((SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)/6||'.'||
 (SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)%6)) AS float) AS overs,

  SUM(total_runs-legbye_runs-bye_runs)/
  CAST((SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)/6||'.'||
(((SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
```

END)%6)*10000)/6)) AS float) AS Economy

    FROM deliveries as d
INNER JOIN matches as m
ON d.matchid=m.id
WHERE bowler LIKE '%Bumrah' AND season=2017
GROUP BY bowler,season HAVING SUM(CASE
    WHEN dismissal_kind='run out' THEN 0
    WHEN dismissal_kind IS NULL THEN 0
    ELSE 1
    END) >0 AND CAST((SUM(CASE
      WHEN noball_runs=0 AND wide_runs=0 THEN 1
      ELSE 0
    END)/6||'.'||
    (SUM(CASE
      WHEN noball_runs=0 AND wide_runs=0 THEN 1
      ELSE 0
    END)%6)) AS float)>10
ORDER BY economy, season desc
LIMIT 100;

Result:

bowler character varying	season integer	wickets bigint	runs bigint	average bigint	balls bigint	strike_rate bigint	overs double precision	economy double precision
1 JJ Bumrah	2017	20	445	22	362	18	60.2	7.37569468270424

Use Case 7

Bowlers economy in IPL 2015 where the bowlers have bowled at least 10 overs

Query:

SELECT bowler, season,
SUM(CASE
    WHEN dismissal_kind='run out' THEN 0
    WHEN dismissal_kind IS NULL THEN 0
    ELSE 1

```
END) AS wickets,
SUM(total_runs-legbye_runs-bye_runs) AS runs,

SUM(total_runs-legbye_runs-bye_runs)/SUM(CASE
WHEN dismissal_kind='run out' THEN 0
WHEN dismissal_kind IS NULL THEN 0
ELSE 1
END) AS average,

SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)
    AS balls,

  SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)/SUM(CASE
WHEN dismissal_kind='run out' THEN 0
WHEN dismissal_kind IS NULL THEN 0
ELSE 1
END) AS strike_rate,

  CAST((SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)/6||'.'||
 (SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
    ELSE 0
  END)%6)) AS float) AS overs,

  SUM(total_runs-legbye_runs-bye_runs)/
  CAST((SUM(CASE
    WHEN noball_runs=0 AND wide_runs=0 THEN 1
```

```sql
                ELSE 0
            END)/6||'.'||
          (((SUM(CASE
                WHEN noball_runs=0 AND wide_runs=0 THEN 1
                ELSE 0
              END)%6)*10000)/6)) AS float) AS Economy

        FROM deliveries as d
INNER JOIN matches as m
ON d.matchid=m.id
WHERE over BETWEEN 1 AND 6 AND season=2016
GROUP BY bowler,season HAVING SUM(CASE
            WHEN dismissal_kind='run out' THEN 0
            WHEN dismissal_kind IS NULL THEN 0
            ELSE 1
            END) >0 AND CAST((SUM(CASE
                WHEN noball_runs=0 AND wide_runs=0 THEN 1
                ELSE 0
              END)/6||'.'||
            (SUM(CASE
                WHEN noball_runs=0 AND wide_runs=0 THEN 1
                ELSE 0
              END)%6)) AS float)>10
ORDER BY economy,season
LIMIT 100;
```

Result:

	bowler character varying	season integer	wickets bigint	runs bigint	average bigint	balls bigint	strike_rate bigint	overs double precision	economy double precision
1	Mustafizur Rahm...	2016	3	79	26	96	32	16	4.9375
2	Sandeep Sharma	2016	6	156	26	162	27	27	5.777777777777778
3	B Kumar	2016	10	213	21	216	21	36	5.916666666666667
4	AD Russell	2016	8	122	15	120	15	20	6.1
5	AB Dinda	2016	6	130	21	126	21	21	6.190476190476191
6	P Kumar	2016	3	247	82	228	76	38	6.5
7	Z Khan	2016	5	187	37	168	33	28	6.678571428571429
8	DS Kulkarni	2016	14	219	15	192	13	32	6.84375
9	SR Watson	2016	4	124	31	108	27	18	6.888888888888889

Total rows: 21 of 21   Query complete 00:00:00.078

Use Case 8
Team batting average in 2011

Query:

SELECT season,SUM(total_runs)/COUNT(DISTINCT matchid) AS
season_average,batting_team
FROM matches as m
INNER JOIN deliveries as d
ON m.id=d.matchid
WHERE season=2010
GROUP BY batting_team,season
ORDER BY season,SUM(total_runs)/COUNT(DISTINCT matchid) DESC;

Result:

	season integer	season_average bigint	batting_team character varying
1	2010	171	Mumbai Indians
2	2010	163	Kings XI Punjab
3	2010	162	Chennai Super Kings
4	2010	155	Rajasthan Royals
5	2010	153	Kolkata Knight Riders
6	2010	153	Delhi Daredevils
7	2010	150	Royal Challengers Bangalo...
8	2010	148	Deccan Chargers

Total rows: 8 of 8   Query complete 00:00:00.087