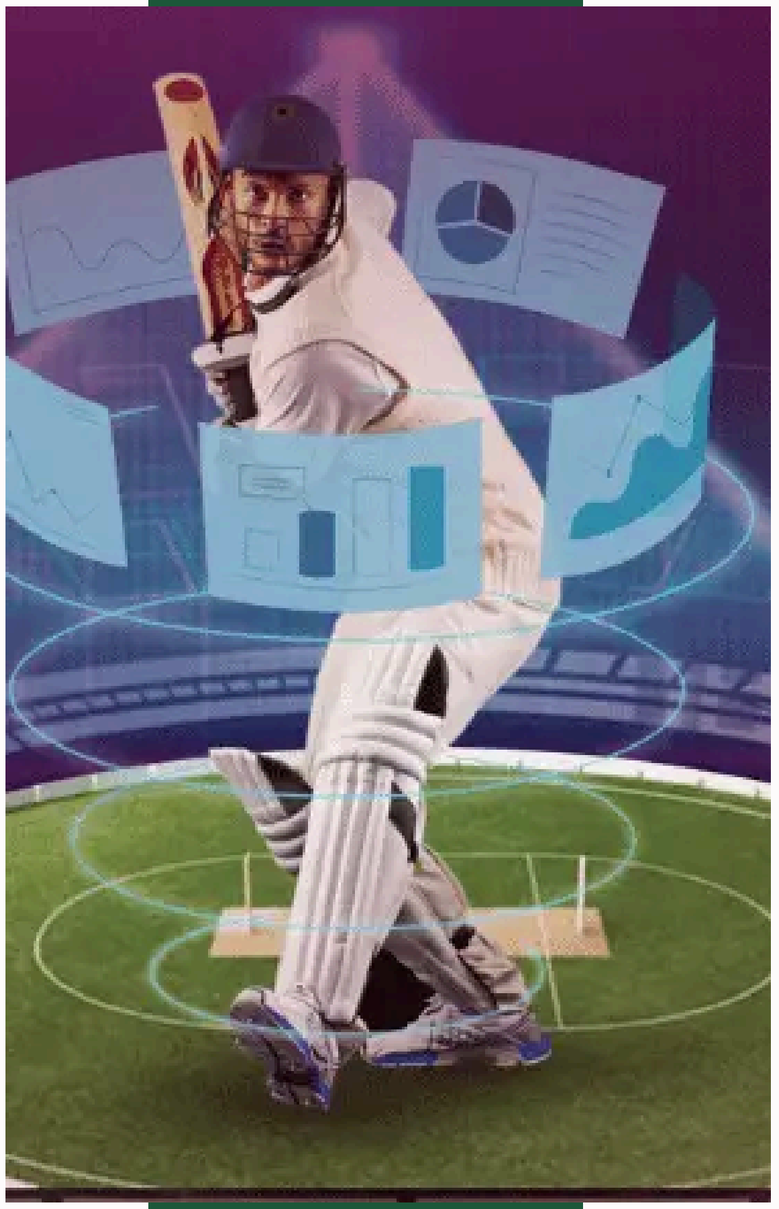
Digital Learning

Presentation 2024

IPL2023 CRICKET ANALYSIS: SQL AT THE CREASE

Presented by Mr. Vaibhav Vishnu Pawar
www.linkedin.com/in/vaibhavpawar000



Basic

PROBLEMS

```sql
-- ### Basic Problems:
-- 1.Retrieve Batsmen Information: Select all columns from the batsmen table
SELECT * FROM IPL2023_Batsman;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

match_no	Batsman	team	run	ball	fours	sixes	out_by
1	Devon Conway	Chennai Super Kings	1	6	0	0	Mohammed Shami
1	Ruturaj Gaikwad	Chennai Super Kings	92	50	4	9	Alzarri Joseph
1	Moeen Ali	Chennai Super Kings	23	17	4	1	Rashid Khan
1	Ben Stokes	Chennai Super Kings	7	6	1	0	Rashid Khan
1	Ambati Rayudu	Chennai Super Kings	12	12	0	1	Joshua Little
1	Shivam Dube	Chennai Super Kings	19	18	0	1	Mohammed Shami
1	Ravindra Jadeja	Chennai Super Kings	1	2	0	0	Alzarri Joseph
1	MS Dhoni	Chennai Super Kings	14	7	1	1	Not Out
1	Mitchell Santner	Chennai Super Kings	1	3	0	0	Not Out
1	Wriddhiman Saha	Gujarat Titans	25	16	2	2	Rajvardhan Hangargekar
1	Shubman Gill	Gujarat Titans	63	36	6	3	Tushar Deshpande
1	Sai Sudharsan	Gujarat Titans	22	17	3	0	Rajvardhan Hangargekar
1	Hardik Pandya	Gujarat Titans	8	11	0	0	Ravindra Jadeja

```sql
-- 2.Calculate Total Runs by Batsman: Calculate the total runs scored by each batsman.
SELECT
    Batsman, Team, SUM(run) AS Total_Runs
FROM
    IPL2023_Batsman
GROUP BY Batsman , Team
ORDER BY Total_Runs DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Batsman	Team	Total_Runs
Shubman Gill	Gujarat Titans	743
Faf du Plessis	Royal Challengers Bangalore	730
Ruturaj Gaikwad	Chennai Super Kings	673
Virat Kohli	Royal Challengers Bangalore	639
Yashasvi Jaiswal	Rajasthan Royals	625
David Warner	Delhi Capitals	516
Suryakumar Yadav	Mumbai Indians	511
Devon Conway	Chennai Super Kings	509
Rinku Singh	Kolkata Knight Riders	474
Heinrich Klaasen	Sunrisers Hyderabad	448
Ishan Kishan	Mumbai Indians	439
Shikhar Dhawan	Punjab Kings	413
Nitish Rana	Kolkata Knight Riders	413

Result 4

```sql
-- 3. Top Batsmen by 6s: Find the top 5 batsmen who hit the most sixes.
SELECT
    Batsman, Team, SUM(sixes) AS TOP_5_Sixer
FROM
    IPL2023_Batsman
GROUP BY Batsman , Team
ORDER BY TOP_5_Sixer DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export:

Batsman	Team	TOP_5_Sixer
Ruturaj Gaikwad	Chennai Super Kings	38
Faf du Plessis	Royal Challengers Bangalore	36
Shivam Dube	Chennai Super Kings	34
Glenn Maxwell	Royal Challengers Bangalore	31
Rinku Singh	Kolkata Knight Riders	29

```sql
-- 4.Bowlers Information: Retrieve all columns from the bowlers table.
SELECT
    *
FROM
    IPL2023_Bowler;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

match_no	Bowler	team	overs	run	wicket	No_ball	ECO
1	Mohammed Shami	Gujarat Titans	4	29	2	1	7.3
1	Hardik Pandya	Gujarat Titans	3	28	0	0	9.3
1	Joshua Little	Gujarat Titans	4	41	1	0	10.3
1	Rashid Khan	Gujarat Titans	4	26	2	0	6.5
1	Alzarri Joseph	Gujarat Titans	4	33	2	0	8.3
1	Yash Dayal	Gujarat Titans	1	14	0	0	14
1	Deepak Chahar	Chennai Super Kings	4	29	0	0	7.3
1	Tushar Deshpande	Chennai Super Kings	3	51	1	1	15.9
1	Rajvardhan Hangargekar	Chennai Super Kings	4	36	3	1	9
1	Mitchell Santner	Chennai Super Kings	4	32	0	0	8
1	Ravindra Jadeja	Chennai Super Kings	4	28	1	0	7
2	Umesh Yadav	Kolkata Knight Riders	4	27	1	0	6.8
2	Tim Southee	Kolkata Knight Riders	4	54	2	0	13.5

IPL2023_Bowler 9

```sql
-- 5. Calculate Bowling Economy: Calculate and display the bowling economy of each bowler
SELECT
    Bowler, Team, SUM(run) / SUM(overs) * 6 AS bowler_economy
FROM
    IPL2023_Bowler
GROUP BY Bowler , Team
ORDER BY bowler_economy
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Bowler	Team	bowler_economy
Ayush Badoni	Lucknow Super Giants	NULL
Matthew Short	Punjab Kings	37.5000
Praveen Dubey	Delhi Capitals	38.0000
Jayant Yadav	Gujarat Titans	39.0000
Glenn Phillips	Sunrisers Hyderabad	40.0000
Mitchell Santner	Chennai Super Kings	40.5000
Reece Topley	Royal Challengers Bangalore	42.0000
David Willey	Royal Challengers Bangalore	42.0000
Dwaine Pretorius	Chennai Super Kings	42.0000
Tilak Varma	Mumbai Indians	42.0000
Marco Jansen	Lucknow Super Giants	42.0000
Tristan Stubbs	Mumbai Indians	42.0000
Joe Root	Rajasthan Royals	42.0000

Result 12

6. Team-wise Runs: Calculate the total runs scored by each team across all matches.

```sql
-- 9. Team-wise Runs: Calculate the total runs scored by each team across all matches.
SELECT
    team, SUM(run) AS Total_Runs
FROM
    IPL2023_Batsman
GROUP BY team
ORDER BY Total_Runs DESC;
```

Result Grid | Filter Rows:

team	Total_Runs
Punjab Kings	2603
Gujarat Titans	2501
Mumbai Indians	2491
Chennai Super Kings	2437
Royal Challengers Bangalore	2401
Kolkata Knight Riders	2338
Rajasthan Royals	2287
Sunrisers Hyderabad	2268
Lucknow Super Giants	2235
Delhi Capitals	2063

7. Find Matches in a City: Retrieve all matches that took place in a specific city.

```sql
SELECT
    match_no, city, date_of_match, venue
FROM
    IPL2023_Matches
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

match_no	city	date_of_match	venue
1	Ahmedabad	3/31/2023	Narendra Modi Stadium
2	Mohali	4/1/2023	Punjab Cricket Association IS Bindra Stadium
3	Lucknow	4/1/2023	Bharat Ratna Shri Atal Bihari Vajpayee Ekana Cr...
4	Hyderabad	4/2/2023	Rajiv Gandhi International Stadium
5	Bengaluru	4/2/2023	M.Chinnaswamy Stadium
6	Chennai	4/3/2023	MA Chidambaram Stadium
7	Delhi	4/4/2023	Arun Jaitley Stadiumi
8	Guwahati	4/5/2023	Barsapara Cricket Stadium
9	Kolkata	4/6/2023	Eden Gardens
10	Lucknow	4/7/2023	Bharat Ratna Shri Atal Bihari Vajpayee Ekana Cr...
11	Guwahati	4/8/2023	Barsapara Cricket Stadium
12	Mumbai	4/8/2023	Wankhede Stadium
13	Ahmedabad	4/9/2023	Narendra Modi Stadium

Intermediate

PROBLEMS

These problems cover a range of SQL concepts such as
basic queries, joins, aggregation functions, conditional filtering, and more.

8. Joining Tables: Join batsman and bowlers tables to get the match-wise performance of both batsman and bowlers.

```sql
SELECT
    bat.match_no,
    bat.Batsman,
    bat.team,
    bat.run,
    bat.fours,
    bat.sixes,
    bat.out_by,
    ball.bowler,
    ball.team,
    ball.overs,
    ball.wicket,
    ball.No_ball,
    ball.ECO
FROM
    IPL2023_batsman AS bat
        JOIN
    IPL2023_bowler AS ball ON bat.match_no = ball.match_no;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

match_no	Batsman	team	run	fours	sixes	out_by	bowler	team	overs	wicket	No_ball	ECO
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Ravindra Jadeja	Chennai Super Kings	4	1	0	7
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Mitchell Santner	Chennai Super Kings	4	0	0	8
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Rajvardhan Hangargekar	Chennai Super Kings	4	3	1	9
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Tushar Deshpande	Chennai Super Kings	3	1	1	15.9
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Deepak Chahar	Chennai Super Kings	4	0	0	7.3
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Yash Dayal	Gujarat Titans	1	0	0	14
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Alzarri Joseph	Gujarat Titans	4	2	0	8.3
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Rashid Khan	Gujarat Titans	4	2	0	6.5
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Joshua Little	Gujarat Titans	4	1	0	10.3
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Hardik Pandya	Gujarat Titans	3	0	0	9.3
1	Devon Conway	Chennai Super Kings	1	0	0	Mohammed Shami	Mohammed Shami	Gujarat Titans	4	2	1	7.3
1	Ruturaj Gaikwad	Chennai Super Kings	92	4	9	Alzarri Joseph	Ravindra Jadeja	Chennai Super Kings	4	1	0	7
1	Ruturaj Gaikwad	Chennai Super Kings	92	4	9	Alzarri Joseph	Mitchell Santner	Chennai Super Kings	4	0	0	8

9.1 Highest Run Scorer: Find the batsman who scored the highest runs in a single match.
9.2 Highest Run Scorer: Find the batsman who scored the highest runs in a each match.
9.3 maximum run scored in each match along with the corresponding batsman, without considering ties.

```sql
-- 7.Highest Run Scorer: Find the batsman who scored the
-- highest runs in a single match.
SELECT
    match_no, batsman, Run
FROM
    IPL2023_Batsman
        JOIN
    (SELECT
        match_no as match_no_MAx, MAX(Run) AS Max_Run
    FROM
        IPL2023_Batsman
    GROUP BY match_no) AS MAX_Score
        ON IPL2023_Batsman.match_no = MAX_Score.match_no_MAx
        AND IPL2023_Batsman.Run = MAX_Score.Max_Run;
```

Result Grid | Filter Rows:

match_no	batsman	Run
1	Ruturaj Gaikwad	92
2	Bhanuka Rajapaksa	50
1	Ruturaj Gaikwad	92
2	Bhanuka Rajapaksa	50
3	Kyle Mayers	73
4	Sanju Samson	55
5	Tilak Varma	84
6	Ruturaj Gaikwad	57
7	Sai Sudharsan	62
8	Shikhar Dhawan	86
9	Shardul Thakur	68
10	Rahul Tripathi	35
10	KL Rahul	35

```sql
-- 7.2 Highest Run Scorer: Find the batsman who scored the
-- highest runs in a each match.
WITH RANK_BATSMEN AS (
    SELECT
        match_no,
        Batsman,
        run,
        row_number() OVER (PARTITION BY match_no ORDER BY run DESC) AS Ranking
    FROM IPL2023_Batsman
    )
SELECT
    match_no,
    Batsman,
    run
    FROM RANK_BATSMEN
    WHERE Ranking = 1;
```

Result Grid | Filter Rows:

match_no	Batsman	run
1	Ruturaj Gaikwad	92
2	Bhanuka Rajapaksa	50
3	Kyle Mayers	73
4	Sanju Samson	55
5	Tilak Varma	84
6	Ruturaj Gaikwad	57
7	Sai Sudharsan	62
8	Shikhar Dhawan	86
9	Shardul Thakur	68
10	Rahul Tripathi	35
11	Jos Buttler	79
12	Ajinkya Rahane	61
13	Venkatesh Iyer	83

Result 25

```sql
-- 7.3 Highest Run Scorer: Find the batsman who scored the
-- highest runs in a each match,without considering ties.
SELECT match_no,Batsman,Max(run) as Max_Run
FROM IPL2023_Batsman
GROUP BY match_no
```

Result Grid | Filter Rows:

match_no	Batsman	Max_Run
1	Devon Conway	92
2	Prabhsimran Singh	50
3	KL Rahul	73
4	Yashasvi Jaiswal	55
5	Rohit Sharma	84
6	Ruturaj Gaikwad	57
7	David Warner	62
8	Prabhsimran Singh	86
9	Rahmanullah Gurbaz	68
10	Anmolpreet Singh	35
11	Yashasvi Jaiswal	79
12	Rohit Sharma	61
13	Wriddhiman Saha	83

Result 29

The three queries have different behaviors and will produce different outputs: key difference lies in how they handle ties:

9.1 The first query returns all batsmen tied for the highest runs in a match.

9.2 The second query returns only one of the batsmen tied for the highest runs in a match arbitrarily.

9.3 The third query only returns the maximum run scored in each match along with the corresponding batsman, without considering ties.

10. Best Bowling Performance: Identify the bowler with the best bowling figures (most wickets in a single match with the lowest economy).

```sql
-- 8.Best Bowling Performance: Identify the bowler with the best bowling figures
-- (most wickets in a single match with the lowest economy).


SELECT
    match_no, Bowler, team, wicket, ECO
FROM
    IPL2023_Bowler
        JOIN
    (SELECT
        match_no AS M_N,
            MAX(wicket) AS Max_Wicket,
            MIN(ECO) AS Min_ECO
    FROM
        IPL2023_Bowler
    GROUP BY M_N) AS BEST_PERFORMER ON IPL2023_Bowler.match_no = BEST_PERFORMER.M_N
        AND IPL2023_Bowler.ECO = BEST_PERFORMER.Min_ECO
GROUP BY match_no
ORDER BY wicket DESC , ECO ASC
LIMIT 1;
```

Result Grid | Filter Rows: | Export:

match_no	Bowler	team	wicket	ECO
3	Mark Wood	Lucknow Super Giants	5	3.5

```sql
SELECT
    Match_no,
    bowler,
    Team,
    MAX(Wicket) AS Max_Wickets,
    MIN(ECO) AS Min_Economy
FROM
    IPL2023_Bowler
GROUP BY Match_no , bowler , Team
ORDER BY Max_Wickets DESC , Min_Economy ASC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

Match_no	bowler	Team	Max_Wickets	Min_Economy
3	Mark Wood	Lucknow Super Giants	5	3.5

If the requirement is simply to find the bowler with the best bowling performance in terms of most wickets and lowest economy rate in each match, 1'st query might suffice.

Advance
PROBLEMS

These advanced problems require a deeper understanding of SQL concepts, as well as statistical analysis and domain knowledge of cricket. They'll challenge us to apply more advanced SQL techniques and perform complex analyses on the dataset.

11. Player Consistency Analysis: Analyze the consistency of batsmen by calculating the coefficient of variation (standard deviation of runs scored divided by the mean runs scored) for each player across all matches.

```sql
WITH PlayseStats AS
(
SELECT
Batsman,
COUNT(DISTINCT(match_no)) AS No_of_Matches_Played,
team,
ROUND(STDDEV(run),3) AS STD_Dev,
ROUND(AVG(run),3) AS AVG_Runs
FROM ipl2023_batsman
GROUP BY Batsman)
SELECT
Batsman,
team,
No_of_Matches_Played,
AVG_Runs,
STD_Dev,
ROUND((STD_Dev/NULLIF(AVG_Runs,0)),2) AS CV
FROM PlayseStats
WHERE No_of_Matches_Played > 1
GROUP BY Batsman
ORDER BY No_of_Matches_Played DESC,CV ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Batsman	team	No_of_Matches_Played	AVG_Runs	STD_Dev	CV
Faf du Plessis	Royal Challengers Bangalore	14	52.143	21.692	0.42
Rinku Singh	Kolkata Knight Riders	14	33.857	20.396	0.6
Shubman Gill	Gujarat Titans	14	49.533	32.786	0.66
Ishan Kishan	Mumbai Indians	14	31.357	20.971	0.67
David Warner	Delhi Capitals	14	36.857	25.746	0.7
Virat Kohli	Royal Challengers Bangalore	14	45.643	32.912	0.72
Nitish Rana	Kolkata Knight Riders	14	29.500	22.828	0.77
Nicholas Pooran	Lucknow Super Giants	14	25.571	20.444	0.8
Yashasvi Jaiswal	Rajasthan Royals	14	44.643	35.668	0.8
Suryakumar Yadav	Mumbai Indians	14	36.500	31.029	0.85
Sanju Samson	Rajasthan Royals	14	25.857	23.271	0.9
Andre Russell	Kolkata Knight Riders	14	16.214	15.048	0.93
Rohit Sharma	Mumbai Indians	14	22.357	21.039	0.94

=> To calculate the players who have played in more than one match,we introduced count(Distinct(match_no)) ,
for accurate result we need to niglify the players who were not played even a single match, we filtered this by 'SELECT' statement by using 'WHERE' Condition
(WHERE No_of_Matches_Played > 1).

=> We use the **NULLIF()** function to handle cases where the mean runs (Mean_Runs) are 0. This prevents division by zero errors and ensures that the CV is calculated correctly for players with non-zero mean runs.

=> NOTE: The NULLIF() function compares expression 1 with expression 2. If the two expressions are equal, the function returns NULL. If they are not equal, the function returns

12. Bowling Strike Rate: Calculate the bowling strike rate for each bowler (average number of balls bowled per wicket taken) and identify the bowler with the best strike rate.

```sql
SELECT
    Bowler,
    team,
    SUM(overs) AS Total_Overs,
    SUM(overs) * 6 AS Total_Balls,
    SUM(wicket) AS Total_Wickets,
    SUM(overs) * 6 / SUM(IFNULL(wicket, 0)) AS Strike_Rate
FROM
    IPL2023_bowler
GROUP BY Bowler
HAVING Strike_Rate >= 1
ORDER BY Strike_Rate ASC;
```

In cricket, a lower bowling strike rate is generally considered better because it indicates that the bowler takes wickets more frequently, meaning they are more effective at dismissing batsmen. Therefore, bowlers with a lower strike rate are often considered more valuable to their team.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Bowler	team	Total_Overs	Total_Balls	Total_Wickets	Strike_Rate
Rajvardhan Hangargekar	Chennai Super Kings	4	24	3	8.0000
Mark Wood	Lucknow Super Giants	16	96	11	8.7273
Mitchell Marsh	Delhi Capitals	20	120	12	10.0000
Michael Bracewell	Royal Challengers Bangalore	11	66	6	11.0000
Navdeep Saini	Rajasthan Royals	6	36	3	12.0000
Reece Topley	Royal Challengers Bangalore	2	12	1	12.0000
Rovman Powell	Delhi Capitals	2	12	1	12.0000
Kulwant Khejroliya	Kolkata Knight Riders	4	24	2	12.0000
Tristan Stubbs	Mumbai Indians	2	12	1	12.0000
Karn Sharma	Royal Ch		126	10	12.6000
Andre Russell	Kolkata Knight Riders	15	90	7	12.8571
Mohit Sharma	Gujarat Titans	35	210	16	13.1250
Rashid Khan	Gujarat Titans	56	336	25	13.4400

Royal Challengers Bangalore

“Rajvardhan Hangargekar” is the bowler with the best strike rate, he belongs to Chennai Super Kings team.

We can achieve it by using limit clause.

```sql
ORDER BY Strike_Rate ASC
Limit 1 ;
```

13. Home Advantage Analysis: Determine whether there is a significant difference in the performance of home teams versus away teams by comparing their average runs scored and wickets taken per match.

```
WITH HomeStats AS (
SELECT AVG(Home_team_run) AS AVG_HOME_TEAM_RUNS,
AVG(Home_team_wickets) AS AVG_HOME_TEAM_Wickets
FROM ipl2023_match_scoreboard),

AwayStats AS (
SELECT AVG(Away_team_run) AS AWAY_TEAM_RUNS,
AVG(Away_team_wickets) AS AWAY_TEAM_WICKETS
FROM ipl2023_match_scoreboard)

SELECT
     "Home" AS TEAM_TYPE, AVG_HOME_TEAM_RUNS AS AVG_Runs,
     AVG_HOME_TEAM_Wickets AS AVG_Wickets
     FROM HomeStats
UNION ALL
SELECT
     "AWAY" AS TEAM_TYPE, AWAY_TEAM_RUNS AS AVG_Runs,
     AWAY_TEAM_WICKETS AS AVG_Wickets
     FROM AwayStats;
```

Result Grid | Filter Rows:

TEAM_TYPE	AVG_Runs	AVG_Wickets
Home	173.8429	6.2286
AWAY	173.1143	6.0143

UNION:
Combines the results of two or more SELECT statements into a single result set,
removing duplicate rows.

UNION ALL:
Similar to UNION, but includes all rows from each SELECT statement, including duplicates.

INTERSECT:
Returns only the rows that appear in both result sets of two SELECT statements.

EXCEPT:
Returns the rows that appear in the first result set but not in the second result set.

14. Best All-Round Performance: Identify the player with the best all-round performance in a single match, considering both batting and bowling performances. You'll need to calculate a composite score for each player based on their runs scored and wickets taken in the same match.

```sql
WITH Player_performance AS (
SELECT Player,AVG (Runs) AS Batting_Average,AVG(Wickets) AS Bowling_Average
FROM (
    SELECT
        Batsman AS player,
        Run AS Runs,
        NULL AS Wickets
    FROM IPL2023_Batsman
    UNION ALL
    SELECT
        Bowler AS player,
        NULL AS Wickets,
        Wicket AS Wickets
    FROM IPL2023_Bowler) AS Combined_Data
 GROUP BY Player
 HAVING Batting_Average >= 1 AND Bowling_Average >=1 ),      # Here we confirm there is all allrounders.

WeightedData  AS (
    SELECT
        Player,
        ((((0.6*Batting_Average) + (0.4 * Bowling_Average))/2) * 100 ) AS Performace_index
    FROM Player_performance
    GROUP BY Player)
SELECT
    Player,
    Performace_index
FROM WeightedData
ORDER BY Performace_index DESC LIMIT 1 ;
```

Player	Performace_index
Amit Mishra	590.000000000

summary

Finding the best all-rounder in cricket based on both runs scored and wickets taken involves a bit of statistical analysis. One common method is to calculate a player's performance index, which takes into account both batting and bowling performances.

Overall, this query effectively identifies the best all-rounder based on their batting and bowling performances, considering both aspects equally in the performance index calculation.

Digital Learning

THANK YOU

Vaibhav Pawar

www.linkedin.com/in/vaibhavpawar000