



ANALYSING A DATASET WITH SQL



Anjan Mahapatra (Cohort 36)

TASK -1 : FIND THE TOP PERFORMERS

SQL CODE - 1

TASK -1

Write a **SQL query** to list the top 5 players who have the highest number of goals scored. Include their name, team, goals scored, and position.

Expected Output:

A table showing player names, their team names, goals scored, and their positions.

SELECT

p.name **AS** player_name,
t.team_name,
p.goals_scored,
p.position

FROM

players p

JOIN

teams t ON p.team_id = t.team_id

ORDER BY

p.goals_scored DESC

LIMIT 5;

OUTPUT

Solution Accepted

Your Answer

player_name	team_name	goals_scored	position
Erling Haaland	Manchester City	40	Forward
Robert Lewandowski	Juventus	35	Forward
Lionel Messi	FC Barcelona	30	Forward
Cristiano Ronaldo	Juventus	28	Forward
Kylian Mbappe	Paris Saint-Germain	27	Forward

EXPLANATION :

- **JOIN** combines the players and teams tables on team_id to get team_name.
- **ORDER BY** p.goals_scored DESC ensures players with the most goals appear first.
- **LIMIT 5** returns only the top 5 records.

TASK -2 : BEST VALUE FOR MONEY PLAYERS

TASK - 2

Write a **SQL query** to identify the top 3 players who provide the best value for money.
Value for money is defined as the number of goals scored per unit of salary (goals/salary).
For readability, scale this ratio by multiplying by 1,000,000, and display the result to one decimal places based on precision.

Show player name, team name, goals scored, salary, and the calculated **value-for-money ratio**.
Sort the output in descending order of **value_for_money_ratio**.

Expected Output:

A table with player names, their team names, goals, salaries, and their value-for-money ratio, sorted in descending order.

SQL CODE - 2

```
SELECT
    p.name AS player_name,
    t.team_name,
    p.goals_scored,
    p.salary,
    ROUND((p.goals_scored * 1000000.0) / p.salary, 1) AS
value_for_money_ratio
FROM
    players p
JOIN
    teams t ON p.team_id = t.team_id
ORDER BY
    value_for_money_ratio DESC
LIMIT 3;
```

EXPLANATION :

- **JOIN** the players table with teams to get team_name.
Calculate value for money as:
(goals_scored * 1000000.0) / salary
- Use ROUND(..., 1) for 1 decimal precision.
- Sort in descending order and limit to top 3.

OUTPUT

Solution Accepted

Your Answer

player_name	team_name	goals_scored	salary	value_for_money_ratio
Erling Haaland	Manchester City	40	30000000	1.3
Mohamed Salah	Paris Saint-Germain	25	25000000	1.0
Kylian Mbappe	Paris Saint-Germain	27	30000000	0.9

TASK -3 : TEAM PERFORMANCE SUMMARY

SQL CODE - 3

TASK - 3

Write a SQL query to generate a summary report of each team's performance. The report should show the team name, number of wins, losses, draws, total goals scored by all its players, and the team's league position.

Expected Output:

A table listing team names, their number of wins, losses, draws, total goals, and league position.

SELECT

```
t.team_name,  
t.wins,  
t.losses,  
t.draws,  
SUM(p.goals_scored) AS total_goals,  
t.league_position
```

FROM

```
teams t
```

JOIN

```
players p ON t.team_id = p.team_id
```

GROUP BY

```
t.team_id, t.team_name, t.wins, t.losses,  
t.draws, t.league_position
```

ORDER BY

```
t.league_position ASC;
```

EXPLANATION :

- **JOIN** the teams and players tables using team_id.
- Group the results by team to aggregate total goals.
- Use **SUM**(p.goals_scored) to calculate each team's total goals.
- Order the final result by league_position ascending.

OUTPUT

Solution Accepted

Your Answer

team_name	wins	losses	draws	total_goals	league_position
Paris Saint-Germain	22	2	3	62	1
Manchester City	21	4	1	65	2
FC Barcelona	20	3	2	33	3
Juventus	18	5	1	63	4

TASK -4 : PLAYER PERFORMANCE ANALYSIS

TASK - 4

Write a **SQL query** to find all players who have been transferred to a different team. Include player name, original team name, new team name, and the transfer fee.

Expected Output:

A table showing player names, their original team, the team they transferred to, and the transfer fee.

SQL CODE - 4

```
SELECT
    p.name AS player_name,
    t1.team_name AS original_team,
    t2.team_name AS new_team,
    tr.transfer_fee
FROM
    transfers tr
JOIN
    players p ON tr.player_id = p.player_id
JOIN
    teams t1 ON tr.from_team_id = t1.team_id
JOIN
    teams t2 ON tr.to_team_id = t2.team_id
ORDER BY
    p.name;
```

EXPLANATION :

- **JOIN** to fetch related team and player info.
- Aliases (**t1, t2**) to reuse the teams table for both old and new teams.
- **ORDER BY** to sort the output.

OUTPUT

Solution Accepted

Your Answer

player_name	original_team	new_team	transfer_fee
Kevin De Bruyne	FC Barcelona	Paris Saint-Germain	80000000
Sergio Ramos	Paris Saint-Germain	FC Barcelona	90000000
Virgil van Dijk	Juventus	Manchester City	70000000

TASK -5 : IDENTIFY POTENTIAL UNDERPERFORMERS

SQL CODE - 5

TASK - 5

Write a **SQL query** to identify players who have played more than 10 matches but have scored fewer than 4 goals. Include their name, team, matches played, and goals scored.

Expected Output:

A table with player names, their team, number of matches played, and their goals scored.

SELECT

p.name **AS** player_name,
t.team_name,
p.matches_played,
p.goals_scored

FROM

players p

JOIN

teams t **ON** p.team_id = t.team_id

WHERE

p.matches_played > **10** **AND**
p.goals_scored < **4**;

OUTPUT

Solution Accepted

Your Answer

player_name	team_name	matches_played	goals_scored
Sergio Ramos	FC Barcelona	21	3

EXPLANATION :

- FROM players p: Start from the players table.
- JOIN teams t ON p.team_id = t.team_id: Join with the teams table to get team names.
- WHERE p.matches_played > 10 AND p.goals_scored < 4: Filter only players who played more than 10 matches and scored fewer than 4 goals.
- SELECT ...: Fetch required columns with readable aliases.

TASK -6 : TEAM BUDGET EFFICIENCY

SQL CODE - 6

TASK - 6

Write a **SQL query** to find the top 3 teams that have achieved the highest number of wins with the lowest budget. The "win-to-budget ratio" is calculated as the number of wins divided by the budget, scaled by a factor of 1,000,000, and rounded to two decimal places. Show the team name, number of wins, and their budget, sorted by the best win-to-budget ratio.

Expected Output:

A table with team names, number of wins, and their budget, sorted by the best win-to-budget ratio.

```
SELECT
    team_name,
    wins,
    budget,
    ROUND((wins * 1000000.0) / budget, 2)
AS win_to_budget_ratio
FROM
    teams
ORDER BY
    win_to_budget_ratio DESC
LIMIT 3;
```

EXPLANATION :

- **ROUND**((wins * 1000000.0) / budget, 2): Scales the ratio by 1,000,000. Ensures floating point division using 1000000.0. Rounds to 2 decimal places.
- **ORDER BY** win_to_budget_ratio DESC: Sorts from highest ratio.
- **LIMIT 3**: Gets top 3 teams only.

OUTPUT

Solution Accepted

Your Answer

team_name	wins	budget	win_to_budget_ratio
Juventus	18	500000000	0.04
FC Barcelona	20	600000000	0.03
Paris Saint-Germain	22	700000000	0.03