

## South African Soccer Dataset – SQL

1. View the first 100 rows of the dataset to understand its structure.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure, including the 'Soccer\_Team\_Analysis' database and its tables. The SQL Query window in the center contains the following query:

```
SELECT *
FROM soccer_team_analysis;

-- 1. View the first 100 rows of the dataset to understand its structure.
SELECT TOP 100 *
FROM soccer_team_analysis;
```

The Results pane at the bottom displays the first 10 rows of the dataset, showing columns: player\_name, team, date\_of\_birth, age, marital\_status, number\_of\_kids, nationality, country\_of\_birth, position, and preferred\_foot.

player_name	team	date_of_birth	age	marital_status	number_of_kids	nationality	country_of_birth	position	preferred_foot
"Siyanda Dlamini"	"Stellenbosch FC"	1995-05-28 00:00:00.0000000	30	"Widowed"	0	"South African"	"South African"	"Defender"	"Right"
"Thabo Ndlovu"	"Cape Town City"	2004-06-01 00:00:00.0000000	21	"Single"	1	"Zimbabwean"	"Zimbabwean"	"Forward"	"Right"
"Vusi Molefe"	"Stellenbosch FC"	2006-09-20 00:00:00.0000000	19	"Single"	0	"Nigerian"	"Nigerian"	"Forward"	"Left"
"Thembi Mahlangu"	"Bloemfontein Celtic"	2004-02-08 00:00:00.0000000	21	"Divorced"	0	"Zambian"	"Zambian"	"Goalkeeper"	"Left"
"Nokuthula Sithole"	"Polokwane City"	2003-03-18 00:00:00.0000000	22	"Divorced"	3	"Nigerian"	"Nigerian"	"Goalkeeper"	"Both"
"Thembi Sithole"	"Kaizer Chiefs"	1989-02-14 00:00:00.0000000	36	"Married"	2	"Nigerian"	"Nigerian"	"Goalkeeper"	"Left"
"Siyanda Mahlangu"	"Chippa United"	1989-12-09 00:00:00.0000000	36	"Single"	0	"Zambian"	"Zambian"	"Defender"	"Both"
"Lerato Mashaba"	"Polokwane City"	1998-01-31 00:00:00.0000000	27	"Widowed"	0	"Ghanaian"	"Ghanaian"	"Forward"	"Right"
"Nomsa Mahlangu"	"Polokwane City"	1991-07-16 00:00:00.0000000	34	"Divorced"	0	"Malawian"	"Malawian"	"Forward"	"Left"
"Tumelo Khumalo"	"Kaizer Chiefs"	1996-08-28 00:00:00.0000000	29	"Married"	4	"Malawian"	"Malawian"	"Forward"	"Both"

2. Count the total number of players in the dataset.

The screenshot shows the SQL Server Enterprise Manager interface. The SQL Query window in the center contains the following query:

```
SELECT *
FROM soccer_team_analysis;

-- 1. View the first 100 rows of the dataset to understand its structure.
SELECT TOP 100 *
FROM soccer_team_analysis;

-- 2. Count the total number of players in the dataset.
SELECT COUNT(*) AS total_players
FROM soccer_team_analysis;
```

The Results pane at the bottom displays the result of the query, showing a single row with the column 'total\_players' and the value 300.

total_players
300

3. List all unique teams in the league.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile(Pretty))'. The main query window shows the following SQL code:

```

1  SELECT *
2  FROM soccer_team_analysis;
3
4  -- 1. View the first 100 rows of the dataset to understand its structure.
5  SELECT TOP 100 *
6  FROM soccer_team_analysis;
7
8  -- 2. Count the total number of players in the dataset.
9  SELECT COUNT(*) AS total_players
10 FROM soccer_team_analysis;
11
12 -- 3. List all unique teams in the league.
13 SELECT DISTINCT team
14 FROM soccer_team_analysis
15 ORDER BY team;

```

The Results pane shows the output of the last query, listing 11 unique teams:

team
1 "Amazulu FC"
2 "Bloemfontein Celtic"
3 "Cape Town City"
4 "Chippa United"
5 "Golden Arrows"
6 "Kaizer Chiefs"
7 "Mamelodi Sundowns"
8 "Moroka Swallows"
9 "Orlando Pirates"
10 "Polokwane City"
11 "Richards Bay FC"

#### 4. Count how many players are in each team.

The screenshot shows the SQL Server Enterprise Manager interface with the following SQL code in the query window:

```

7  -- 2. Count the total number of players in the dataset.
8  SELECT COUNT(*) AS total_players
9  FROM soccer_team_analysis;
10
11 -- 3. List all unique teams in the league.
12 SELECT DISTINCT team
13 FROM soccer_team_analysis
14 ORDER BY team;
15
16 -- 4. Count how many players are in each team.
17 SELECT team, COUNT(*) AS player_count
18 FROM soccer_team_analysis
19 GROUP BY team
20 ORDER BY player_count DESC;

```

The Results pane shows the output of the last query, listing 11 teams with their player counts:

team	player_count
1 "Polokwane City"	27
2 "Golden Arrows"	23
3 "Amazulu FC"	22
4 "Chippa United"	21
5 "Stellenbosch FC"	20
6 "SuperSport United"	20
7 "TS Galaxy"	20
8 "Kaizer Chiefs"	20
9 "Mamelodi Sundowns"	19
10 "Moroka Swallows"	19
11 "Royal AM"	18

#### 5. Identify the top 10 players with the most goals.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile/Pretty)'. The main query window shows the following SQL code:

```

ORDER BY team;
-- 4. Count how many players are in each team.
SELECT team, COUNT(*) AS player_count
FROM soccer_team_analysis
GROUP BY team
ORDER BY player_count DESC;
-- 5. Identify the top 10 players with the most goals.
SELECT TOP 10 player_name, goals
FROM soccer_team_analysis
ORDER BY goals DESC;

```

The Results pane shows the output of the query:

player_name	goals
1 "Vusi Molefe"	99
2 "Thabo Ndlovu"	98
3 "Gugu Hongwane"	98
4 "Thabo Sithole"	98
5 "Mandla Mabena"	98
6 "Mpho Mahlangu"	97
7 "Botumelo Nkosi"	96
8 "Mpho Radebe"	92
9 "Khanyi Nkosi"	92
10 "Sipho Phiri"	91

The status bar at the bottom indicates: PHINDILE (16.0 RTM) | Phindile/Pretty (56) | Soccer\_Team\_Analysis | 00:00:00 | Row: 1, Col: 1 | 10 rows

6. Find the average salary for players in each team.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile/Pretty)'. The main query window shows the following SQL code:

```

ORDER BY goals DESC;
-- 6. Find the average salary for players in each team.
SELECT team, AVG(average_salary_zar) AS avg_salary
FROM soccer_team_analysis
GROUP BY team
ORDER BY avg_salary DESC;

```

The Results pane shows the output of the query:

team	avg_salary
1 "Royal AM"	209078.331111111
2 "Cape Town City"	208407.432
3 "Golden Arrows"	199057.643913043
4 "SuperSport United"	195482.9785
5 "Mamelodi Sundowns"	194404.484736842
6 "Richards Bay FC"	193006.199230769
7 "Kaizer Chiefs"	188954.4615
8 "Moroka Swallows"	186786.827894737
9 "TS Galaxy"	184719.574
10 "Stellenbosch FC"	182326.7355
11 "Chippa United"	180947.634761905

The status bar at the bottom indicates: PHINDILE (16.0 RTM) | Phindile/Pretty (56) | Soccer\_Team\_Analysis | 00:00:00 | Row: 3, Col: 2 | 16 rows

7. Retrieve the top 10 players with the highest market value.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile/Pretty)'. The main query window shows two queries. The first query calculates the average salary for each team. The second query retrieves the top 10 players by market value.

```

24 SELECT team, AVG(average_salary_zar) AS avg_salary
25 FROM soccer_team_analysis
26 GROUP BY team
27 ORDER BY avg_salary DESC;
28 -- 7. Retrieve the top 10 players with the highest market value.
29 SELECT TOP 10 player_name, market_value_zar
30 FROM soccer_team_analysis
31 ORDER BY market_value_zar DESC;
32
33

```

The results pane shows the output of the second query, displaying the top 10 players by market value.

player_name	market_value_zar
1 "Ayanda Mabaso"	24979190.35
2 "Lebogang Mabena"	24887821.58
3 "Gugu Mashaba"	24868294.94
4 "Kagiso Mokoena"	24550182.81
5 "Botumelo Radebe"	24540830.44
6 "Thabo Ndlovu"	24380712.61
7 "Spho Mashaba"	24342131.59
8 "Lerato Ngobeni"	24231618.07
9 "Thabo Tshabalala"	24229791.64
10 "Lerato Tshabalala"	24226785.17

## 8. Calculate the average passing accuracy for each position.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile/Pretty)'. The main query window shows two queries. The first query calculates the average salary for each team. The second query calculates the average passing accuracy for each position.

```

24 SELECT team, AVG(average_salary_zar) AS avg_salary
25 FROM soccer_team_analysis
26 GROUP BY team
27 ORDER BY avg_salary DESC;
28 -- 7. Retrieve the top 10 players with the highest market value.
29 SELECT TOP 10 player_name, market_value_zar
30 FROM soccer_team_analysis
31 ORDER BY market_value_zar DESC;
32 -- 8. Calculate the average passing accuracy for each position.
33 SELECT position, AVG(passing_accuracy) AS avg_passing_accuracy
34 FROM soccer_team_analysis
35 GROUP BY position
36 ORDER BY avg_passing_accuracy DESC;
37

```

The results pane shows the output of the second query, displaying the average passing accuracy for each position.

position	avg_passing_accuracy
1 "Goalkeeper"	83.9397058823529
2 "Forward"	83.210843373494
3 "Midfielder"	82.8177215189873
4 "Defender"	82.6642857142857

## 9. Compare shot accuracy with goals to find correlations.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile(Pretty))'. The main query window, titled 'SQLQuery2.s...Pretty (56)', contains the following SQL code:

```

-- 9. Compare shot accuracy with goals to find correlations.
SELECT
    AVG(CAST(shot_accuracy AS FLOAT)) AS avg_shot_accuracy,
    AVG(CAST(goals AS FLOAT)) AS avg_goals,
    COUNT(*) AS total_players
FROM soccer_team_analysis;

```

The Results pane shows the output of the query:

	avg_shot_accuracy	avg_goals	total_players
1	65.1593333333334	33.1466666666667	300

The status bar at the bottom indicates 'Query executed successfully.' and 'PHINDILE (16.0 RTM) | Phindile(Pretty (56)) | Soccer\_Team\_Analysis | 00:00:00 | Row: 1, Col: 1 | 1 rows'.

## 10. Compute total goals and assists for each team

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile(Pretty))'. The main query window, titled 'SQLQuery2.s...Pretty (56)', contains the following SQL code:

```

-- 9. Compare shot accuracy with goals to find correlations.
SELECT
    AVG(CAST(shot_accuracy AS FLOAT)) AS avg_shot_accuracy,
    AVG(CAST(goals AS FLOAT)) AS avg_goals,
    COUNT(*) AS total_players
FROM soccer_team_analysis;
-- 10. Compute total goals and assists for each team.
SELECT
    team,
    SUM(CAST(goals AS FLOAT)) AS total_goals,
    SUM(CAST(assists AS FLOAT)) AS total_assists
FROM soccer_team_analysis
GROUP BY team
ORDER BY total_goals DESC;

```

The Results pane shows the output of the query:

	team	total_goals	total_assists
1	"Polokwane City"	1300	783
2	"Sekhukhune United"	842	608
3	"Moroka Swallows"	729	452
4	"Stellenbosch FC"	725	514
5	"Chippa United"	684	205
6	"Golden Arrows"	676	519
7	"Bloemfontein Celtic"	613	566
8	"Kaizer Chiefs"	597	481
9	"SuperSport United"	580	514
10	"Mamelodi Sundowns"	570	475
11	"AmaZulu FC"	546	568

The status bar at the bottom indicates 'Query executed successfully.' and 'PHINDILE (16.0 RTM) | Phindile(Pretty (56)) | Soccer\_Team\_Analysis | 00:00:00 | Row: 1, Col: 1 | 16 rows'.

## 11. Count players by their marital status.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile/Pretty)'. The main query window shows the following SQL code:

```

46 SUM(CAST(goals AS FLOAT)) AS total_goals,
47 SUM(CAST(assists AS FLOAT)) AS total_assists
48 FROM soccer_team_analysis
49 GROUP BY team
50 ORDER BY total_goals DESC;
51 -- 11. Count players by their marital status.
52 SELECT marital_status, COUNT(*) AS count_players
53 FROM soccer_team_analysis
54 GROUP BY marital_status
55 ORDER BY count_players DESC;
56
57
58

```

The query results are displayed in a table with two columns: 'marital\_status' and 'count\_players'.

marital_status	count_players
"Single"	79
"Widowed"	78
"Divorced"	78
"Married"	65

The status bar at the bottom indicates 'Query executed successfully.' and 'PHINDILE (16.0 RTM) | Phindile/Pretty (56) | Soccer\_Team\_Analysis | 00:00:00 | Row: 1, Col: 1 | 4 rows'.

## 12. Count players by nationality.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile/Pretty)'. The main query window shows the following SQL code:

```

47 SUM(CAST(assists AS FLOAT)) AS total_assists
48 FROM soccer_team_analysis
49 GROUP BY team
50 ORDER BY total_goals DESC;
51 -- 11. Count players by their marital status.
52 SELECT marital_status, COUNT(*) AS count_players
53 FROM soccer_team_analysis
54 GROUP BY marital_status
55 ORDER BY count_players DESC;
56 -- 12. Count players by nationality.
57 SELECT nationality, COUNT(*) AS count_players
58 FROM soccer_team_analysis
59 GROUP BY nationality
60 ORDER BY count_players DESC;
61

```

The query results are displayed in a table with two columns: 'nationality' and 'count\_players'.

nationality	count_players
"Ghanaian"	47
"South African"	46
"Zimbabwean"	45
"Zambian"	44
"Malawian"	42
"Nigerian"	39
"Mozambican"	37

The status bar at the bottom indicates 'Query executed successfully.' and 'PHINDILE (16.0 RTM) | Phindile/Pretty (56) | Soccer\_Team\_Analysis | 00:00:00 | Row: 1, Col: 1 | 7 rows'.

## 13. Find average market value grouped by nationality

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile(Pretty))'. The main window shows two SQL queries. The first query, 'SQLQuery2.s...Pretty (56)', is a comment. The second query, 'SQLQuery1.sql...Pretty (52)', is a SELECT statement that counts players by nationality and finds the average market value grouped by nationality. The results of the second query are displayed in a table with 7 rows.

```

-- 12. Count players by nationality.
SELECT nationality, COUNT(*) AS count_players
FROM soccer_team_analysis
GROUP BY nationality
ORDER BY count_players DESC;

-- 13. Find average market value grouped by nationality.
SELECT nationality, AVG(market_value_zar) AS avg_market_value_zar
FROM soccer_team_analysis
GROUP BY nationality
ORDER BY avg_market_value_zar DESC;

```

nationality	avg_market_value_zar
"Nigerian"	15065261.3964103
"Mozambican"	14736203.5943243
"Zambian"	12777043.0611364
"Ghanaian"	12298308.0365957
"South African"	12037870.2363043
"Malawian"	11727296.0554762
"Zimbabwean"	10367909.5322222

14. Determine how many player contracts end in each year.

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'PHINDILE (SQL Server 16.0.1000.6 - Phindile(Pretty))'. The main window shows a SQL query, 'SQLQuery1.sql...Pretty (52)', which is a SELECT statement that determines how many player contracts end in each year. The results of the query are displayed in a table with 1 row.

```

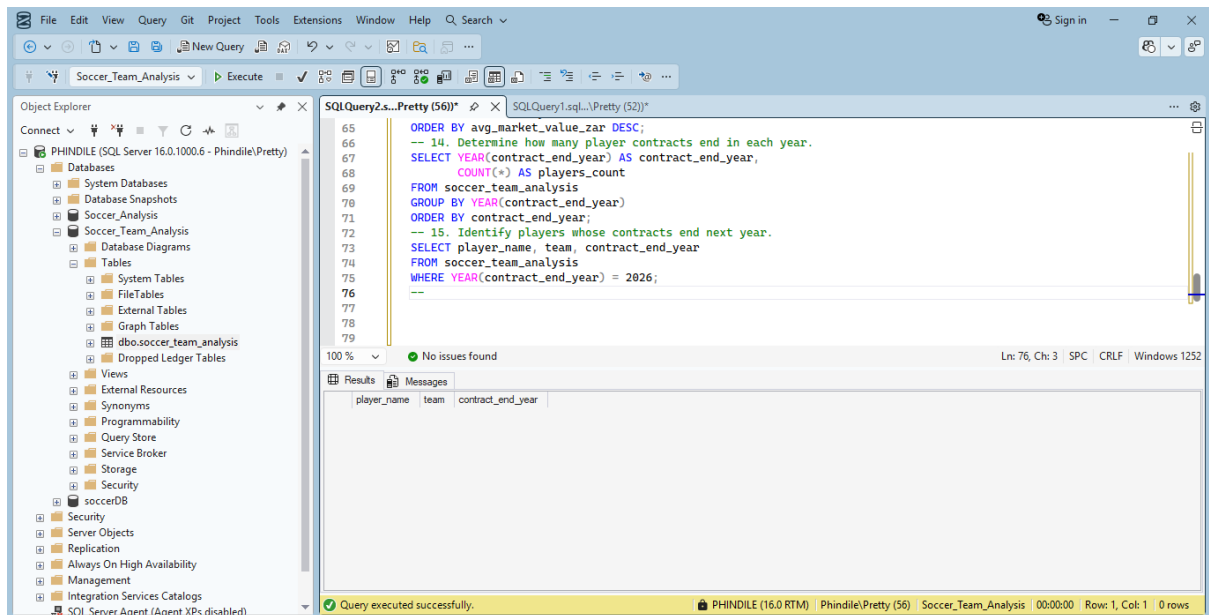
-- 13. Find average market value grouped by nationality.
SELECT nationality, AVG(market_value_zar) AS avg_market_value_zar
FROM soccer_team_analysis
GROUP BY nationality
ORDER BY avg_market_value_zar DESC;

-- 14. Determine how many player contracts end in each year.
SELECT YEAR(contract_end_year) AS contract_end_year,
COUNT(*) AS players_count
FROM soccer_team_analysis
GROUP BY YEAR(contract_end_year)
ORDER BY contract_end_year;

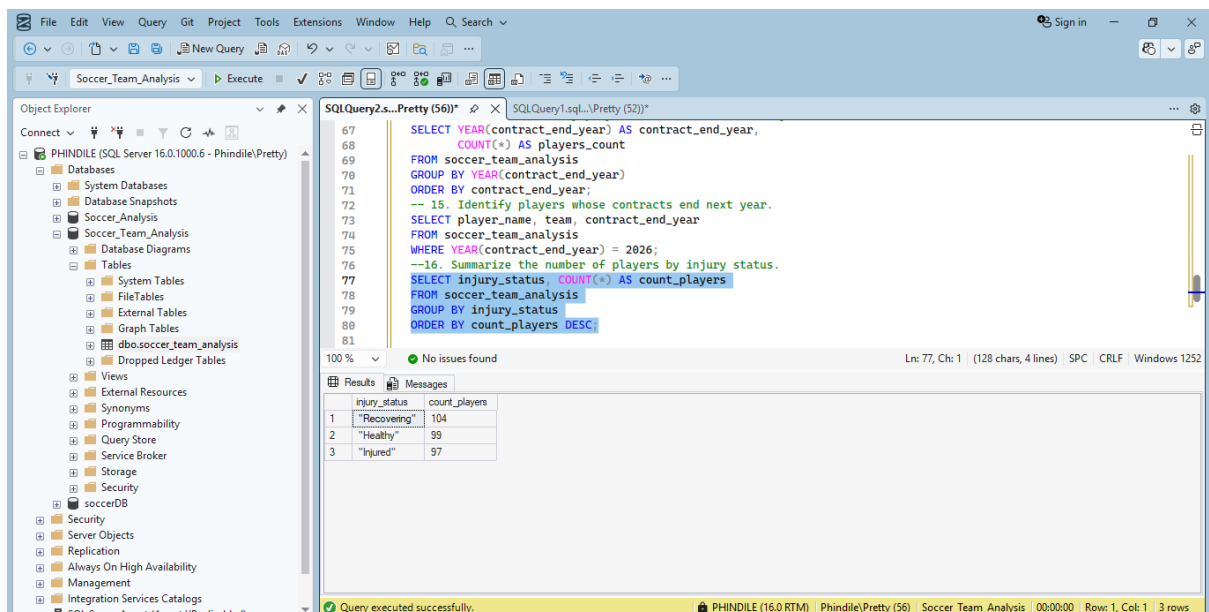
```

contract_end_year	players_count
1905	300

15. Identify players whose contracts end next year.



16. Summarize the number of players by injury status.



17. Calculate goals per match ratio for each player.



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'Soccer\_Team\_Analysis' database selected. The right pane shows a SQL query window with the following code:

```

--16. Summarize the number of players by injury status.
SELECT injury_status, COUNT(*) AS count_players
FROM soccer_team_analysis
GROUP BY injury_status
ORDER BY count_players DESC;
-- 17. Calculate goals per match ratio for each player.
SELECT
    player_name,
    goals,
    matches_played,
    CAST(goals AS FLOAT) / NULLIF(matches_played, 0) AS goals_per_match
FROM soccer_team_analysis
ORDER BY goals_per_match DESC;

```

The query results are displayed in a table with the following data:

player_name	goals	matches_played	goals_per_match
1 "Thembi Zulu"	72	3	24
2 "Thabo Mhembu"	88	5	17.6
3 "Gugu Tahabalala"	57	9	6.33333333333333
4 "Lindwe Mabaso"	67	13	5.15384615384615
5 "Mpho Zulu"	5	1	5
6 "Spho Tahabalala"	110	26	4.23076923076923
7 "Botumelo Nkosi"	96	25	3.84
8 "Karabo Tahabalala"	106	30	3.53333333333333
9 "Lindwe Zulu"	34	11	3.09090909090909
10 "Spho Phiri"	91	33	2.75757575757576
11 "Spho Dlamini"	43	22	1.95454545454545

The status bar at the bottom indicates: Query executed successfully. PHINDILE (16.0 RTM) | Phindile/Pretty (56) | Soccer\_Team\_Analysis | 00:00:00 | Row: 1, Col: 1 | 300 rows

18. Count how many players are managed by each agent.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the 'Soccer\_Team\_Analysis' database selected. The right pane shows a SQL query window with the following code:

```

matches_played,
CAST(goals AS FLOAT) / NULLIF(matches_played, 0) AS goals_per_match
FROM soccer_team_analysis
ORDER BY goals_per_match DESC;
-- 18. Count how many players are managed by each agent.
SELECT agent, COUNT(*) AS player_count
FROM soccer_team_analysis
GROUP BY agent
ORDER BY player_count DESC;

```

The query results are displayed in a table with the following data:

agent	player_count
1 "PlayerFirst"	63
2 "ProSpot"	62
3 "nan"	62
4 "SoccerLink Africa"	62
5 "SA Elite Agents"	51

The status bar at the bottom indicates: Query executed successfully. PHINDILE (16.0 RTM) | Phindile/Pretty (56) | Soccer\_Team\_Analysis | 00:00:00 | Row: 1, Col: 1 | 5 rows

19. Calculate average height and weight by player position.

SQL Query2.s...Pretty (56)\*

```

ORDER BY goals_per_match DESC;
-- 18. Count how many players are managed by each agent.
SELECT agent, COUNT(*) AS player_count
FROM soccer_team_analysis
GROUP BY agent
ORDER BY player_count DESC;
-- 19. Calculate average height and weight by player position.
SELECT
    position,
    AVG(height_cm) AS avg_height_cm,
    AVG(weight_kg) AS avg_weight_kg
FROM soccer_team_analysis
GROUP BY position
ORDER BY avg_height_cm DESC;

```

100 % No issues found Ln: 95, Ch: 1 (175 chars, 7 lines) SPC CRLF Windows 1252

position	avg_height_cm	avg_weight_kg
1 "Defender"	182	78
2 "Forward"	179	78
3 "Goalkeeper"	179	78
4 "Midfielder"	179	76

Query executed successfully. PHINDILE (16.0 RTM) Phindile/Pretty (56) Soccer\_Team\_Analysis 00:00:00 Row: 1, Col: 1 4 rows

## 20. Identify players with the highest combined goals and assists.

SQL Query2.s...Pretty (56)\*

```

position,
AVG(height_cm) AS avg_height_cm,
AVG(weight_kg) AS avg_weight_kg
FROM soccer_team_analysis
GROUP BY position
ORDER BY avg_height_cm DESC;
-- 20. Identify players with the highest combined goals and assists.
SELECT
    player_name,
    goals,
    assists,
    (goals + assists) AS total_contribution
FROM soccer_team_analysis
ORDER BY total_contribution DESC;

```

100 % No issues found Ln: 102, Ch: 1 (228 chars, 8 lines) SPC CRLF Windows 1252

player_name	goals	assists	total_contribution
1 "Vusi Molefe"	99	77	9977
2 "Gugu Hlongwane"	98	38	9838
3 "Mandla Mabena"	98	37	9837
4 "Thabo Ndlovu"	98	28	9828
5 "Thabo Sithole"	98	2	982
6 "Khanyi Ndlovu"	9	75	975
7 "Mpho Mahlangu"	97	4	974
8 "Tumelo Dlamini"	9	7	97
9 "Mandla Sithole"	9	7	97
10 "Botumelo Nkosi"	96	54	9654
11 "Nomasa Mahlangu"	9	6	96

Query executed successfully. PHINDILE (16.0 RTM) Phindile/Pretty (56) Soccer\_Team\_Analysis 00:00:00 Row: 1, Col: 1 300 rows