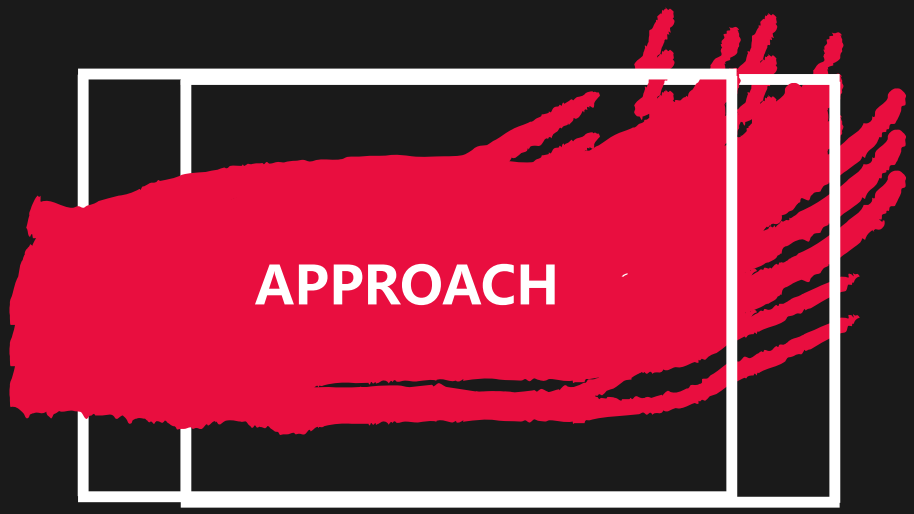
FINAL PROJECT

SOCCER
PLAYER PERFORMANCE ANALYSIS

Learner:

Kanimozhi Mani

BADM - MBE10



1

Data Preprocessing

- **Dataset Cleaning:** Duplicate rows identified with the help of the column ID and the duplicates were removed. (as the ID is an unique identifier and there won't be any identical IDs)

  In the original dataset, there were 17982 rows and duplicates were removed. After removing there were 17929 rows.

2

Exploratory Data Analysis (EDA)

- **Analysis:** Utilized Microsoft Power BI to visualize player statistics and trends.

3

Feature Engineering

- **Feature Creation:** Developed new metrics that enhance player evaluation and comparison.

4

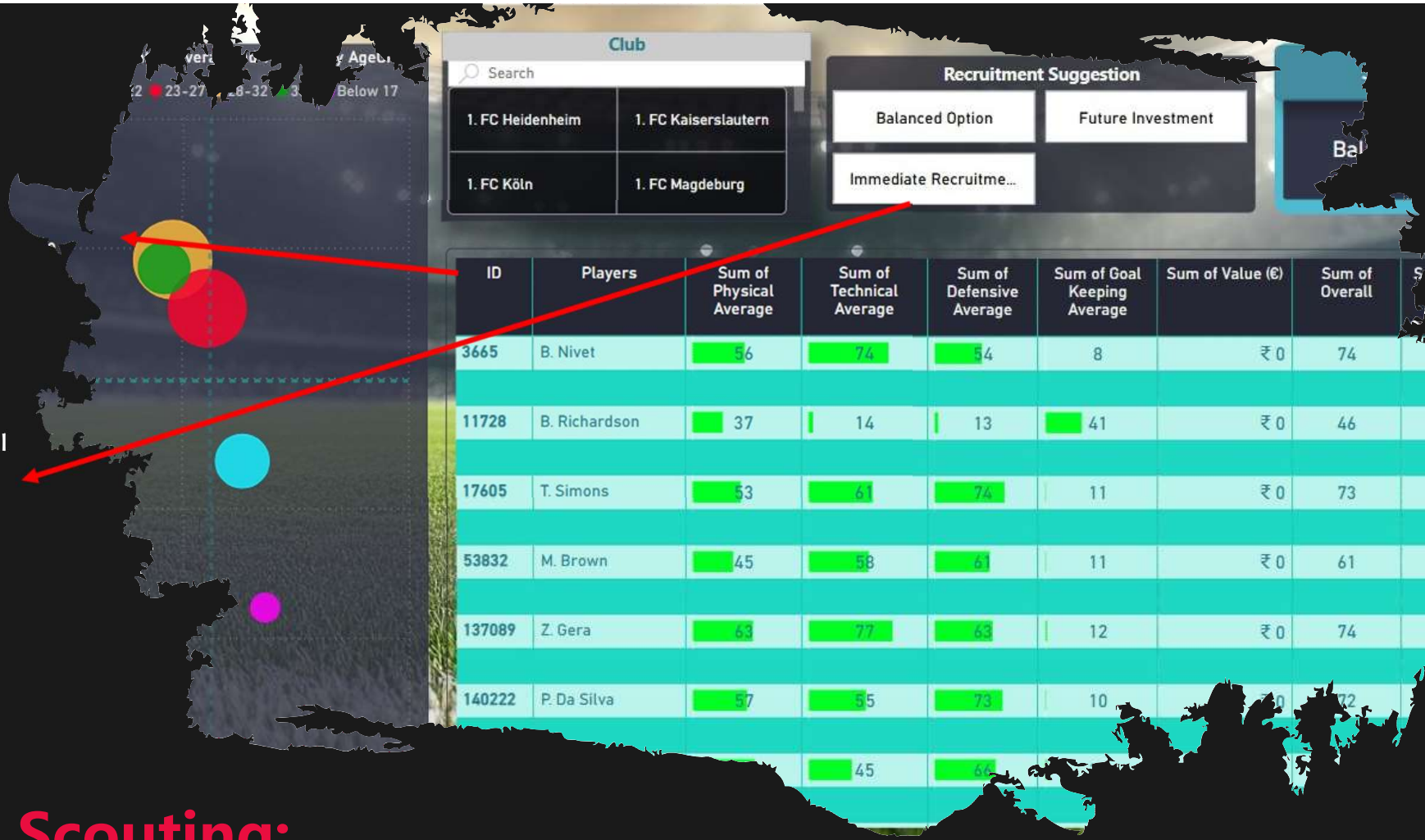
Visualization and Reporting

- Created interactive dashboards in Power to showcase insights and recommendations.

BUSINESS USE CASES

The chart ranks players by their **overall performance score** and future **potential rating**.

These buttons helps in identifying **high-potential players** for recruitment by showcasing their ratings, making it easier to prioritize targets.



ID	Players	Sum of Physical Average	Sum of Technical Average	Sum of Defensive Average	Sum of Goal Keeping Average	Sum of Value (€)	Sum of Overall
3665	B. Nivet	56	74	54	8	₹ 0	74
11728	B. Richardson	37	14	13	41	₹ 0	46
17605	T. Simons	53	61	74	11	₹ 0	73
53832	M. Brown	45	58	61	11	₹ 0	61
137089	Z. Gera	63	77	63	12	₹ 0	74
140222	P. Da Silva	57	55	73	10		
			45	66			

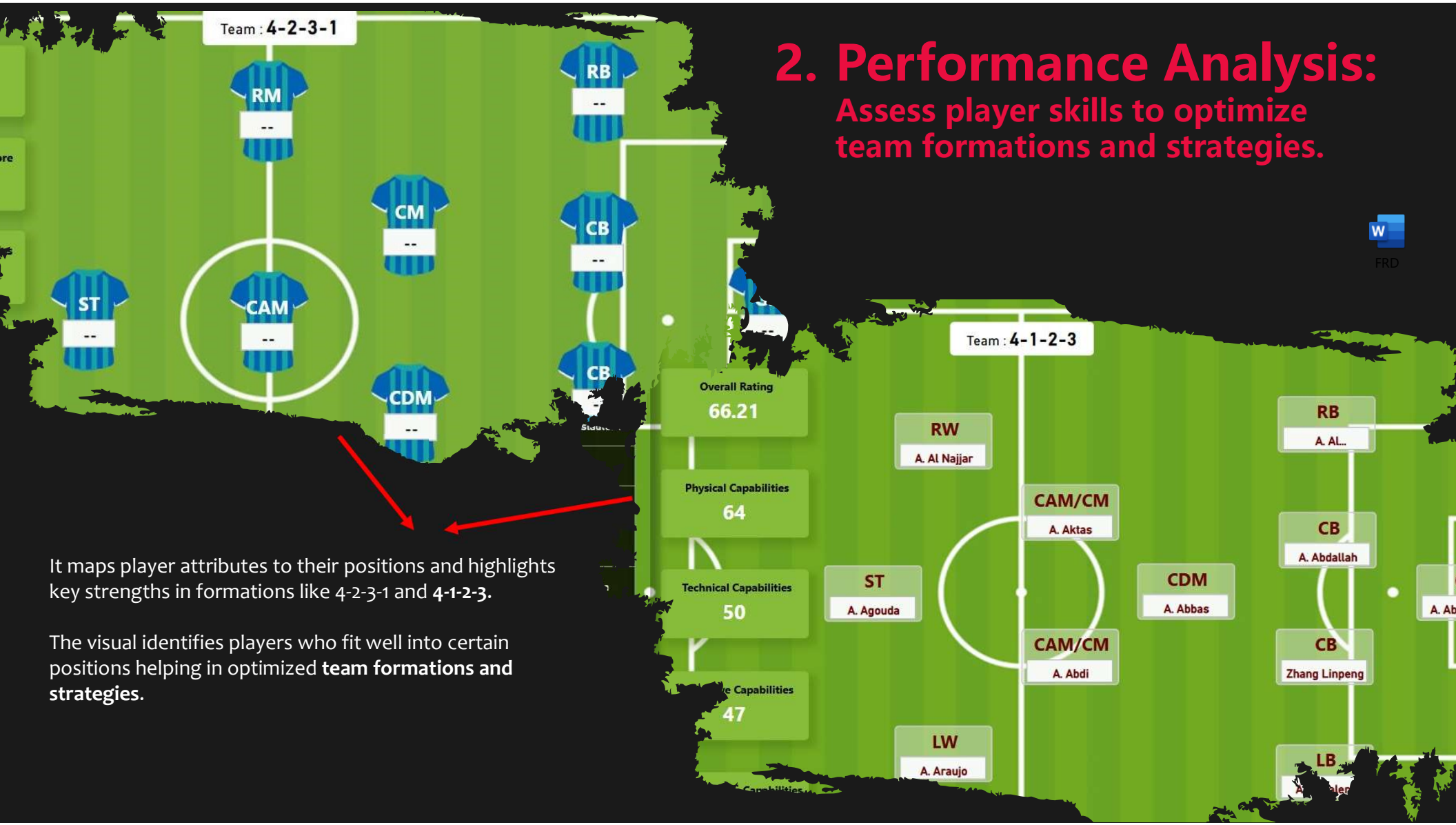
1. Player Scouting:

Identify high-potential players based on performance metrics for recruitment.

FRD

2. Performance Analysis:

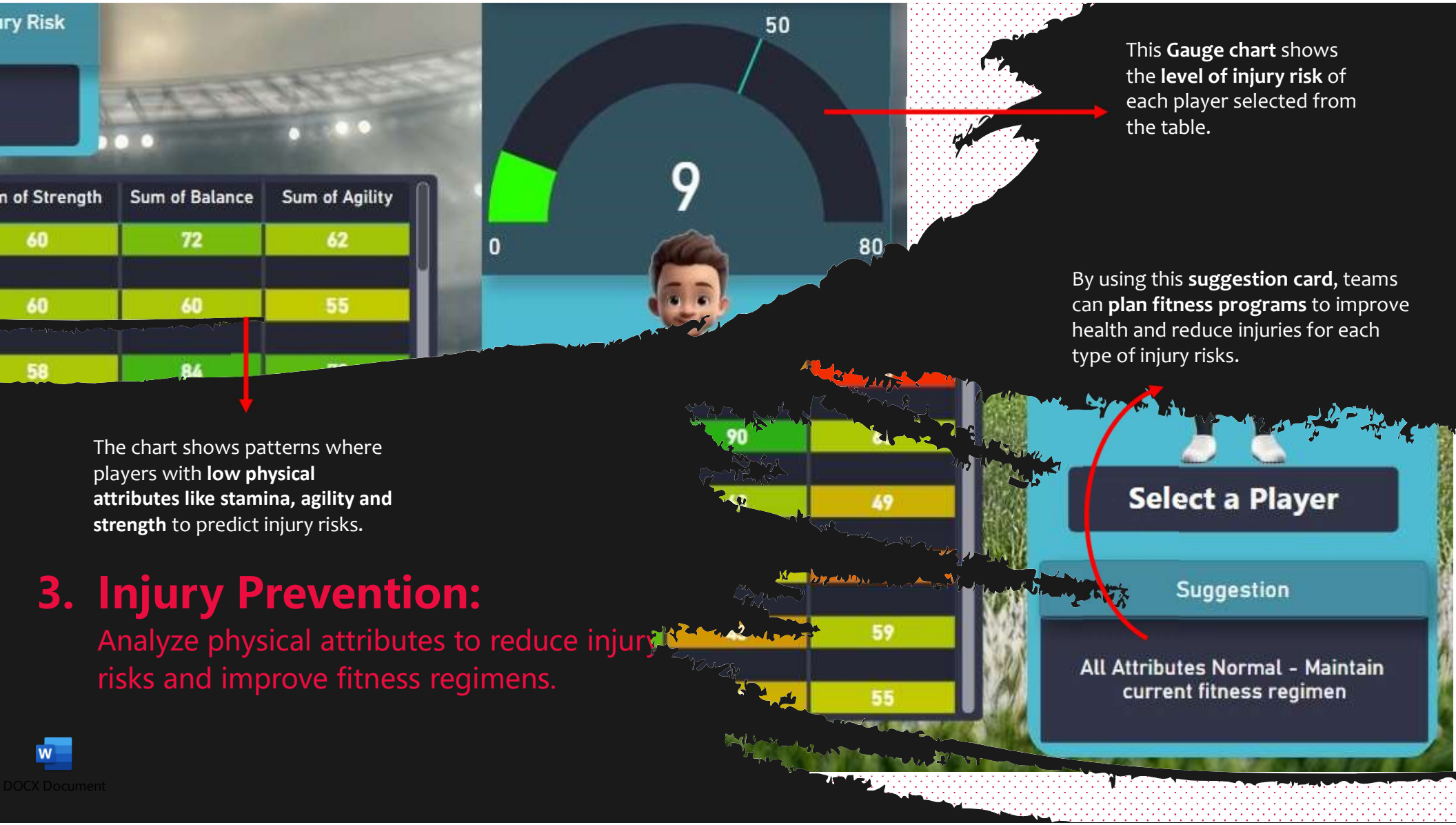
Assess player skills to optimize team formations and strategies.



FRD

It maps player attributes to their positions and highlights key strengths in formations like 4-2-3-1 and **4-1-2-3.**

The visual identifies players who fit well into certain positions helping in optimized **team formations and strategies.**



3. Injury Prevention:

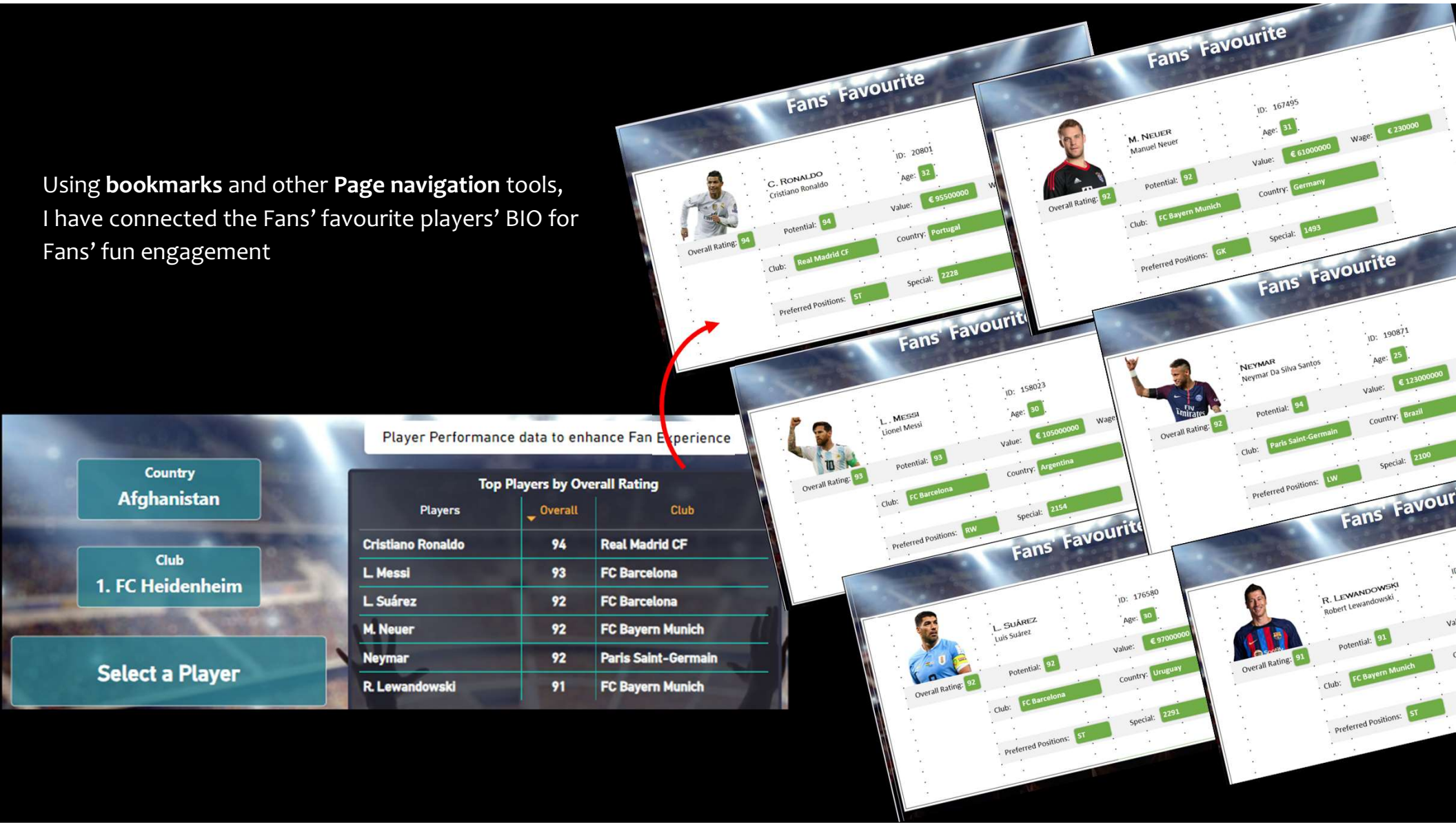
Analyze physical attributes to reduce injury risks and improve fitness regimens.

DOCX Document

4. Fan Engagement:

Tailor marketing strategies using player performance data to enhance fan experience.

Using **bookmarks** and other **Page navigation** tools, I have connected the Fans' favourite players' BIO for Fans' fun engagement



Teams can justify salary increases for high performers or renegotiate contracts for those who are underperforming using this chart data.

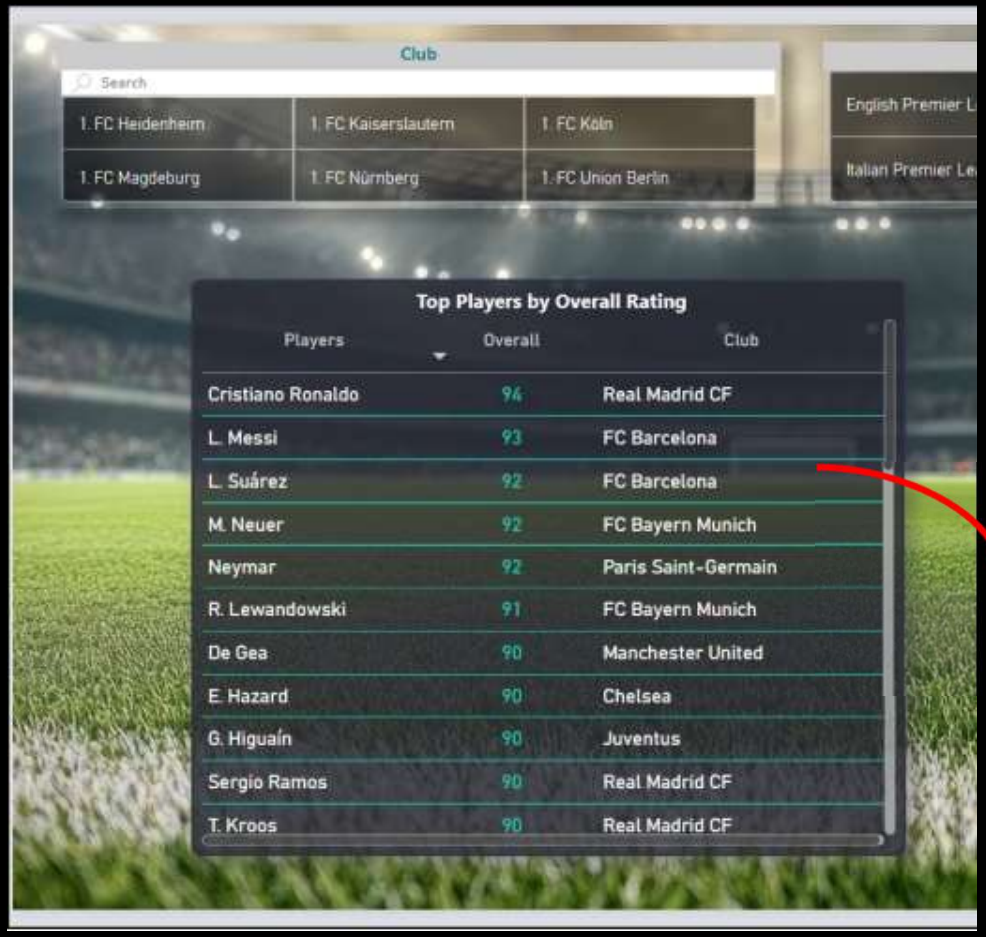
These buttons highlights whether players are **underpaid or overpaid** compared to their performance and value.

5. Contract Negotiation:

Use performance insights to justify player salaries and contract terms.

FRD

TASKS

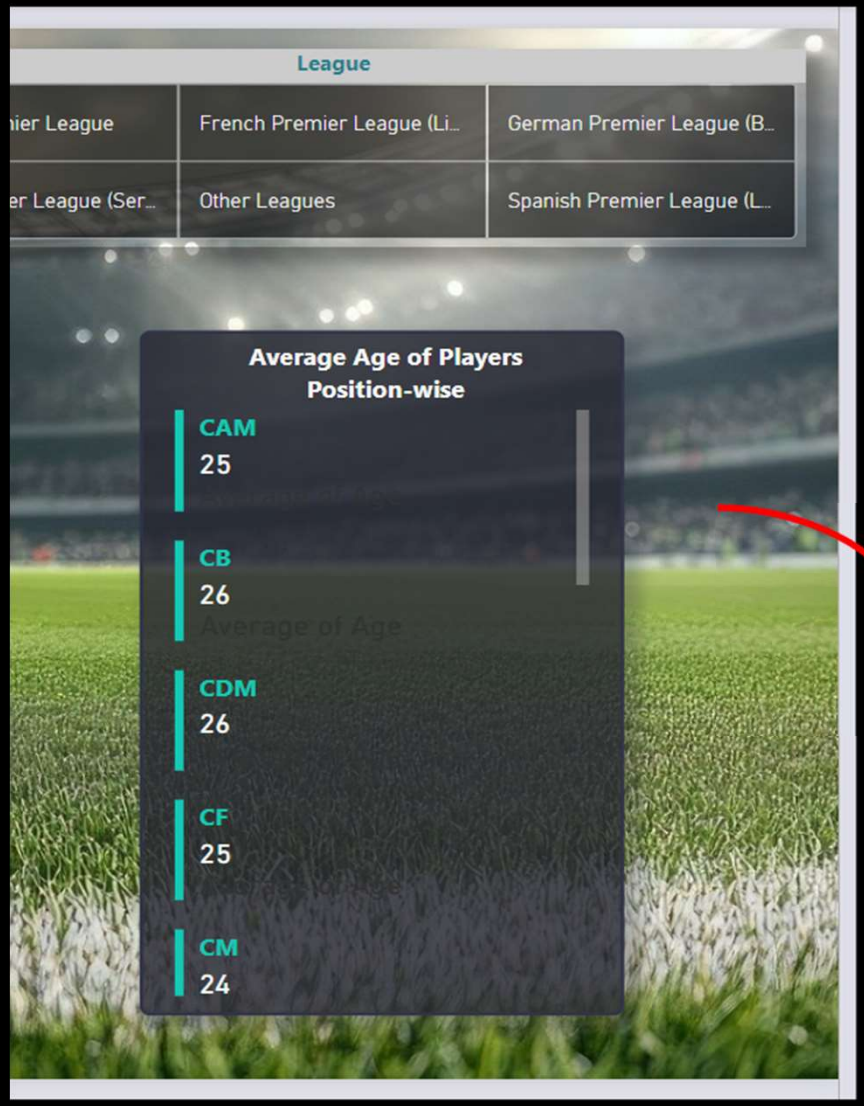


Task 1

Identify the top players by overall rating.

Top Players by Overall Rating: Cristiano Ronaldo leads with a rating of 94, followed by Messi and Suárez at 93 and 92 respectively.

Recommendation: Teams can focus on these top-rated players for important matches to improve overall performance.



Task 2

Calculate the average age of players in each position.

Top Players by Overall Rating: Cristiano Ronaldo leads with a rating of 94, followed by Messi and Suárez at 93 and 92 respectively.

Recommendation: Teams can focus on these top-rated players for important matches to improve overall performance.

Task 3

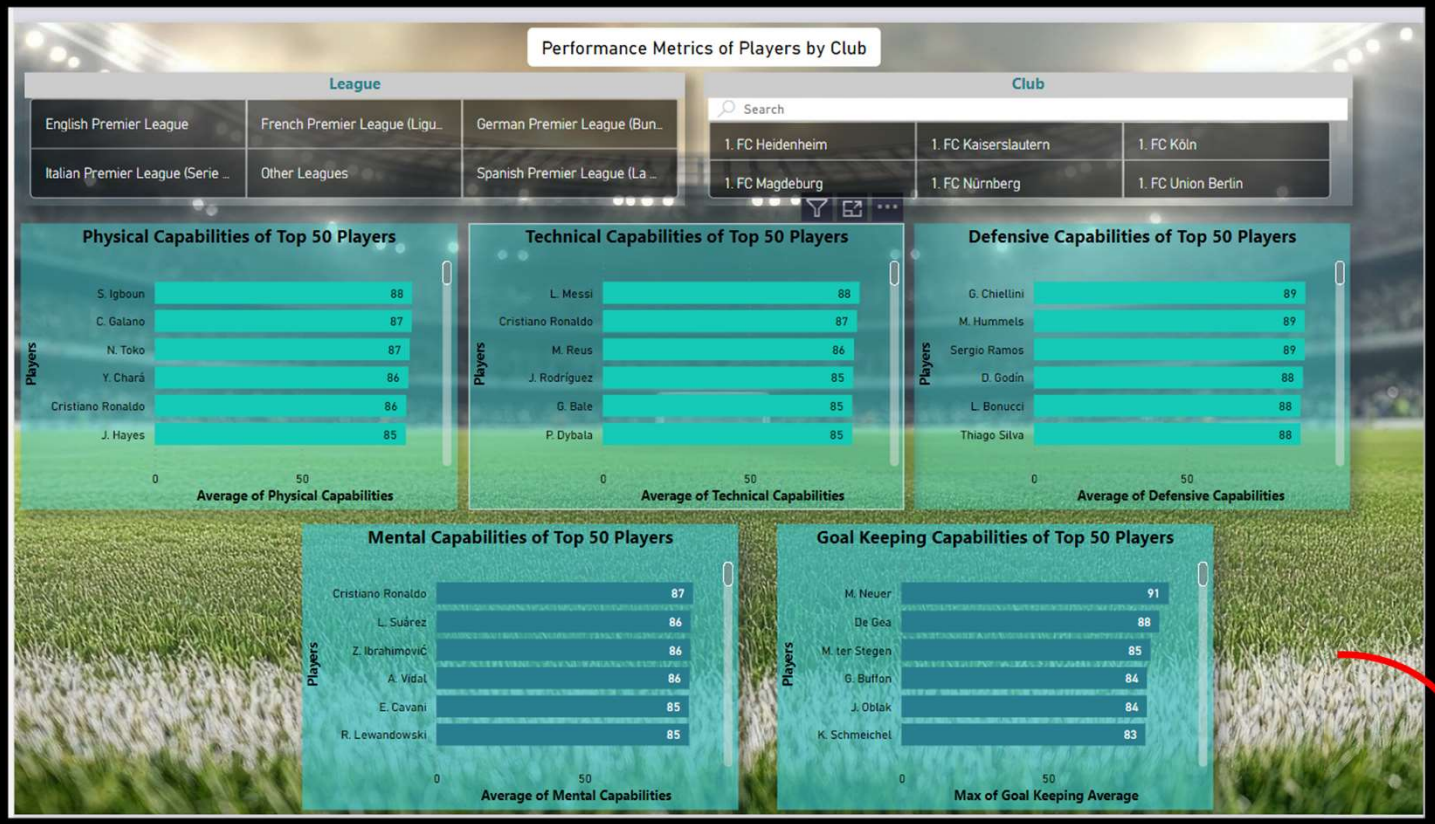
Determine the market value distribution among different nationalities.

Insight: Market value varies across nationalities, with some players having significantly higher value.

Recommendation: Focus on nationalities producing high-value players for recruitment opportunities.

Insight: Market value distribution has been done across Continent-wise too.

I have manually included continents data column for each country present in the dataset and matched to its respective continent to get more broad results.

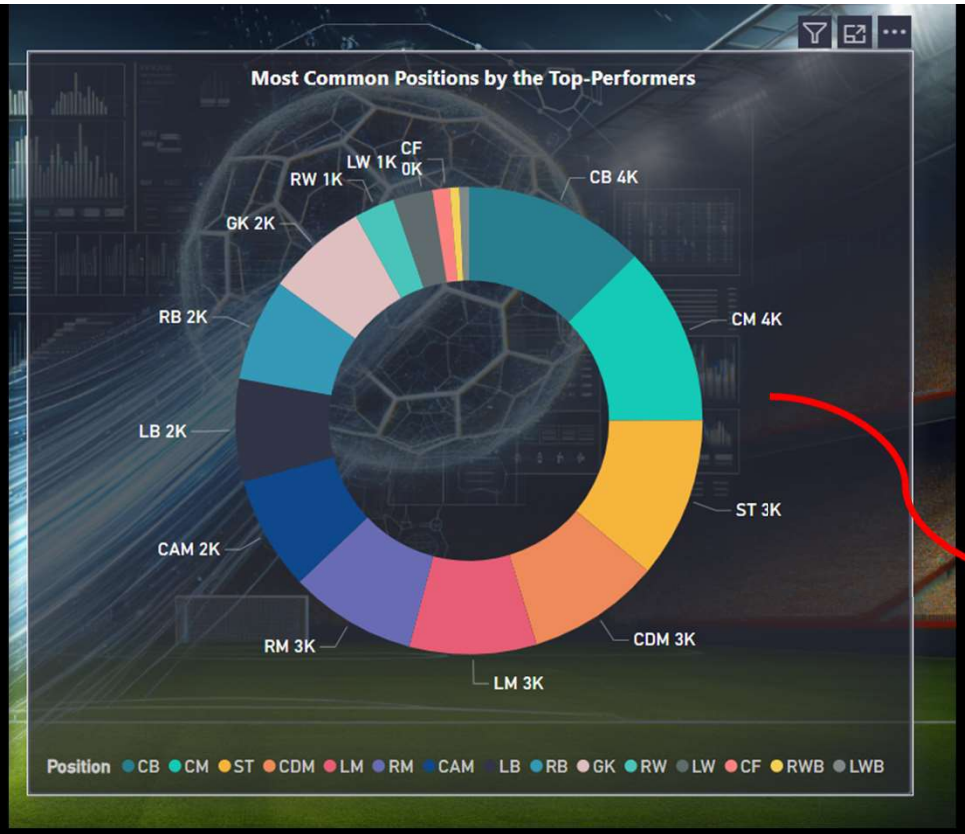


Task 4

Analyze performance metrics of players by club.

Insight: Clubs like Real Madrid and FC Barcelona have top-performing players based on metrics.

Suggestion: Analyze clubs with lower performance to identify areas for improvement.



Find the most common positions played by top-performing players.

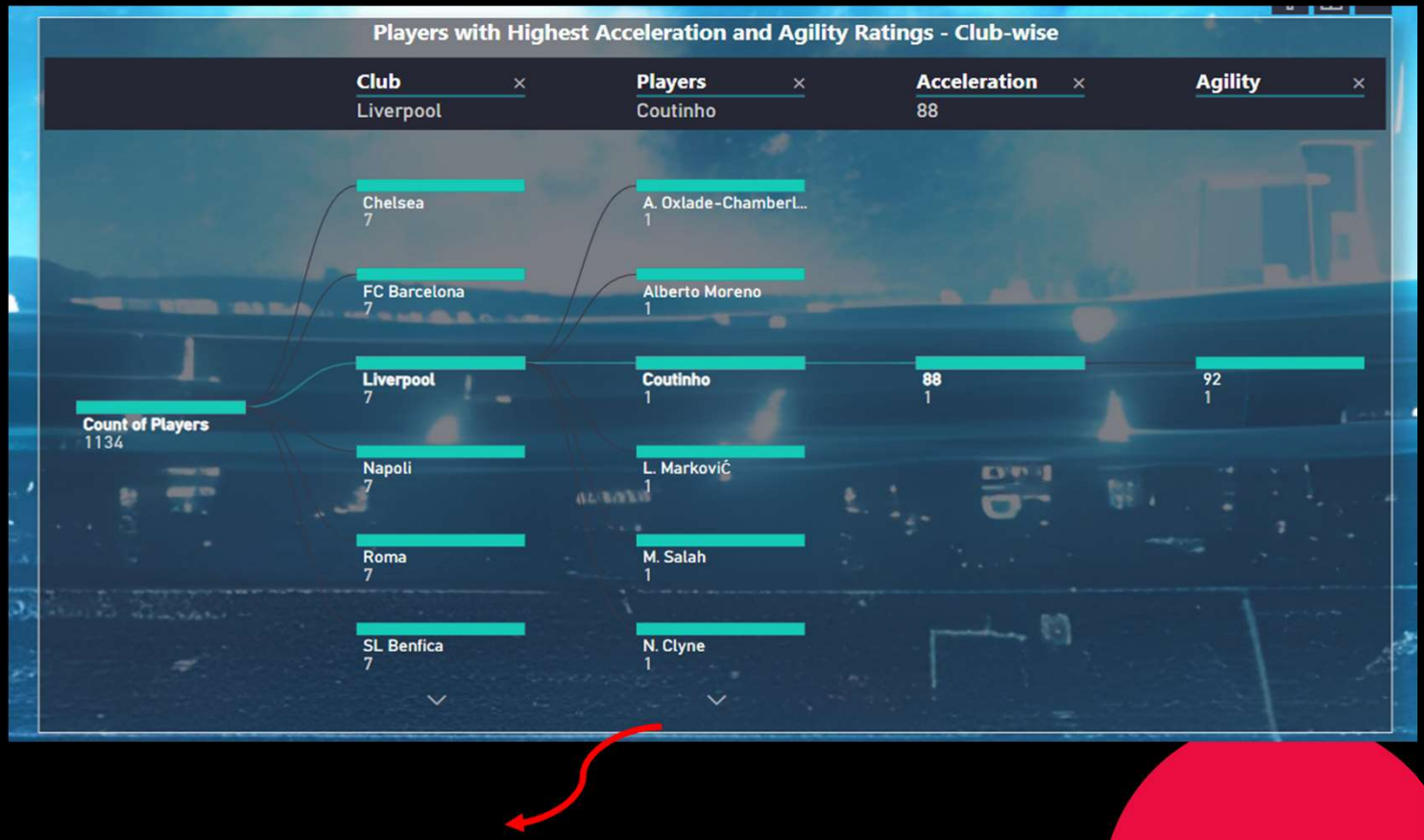
Insight: Top-performing players often occupy attacking positions like CF and CAM.

Recommendation: Teams can strengthen their attacking list to increase scoring.

I have created a separate table where I have clubbed all the positions of the players into separate rows and then counted each position. With this data, I have created this doughnut chart to find the most common position by Top-performers.

Task 5

Identify players with the highest acceleration and agility ratings.



Insight: Players with high acceleration and agility have better ball control and speed.

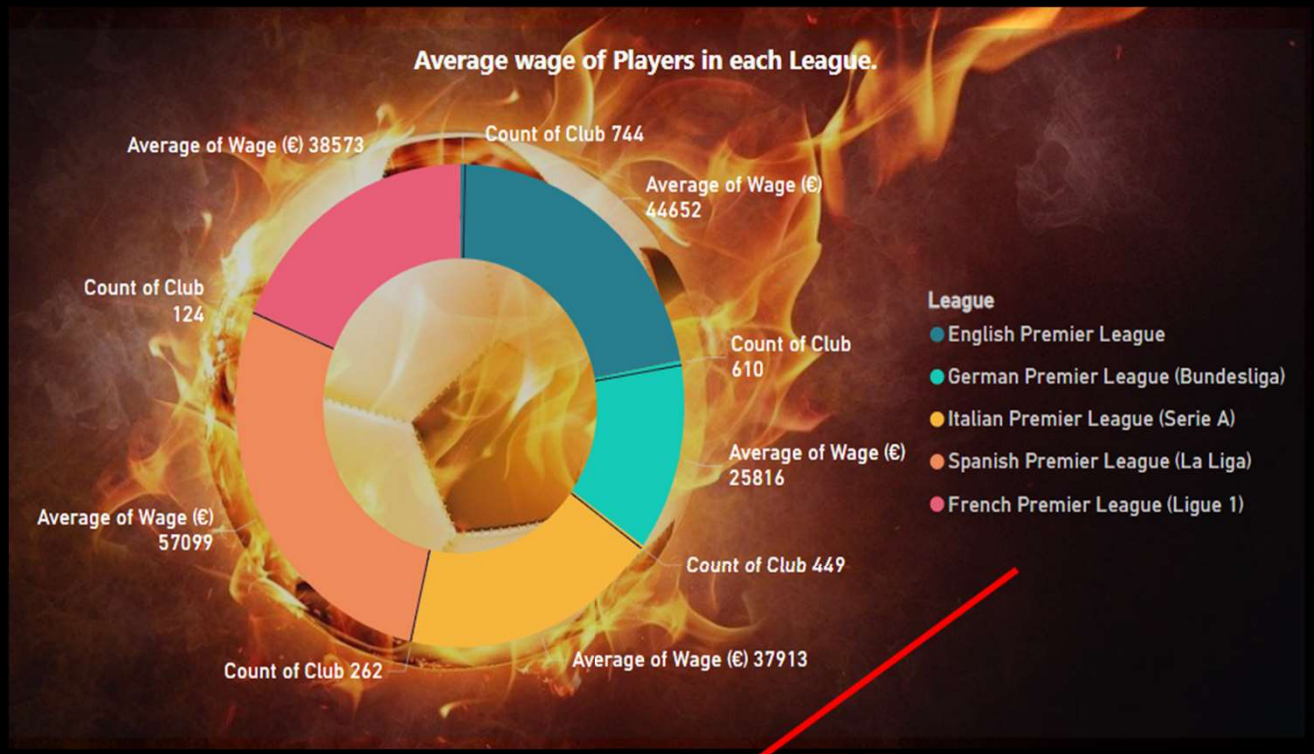
Suggestion: These players should be prioritized for fast-paced games.

Task 6

Calculate the average wage for players in each league.

Insight: Wages differ significantly across leagues, with top leagues paying the most.

Recommendation: Teams must budget wisely to attract talented players from high-paying leagues



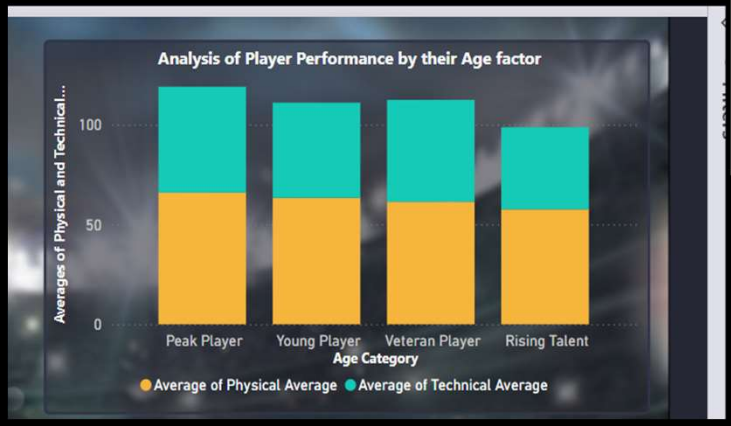
Since the given dataset did not include league information,
I manually created leagues and grouped the clubs into **five major leagues** for analysis.

Task 7

Discover trends in player performance over the last few seasons.

Insight: Since season data is unavailable, players' age was considered to analyze performance trends. Younger players tend to show better growth potential while older players maintain consistent performance levels.

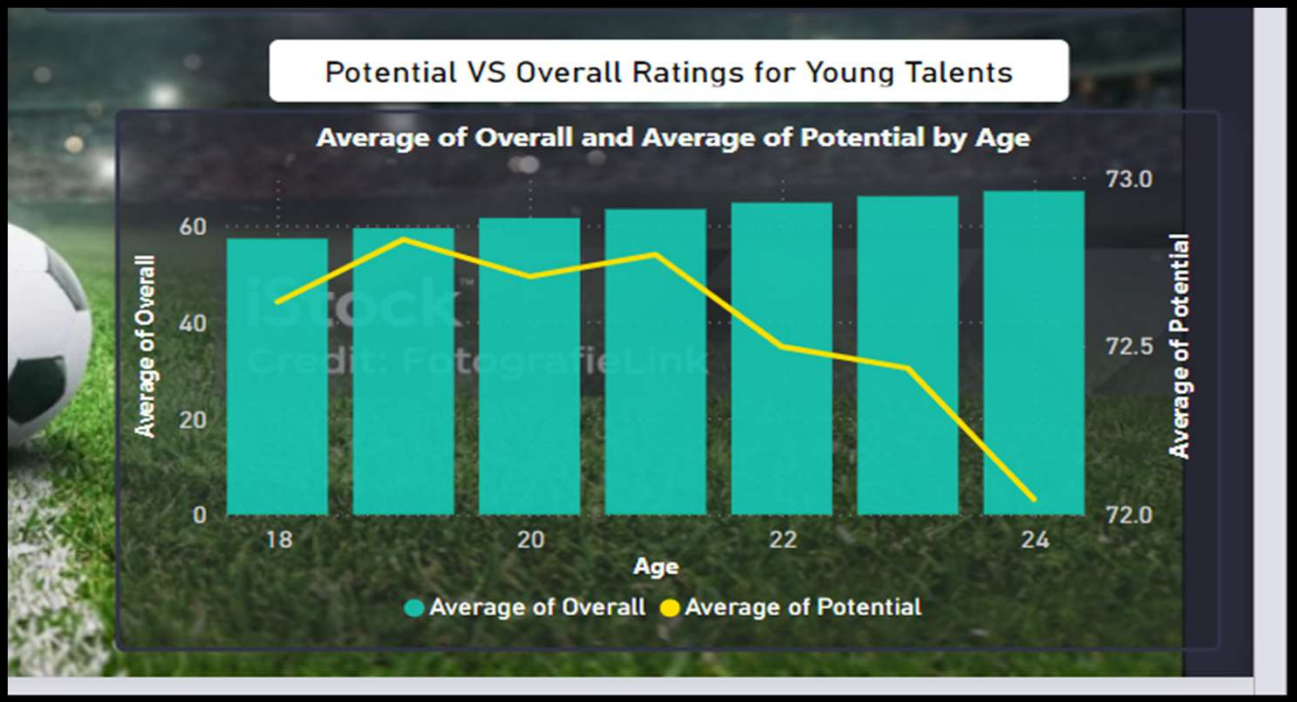
Suggestion: Teams should invest in younger players for long-term development while leveraging experienced players for stability and leadership on the field.



Task 8

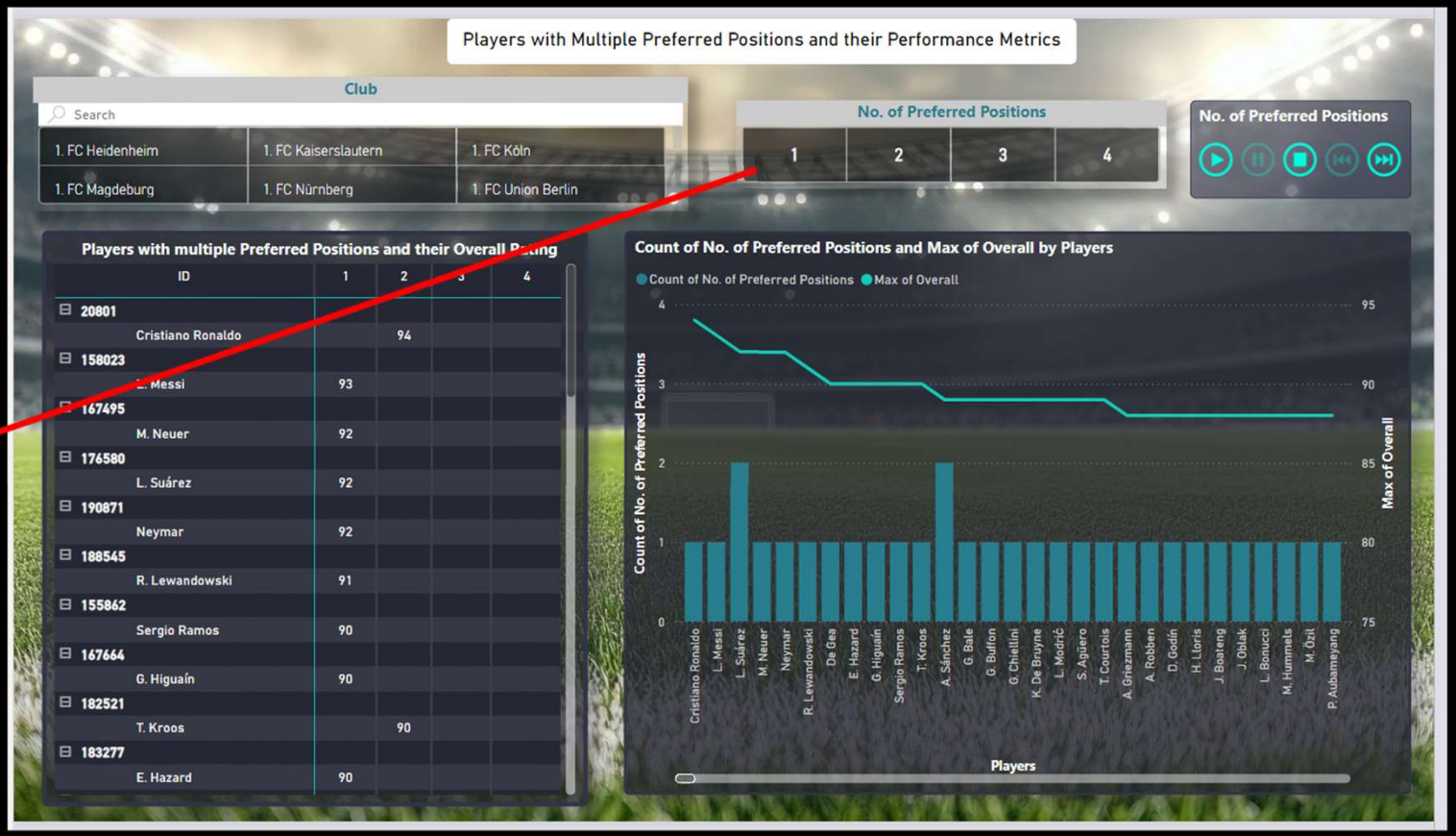
Determine the potential vs. overall rating for young talents.

Task 9



Identify players with multiple preferred positions and their performance metrics.

Task 10

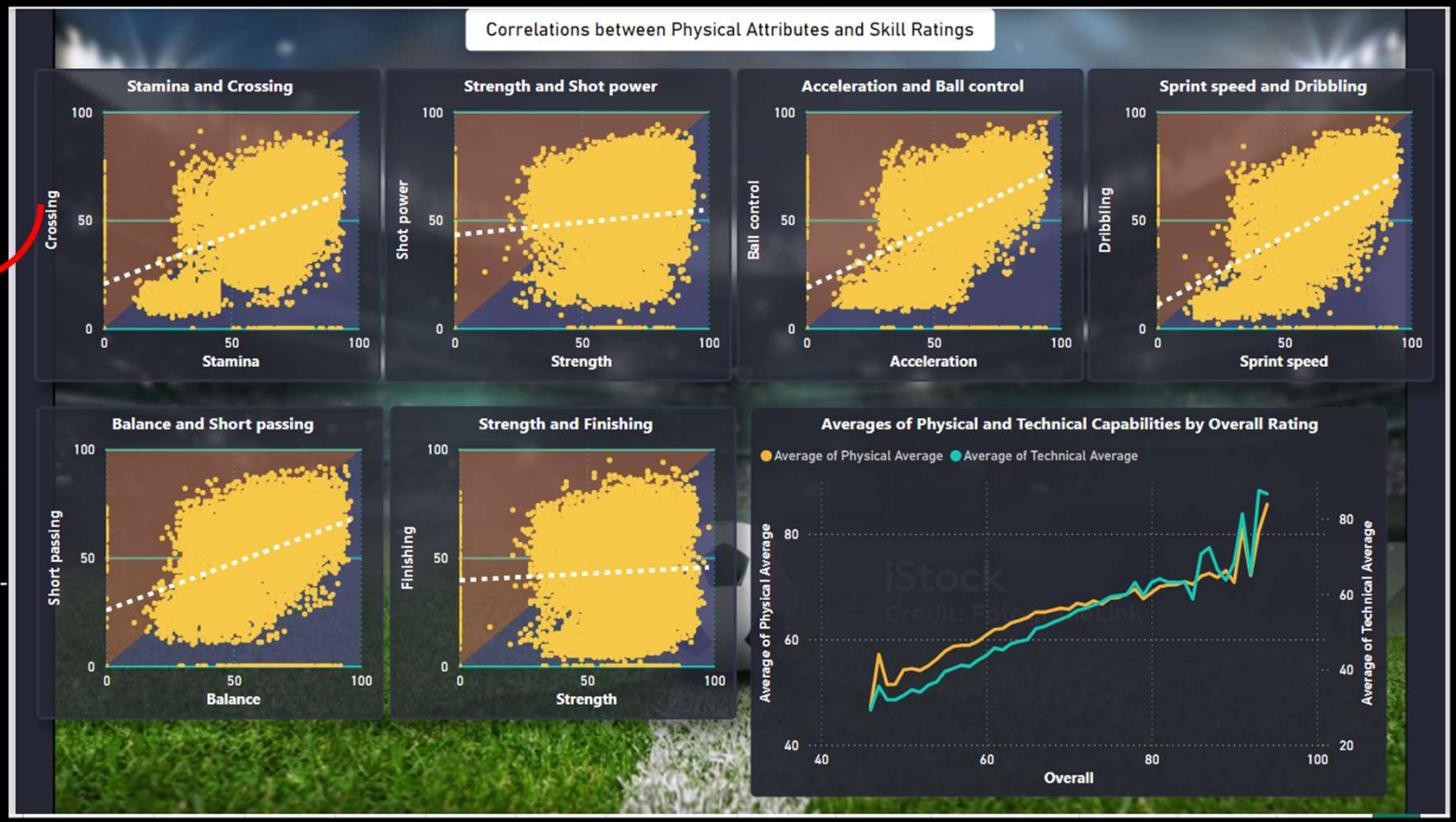


Insight: Versatile players perform well in multiple positions.

Suggestion: Utilize these players strategically to fill gaps in team formations.

Analyze correlations between physical attributes and skill ratings.

Task 11



Insight: Strong physical attributes improve skill ratings like dribbling and shot power.

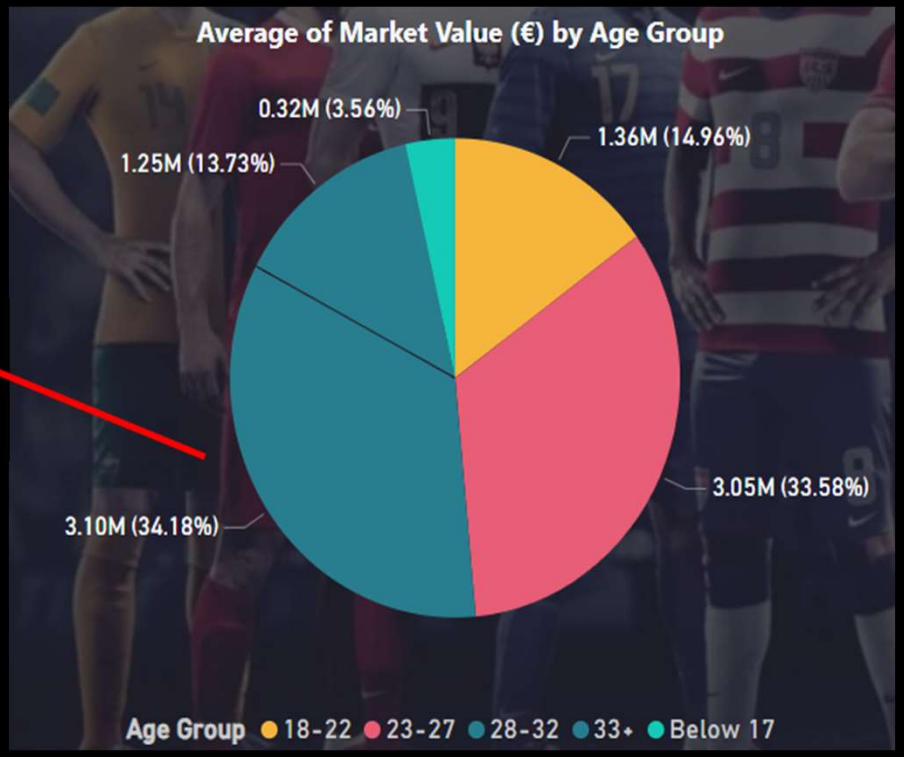
Recommendation: Train players to enhance physical fitness for better on-field performance.

Task 12

Calculate the average market value of players by age group.

Insight: Players aged 24-28 have the highest market value.

Recommendation: Teams should focus on this age group for immediate returns on investment.

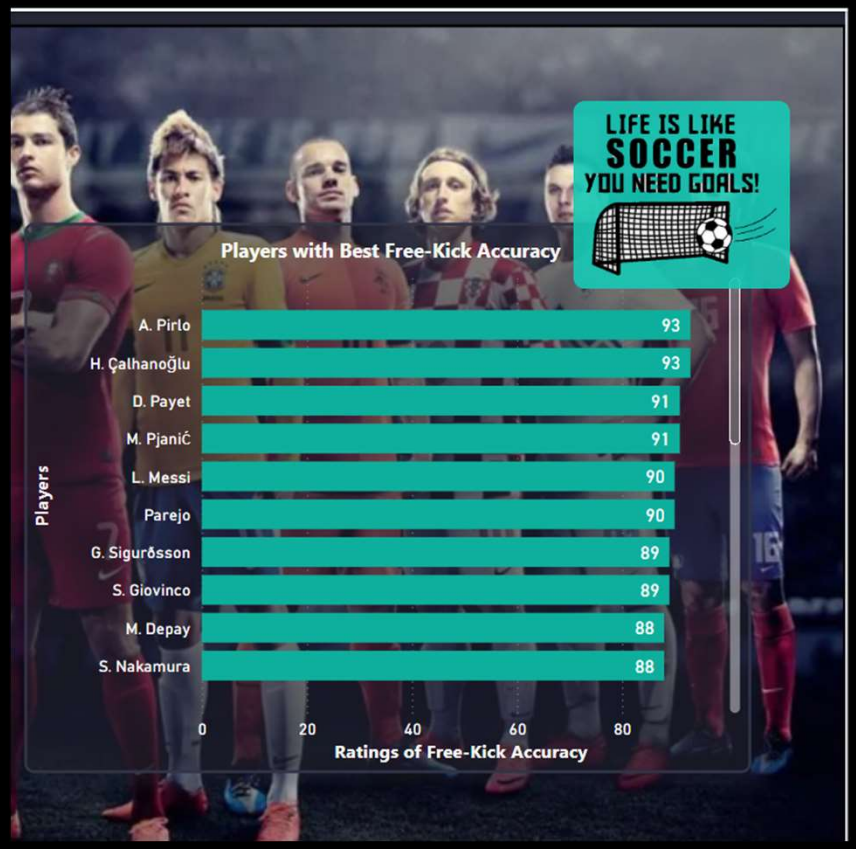


Task 13

Identify players with the best free-kick accuracy.

Insight: Players with high free-kick accuracy are valuable for set-piece goals.

Recommendation: Use these players in critical set-piece situations to increase goal opportunities.



Create visualizations for player statistics comparisons by club.

Task 14

Insight: Clubs vary in player statistics, with top clubs excelling in overall performance.

Suggestion: Lower-performing clubs should analyze top clubs' strategies to improve results.

Insight: Players with declining performance are at higher risk of transfer.

Recommendation: Monitor underperforming players and provide additional training to retain them.

These buttons help to identify if the selected player is **in risk** or **safe** position.

Develop queries to identify players at risk of being transferred based on performance.

Task 15


SQL QUERIES

SQL Commands used

DDL – Data Definition Language - Create, Drop, Alter

DML – Data Manipulation Language - Insert, Update, Delete

DQL – Data Query Language - Select

Clauses used

where, having

order by, group by

limit

Operators used

LOGICAL – and, or, in

COMPARISON – between

NOT IN

Aggregate functions used

MAX, MIN

SUM, AVG

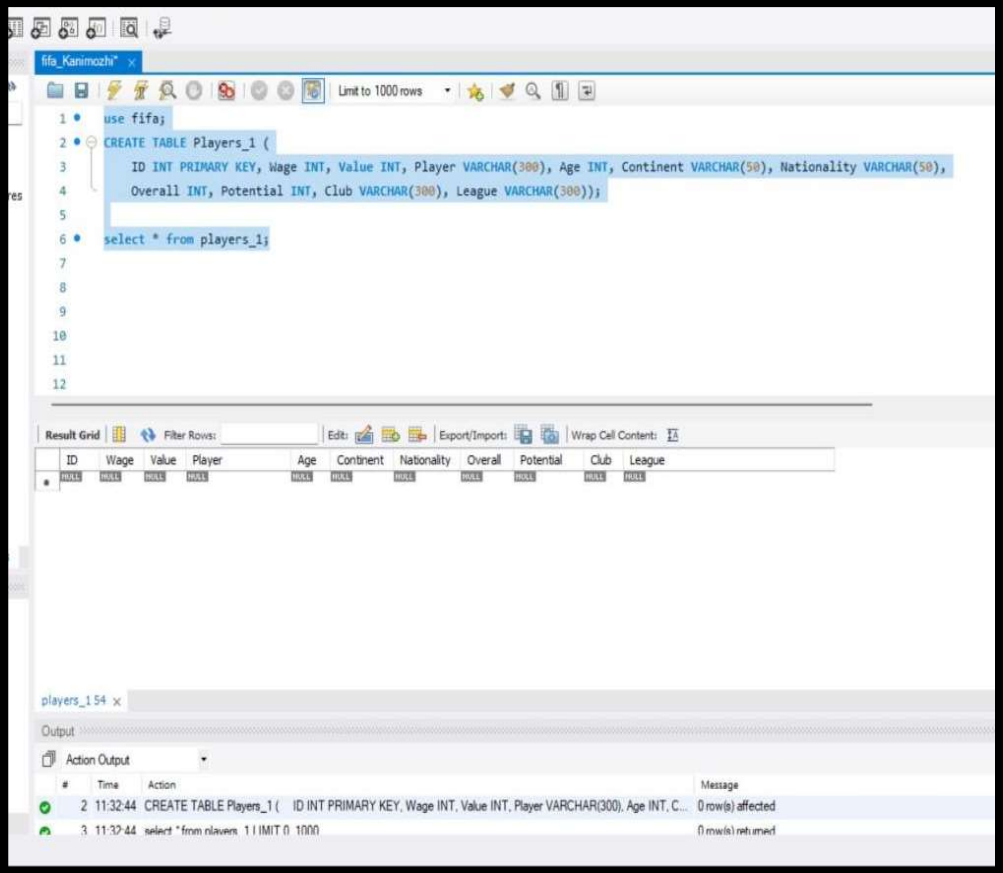
COUNT

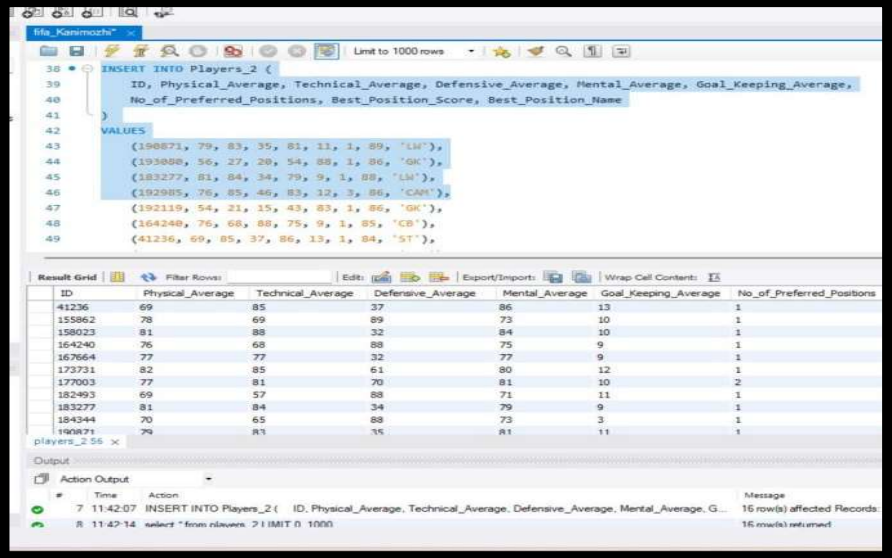
Joins used

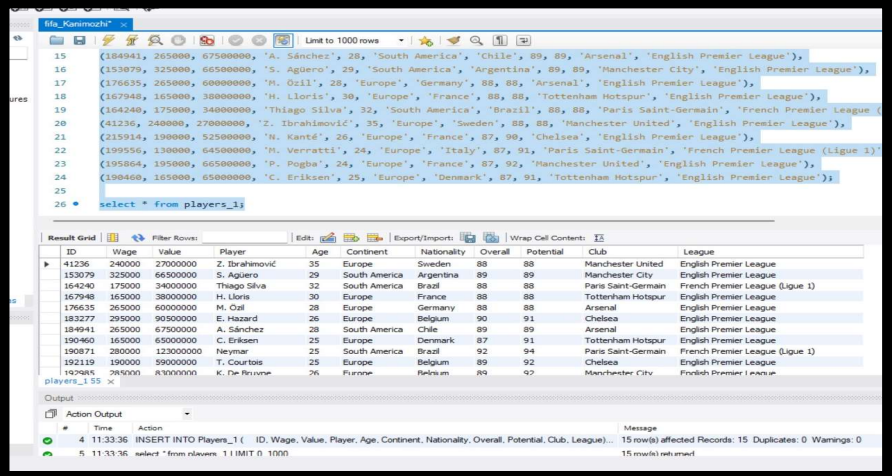
Inner Join, Union

Right Join, Left Join

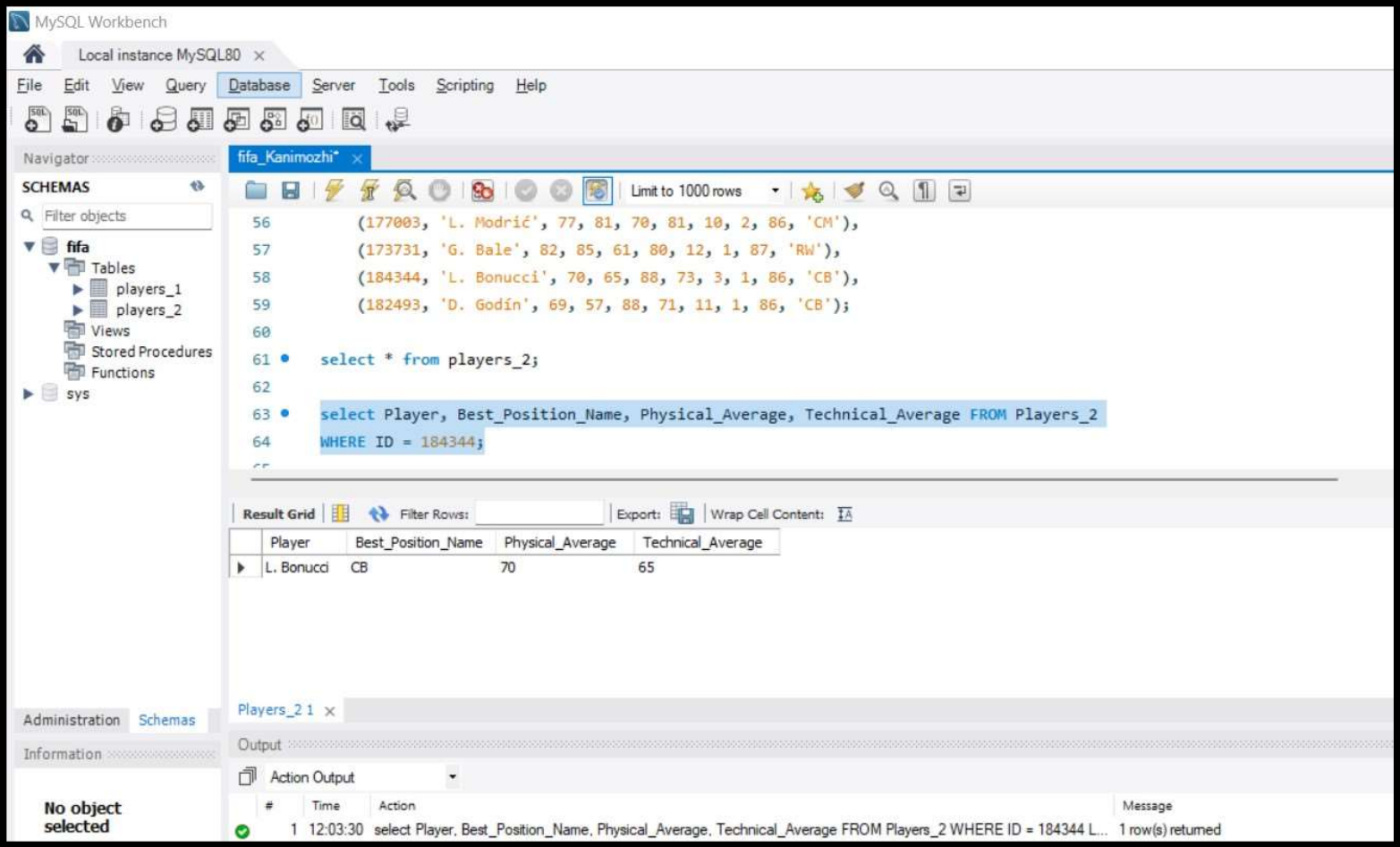
Cross Join, Self-Join



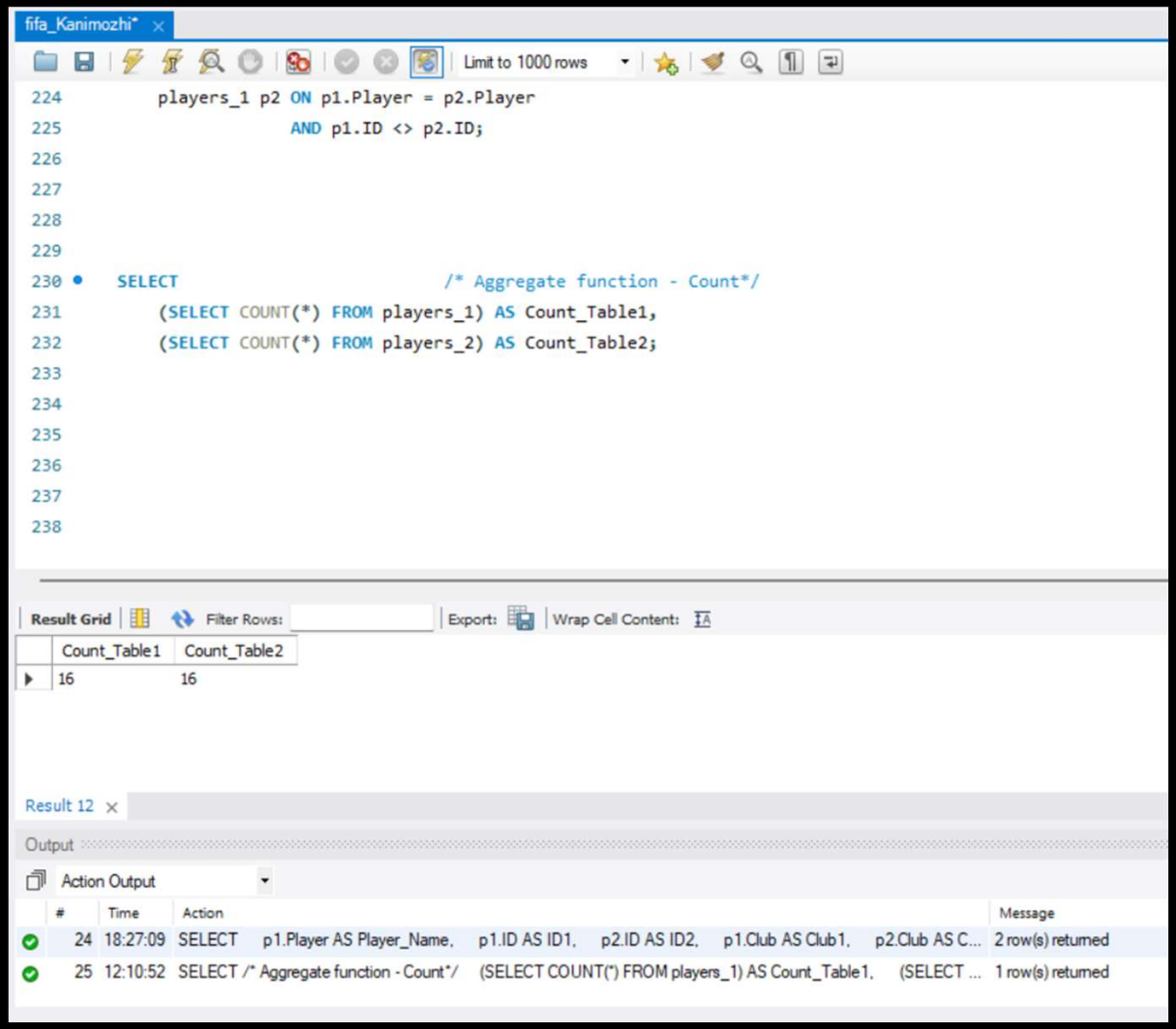




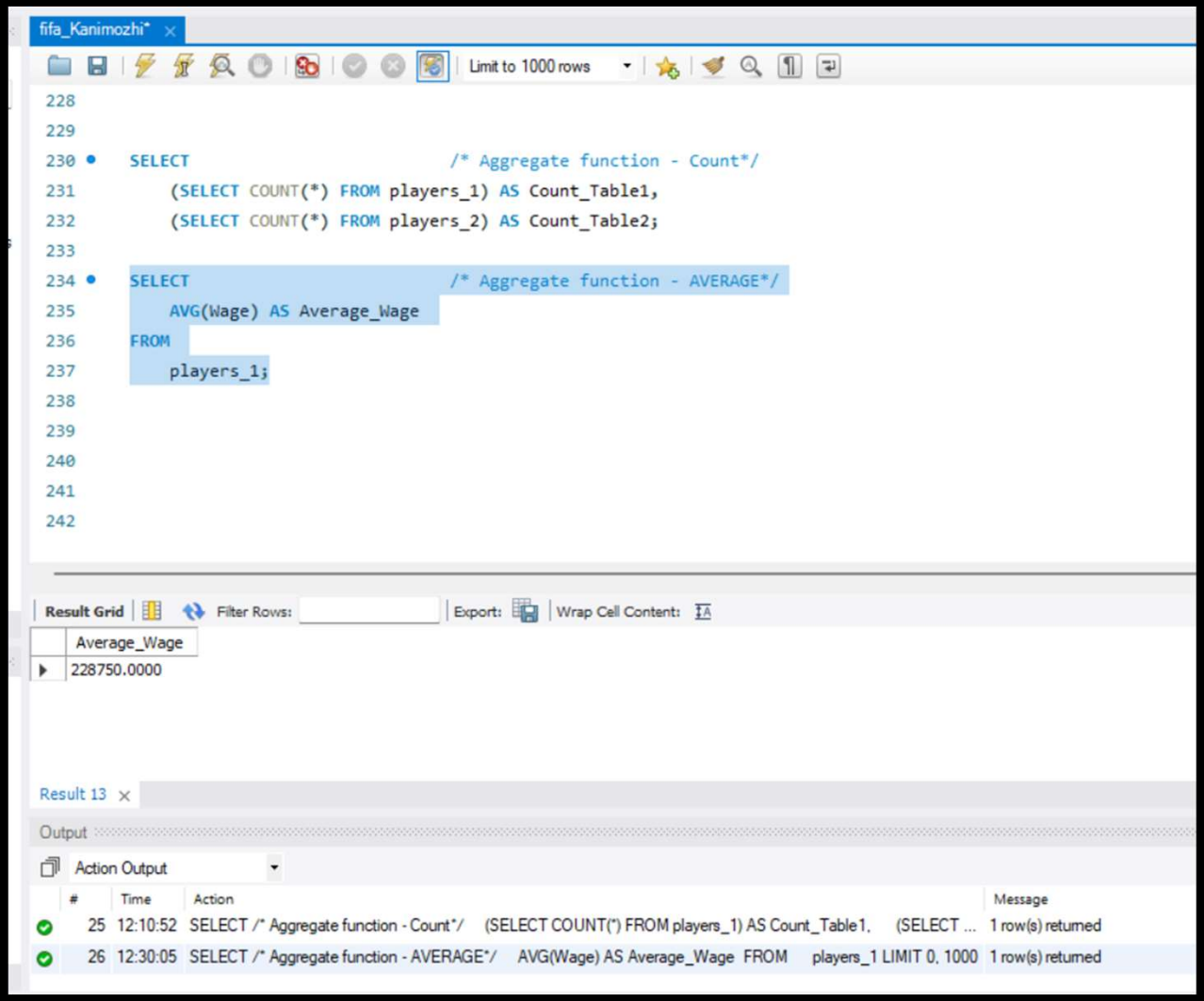
- Creating database and tables 1 and 2.
- Inserting records into the tables and retrieving them all using ‘SELECT’ command.



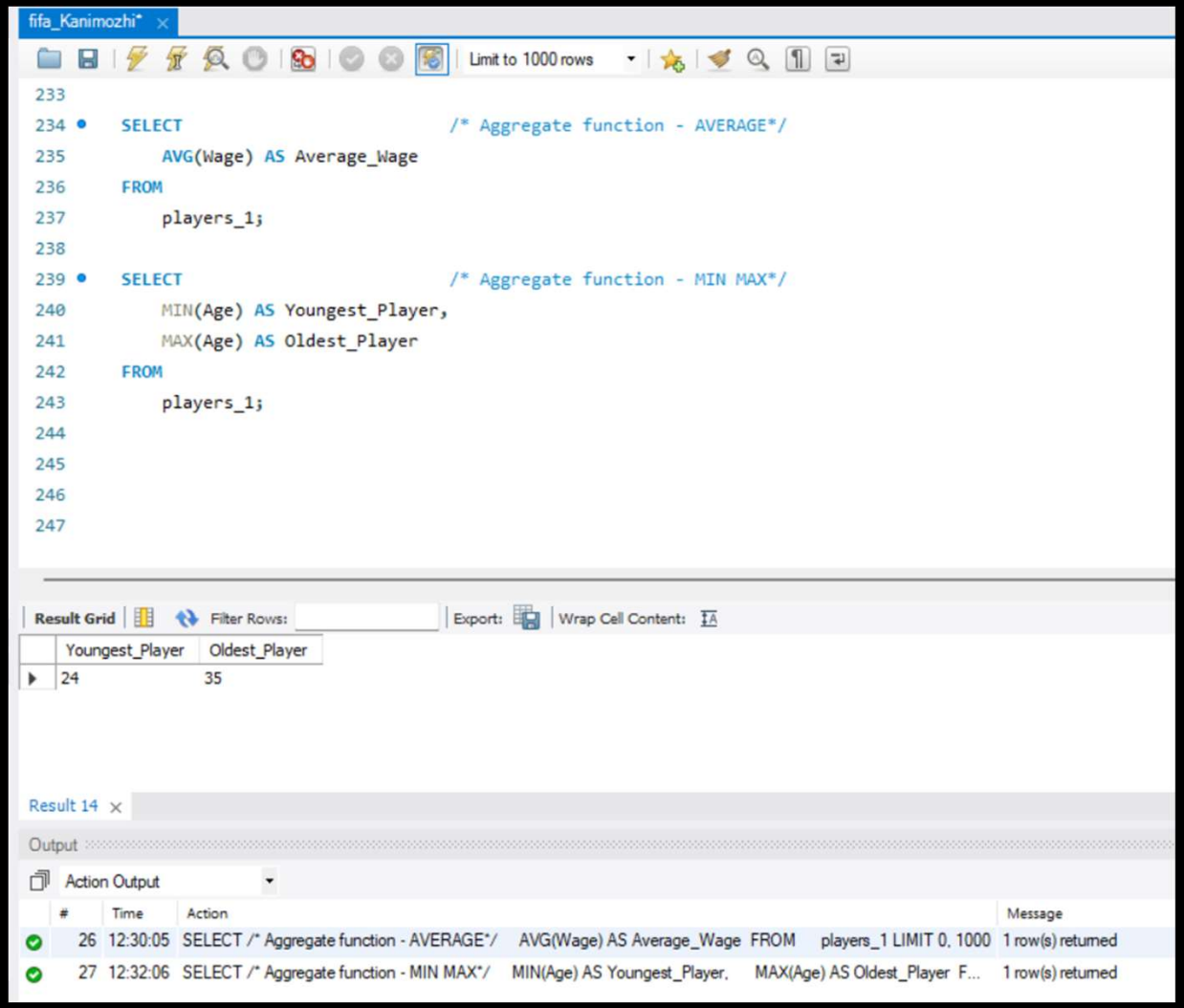
- Retrieves specific columns for a player with ID 184344.
- SELECT command with a WHERE clause for filtering rows.



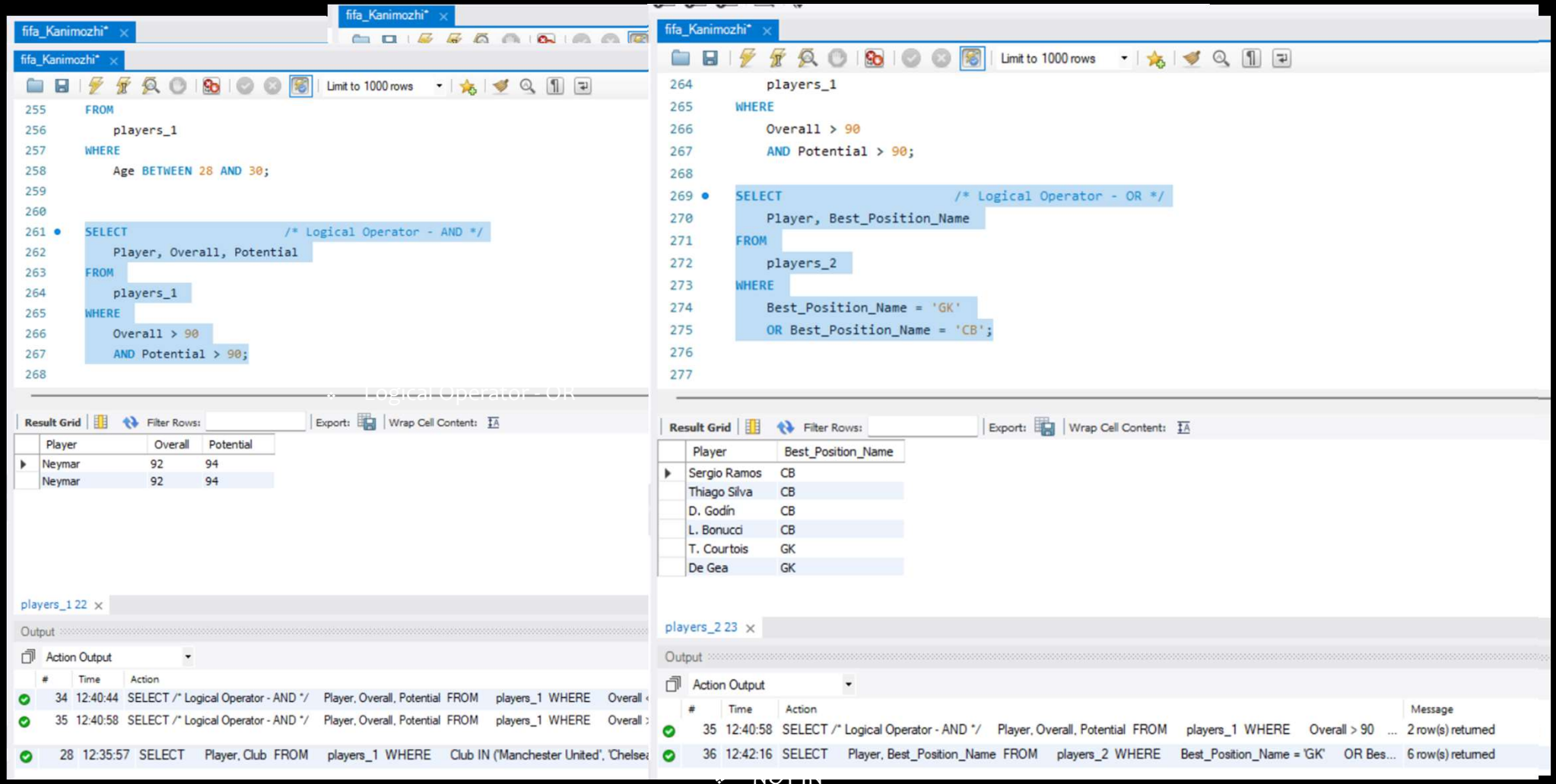
- Count function

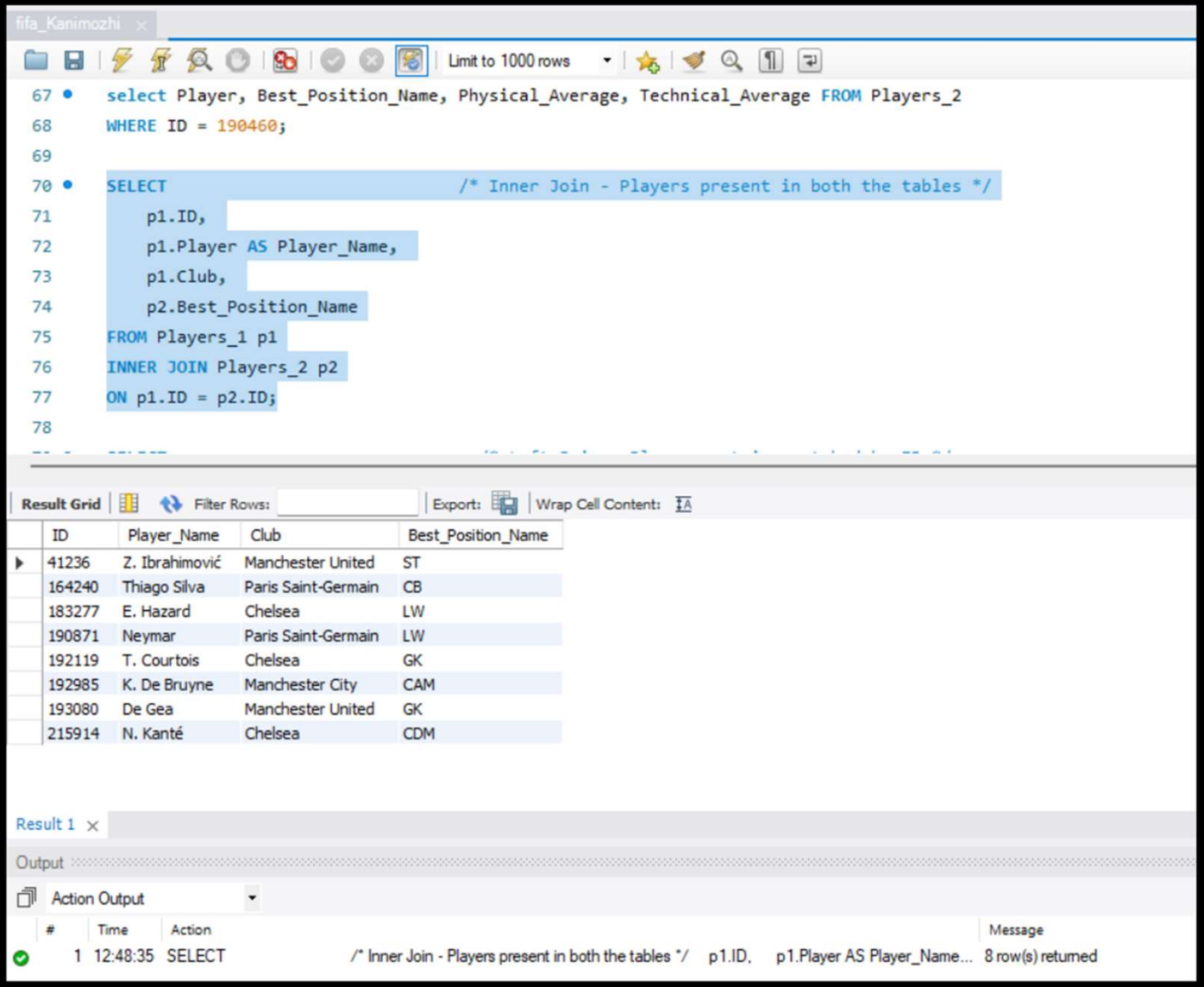


- AVERAGE function

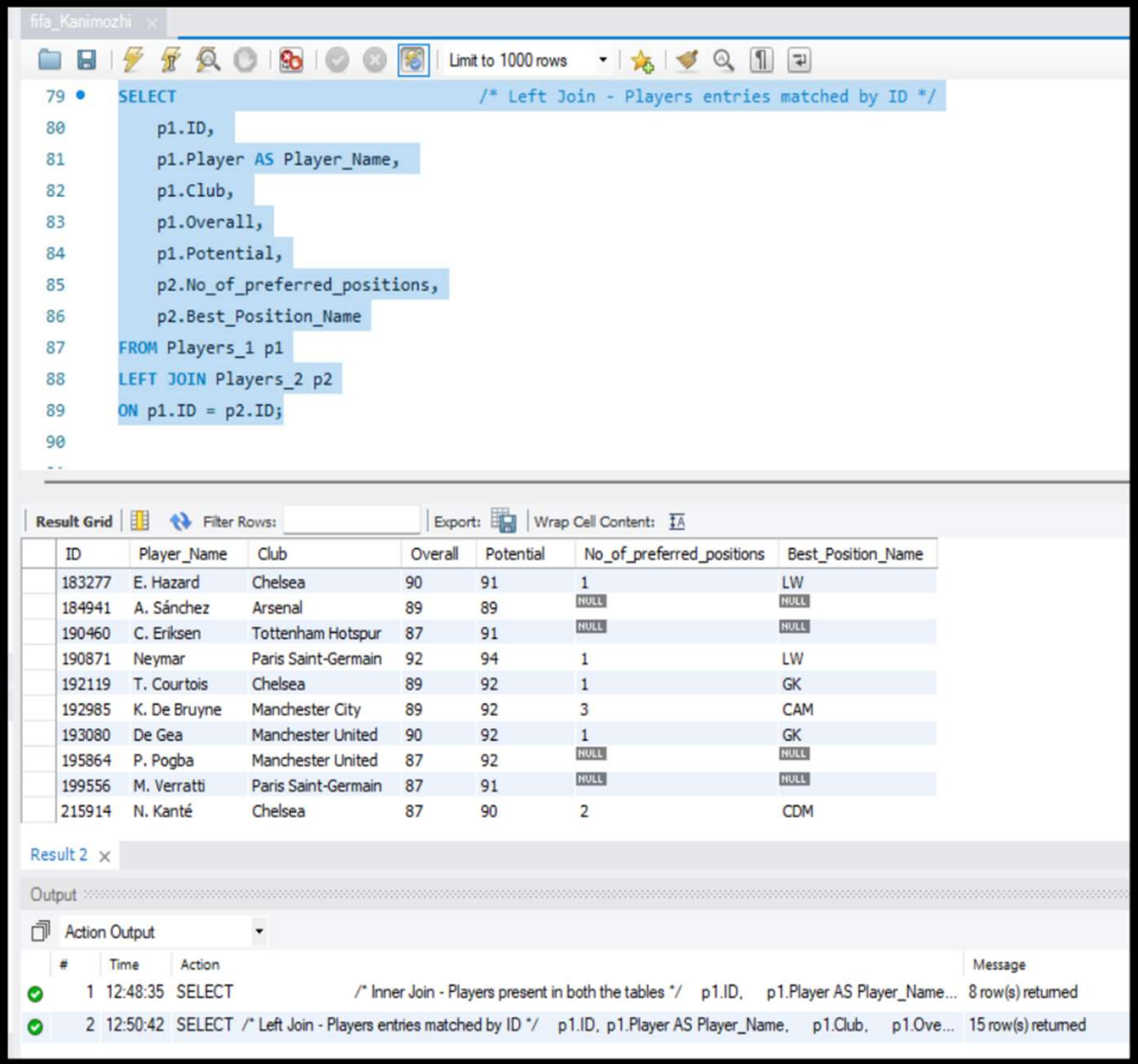


- Minimum and Maximum function

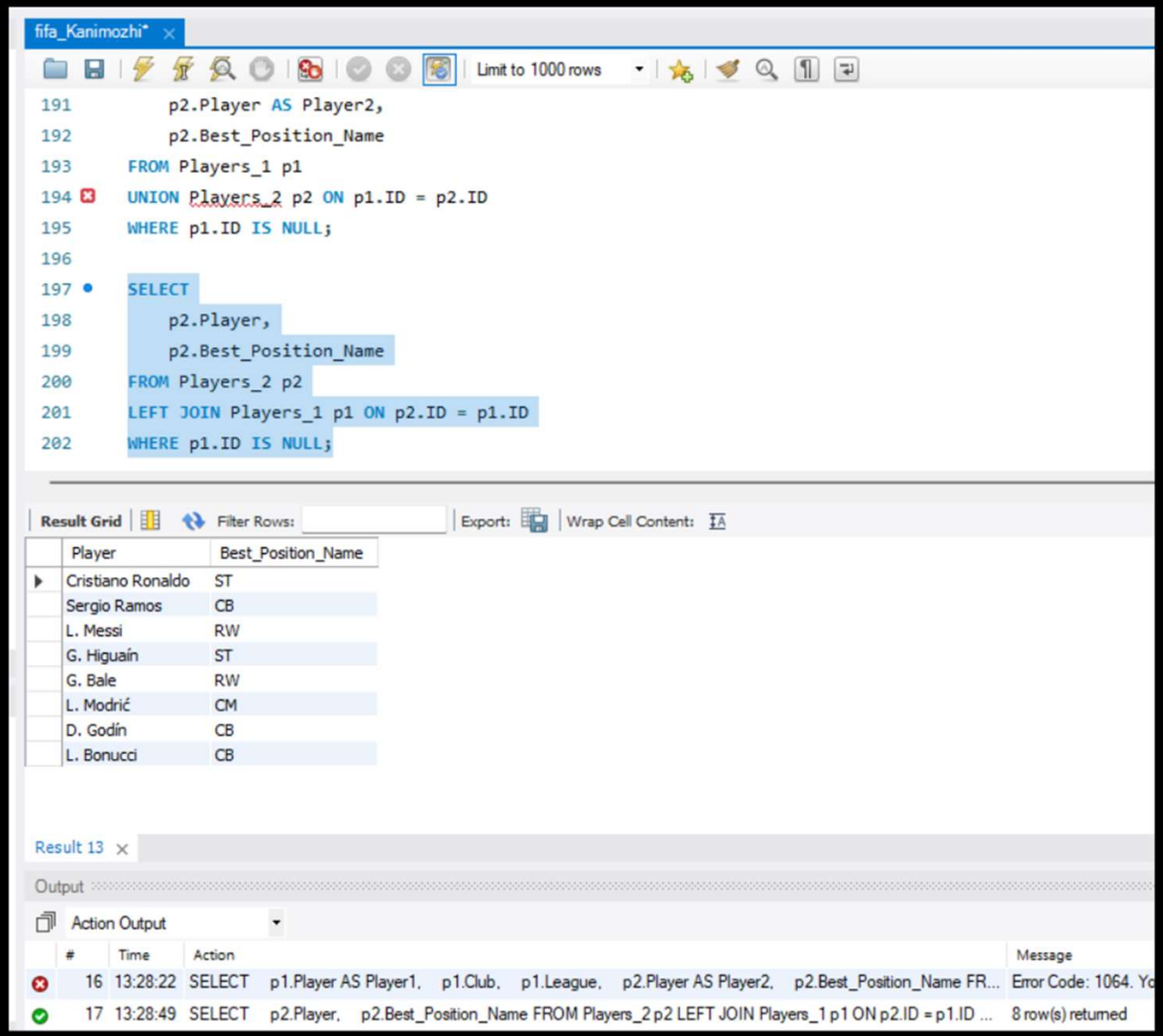

fifa_Kanimozhi*
Limit to 1000 rows
FROM
players_1
WHERE
Age BETWEEN 28 AND 30;
SELECT /* Logical Operator - AND */
Player, Overall, Potential
FROM
players_1
WHERE
Overall > 90
AND Potential > 90;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Player Overall Potential
Neymar 92 94
Neymar 92 94
players_1 22
Output
Action Output
Time Action
34 12:40:44 SELECT /* Logical Operator - AND */ Player, Overall, Potential FROM players_1 WHERE Overall
35 12:40:58 SELECT /* Logical Operator - AND */ Player, Overall, Potential FROM players_1 WHERE Overall
28 12:35:57 SELECT Player, Club FROM players_1 WHERE Club IN ('Manchester United', 'Chelse
players_1
WHERE
Overall > 90
AND Potential > 90;
SELECT /* Logical Operator - OR */
Player, Best_Position_Name
FROM
players_2
WHERE
Best_Position_Name = 'GK'
OR Best_Position_Name = 'CB';
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Player Best_Position_Name
Sergio Ramos CB
Thiago Silva CB
D. Godín CB
L. Bonucci CB
T. Courtois GK
De Gea GK
players_2 23
Output
Action Output
Time Action Message
35 12:40:58 SELECT /* Logical Operator - AND */ Player, Overall, Potential FROM players_1 WHERE Overall > 90 ... 2 row(s) returned
36 12:42:16 SELECT Player, Best_Position_Name FROM players_2 WHERE Best_Position_Name = 'GK' OR Bes... 6 row(s) returned



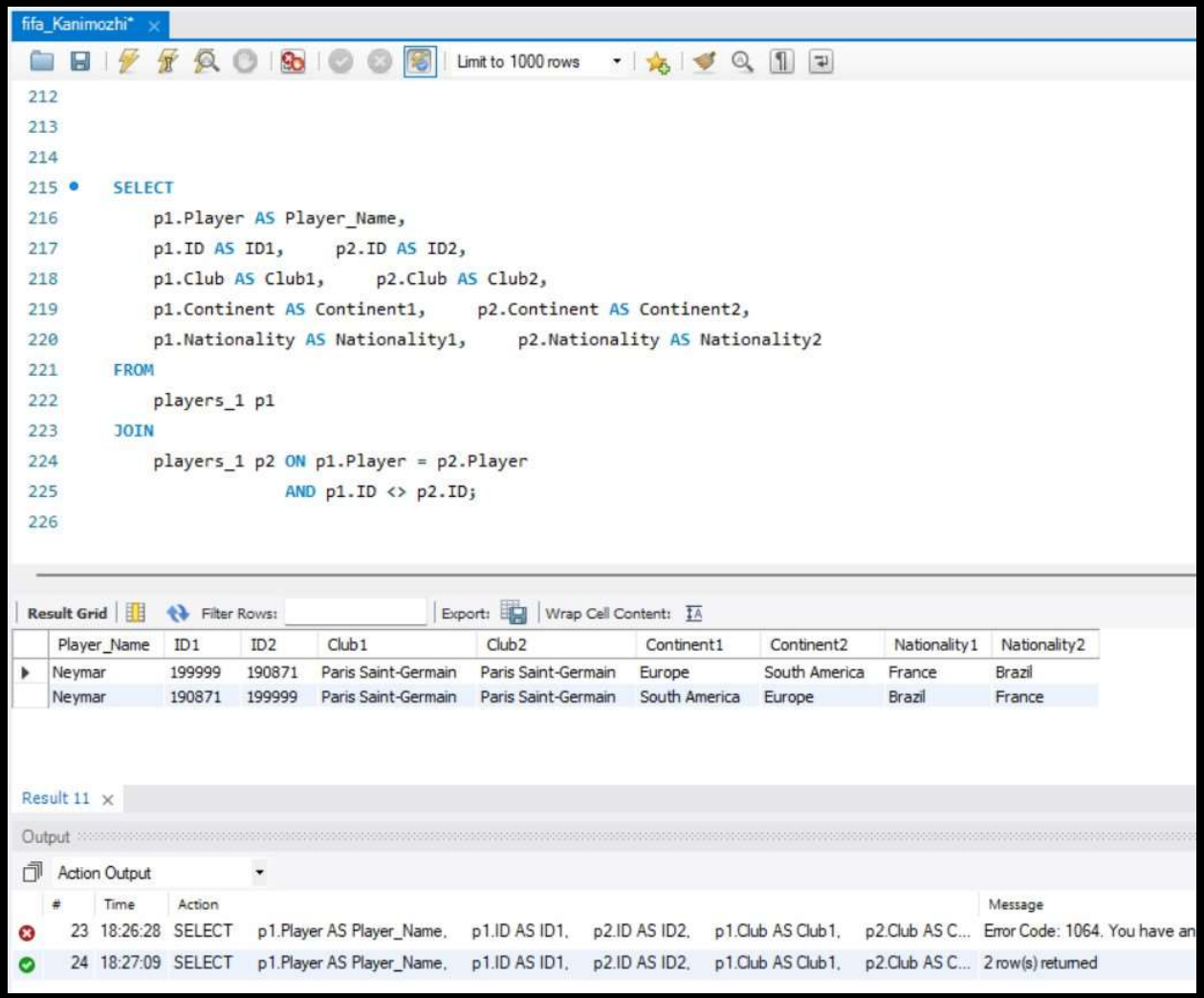
- Displays players present in both tables with their club and best position.
- Uses INNER JOIN to match entries based on ID.



- This query retrieves all players from Players_1 and matches them with players in Players_2. If a match does not exist in Players_2, the respective columns will have NULL values.
- Uses LEFT JOIN to include all records from the left table (Players_1), whether or not they have a matching entry in Players_2.

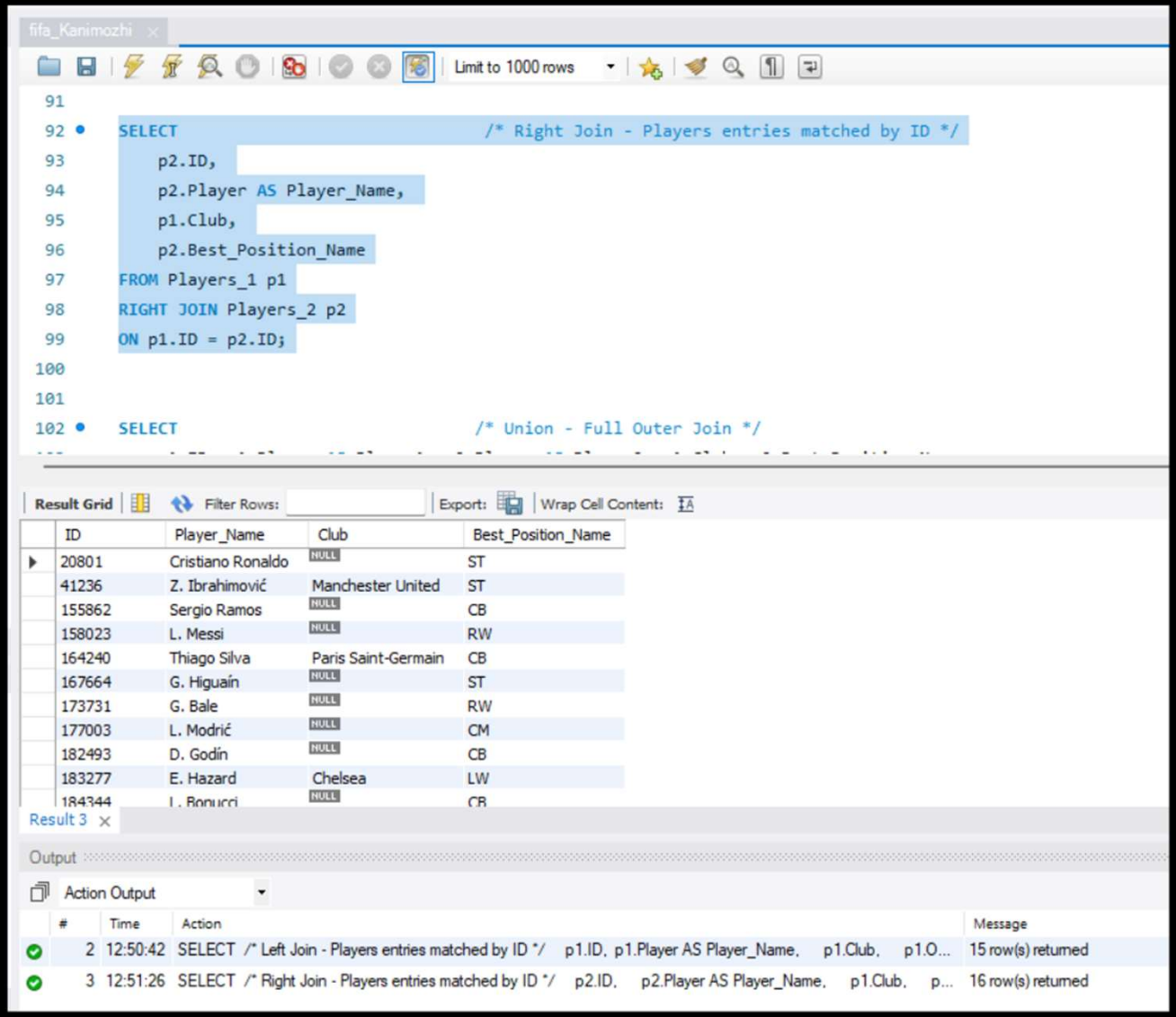


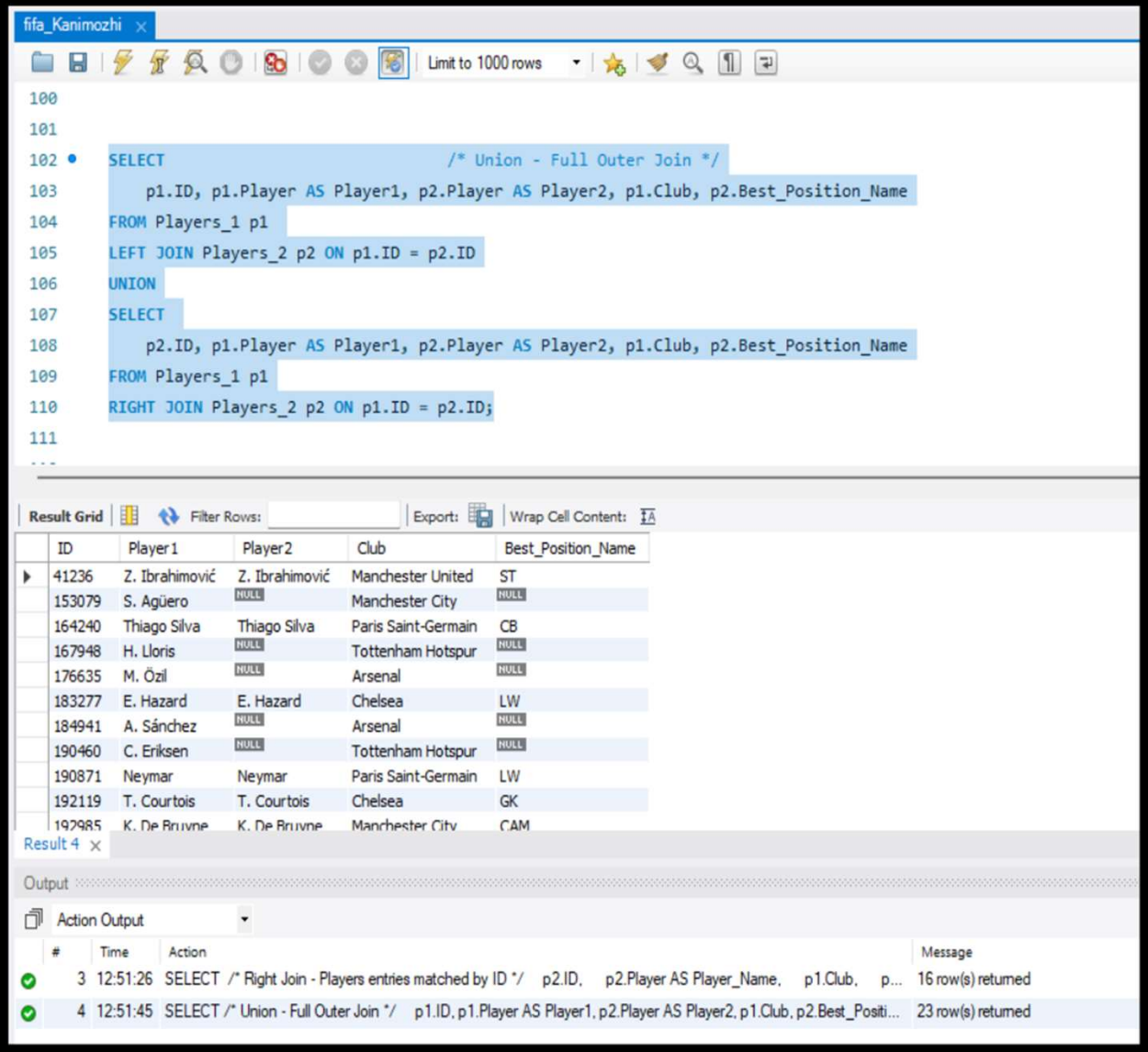
- Search for the players present in Players_2 but missing from Players_1.
- LEFT JOIN with a WHERE clause checks for NULL values in Players_1.



- Self-JOIN is like comparing data within the same table.
- Here, it compares players with the same Name but with other attributes being different and identifies them.

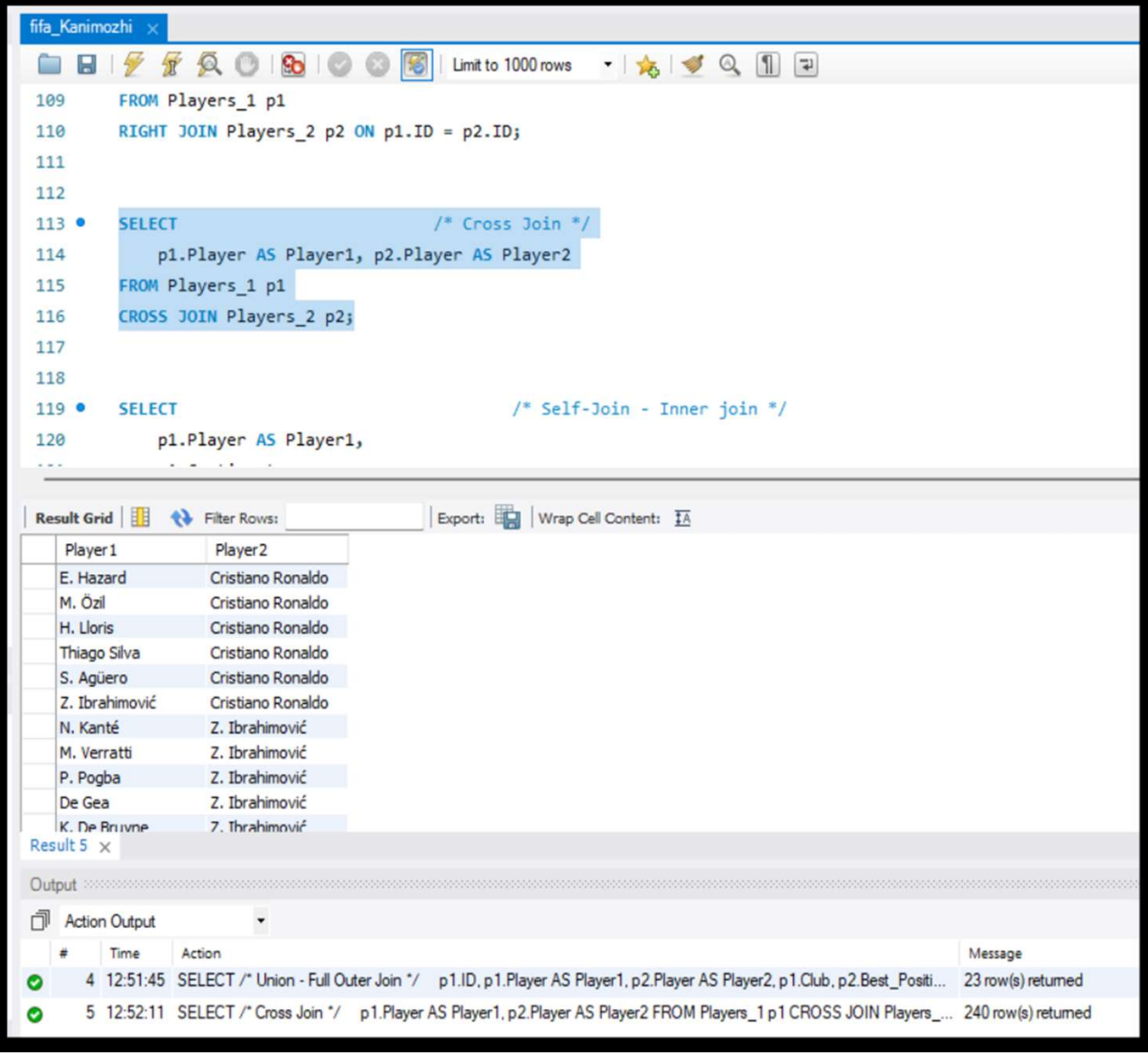
- Retrieves all records from Players_2 and matches those in Players_1, showing NULL for unmatched entries from Players_1.
- Uses RIGHT JOIN to prioritize Players_2, ensuring its unmatched entries are included.

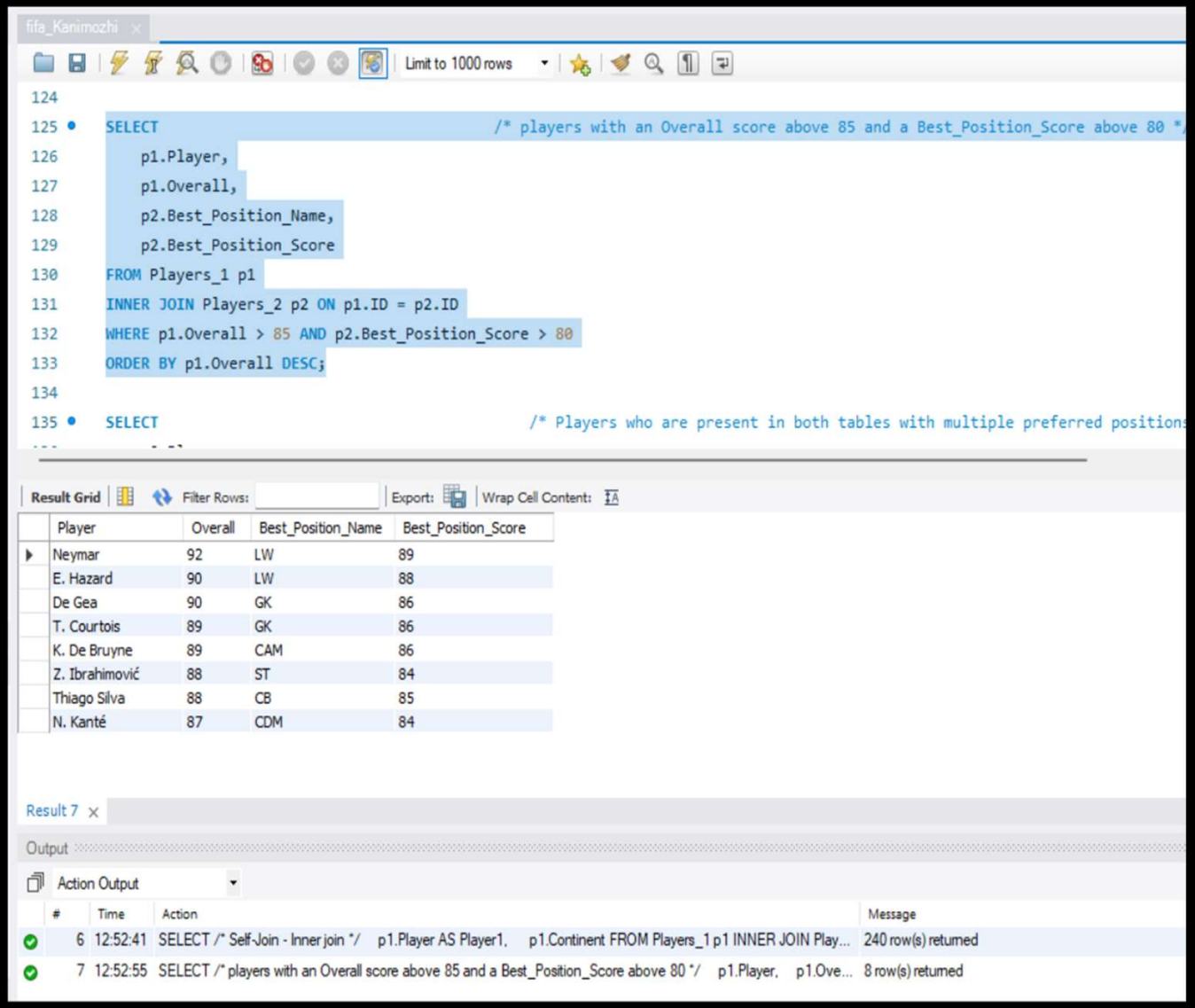




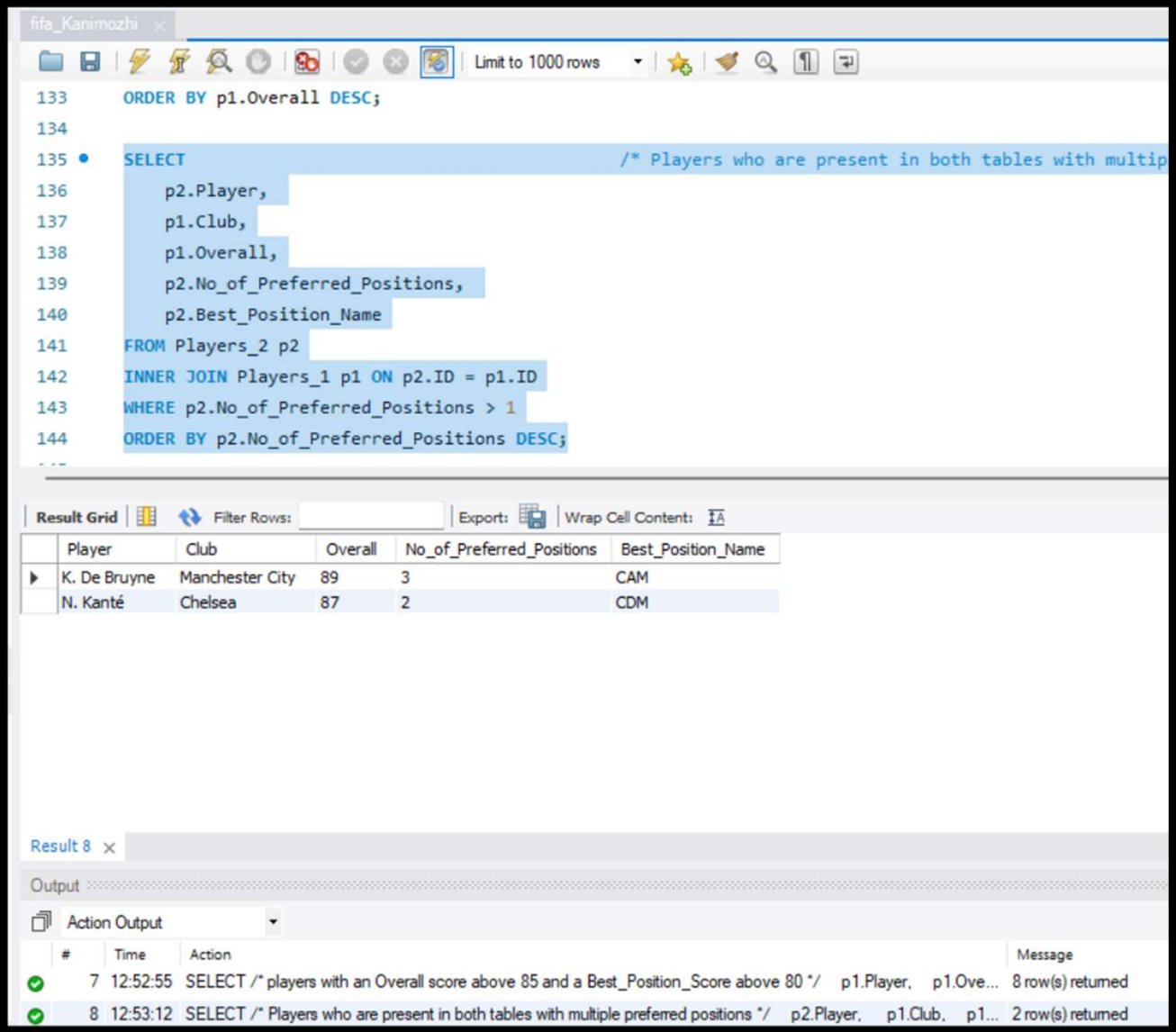
- Combines LEFT JOIN and RIGHT JOIN with UNION to include all rows from both tables and identify mismatches.
- Similar to FULL OUTER JOIN functionality using a combination of joins.

- Generates all possible player combinations between Players_1 and Players_2 without any filtering.
- Uses CROSS JOIN to produce a cartesian product of the two tables.

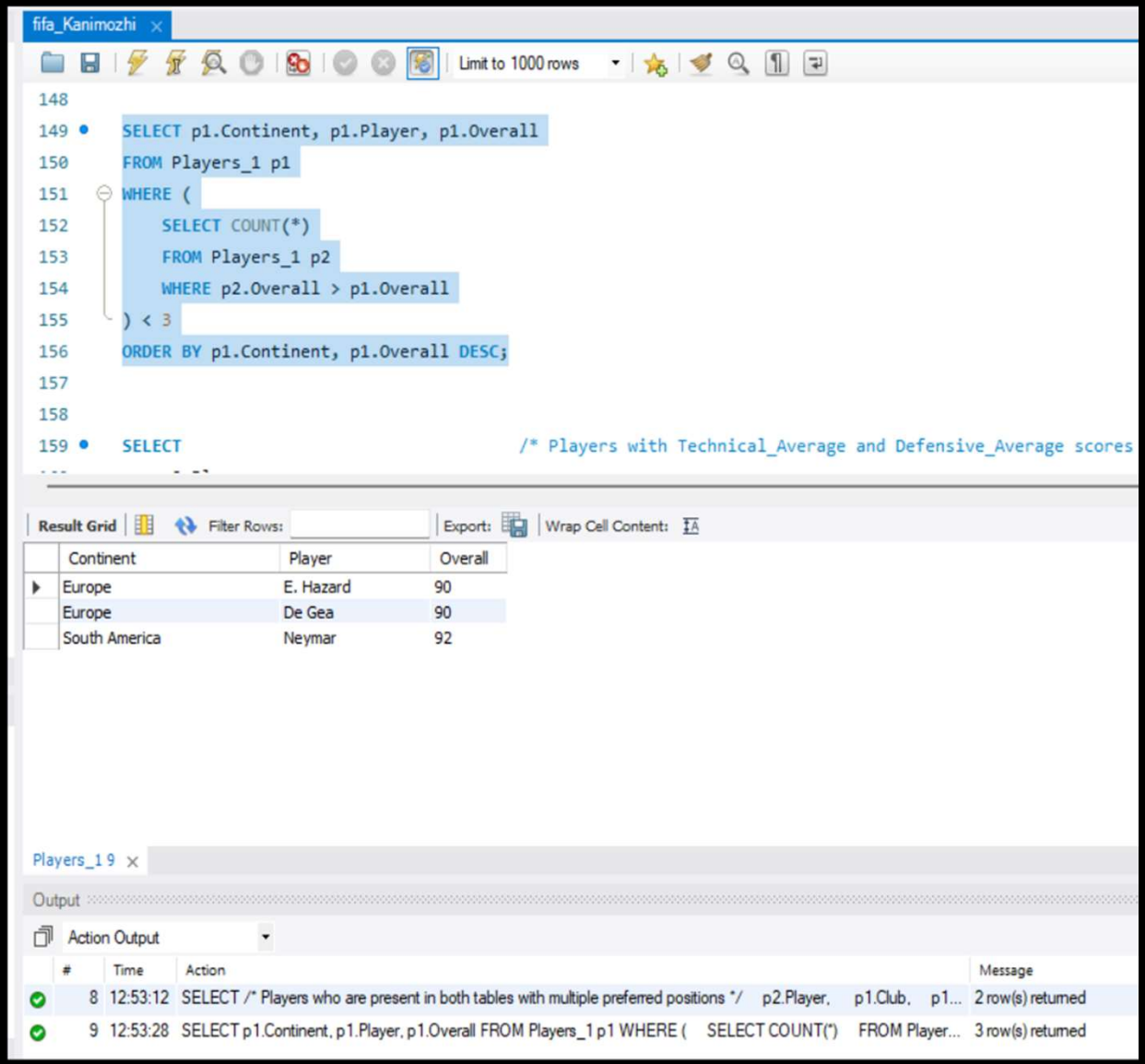




- Finds players with an overall score above 85 and preferred position score above 80, sorted by overall score.
- Combined filtering with WHERE and sorting using ORDER BY clauses.

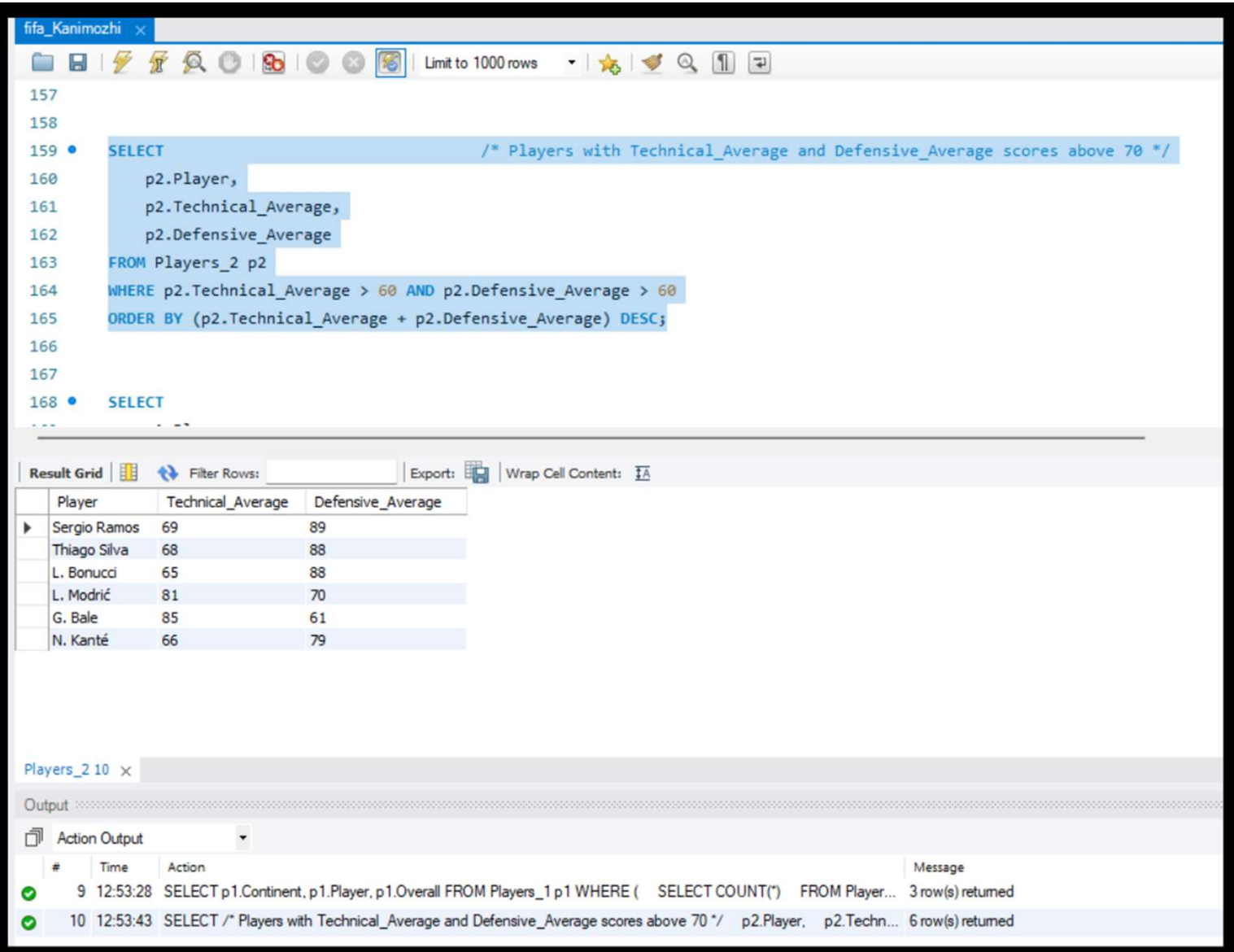


- Results players with more than one preferred position, sorted by the number of positions.
- Filtering and sorting with WHERE and ORDER BY clauses.



- Finds the top 3 players by overall rating within each continent by counting players with a higher rating.
- Combines a WHERE clause with a sub-query and ORDER BY for sorting results.

- Retrieves players whose Technical_Average and Defensive_Average exceed 60, ordered by their combined scores in descending order.
- Filtering with WHERE and custom sorting using an expression in ORDER BY.



THANK YOU
for your valuable time!

You suggestions help me grow better!