

## Tennis Data Analytics: Sport Radar API & Streamlit Dashboard

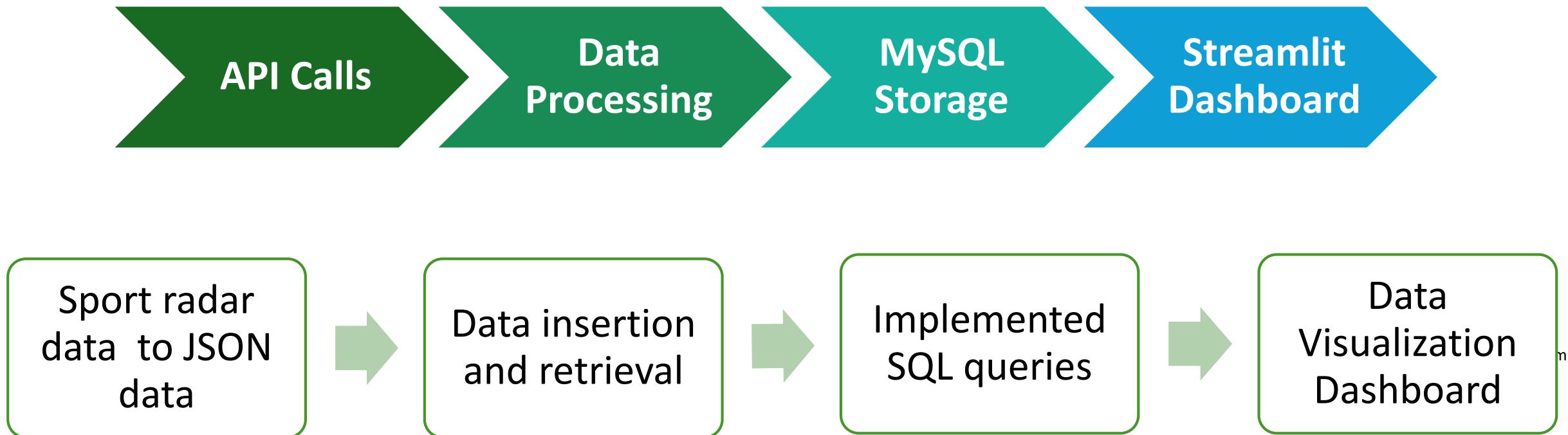
Name : Sham Prasath Kannaiyan  
Batch : DS-C-WD-E-B34  
Batch Type : WD

## Problem Statement

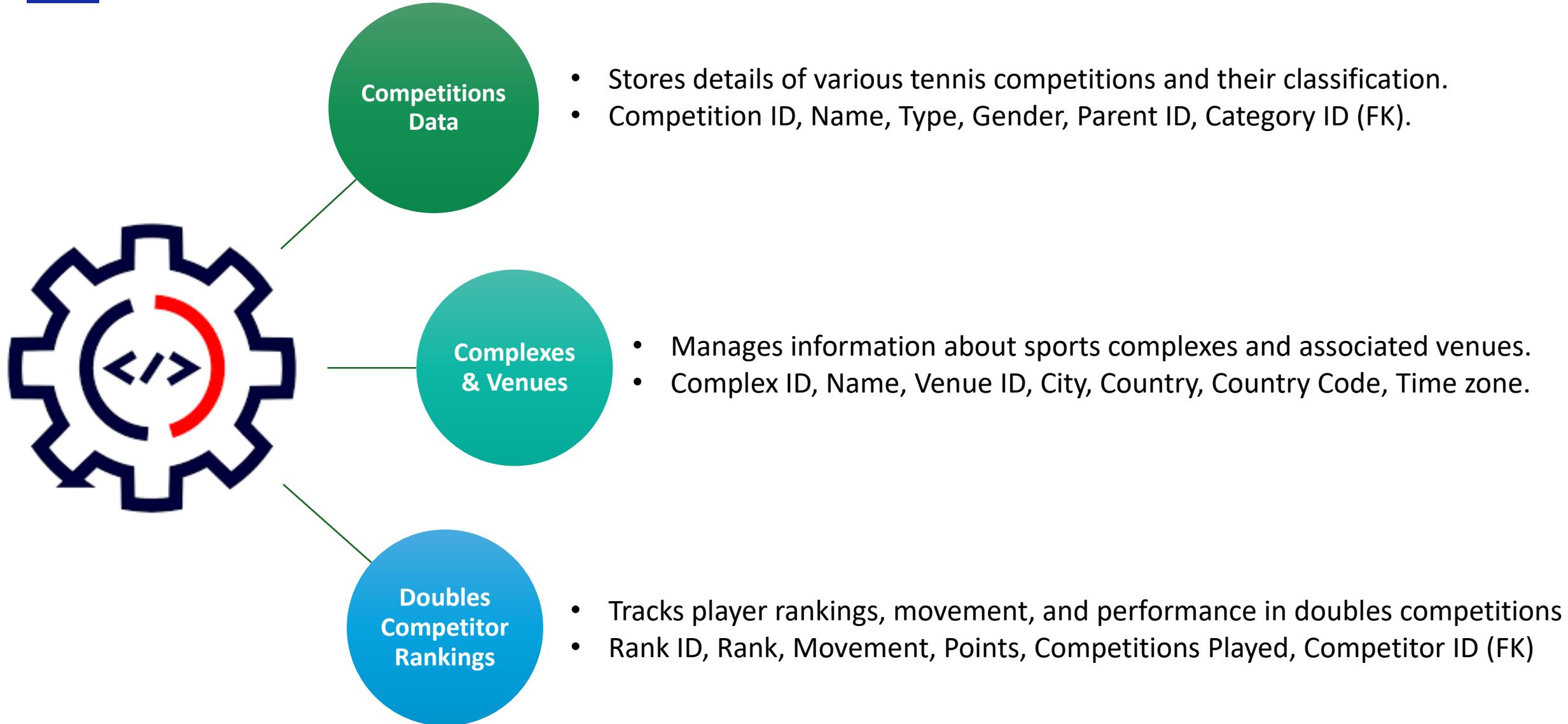
- ◇ **Comprehensive Sports Data Solution** – Manage, visualize, and analyse sports competition data from the Sport Radar API.
- ◇ **Automated Data Processing** – Parse **JSON** data, structure it in a **relational database**, and ensure efficient storage.
- ◇ **Tournament & Event Insights** – Provide **hierarchical competition structures**, event breakdowns, and key analytics.
- ◇ **Real-Time Analytics & Trends** – Assist analysts, organizations, and enthusiasts in exploring **competition trends** dynamically.
- ◇ **Interactive Visualization** – Enable **intuitive dashboards** for deeper insights into tournaments and rankings.
- ◇ **Enhanced Decision-Making** – Deliver **detailed, structured data** to facilitate better sports analysis and engagement.



## System Architecture Diagram



## Data Collection (API Integration)



## Data Processing & Transformation

### API Integration

Used Python requests to fetch real-time data from Sport Radar API

### Data Parsing

Extracted JSON response and converted it into structured pandas Data Frames

### Data Cleaning

Handled missing values, normalized formats, and ensured consistency

### Database Storage

Inserted cleaned data into MySQL tables using MySQL Connector

### Visualization

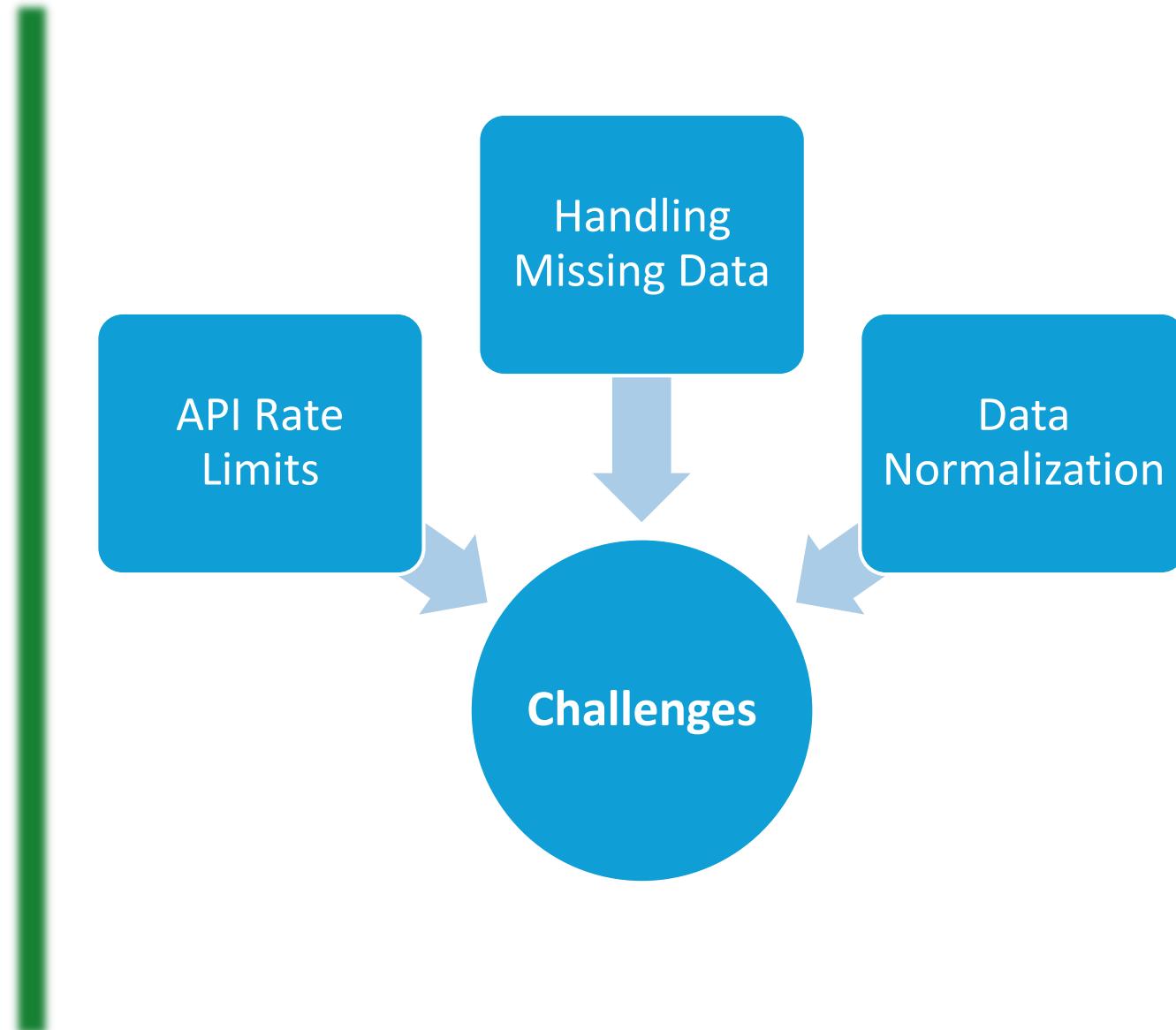
Connected data to Streamlit for interactive dashboards.

### API Rate Limits

### Handling Missing Data

### Data Normalization

### Challenges



# Python code & SQL Data

```

import requests
import json

# API Configuration
API_KEY = "vPAg0e8ZLcCjUWQdDrIlxcQ60PqH81i4DqzrjAoX"
BASE_URL = "https://api.sportradar.com/tennis/trial/v3/en"

def fetch_data(endpoint): 1usage
    """Fetch data from SportRadar API and return JSON."""
    url = BASE_URL + "/" + endpoint + ".json?api_key=" + API_KEY
    response = requests.get(url)

    if response.status_code == 200:
        return response.json()

    print("Error:", response.status_code, response.text)
    return None

# Fetch & Display Competitions
competitions = fetch_data("competitions")
if competitions:
    print(json.dumps(competitions, indent=2))
else:
    print("No data received.")

```



```

import mysql.connector
from mysql.connector import Error
try:
    # Provide your correct credentials here
    connection = mysql.connector.connect(
        host="localhost",
        user="root",
        password="Sham@1234"
    )
    if connection.is_connected():
        print("Connection to MySQL server was successful!")

        # Create a cursor object to execute SQL queries
        cursor = connection.cursor()

        # SQL query to create a new database
        create_database_query = "CREATE DATABASE IF NOT EXISTS tennis_db"
        cursor.execute(create_database_query)
        print("Database 'tennis_db' created successfully!")

except Error as e:
    print(f"Error: {e}")
finally:
    # Close the connection
    if 'connection' in locals() and connection.is_connected():
        connection.close()
        print("MySQL connection is closed.")

```

```

import mysql.connector
import requests

# ✓ API Configuration
API_KEY = "vPAg0e8ZLcCjUWQdDrIlxcQ60PqH81i4DqzrjAoX"
BASE_URL = "https://api.sportradar.com/tennis/trial/v3/en"

# ✓ Connect to MySQL
def get_db_connection(): 3 usages
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="Sham@1234",
        database="tennis_db"
    )

# ✓ Create Tables
def create_competitions_tables(): 1 usage
    connection = get_db_connection()
    cursor = connection.cursor()

    # Create Categories Table
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS categories (
            category_id VARCHAR(50) PRIMARY KEY,
            category_name VARCHAR(100) NOT NULL
    """)

```

	competition_id	competition_name	parent_id	type	gender	category_id
▶	sr:competition:10025	ITF Men Stara Zagora, Bulgaria Men Singles	sr:competition:10023	singles	men	sr:category:785
	sr:competition:10027	ITF Men Stara Zagora, Bulgaria Men Doubles	sr:competition:10023	doubles	men	sr:category:785
	sr:competition:10031	ITF Men Sibiu, Romania Men Singles	sr:competition:10029	singles	men	sr:category:785
	sr:competition:10033	ITF Men Sibiu, Romania Men Doubles	sr:competition:10029	doubles	men	sr:category:785
	sr:competition:10037	ITF Men Busto Arsizio, Italy Men Singles	sr:competition:10035	singles	men	sr:category:785
	sr:competition:10039	ITF Men Busto Arsizio, Italy Men Doubles	sr:competition:10035	doubles	men	sr:category:785
	sr:competition:10043	ITF Men Sabac, Serbia Men Singles	sr:competition:10041	singles	men	sr:category:785

# Streamlit Dashboard Features

**Filters & Navigation**

**Filter Competitors**

Search Competitor by Name

Filter by Rank: 1 to 501

Filter by Points: 107 to 10750

Filter by Country: All

**Run SQL Queries**

Enter SQL Query:

```
SELECT * FROM competitors WHERE country = 'Spain'
```

**Tennis Competitor Rankings Dashboard**

Total Competitors: 1000 Countries Represented: 80 Highest Points: 10750

**Competitor Data**

competitor_id	name	country	country_code	abbreviation	rank	movement	points	competitions_played	
0	sr:competitor:49363	Pavic, Mate	Croatia	HRV	PAV	1	0	7,620	23
1	sr:competitor:51836	Arevalo-Gonzalez, Marcelo	El Salvador	SLV	ARE	1	0	7,620	23
2	sr:competitor:637970	Patten, Henry	Great Britain	GBR	PAT	3	0	7,355	28
3	sr:competitor:14898	Heliovaara, Harri	Finland	FIN	HEL	4	0	7,205	26
4	sr:competitor:36593	Krawietz, Kevin	Germany	DEU	KRA	5	0	6,330	21
5	sr:competitor:52293	Putz, Tim	Germany	DEU	PUT	6	0	6,240	20

**Country-Wise Analysis**

	country	total_competitors	avg_points
0	USA	93	956.1935
1	Neutral	65	627.2462
2	France	54	567.537
3	Japan	54	434.037
4	Great Britain	52	849.2115
5	Australia	49	786.1633
6	Czechia	41	766.7805
7	Italy	40	879.175
8	Netherlands	33	756.0909
9	China	33	771.8788

**Competitor Details**

Select a Competitor: Granollers, Marcel

Name: Granollers, Marcel

Rank: 9

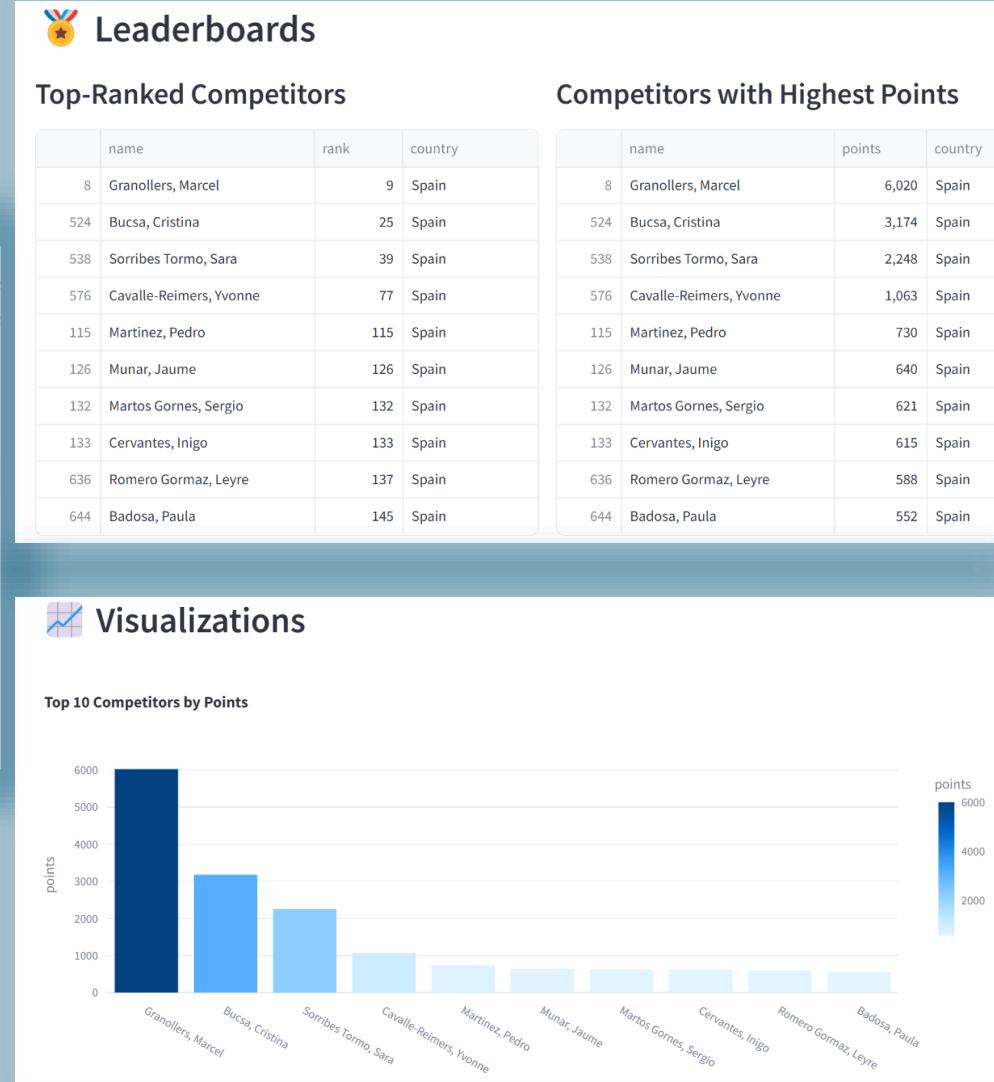
Movement: 0

Points: 6020

Competitions Played: 14

Country: Spain

id	sr:competitor	name
644	sr:competitor:133662	Badosa, Paula
667	sr:competitor:119216	Herrero Linana, Alicia



## Future Scope



**Advanced Optimization:** Ai-Driven Predictive analytics for player performance trends and match outcomes.



**UI Enhancements:** Real-time data visualization updates and interactive filtering for better user experience



**Data Integration:** Seamless AI expansion to include historical data and advanced event insights.



**Automated Ranking System Generation:** Machine learning-based ranking adjustments and competitor performance tracking.



**Standards Compliance:** Ensuring compatibility with global sports data standards and federations.



**Cloud-Based Collaboration:** Real-time database access and multi-user dashboard for enhanced analytics.





# Thank you!

**Data is the new game-changer—unlocking insights, elevating strategy, and redefining tennis analytics**