Project Title	Game Analytics: Unlocking Tennis Data with SportRadar API
Skills take away From This Project	Python scripting, Data Collection using API integration, Data Management using SQL, Streamlit
Domain	Sports/Data Analytics

Problem Statement:

The **SportRadar Event Explorer** project aims to develop a comprehensive solution for managing, visualizing, and analyzing sports competition data extracted from the Sportradar API. The application will parse JSON data, store structured information in a relational database, and provide intuitive insights into tournaments, competition hierarchies, and event details. This project is designed to assist sports enthusiasts, analysts, and organizations in understanding competition structures and trends while exploring detailed event-specific information interactively.

Business Use Cases:

1. **Event Exploration**: Enable users to navigate through competition hierarchies (e.g., ATP Vienna events).
2. **Trend Analysis**: Visualize the distribution of events by type, gender, and competition level.
3. **Performance Insights**: Analyze player participation across singles and doubles events.
4. **Decision Support**: Offer data-driven insights to event organizers or sports bodies for resource allocation.

Approach:

Data Extraction

- Parse and extract data from Sportradar JSON responses.(using API)
- Transform nested JSON structures into a flat relational schema for analysis.

Data Storage:

- Create a SQL database with well-designed schema (e.g., defining appropriate data types and primary keys).   The data to be collected is provided below.

Data Collection:

1) **COLLECT THE COMPETITION DATA FROM THE API ENDPOINTS-**

   https://developer.sportradar.com/tennis/reference/competitions

Table Structure

1. Categories Table

This table stores information about categories.

Column Name	Data Type	Constraints	Description
category_id	VARCHAR(50)	PRIMARY KEY	Unique ID for the category
category_name	VARCHAR(100)	NOT NULL	Name of the category

2. Competitions Table

This table stores information about competitions and links to the Categories table.

Column Name	Data Type	Constraints	Description
competition_id	VARCHAR(50)	PRIMARY KEY	Unique ID for the competition
competition_name	VARCHAR(100)	NOT NULL	Name of the competition
parent_id	VARCHAR(50)	NULLABLE	Parent competition ID
type	VARCHAR(20)	NOT NULL	Type of competition (e.g., doubles)
gender	VARCHAR(10)	NOT NULL	Gender of participants (e.g., men)
category_id	VARCHAR(50)	FOREIGN KEY REFERENCES Categories(category_id)	Links to the category table

Data Analysis:

Execute the Following SQL queries;

1) List all competitions along with their category name
2) Count the number of competitions in each category

3) Find all competitions of type 'doubles'
4) Get competitions that belong to a specific category (e.g., ITF Men)
5) Identify parent competitions and their sub-competitions
6) Analyze the distribution of competition types by category
7) List all competitions with no parent (top-level competitions)

2) COLLECT THE COMPLEXES DATA FROM THE API ENDPOINTS-

https://developer.sportradar.com/tennis/reference/complexes

Table Structure

1. Complexes Table

This table stores information about sports complexes.

Column Name	Data Type	Constraints	Description
complex_id	VARCHAR(50)	PRIMARY KEY	Unique ID for the complex
complex_name	VARCHAR(100)	NOT NULL	Name of the sports complex

2. Venues Table

This table stores information about venues linked to a complex.

Column Name	Data Type	Constraints	Description
venue_id	VARCHAR(50)	PRIMARY KEY	Unique ID for the venue

venue_name	VARCHAR(100)	NOT NULL		Name of the venue
city_name	VARCHAR(100)	NOT NULL		Name of the city
country_name	VARCHAR(100)	NOT NULL		Name of the country
country_code	CHAR(3)	NOT NULL		ISO country code
timezone	VARCHAR(100)	NOT NULL		Timezone of the venue
complex_id	VARCHAR(50)	FOREIGN KEY REFERENCES Complexes(complex_id)		Links to the complexes table

Execute the following SQL queries:

1) List all venues along with their associated complex name
2) Count the number of venues in each complex
3) Get details of venues in a specific country (e.g., Chile)
4) Identify all venues and their timezones
5) Find complexes that have more than one venue
6) List venues grouped by country
7) Find all venues for a specific complex (e.g., Nacional)

3) COLLECT THE DOUBLES COMPETITOR RANKINGS DATA FROM THE API ENDPOINTS-

https://developer.sportradar.com/tennis/reference/doubles-competitor-rankings

1. Competitor_Rankings Table

This table will store ranking-related information about competitors.

Column Name	Data Type	Constraints	Description
rank_id	INT	PRIMARY KEY AUTO_INCREMENT	Unique ID for each ranking record
rank	INT	NOT NULL	Rank of the competitor
movement	INT	NOT NULL	Rank movement compared to the previous week
points	INT	NOT NULL	Total ranking points
competitions_played	INT	NOT NULL	Number of competitions played

competitor_id	VARCHAR(50)	FOREIGN KEY REFERENCES Competitors(competitor_id)	Links to competitor details

2. Competitors Table

This table will store detailed information about each competitor.

Column Name	Data Type	Constraints	Description
competitor_id	VARCHAR(50)	PRIMARY KEY	Unique ID for each competitor
name	VARCHAR(100)	NOT NULL	Name of the competitor
country	VARCHAR(100)	NOT NULL	Competitor's country
country_code	CHAR(3)	NOT NULL	ISO country code
abbreviation	VARCHAR(10)	NOT NULL	Shortened name/abbreviation of competitor

Execute the following SQL queries:

1) Get all competitors with their rank and points.
2) Find competitors ranked in the top 5

3) List competitors with no rank movement (stable rank)
4) Get the total points of competitors from a specific country (e.g., Croatia)
5) Count the number of competitors per country
6) Find competitors with the highest points in the current week

Build a Streamlit Application:

● Connect the Streamlit app with the SQL database for real-time query execution.
● Display analysis results in the form of tables, charts, or dashboards within the app.**User Interface**: Interactive dashboards with filters for competition types, levels, and categories.

Suggested Features for the Application (Use your creativity)

1. **Homepage Dashboard:**
   ○ Summary statistics like:
      ■ Total number of competitors.
      ■ Number of countries represented.
      ■ Highest points scored by a competitor.
2. **Search and Filter Competitors:**
   ○ Allow users to search for a competitor by name.
   ○ Filter competitors by rank range, country, or points threshold.
3. **Competitor Details Viewer:**
   ○ Display detailed information about a selected competitor, including:
      ■ Rank, movement, competitions played, and country.
4. **Country-Wise Analysis:**
   ○ List countries with the total number of competitors and their average points.
5. **Leaderboards:**
   ○ Show tables for:
      ■ Top-ranked competitors.
      ■ Competitors with the highest points.

Sample User Interface: (Streamlit)
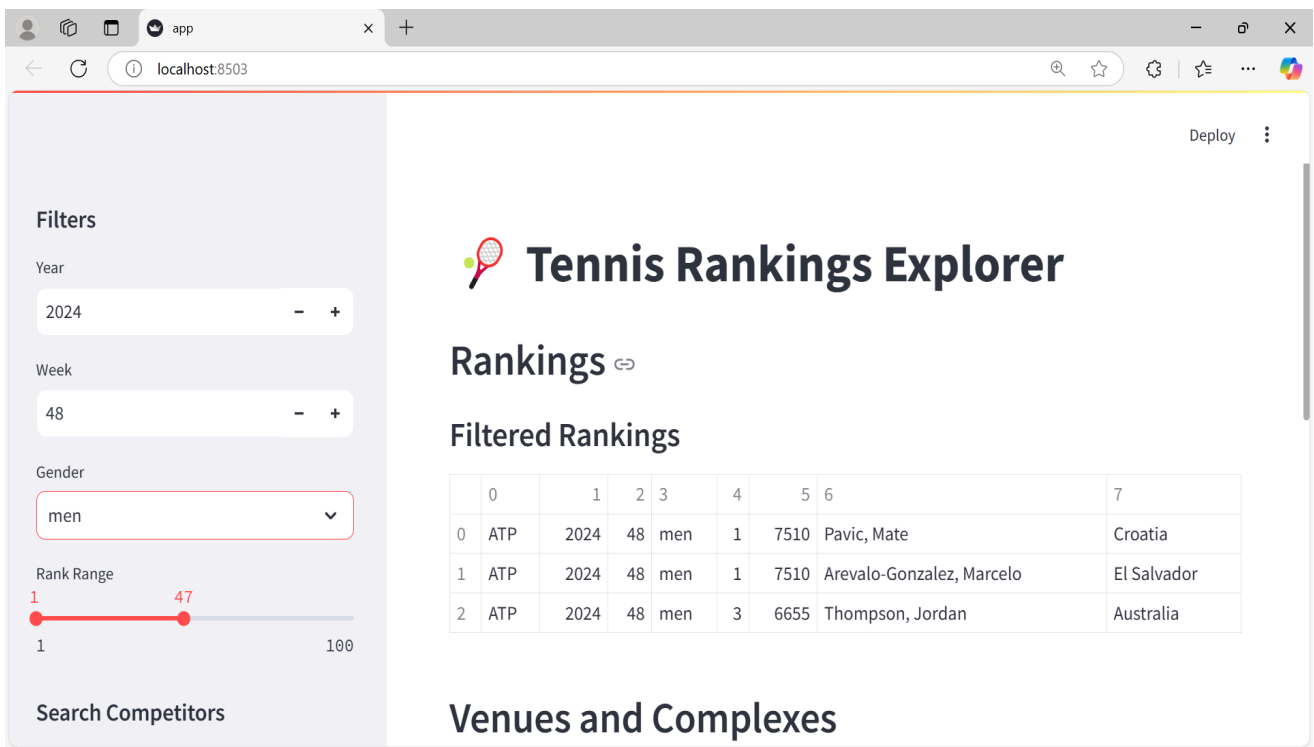
Results:

The expected outcome is a fully functional web application capable of parsing and visualizing sports event data. Users can navigate competition hierarchies, filter data, and generate meaningful insights about sports events. This project will enhance proficiency in working with APIs, databases, and visualization tools.

Skills Takeaway:

1. **Data Extraction:** API calls or web scraping.
2. **SQL Database Management:** Designing tables and schema.
3. **Data Analysis:** Writing SQL queries to derive insights.

Project Evaluation metrics:

1. **Data Extraction Accuracy:** Successful retrieval of the desired data from APIs or sources without errors or missing values.

2. **SQL Database Design:**Proper structuring of tables with normalized data and correct relationships between them.
3. **Query Efficiency:**Ability to write optimized SQL queries that retrieve relevant insights quickly.
4. **Streamlit Application Functionality:**Smooth navigation and interactive features for data display and analysis.
5. **Project Completeness:**End-to-end workflow execution from data extraction to SQL storage and application using streamlit.
6. **Documentation Quality:**Clearly upload all files in github.
7. **Presentation and Usability:**An intuitive user interface with meaningful outputs and actionable insights.
8. **Error Handling:**Ability to manage API rate limits, database constraints, or user input errors effectively.
9. **Innovation and Creativity:**Unique implementation ideas, or insightful SQL queries.

Technical Tags:

1. **Languages**: Python
2. **Database**: MySQL/PostgreSQL
3. **Application**: Streamlit
4. **API Integration**: Sportradar API

Project Deliverables:

1. **SQL Database**: Populated with structured sports event data from the Sportradar API.
2. **API Scripts**: Automate data extraction and transform JSON into a relational format.

3. **Streamlit App**: Interactive tool for exploring and visualizing competition data.

4. **Documentation**: Detailed report on workflow, schema design, challenges, and insights.

** *In github, upload all your code files. Also upload a document that contains all the SQL queries.*

Project Guidelines:

1. **Coding Standards**
   ○ Use meaningful names: Variables, functions, and database tables should have descriptive names.

- ○ Follow PEP 8 (for Python): Maintain consistent formatting with proper indentation and spacing.
- ○ Modularize your code: Break your code into functions or classes to enhance readability and reusability.
- ○ Error handling: Implement try-except blocks for handling API errors and SQL exceptions.
- ○ Document your code: Include docstrings and comments to explain logic and functions.

2. **SQL Database Practices**
   - ○ Normalize tables: Avoid redundancy and ensure efficient data storage.
   - ○ Use indexes: Optimize query performance with appropriate indexing.
   - ○ Follow naming conventions: Use consistent and descriptive names for tables and fields.

3. **Streamlit Application Development**
   - ○ Interactive features: Ensure the UI is responsive, with interactive widgets for filters.
   - ○ Minimalist design: Keep the layout simple for a smooth user experience.
   - ○ Performance optimization: Avoid loading all data at once—use pagination or batch processing where possible.

4. **General Best Practices**
   - ○ Test frequently: Regularly test each component (e.g., API requests, SQL queries, Streamlit app) during development.
   - ○ Backup your data: Maintain backups of your SQL database and code.
   - ○ Documentation: Provide a README file with setup instructions, project objectives, and a demo walkthrough.

Reference:

Streamlit Doc	https://docs.streamlit.io/library/api-reference
SportRadar API key	https://console.sportradar.com/signup
Streamlit recording (Tamil)	**Special Session for STREAMLIT Tamil**
Streamlit recording (English)	🗒 Special session for STREAMLIT(11/08/2…

Project Live Evaluation Metrics	📄 Project Live Evaluation
Capstone Explanation Guideline	📄 Capstone Explanation Guideline
GitHub Reference	🅿 How to Use GitHub.pptx
Project Orientation (English)	
Project Orientation (Tamil)	📄 Project Orientation Session :Unlocking …

Timeline:

The project must be completed and submitted within **7 days from the assigned date.**

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday to Saturday (4:00PM to 5:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.
Note: This form will Open on Saturday and Sunday Only on Every Week

Timing: Monday-Saturday (5:30 PM to 6:30 PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA