Game Analytics: Unlocking Tennis Data with SportRadar API

This presentation explores a data science project focused on processing and visualizing tennis data using the SportRadar API. Discover how I leveraged Python, SQL, and Streamlit to extract, analyze, and present valuable insights.

by Gnanasekar S

Project Overview

The project involved fetching real-time tennis data from the SportRadar API, a comprehensive sports data platform.

I cleaned and processed the raw data using the powerful pandas library, preparing it for analysis and storage.

The structured data was then stored in a MySQL database, ensuring data integrity and efficiency.

I ran SQL queries to extract insights from the database, leveraging JOINs and filtering conditions for data analysis.

Tech Stack

1. Python was my primary tool, utilizing libraries such as pandas for data manipulation, mysql.connector for database interaction, and requests for API communication.

2. MySQL served as the data storage platform, providing a structured and reliable environment for my tennis data.

3. Streamlit enabled us to build a dynamic and interactive web app, offering a user-friendly interface for data visualization.

4. I integrated the SportRadar API to access real-time tennis data, fueling my analysis and visualization efforts.

Database Schema

I designed a well-structured database schema with six tables for storing tennis data.

The tables were carefully linked using primary and foreign keys, ensuring data integrity and relationship maintenance.

This approach maintained data consistency and facilitated efficient querying and analysis.

Data Processing Workflow

I fetched tennis data directly from the SportRadar API using Python's requests library.

The raw data was cleaned and preprocessed using pandas to handle inconsistencies and prepare it for storage.

Finally, I stored the structured data in the MySQL tables, ensuring data integrity and organization.

SQL Queries & Insights

1. I executed SQL queries to extract meaningful insights from the stored tennis data.

2. Leveraged JOINs and filtering conditions to analyze player performance, match statistics, and tournament trends.

3. Optimized query performance to retrieve data efficiently and provide rapid insights.

Streamlit Dashboard Features

1. I built an interactive Streamlit dashboard to visualize the processed tennis data.

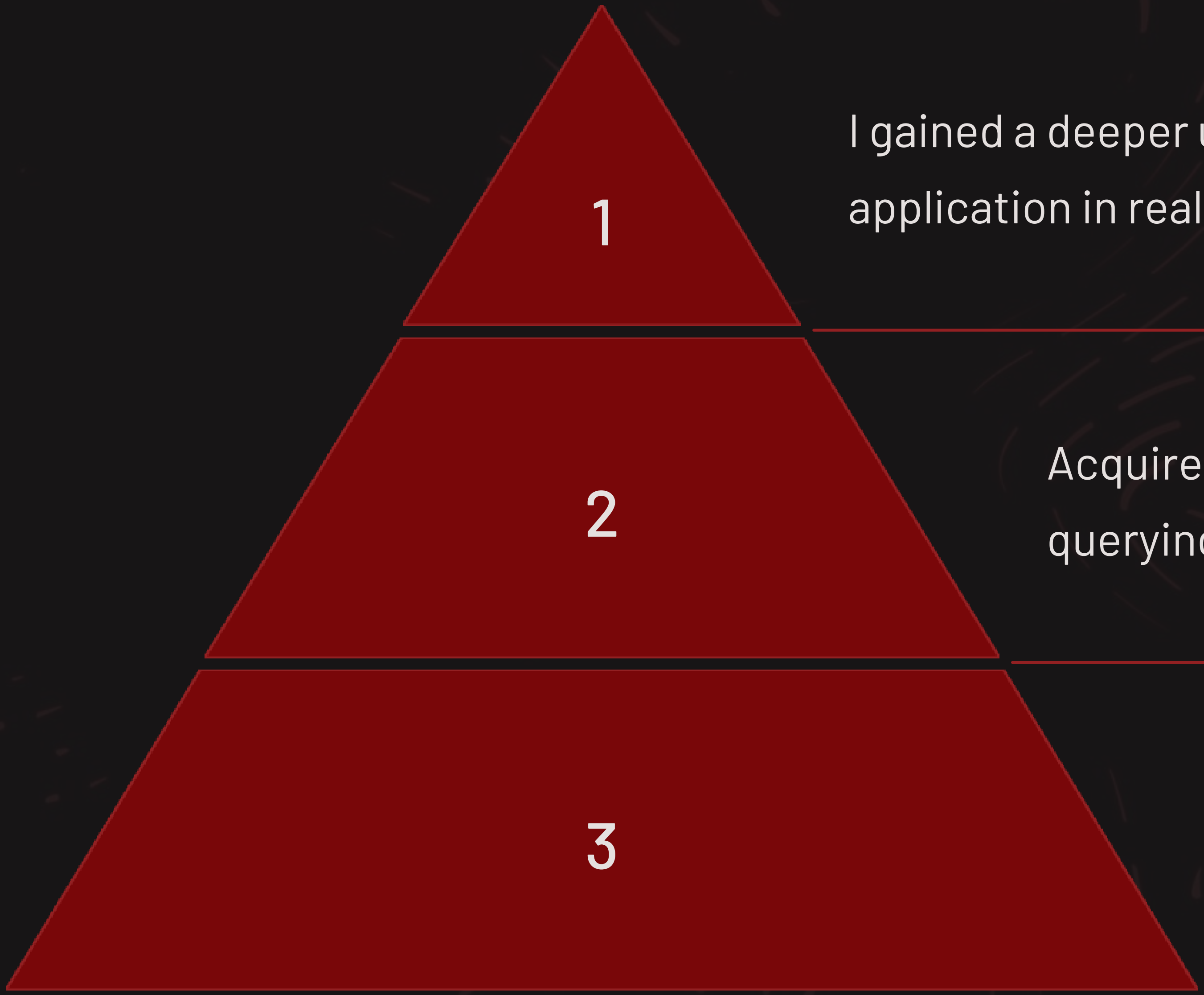
2. The dashboard utilized Plotly and pandas to generate visually appealing charts and tables.

3. Users could easily filter and interact with the data, exploring insights in a user-friendly manner.

Challenges & Solutions

- I encountered challenges related to data inconsistencies, but resolved them using pandas data cleaning techniques.
- Ensured database normalization for optimal efficiency and performance.
- Optimized API calls to avoid exceeding rate limits, ensuring smooth data fetching.

Conclusion & Learnings



1. I gained a deeper understanding of data processing techniques and their application in real-world scenarios.

2. Acquired practical experience with API integration and SQL querying, enhancing my data analysis skills.

3. Developed a fully functional data analytics project which gave a very good understanding on how real time data analytics project are.