Tennis Rankings Dashboard

Introduction

This document provides a professional and detailed explanation of a **Streamlit** dashboard application for visualizing tennis player data stored in a **MySQL** database. The app supports interactive visualizations, metric cards, search filters, and leaderboard analytics.

Technologies Used

- **Python**
- **Streamlit** (for UI)
- **MySQL** (for database)
- **Pandas** (for data manipulation)
- **Plotly** (for data visualization)
- **MySQL Connector** (for Python-to-MySQL communication)

---

1. Setup and Configuration

Streamlit Configuration

```
st.set_page_config(layout="wide")
st.title("Tennis Rankings Dashboard")
```

This sets the layout of the Streamlit dashboard to full width and defines the main page title.

Database Connection

```
def get_connection():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="root",
        database="sportsradar"
    )
```

Establishes a connection to the local MySQL database named `sportsradar` using root credentials.

SQL Query Wrapper

```
def query_df(sql):
    conn = get_connection()
    df = pd.read_sql(sql, conn)
    conn.close()
    return df
```

Executes an SQL query and returns the result as a Pandas DataFrame.

---

2. Overview Section

Displays key performance indicators (KPIs):

```
col1, col2, col3, col4 = st.columns(4)
```

Splits the layout into 4 columns for parallel display.

Metrics Queries:

```
SELECT COUNT(*) AS total FROM Competitors;
SELECT COUNT(DISTINCT competition_name) FROM competitions;
SELECT COUNT(DISTINCT country) FROM Competitors;
SELECT MAX(points) FROM Rankings;
```

These queries calculate total competitors, competitions, unique countries, and max points.

---

3. Filter/Search Panel

This section is embedded in the sidebar:

```
name_filter = st.sidebar.text_input("Search by name")

country_filter = st.sidebar.selectbox("Select Country", ["All"] + ... )

rank_range = st.sidebar.slider("Rank Range", 1, 100, (1, 10))
min_points = st.sidebar.slider("Minimum Points", 0, 10000, 1000)
```

Filter Query Example:

```
SELECT c.name, c.country, r.rank, r.points, r.movement, r.competitions_played
FROM Competitors c
JOIN Rankings r ON c.competitor_id = r.competitor_id
WHERE r.rank BETWEEN X AND Y
  AND r.points >= Z
  AND c.name LIKE '%value%'
  AND c.country = 'selected_country';
```

4. Venues by Country

User enters a country name to fetch venue info:

```
venue_df = query_df(f"""
   SELECT venue_name, timezone, country_name FROM venues
   WHERE country_name = '{user_input}'
""")
```

Displays venue data or a warning if no results are found.

---

5. Country Analysis

Shows average performance by country:

```
SELECT c.country, COUNT(*) AS competitors, AVG(r.points) AS avg_points
FROM Competitors c
JOIN Rankings r ON c.competitor_id = r.competitor_id
GROUP BY c.country
ORDER BY competitors DESC;
```

Two bar charts are shown using Plotly:

- Competitors per Country
- Average Points per Country

---

6. Leaderboards

Leaderboard with three tabs:

→ Top Rank
ORDER BY r.rank ASC LIMIT 10;

→Top Points
ORDER BY r.points DESC LIMIT 10;

→ Top Movement
ORDER BY r.movement DESC LIMIT 10;

Each query joins `Competitors` and `Rankings` to display player info.

---

7. Dynamic Sorting

Allows users to dynamically choose how to sort the leaderboard:

```
sort_by = st.selectbox("Sort Leaderboards by", ["Rank", "Points", "Movement"])
```

SQL adapts based on user choice:

```
ORDER BY r.{sort_by} DESC LIMIT 10;
```

Conclusion

This Streamlit app is a complete interactive dashboard for analyzing tennis data from a MySQL backend. It leverages filters, charts, SQL queries, and user interactivity to provide valuable insights into tennis rankings and competition data.

Suggested Enhancements

- Add login authentication for security.
- Enable chart downloads.
- Integrate historical trend analysis with time-series charts.
- Add pagination for large data tables.