OLYMPICS

UNVEILING OLYMPIC INSIGHTS: DATA-DRIVEN GOLD MEDALS

Data Olympians

I320D: Data Engineering

Professor Young

EVE LE, SHAMEZ ARAB, PRANAV REDDY, JIHAN SONG

INTRODUCTION

We were given a set of CSV files for all Olympic events since 1896.

The Olympics CTO has been asked to provide some statistics about:

1 Countries

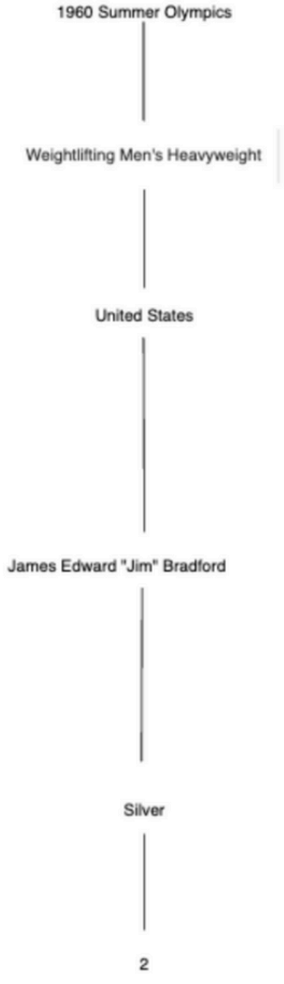
2 Teams

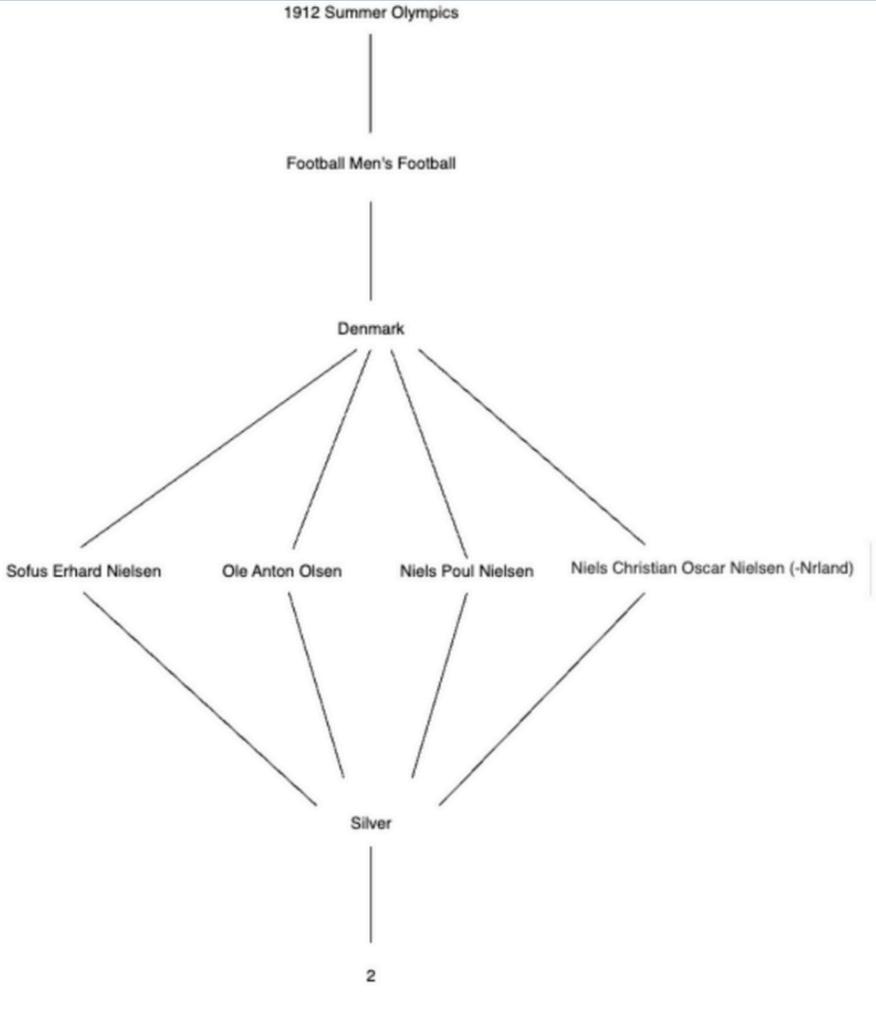
3 Athletes

to provide to the board of the International Olympic Comittee.

DATA INGESTION

Collecting, Importing, and Processing Our Raw Data

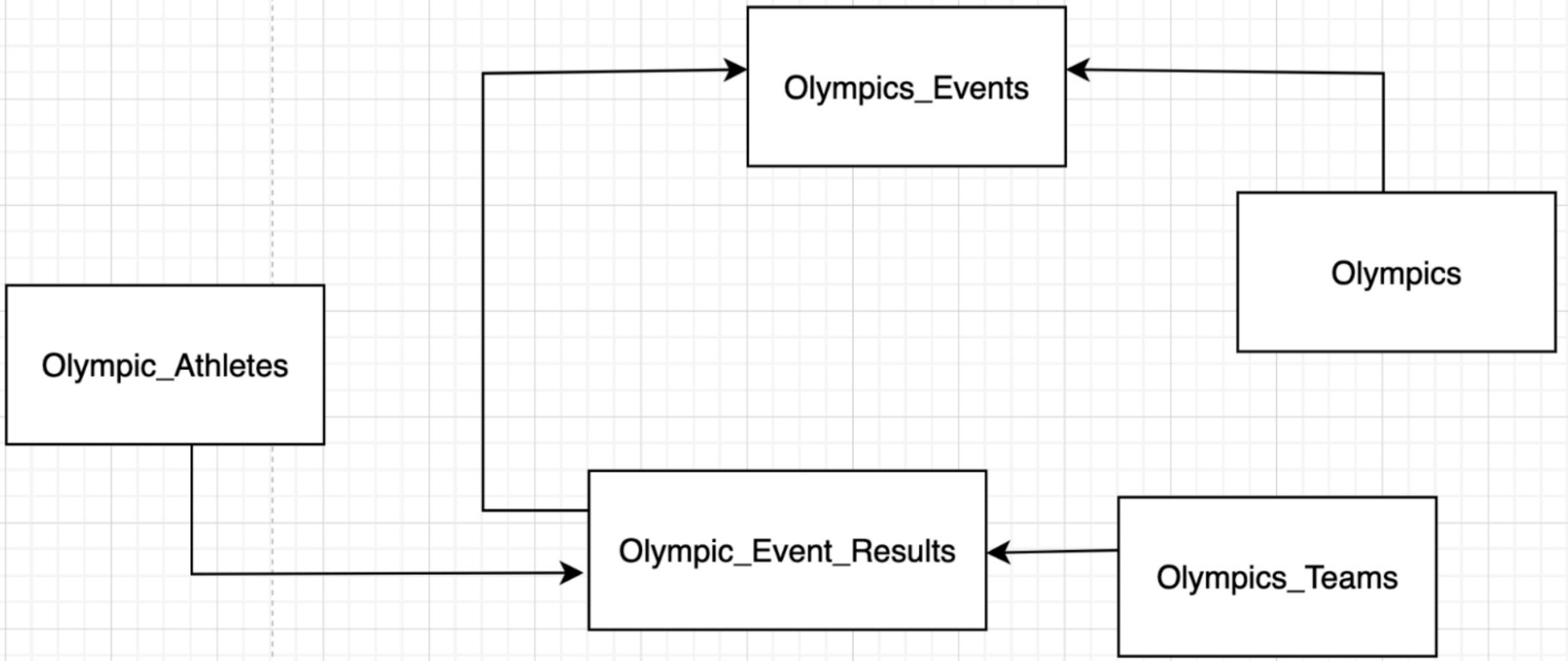




Concrete Data

DATA INGESTION

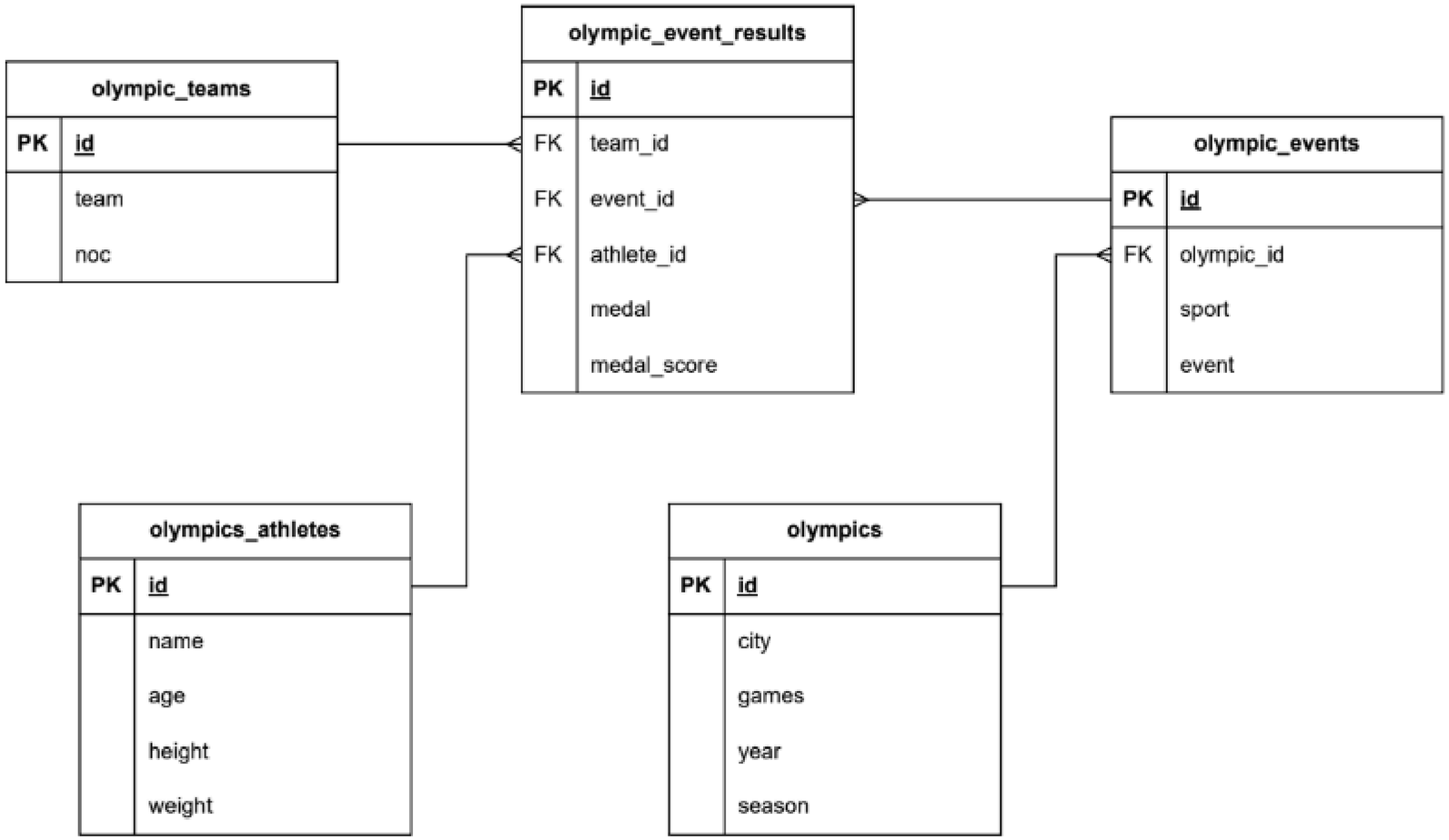
Collecting, Importing, and Processing Our Raw Data



Conceptual ER Diagram

DATA INGESTION

Collecting, Importing, and Processing Our Raw Data



Physical ER Diagram

STAGING TABLES

The purpose of creating staging tables is to create the basic structure with tables so that we can load the actual data from CSV files.

Then we bash to run the shell file named "create_staging_tables.sh" in the terminal.

```
cd projects/olympics/src/scripts/

bash create_staging_tables.sh
```

```
#!/usr/bin/env bash

. ${HOME}/venv/bin/activate
rm -rf ./.ipynb_checkpoints
psql -U ${PROJECT}_read_write -h localhost -d ${PROJECT} -f ./staging_tables.sql
```

```sql
# First, have all the DROP TABLE IF EXISTS commands
DROP TABLE IF EXISTS staging.olympic_athletes;

DROP TABLE IF EXISTS staging.olympic_event_results;

DROP TABLE IF EXISTS staging.olympic_events;

DROP TABLE IF EXISTS staging.olympic_teams;

DROP TABLE IF EXISTS staging.olympics;

# Then have all of the CREATE TABLE IF NOT EXISTS commands

CREATE TABLE IF NOT EXISTS staging.olympic_athletes(
    id SERIAL,
    name TEXT,
    age INTEGER,
    height INTEGER,
    weight NUMERIC
);

CREATE TABLE IF NOT EXISTS staging.olympic_event_results(
    id SERIAL,
    team_id INTEGER,
    event_id INTEGER,
    athlete_id INTEGER,
    medal TEXT,
    medal_score INTEGER
);
```

LOAD SCRIPT WITH \COPY COMMANDS

First, Truncate table is to remove all the possible data that might exist in newly created table, without modifying its structure.

Then, \copy command allows us to copy the whole data from csv to our staging tables. This bulk load is much faster than using line by line languages such as python.

Then we bash to run the shell file named "load_staging_data.sh" in the terminal to terminate the shell file.

```
TRUNCATE TABLE staging.olympic_athletes;
TRUNCATE TABLE staging.olympic_event_results;
TRUNCATE TABLE staging.olympic_events;
TRUNCATE TABLE staging.olympic_teams;
TRUNCATE TABLE staging.olympics;

\copy staging.olympic_athletes FROM '../../data/olympic_athletes.csv' WITH HEADER CSV;
\copy staging.olympic_event_results FROM '../../data/olympic_event_results.csv' WITH HEADER CSV;
\copy staging.olympic_events FROM '../../data/olympic_events.csv' WITH HEADER CSV;
\copy staging.olympic_teams FROM '../../data/olympic_teams.csv' WITH HEADER CSV;
\copy staging.olympics FROM '../../data/olympics.csv' WITH HEADER CSV;
```

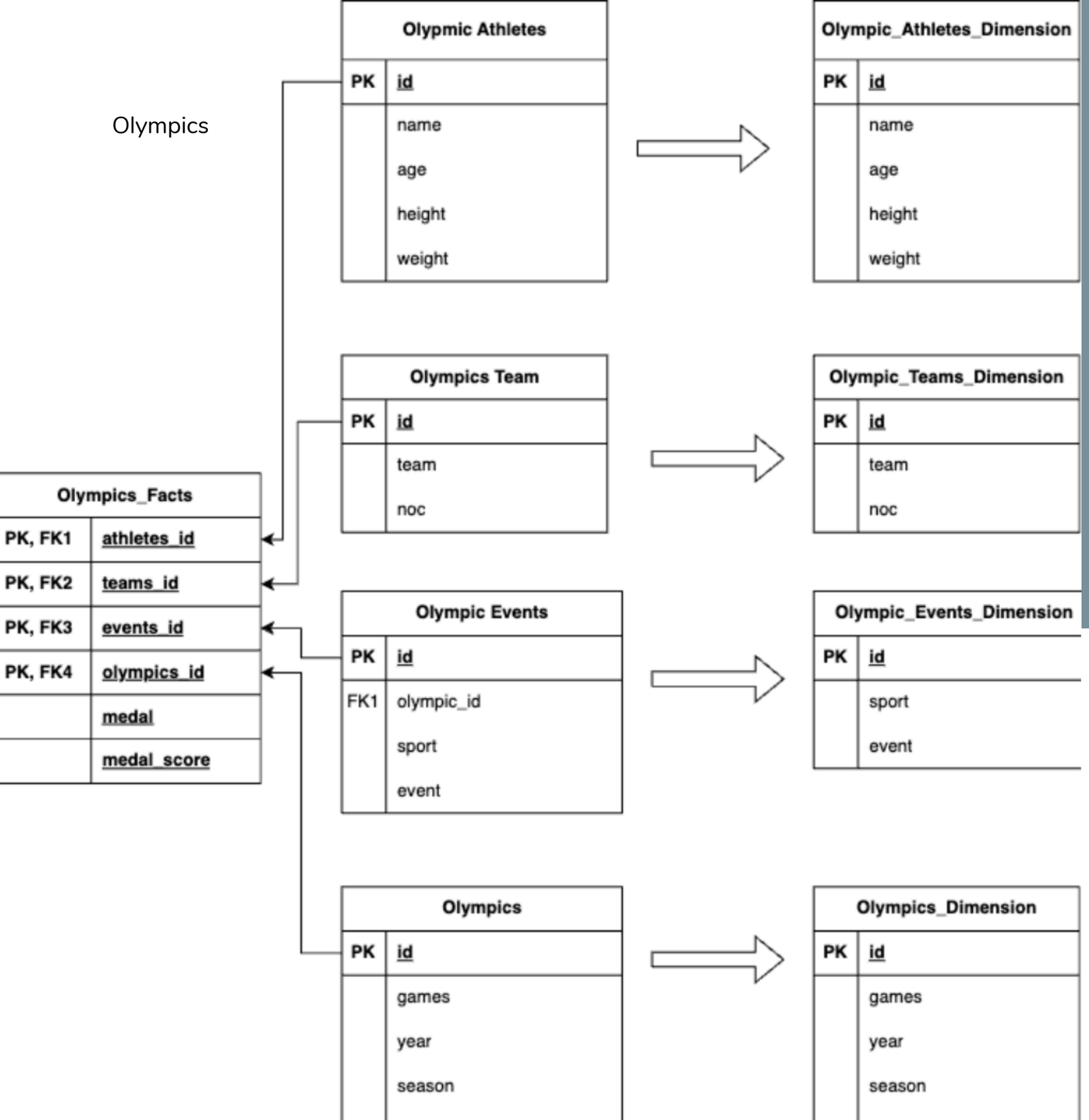
```
cd olympics/src/scripts
bash load_staging_data.sh

#!/usr/bin/env bash

. ${HOME}/venv/bin/activate
psql -U ${PROJECT}_read_write -h localhost -d
${PROJECT} -f ./copy_staging_data.sql
```

DATA TRANSFORMATION

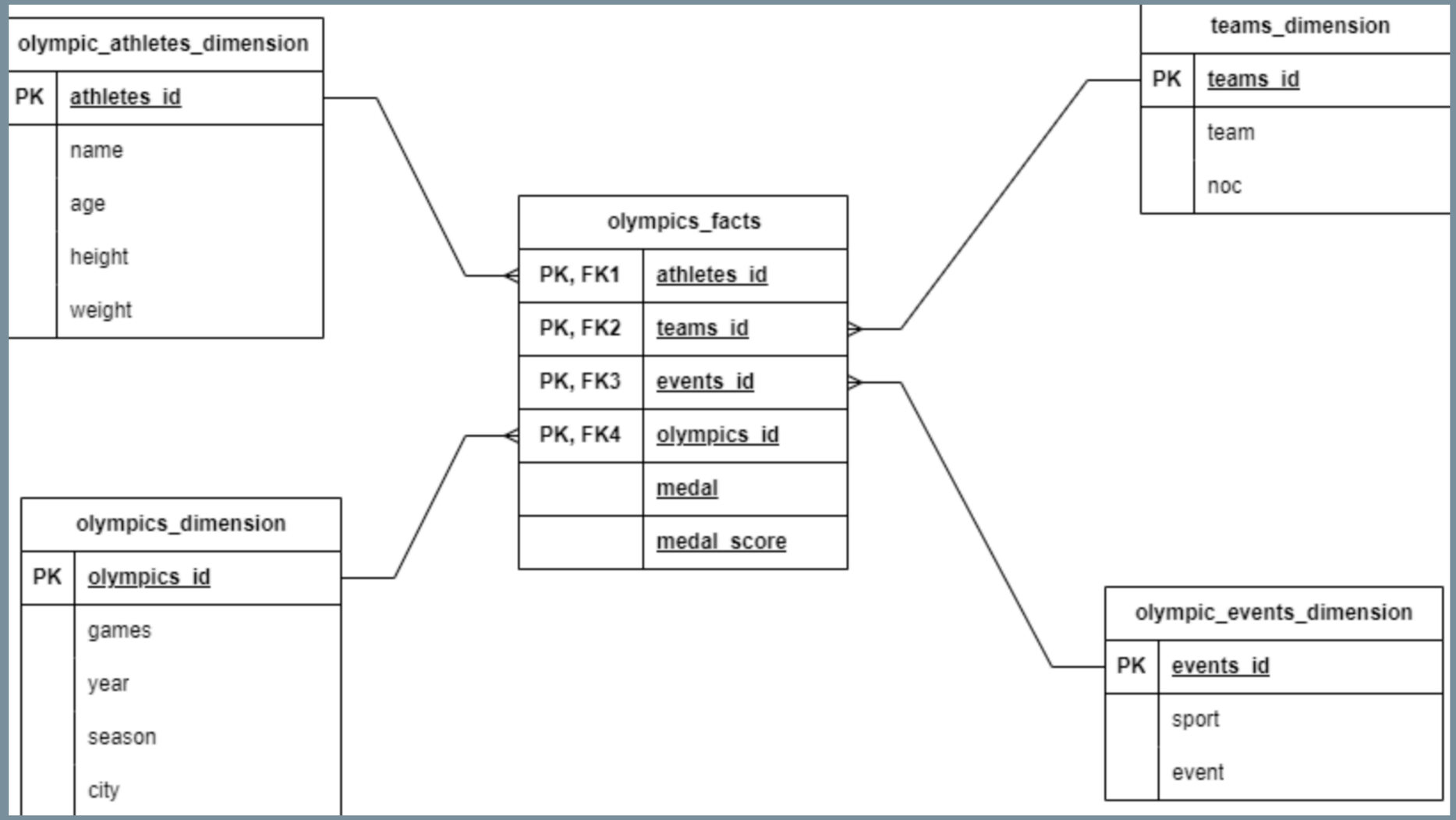
Olympics



athletes_dimension

```
{{ config(
    materialized="table"
) }}
SELECT id AS athlete_id,
       name,
       age,
       height,
       weight
FROM   staging.olympic_athletes
```

PHYSICAL DIAGRAM OF STAR SCHEMA



DATA PRODUCTS

- The query is selecting columns **(noc, medal_score, year, athlete_id, name, team)** from a set of joined tables **(olympics_facts, teams_dimension, olympics_dimension, athletes_dimension)** related to Olympic data

- The join conditions are based on matching keys **(team_id, olympic_id, athlete_id)**.

- The result of this query is likely used to create a data product providing information about Olympic events, teams, and athletes.

dbt model files to create and load the data into your data product(s)

```
{{ config
(
    materialized="table"
)
}}
SELECT noc,
       medal_score,
       year,
       athletes_dimension.athlete_id,
       name,
       team
FROM olympics_facts
JOIN teams_dimension ON (olympics_facts.team_id = teams_dimension.team_id)
JOIN olympics_dimension ON (olympics_facts.olympic_id = olympics_dimension.olympic_id)
JOIN athletes_dimension ON (olympics_facts.athlete_id = athletes_dimension.athlete_id)
```

HOW WE DECIDED OUR DATA PROCUCT

Originally, we planned on making individual data products for each specific question we had to answer, however, after cafeully analyzing the question we found that most question required similar features to answer them

As we can see on the table to the right, **medal_score** can be used to answer all the questions . **Year, noc, and team** are used less frequently but multiple times. **name** is the only attribute that is unique to a question. Therefore, we decided it would be more effecient to incorporate all the attributes into one single data product instead of making multiple.

Requirements

The Olympic CTO and her staff would like to you to analyze Olympic event results and answer the following questions:

1. What country scored the most points:
   - Overall, i.e., in all olympic games (top 3)
   - in each of the last 5 Olympic Games
2. What country has take the most gold medals:
   - Overall (top 3)
   - In each of the last 5 Olympic Games
3. What teams have won the most medals (all medals):
   - Overall (top 3)
   - In each the last 5 Olympic games
4. What athletes have won the most medals? (top 10)

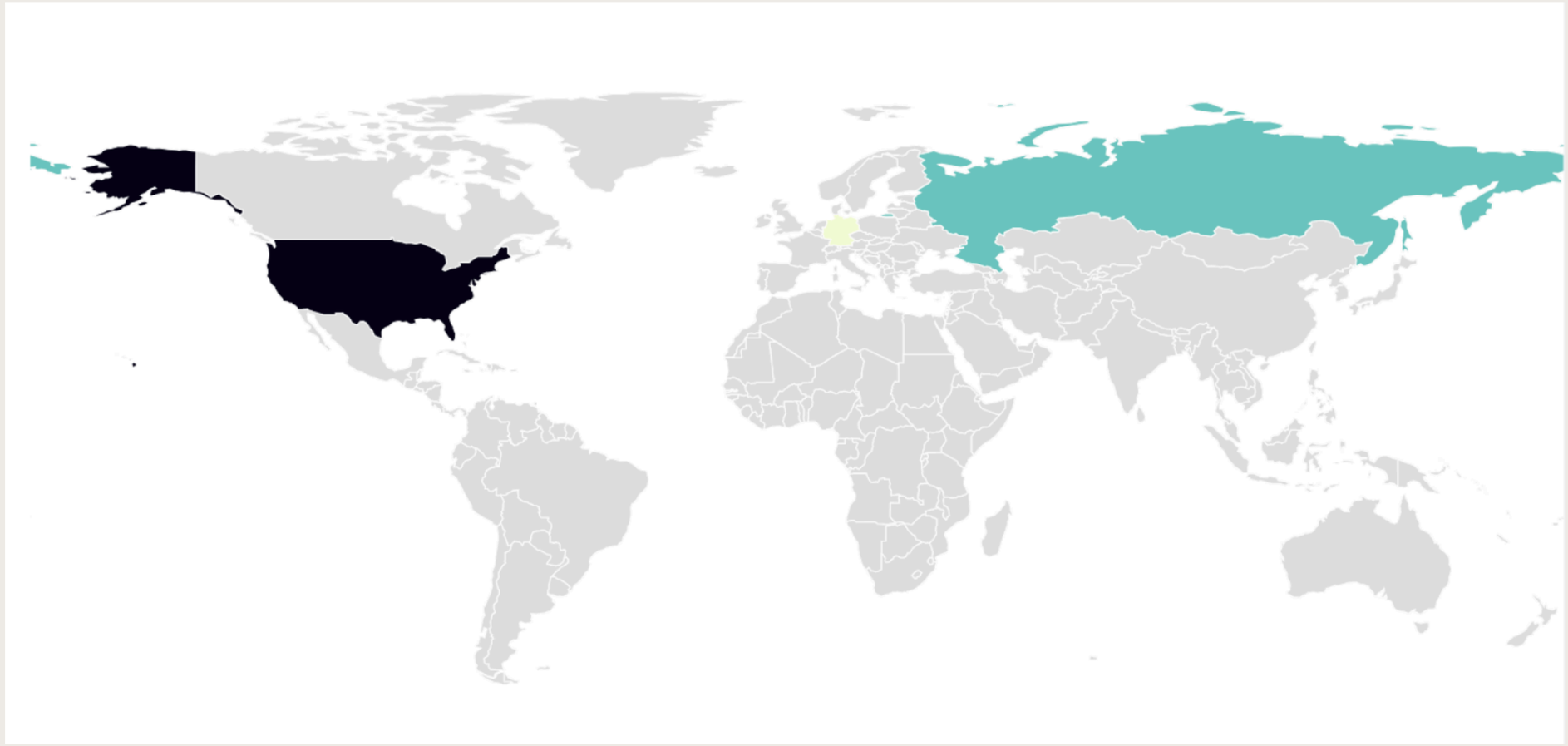
Question	Required Attributes
What country scored the most points overall, i.e., in all olympic games (top 3)	medal_score, noc
What country scored the most points in each of the last 5 Olympic Games	noc, year, medal_score
What country has taken the most gold medals: overall (top 3)	medal_score, noc
What country has take the most gold medals: In each of the last 5 Olympic Games	noc, year, medal_score
What teams have won the most medals (all medals): Overall (top 3)	medal_score, team
What teams have won the most medals (all medals): In the last 5 years	team, year, medal_score
What athletes have won the most medals? (top 10)	name, medal_score

VISUALIZATIONS

We relied on 6 columns/aliases within our data product to help us answer our questions.

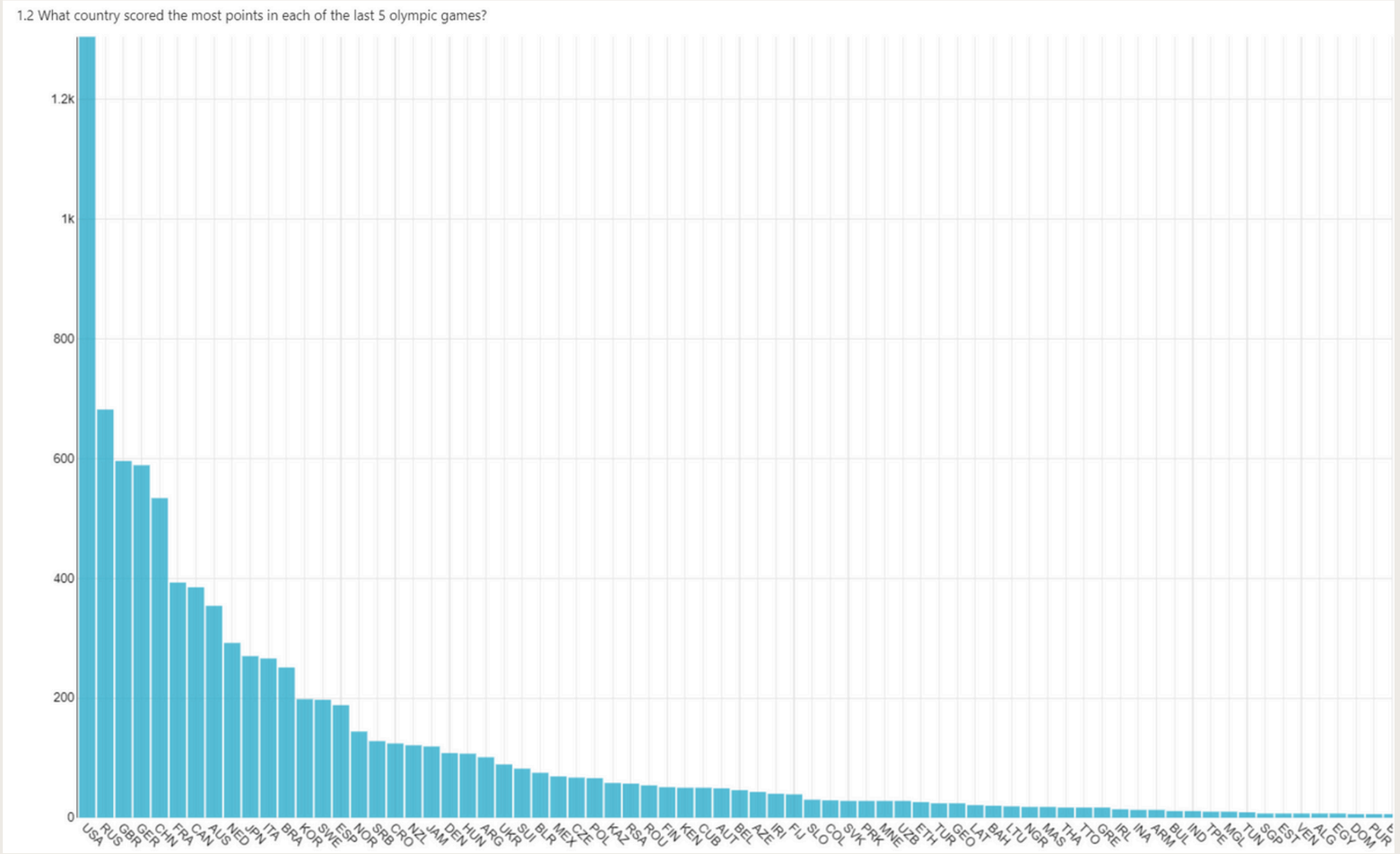
year to limit scores/medals to the last five games	**athlete_id** to link back to the column name	**name** to have athletes' actual name
noc name of country that allows us to group by	**team** to answer questions about specific competition teams	**medal_score** to perform aggregate functions on to calculate scores and rankings

VISUALIZATIONS



Q: What country scored the most points overall? (i.e. all olympic games, top 3)
A: USA (12,554 points), Soviet Union (5,399 points), and Germany (4,329 points)

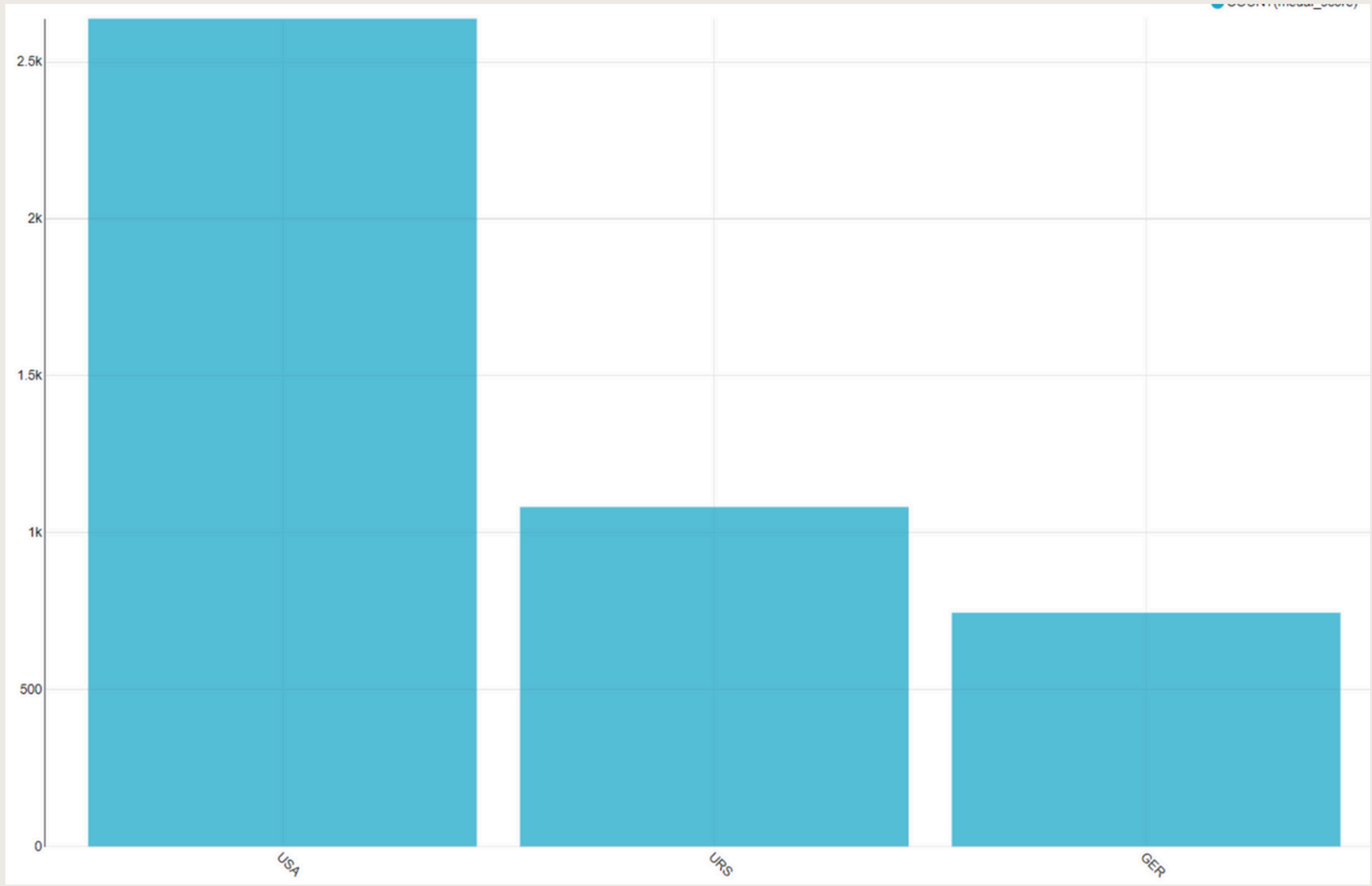
VISUALIZATIONS



Q: What country scored the most points in the last 5 olympic games?

A: USA (1,310 points)

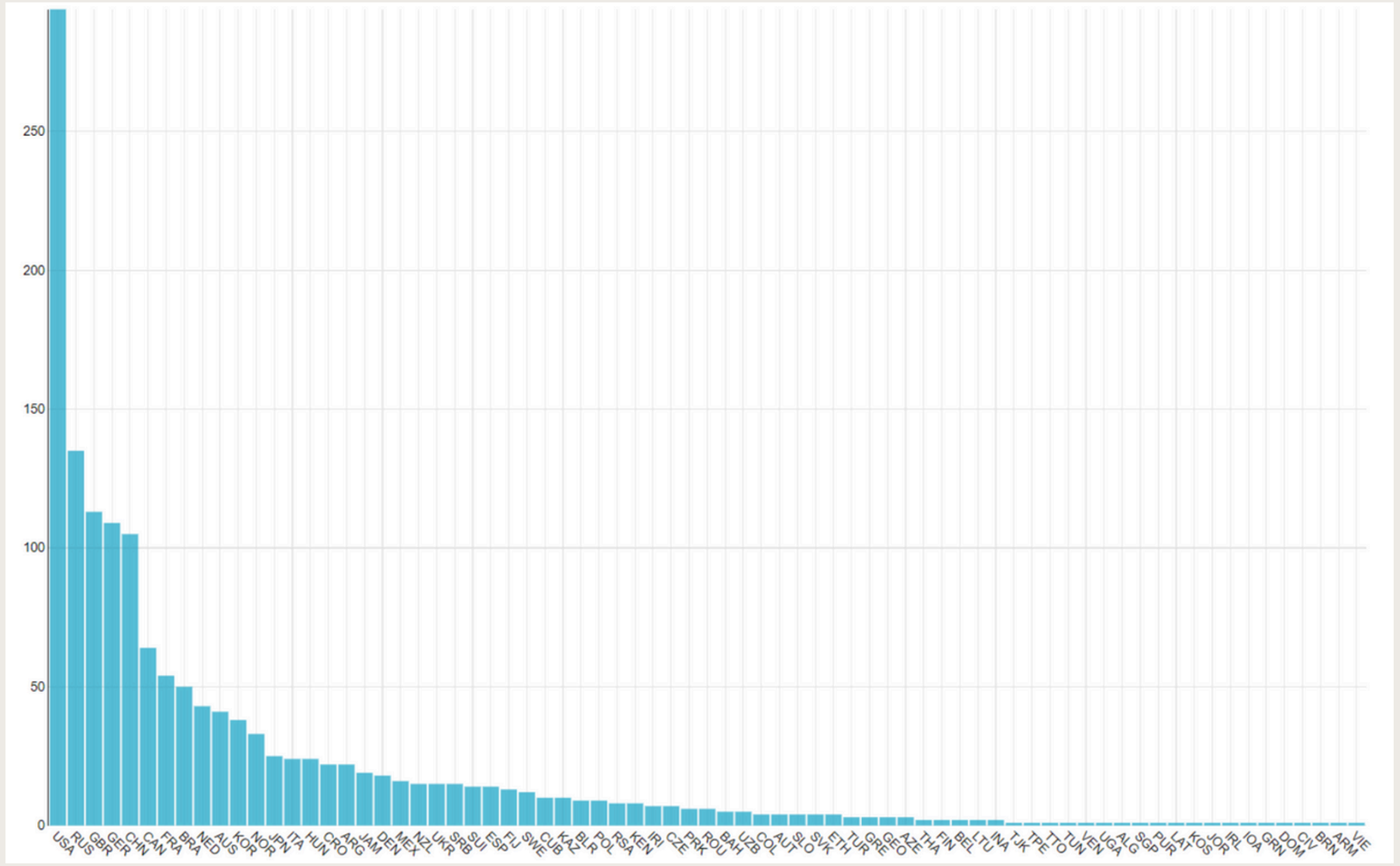
VISUALIZATIONS



Q: What country scored the most gold medals overall? (Top 3)

A: USA (2.64k), Soviet Union (1.08k), and Germany (745)

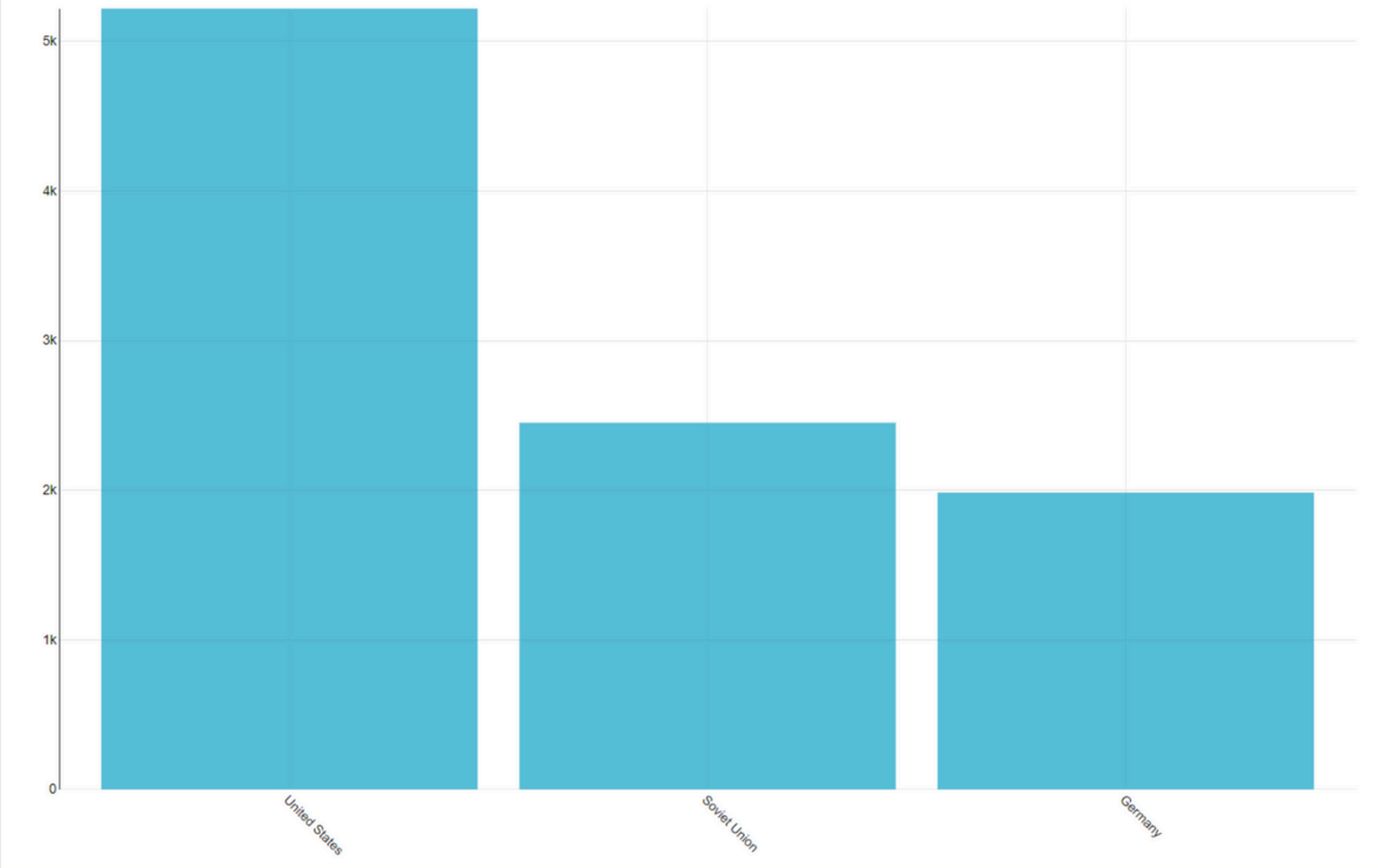
VISUALIZATIONS



Q: What country scored the most gold medals in each of the last 5 Olympics games?

A: USA (294 gold medals)

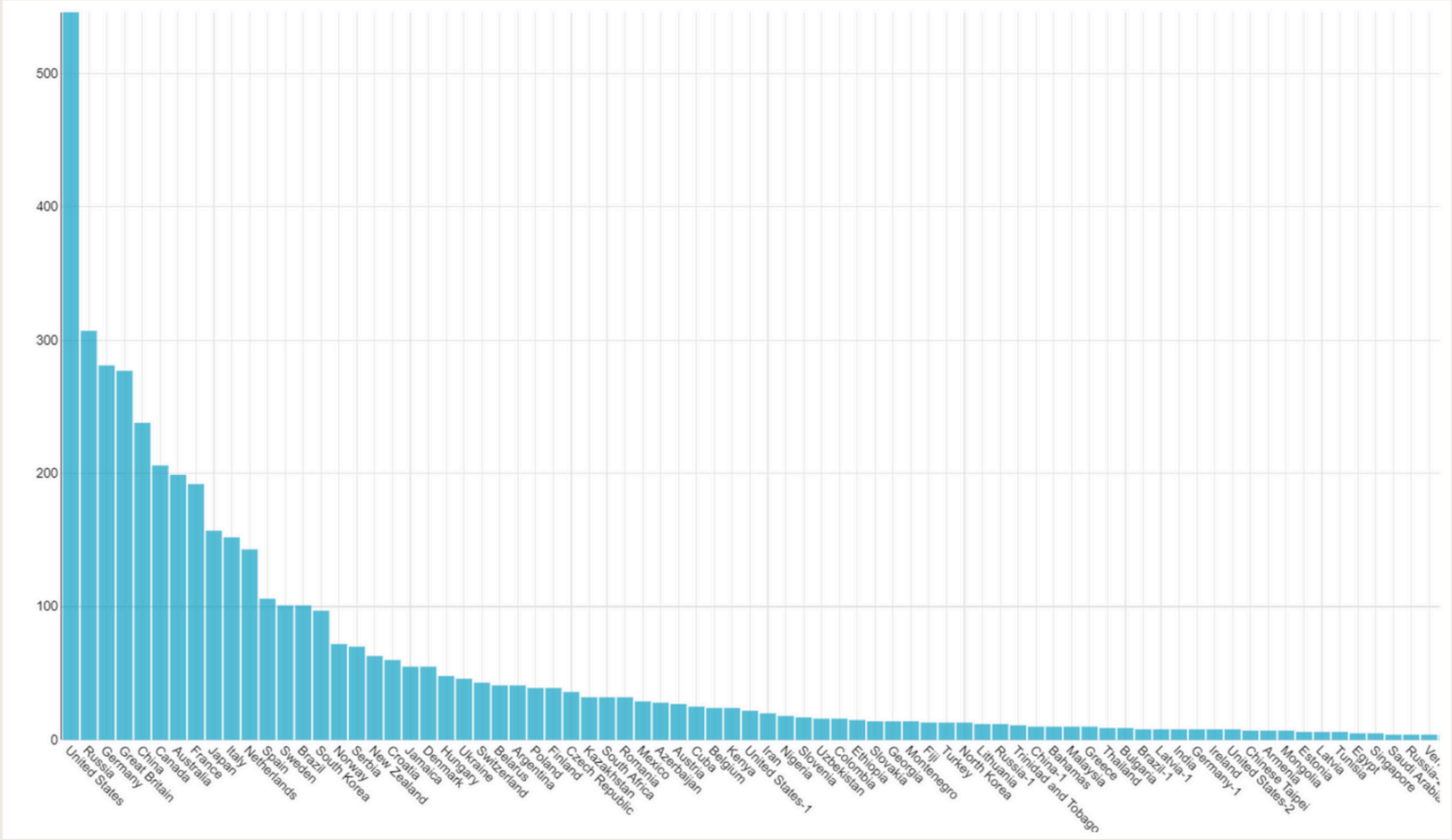
VISUALIZATIONS



Q: What teams have won the most medals (all medals) overall? (Top 3)

A: USA (5.22k medals), Soviet Union (2.45k medals), Germany (1.98k medals)

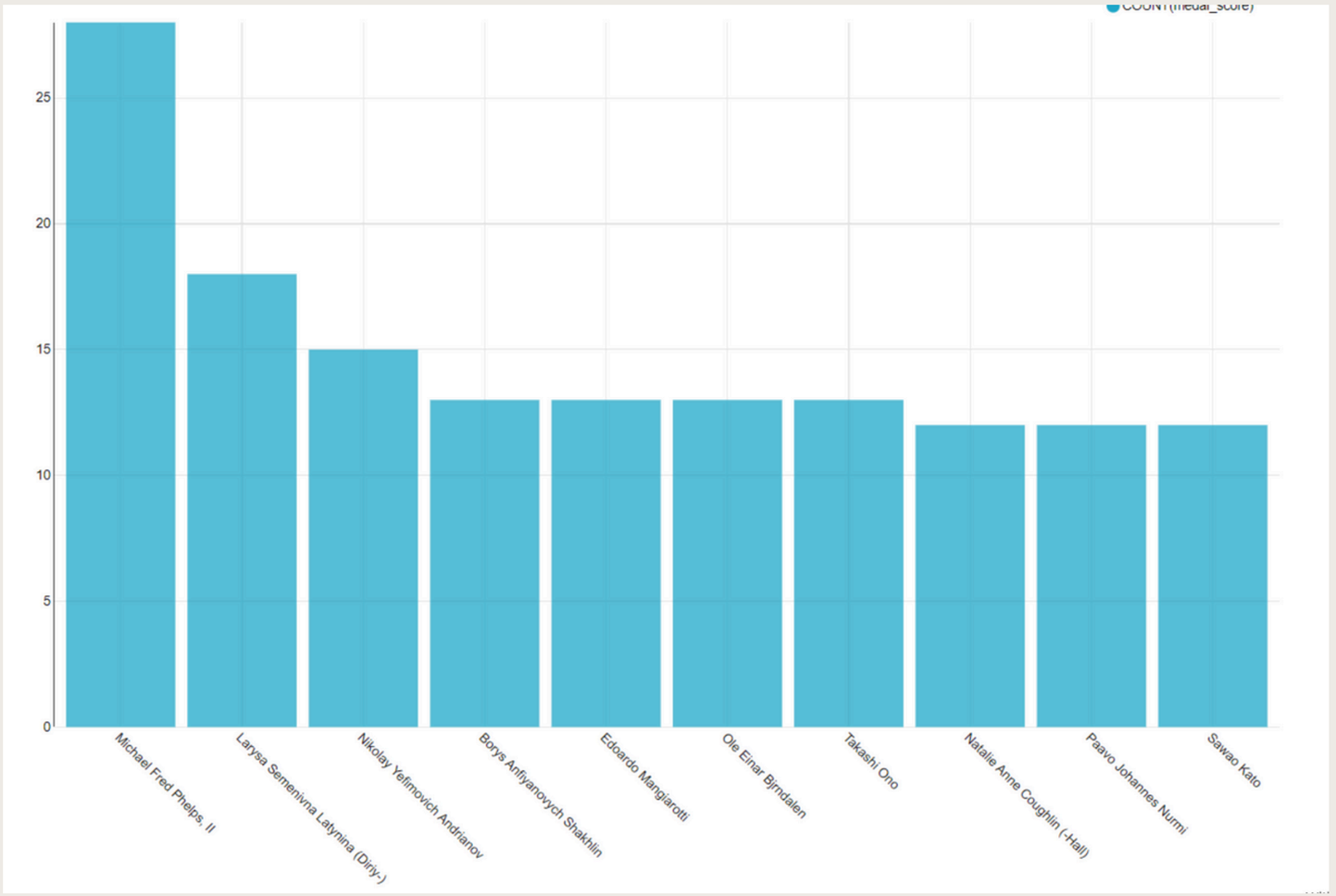
VISUALIZATIONS



Q: What teams have won the most medals (all medals) in each of the last 5 Olympic games?

A: USA (546 medals)

VISUALIZATIONS



Q: What athletes have won the most medals? (Top 10)

A: Michael Phelps...

THANK YOU!

Any questions, comments, or concerns?