**Merge two unrelated datasets COVID-19 case data and economic indicators and resolve schema conflicts**

**Objective**

To integrate two unrelated datasets—COVID-19 global data and economic indicators—into a unified PostgreSQL database that allows coherent analysis through structured schema design, data cleaning, normalization, and a unified analytical view.

**Database Setup**

**Create and Connect to Database**

*CREATE DATABASE data\_merge;*

*\c merging\_db;*

**Data Staging**

**Create Staging Table for Economic Indicators**

*CREATE TABLE stg\_economic\_indicators (*

*country TEXT,*

*indicator TEXT,*

*year INT,*

*value NUMERIC*

*);*

**Prepare the Economic Dataset File**

*sudo cp /home/main/Downloads/economic\_indicators\_dataset\_2010\_2023.csv /tmp/*

*sudo chmod 644 /tmp/economic\_indicators\_dataset\_2010\_2023.csv*

**Load Data into PostgreSQL**

*psql -U postgres -d merging\_db -h localhost \*

*-c "\copy stg\_covid\_worldwide \*

*FROM '/tmp/economic\_indicators\_dataset\_2010\_2023.csv ' \”*

**Clean COVID Dataset Using Python**

The COVID dataset had irregularities like "N/A" and empty strings. A Python script was used to clean the data.

*import csv*

*input\_path = "/tmp/covid\_worldwide.csv"*

*output\_path = "/tmp/covid\_worldwide\_clean.csv"*

*with open(input\_path, "r") as infile, open(output\_path, "w", newline="") as outfile:*

*reader = csv.reader(infile)*

*writer = csv.writer(outfile)*

*header = next(reader)*

*writer.writerow(header)*

*for row in reader:*

*cleaned = []*

*for cell in row:*

*cell = cell.strip()*

*if cell.upper() == "N/A" or cell == "":*

*cleaned.append("")*

*elif cell.replace(",", "").replace(".", "").isdigit():*

*cleaned.append(cell.replace(",", ""))*

*else:*

*cleaned.append(cell)*

*writer.writerow(cleaned)*

**Create Staging Table for Cleaned COVID Data**

*CREATE TABLE stg\_covid\_worldwide (*

*serial\_number INTEGER,*

*country TEXT,*

*total\_cases BIGINT,*

*total\_deaths BIGINT,*

*total\_recovered BIGINT,*

*active\_cases BIGINT,*

*total\_tests BIGINT,*

*population BIGINT*

*);*

**Load Cleaned Data into PostgreSQL**

*psql -U postgres -d merging\_db -h localhost \*

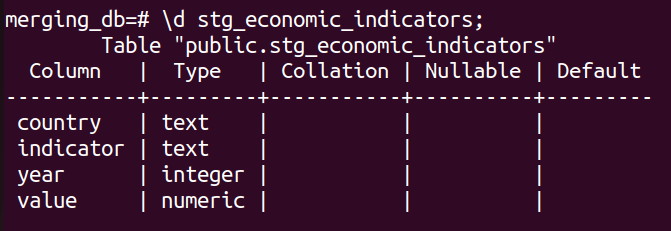
*-c "\copy stg\_covid\_worldwide \*

*FROM '/tmp/covid\_worldwide\_clean.csv' \*

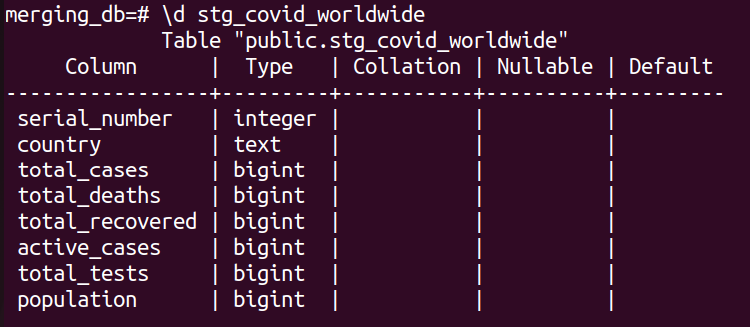
*WITH CSV HEADER NULL ''"*

**Schema Inspection and Verification**

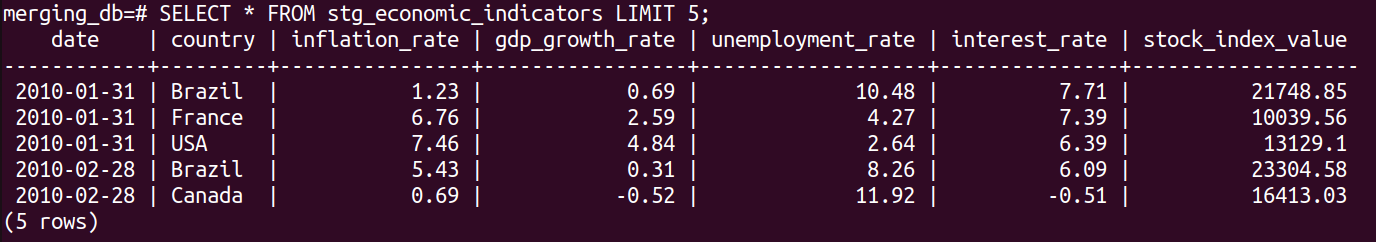
*\d stg\_economic\_indicators;*



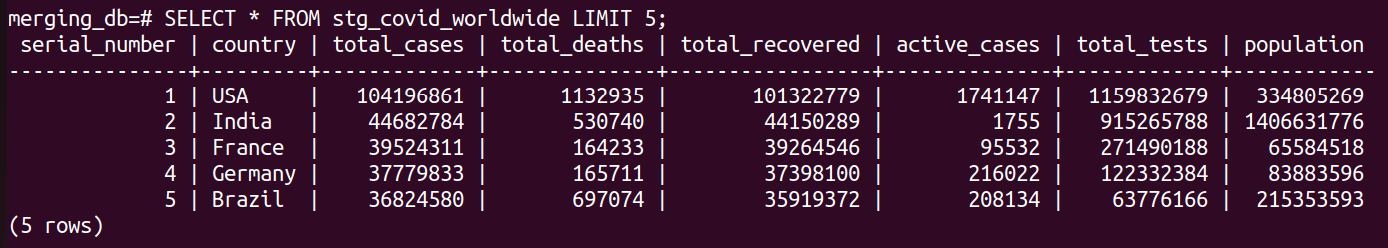
*\d stg\_covid\_worldwide;*



*SELECT \* FROM stg\_economic\_indicators LIMIT 5;*



*SELECT \* FROM stg\_covid\_worldwide LIMIT 5;*

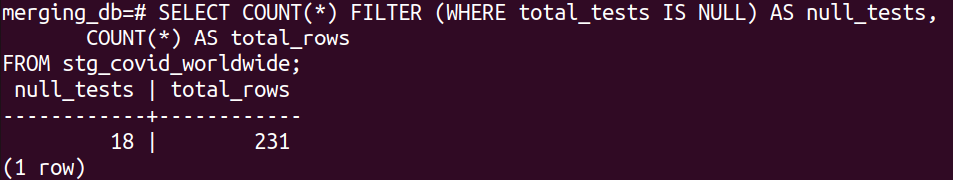


-- Check for NULLs

*SELECT COUNT(\*) FILTER (WHERE total\_tests IS NULL) AS null\_tests,*

*COUNT(\*) AS total\_rows*

*FROM stg\_covid\_worldwide;*



**Dimension Tables**

**Country Dimension Table**

*CREATE TABLE dim\_country (*

*country\_id SERIAL PRIMARY KEY,*

*country\_name TEXT UNIQUE*

*);*

**Populate with Union of Countries**

*INSERT INTO dim\_country (country\_name)*

*SELECT DISTINCT country FROM stg\_covid\_worldwide*

*UNION*

*SELECT DISTINCT country FROM stg\_economic\_indicators*

*ON CONFLICT DO NOTHING;*

**Date Dimension Table**

*CREATE TABLE dim\_date (*

*date\_id SERIAL PRIMARY KEY,*

*date DATE UNIQUE,*

*year INT,*

*month INT,*

*quarter INT*

*);*

**Populate from Economic Indicators**

*INSERT INTO dim\_date (date, year, month, quarter)*

*SELECT DISTINCT*

*date,*

*EXTRACT(YEAR FROM date)::INT,*

*EXTRACT(MONTH FROM date)::INT,*

*EXTRACT(QUARTER FROM date)::INT*

*FROM stg\_economic\_indicators*

*ON CONFLICT DO NOTHING;*

**Fact Tables**

**Economic Indicators Fact Table**

*CREATE TABLE fact\_economics (*

*country\_id INT REFERENCES dim\_country(country\_id),*

*date\_id INT REFERENCES dim\_date(date\_id),*

*inflation\_rate NUMERIC,*

*gdp\_growth\_rate NUMERIC,*

*unemployment\_rate NUMERIC,*

*interest\_rate NUMERIC,*

*stock\_index\_value NUMERIC*

*);*

**Populate Fact Table**

*INSERT INTO fact\_economics (country\_id, date\_id, inflation\_rate, gdp\_growth\_rate, unemployment\_rate, interest\_rate, stock\_index\_value)*

*SELECT*

*dc.country\_id,*

*dd.date\_id,*

*sei.inflation\_rate,*

*sei.gdp\_growth\_rate,*

*sei.unemployment\_rate,*

*sei.interest\_rate,*

*sei.stock\_index\_value*

*FROM stg\_economic\_indicators sei*

*JOIN dim\_country dc ON dc.country\_name = sei.country*

*JOIN dim\_date dd ON dd.date = sei.date;*

**COVID Data Fact Table**

*CREATE TABLE fact\_covid (*

*country\_id INT REFERENCES dim\_country(country\_id),*

*total\_cases BIGINT,*

*total\_deaths BIGINT,*

*total\_recovered BIGINT,*

*active\_cases BIGINT,*

*total\_tests BIGINT,*

*population BIGINT*

*);*

**Populate Fact Table**

*INSERT INTO fact\_covid (country\_id, total\_cases, total\_deaths, total\_recovered, active\_cases, total\_tests, population)*

*SELECT*

*dc.country\_id,*

*scw.total\_cases,*

*scw.total\_deaths,*

*scw.total\_recovered,*

*scw.active\_cases,*

*scw.total\_tests,*

*scw.population*

*FROM stg\_covid\_worldwide scw*

*JOIN dim\_country dc ON dc.country\_name = scw.country;*

**Unified Analytical View**

**Create View for Unified Access**

*CREATE VIEW view\_country\_snapshot AS*

*SELECT*

*dc.country\_name,*

*dd.year,*

*dd.month,*

*fe.inflation\_rate,*

*fe.gdp\_growth\_rate,*

*fe.unemployment\_rate,*

*fe.interest\_rate,*

*fe.stock\_index\_value,*

*fc.total\_cases,*

*fc.total\_deaths,*

*fc.total\_recovered,*

*fc.active\_cases,*

*fc.total\_tests,*

*fc.population*

*FROM fact\_economics fe*

*JOIN dim\_country dc ON fe.country\_id = dc.country\_id*

*JOIN dim\_date dd ON fe.date\_id = dd.date\_id*

*JOIN fact\_covid fc ON fc.country\_id = dc.country\_id*

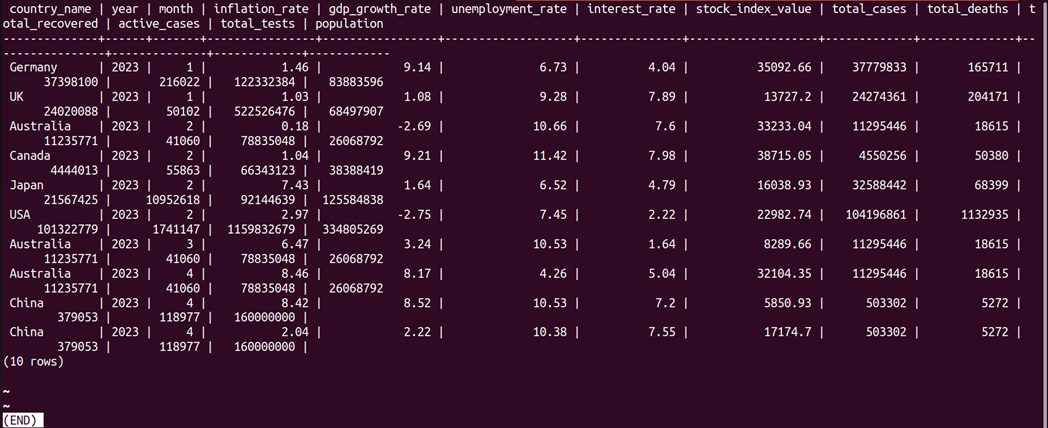
*WHERE dd.year = 2023;*

**Validation and Querying the Unified View**

**Preview Unified Data**

*psql -U postgres -d merging\_db -h localhost \*

*-c "SELECT \* FROM view\_country\_snapshot LIMIT 10;"*



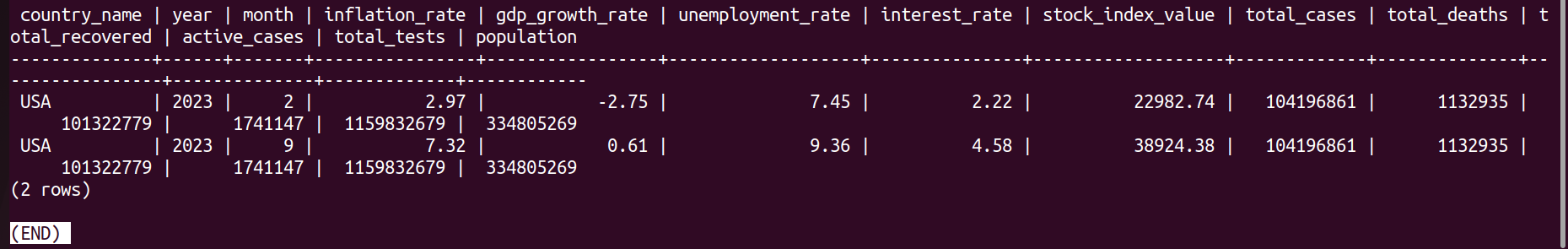
**Filter by Country and Time**

*SELECT \**

*FROM view\_country\_snapshot*

*WHERE country\_name = 'USA' AND year = 2023*

*ORDER BY month;*



**Aggregate Analysis**

*SELECT*

*country\_name,*

*AVG(gdp\_growth\_rate) AS avg\_gdp\_growth,*

*total\_cases*

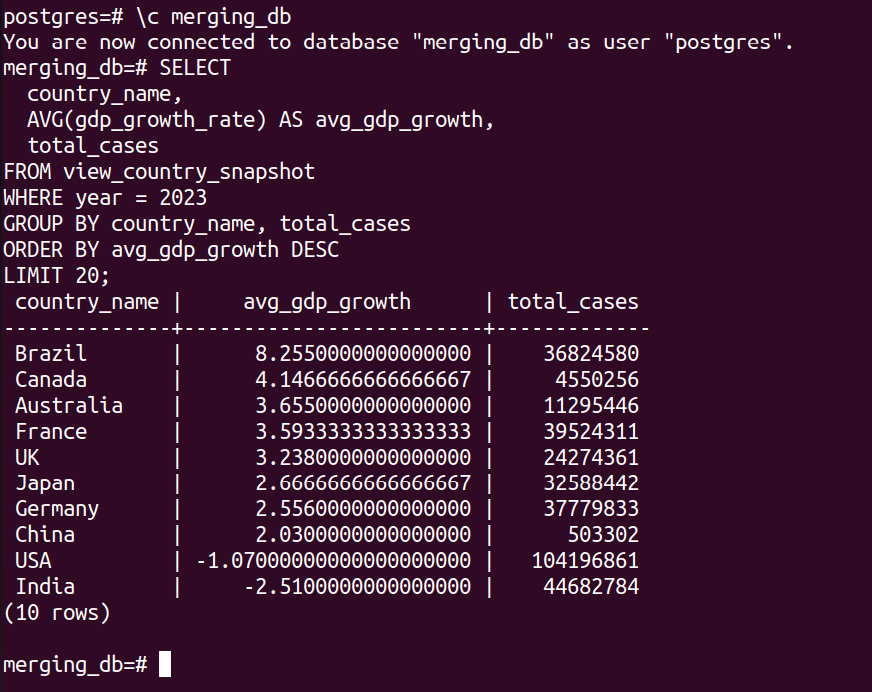
*FROM view\_country\_snapshot*

*WHERE year = 2023*

*GROUP BY country\_name, total\_cases*

*ORDER BY avg\_gdp\_growth DESC*

*LIMIT 20;*



**Conclusion**

Through structured staging, transformation, and modeling, the unrelated COVID and economic datasets have been successfully merged into a single PostgreSQL-based data warehouse. Analysts can now seamlessly explore temporal and regional relationships between public health metrics and economic performance from a unified interface.