Data-Driven Insights: Modeling the World’s Relations with SQL

Babiker Babiker

Jason Bravo   
Vlad Korenevskyi  
Homayoun Nistanaki

Department of Computer Science, California State University, Fullerton

CPSC 332 File Structure & Databases

Prof. Nader Rashad

December7th 2024

Introduction

Understanding the relationships between countries, their economies, and their cultural frameworks is vital in grasping how our interconnected world operates. This project uses SQL to build a comprehensive database that captures the intricate dynamics between nations, encompassing their borders, currencies, time zones, languages, and trade networks.

Countries do not exist in isolation—they are defined by their connections and interactions with neighboring nations, shared linguistic ties, overlapping time zones, and mutual economic dependencies. For instance, trade relationships often rely on shared currencies or time zone compatibility, while cultural connections like common languages can foster closer political and social collaboration.

The primary objective of this project is to model these relationships in a structured and scalable database, enabling us to analyze how various elements interact to shape a country's global identity. By creating tables and establishing links between them, we aim to simulate the complexity of real-world international systems. This database not only demonstrates the power of SQL in handling and visualizing relational data but also provides a framework to explore broader global questions, such as the impact of shared borders on trade, the influence of language on diplomatic alliances, and the role of time zones in international cooperation.

Through this database, we aim to highlight the importance of data modeling in understanding global systems and to showcase how structured information can unravel the complexities of relationships that define our world.

**Database Development Process**

**Figure 1**

Database Schema

A screenshot of a computer

Description automatically generated

The database schema is designed to represent country-related data in a structured, normalized manner. Each table is labeled with its attributes and necessary constraints, ensuring the integrity and consistency of the stored data. This schema supports efficient data storage, retrieval, and management while adhering to best practices in database design.

The schema includes additional tables — Currency, CountryBorder, and TimeZone — to enhance the granularity and comprehensiveness of the information represented. These tables are closely linked to the central Country table through foreign key relationships, forming a relational database structure.

**Key Tables**

* Currency Table
  + CurrencyCode char(3) PRIMARY KEY, FOREIGN KEY
  + CurrencyName varchar(50)
* CurrencyCountry Table
  + CurrencyCode char(3) FOREIGN KEY
  + CountryCode char(3) FOREIGN KEY
  + (CurrencyCode, CountryCode) PRIMARY KEY
* CountryBorder Table
  + CountryCode char(3) FOREIGN KEY
  + NeighborCode char(3) FOREIGN KEY
  + (CountryCode, NeighborCode) PRIMARY KEY
  + BorderLength float
* CountryTimeZone Table
  + Offset float FOREIGN KEY
  + CountryCode FOREIGN KEY
  + (CountryCode, Offset) PRIMARY KEY
* TimeZone Table
  + Offset PRIMARY KEY

**Figure 2**

Physical Model

A screenshot of a computer

Description automatically generated

Displayed above is our physical model which displays each table in full depth. It’s showing the original 3 tables given, plus our added tables along with necessary attributes and constraints.

**Queries**

USE world;

-- Find the cities with the highest population in each country

SELECT

City.Name AS CityName,

City.Population,

Country.Name AS CountryName

FROM

City

JOIN

Country

ON

City.CountryCode = Country.Code

WHERE

City.Population = (

SELECT MAX(Population)

FROM City AS SubCity

WHERE SubCity.CountryCode = City.CountryCode

);

-- List countries with more than one official language

SELECT

Country.Name AS CountryName,

(SELECT COUNT(\*)

FROM CountryLanguage AS SubLang

WHERE SubLang.CountryCode = Country.Code

AND SubLang.IsOfficial = 'T') AS OfficialLanguageCount

FROM

Country

WHERE

(SELECT COUNT(\*)

FROM CountryLanguage AS SubLang

WHERE SubLang.CountryCode = Country.Code

AND SubLang.IsOfficial = 'T') > 1;

-- Find countries where the average city population is greater than 1 million

SELECT

Country.Name AS CountryName,

(SELECT AVG(Population)

FROM City AS SubCity

WHERE SubCity.CountryCode = Country.Code) AS AvgCityPopulation

FROM

Country

WHERE

(SELECT AVG(Population)

FROM City AS SubCity

WHERE SubCity.CountryCode = Country.Code) > 1000000;

-- List countries and their total population for those with at least one city having a population over 5 million

SELECT

Country.Name AS CountryName,

Country.Population AS TotalPopulation

FROM

Country

WHERE

EXISTS (

SELECT 1

FROM City

WHERE City.CountryCode = Country.Code

AND City.Population > 5000000

);

-- Find countries with a time zone offset of "UTC" and their capital city

SELECT

Country.Name AS CountryName,

(SELECT City.Name

FROM City

WHERE City.ID = Country.Capital) AS CapitalCity,

CountryTimeZone.Offset

FROM

Country

JOIN

CountryTimeZone

ON

Country.Code = CountryTimeZone.CountryCode

WHERE

CountryTimeZone.Offset = 'UTC';

-- Countries that use USD

SELECT

Country.Name AS CountryName,

Currency.CurrencyName

FROM

Country

JOIN

CurrencyCountry

ON

Country.Code = CurrencyCountry.CountryCode

JOIN

Currency

ON

CurrencyCountry.CurrencyCode = Currency.CurrencyCode

WHERE

Currency.CurrencyCode = 'USD';

-- Countries that share border and language

SELECT DISTINCT

c1.Name AS Country1,

c2.Name AS Country2,

cl1.language AS SharedLanguage

FROM

CountryBorder cb

JOIN

Country c1 ON cb.country1Code = c1.Code

JOIN

Country c2 ON cb.country2Code = c2.Code

JOIN

CountryLanguage cl1 ON c1.Code = cl1.countryCode AND cl1.isOfficial = TRUE

JOIN

CountryLanguage cl2 ON c2.Code = cl2.countryCode AND cl2.isOfficial = TRUE

AND cl1.language = cl2.language

ORDER BY

c1.Name, c2.Name;

-- Countries which share borders but have different time zones

SELECT DISTINCT

LEAST(c1.Name, c2.Name) AS Country1,

GREATEST(c1.Name, c2.Name) AS Country2

FROM

CountryBorder cb

JOIN

Country c1 ON cb.country1Code = c1.Code

JOIN

Country c2 ON cb.country2Code = c2.Code

WHERE

NOT EXISTS (

SELECT 1

FROM CountryTimeZone t1

JOIN CountryTimeZone t2

ON t1.countryCode = c1.Code AND t2.countryCode = c2.Code

WHERE t1.offset = t2.offset

)

ORDER BY

Country1, Country2;

-- List the time zones of countries along with the head of state

SELECT

City.Name AS CityName,

Country.Name AS CountryName,

City.Population AS CityPopulation

FROM

City

JOIN

Country

ON

City.CountryCode = Country.Code

WHERE

City.Population > 1000000;

-- Find the cities and countries that have populations greater than 1 million

SELECT

City.Name AS CityName,

Country.Name AS CountryName,

City.Population AS CityPopulation

FROM

City

JOIN

Country

ON

City.CountryCode = Country.Code

WHERE

City.Population > 1000000;

-- List country names for each border

SELECT

c1.Name AS Country1,

c2.Name AS Country2,

cb.BorderLength

FROM

CountryBorder cb

JOIN Country c1 ON cb.Country1Code = c1.Code

JOIN Country c2 ON cb.Country2Code = c2.Code;

-- Find the names of countries that share a border with more than two countries.

SELECT c.Name AS CountryName

FROM Country c

WHERE c.Code IN (

SELECT cb.Country1Code

FROM CountryBorder cb

GROUP BY cb.Country1Code

HAVING COUNT(cb.Country2Code) > 2

);

-- List all countries that have a border longer than 1000 km.

SELECT DISTINCT c.Name AS CountryName

FROM Country c

WHERE c.Code IN (

SELECT cb.Country1Code

FROM CountryBorder cb

WHERE cb.BorderLength > 1000

);

-- Identify countries with cities where the population is greater than 10% of the country's total population.

SELECT Name AS CountryName

FROM Country

WHERE Code IN (

SELECT ci.CountryCode

FROM City ci

WHERE ci.Population > (

SELECT Population \* 0.1

FROM Country c

WHERE c.Code = ci.CountryCode

)

);

-- Find countries where at least one official language has a percentage greater than 50%.

SELECT Name AS CountryName

FROM Country

WHERE Code IN (

SELECT CountryCode

FROM CountryLanguage

WHERE IsOfficial = 'T' AND Percentage > 50

);

-- Find the name of the currency used by countries in the 'Middle East' region.

SELECT DISTINCT cu.CurrencyName

FROM Currency cu

WHERE cu.CurrencyCode IN (

SELECT cc.CurrencyCode

FROM CurrencyCountry cc

WHERE cc.CountryCode IN (

SELECT c.Code

FROM Country c

WHERE c.Region = 'Middle East'

)

);

-- List all cities in the City table.

SELECT Name AS CityName

FROM City;

-- Find the total number of countries in the world.

SELECT COUNT(\*) AS TotalCountries

FROM Country;

-- Retrieve the names of all continents.

SELECT DISTINCT Continent

FROM Country;

-- List all countries in Asia.

SELECT Name AS CountryName

FROM Country

WHERE Continent = 'Asia';

**Summary**

This project showcases the power of SQL in modeling and analyzing the complex relationships that define global interactions between nations. By incorporating data on borders, currencies, languages, and time zones, the database captures the interconnected systems that influence cultural, economic, and political ties. The carefully designed, normalized schema ensures efficient data storage and retrieval, providing a scalable framework to explore how shared characteristics—such as common languages or overlapping time zones—shape international dynamics.

Through this database, we gain a deeper understanding of the relationships that define our interconnected world. It highlights the role of structured data in unraveling complex global systems, enabling analyses of critical questions like the impact of shared borders on trade or the influence of linguistic and temporal compatibility on international cooperation. This project not only emphasizes the importance of relational database design but also demonstrates its potential in advancing our understanding of global interactions.

**References**

Last Name, A. B. (Year). Article Title. Journal Title, Pages #-#. URL. URL.

Last Name, C. D. (Year). Book TitleBook Title URL.

Last Name, D. E., Last Name, F. G. (Year). Report TitleReport Title URL.

Last Name, H. I. (Year, Month Day). Article Title/Headline. Periodical.Periodical.

Organization Name. (Year, Month Day).Webpage Title. URL.