EU GDP Visualizer

Introduction

Over the past few years, economic inequality within the European Union has become a fundamental issue alongside the gaps in opportunities and living standards among the members of the bloc. But these inequalities must be viewed in context where satisfaction on the effectiveness of EU cohesion policies or precise areas requiring action can be better regarded. I will analyze a few important economic parameters on 27 European Union countries during the period 2012 to 2023 such as GDP and GDP growth. In doing so, the gaps regarding the changes in dynamics of economic differences can be reflected.

Stage 1 - Find and critique a dataset

Source of open and real data

I opted to concentrate on two datasets on GDP available from Eurostat that shows all of the member states economic performance within a region within a time the time period of 2012 to 2023. The datasets include:

1. **GDP at market prices**

   Link - https://ec.europa.eu/eurostat/databrowser/view/tec00001/default/table?lang=en&category=t_na10.t_nama10.t_nama_10_ma

2. **Real GDP growth rate**

   Link - https://ec.europa.eu/eurostat/databrowser/view/tec00115/default/line?lang=en&category=t_na10.t_nama10.t_nama_10_ma

The data source is open and real. All datasets are from Eurostat which is the official statistics office for the European Union, which provides open access to economic data.

Data assessment

Quality

These datasets are of perfect quality, since they are obtained and validated by Eurostat which is the official statistical office of the EU. The data is using the Eurostat guidelines fully through the application of the specific European System of Accounts (ESA 2010) methods, which guarantees conformity of every EU member state. Quality control procedures have always been carried out together with validation processes and all changes or improvements made are well documented. This is ensuring dependability and relative evaluation.

Level of Detail

- Coverage of all EU member states
- Yearly data entry from 2012 to 2023
- Clear unit measurements

Documentation

- Comprehensive metadata
- Explanation of data collection
- Definitions of all variables and indicators

Interrelation

- Relationships between GDP and growth rates
- Ability to link with other economic datasets

Discoverability

- Eurostat website
- Download links
- API access

Data selected overview & Interests and key questions

I used real open data that was derived from the website of Eurostat's Economic Indicators Database. Eurostat has many economic and social indicators, but I decided to look at GDP indicators because they are important for determining the level of economic and development in the countries of the European Union. Out of the many datasets, I took out two as they reflect in complementary forms the economic development of European countries.

Considering the objective of the work, I looked on the annual data from 2012 till 2023, allowing for a holistic understanding of the economic development over this eleven year period. This period also captures a number of crucial activities that affect the economy and therefore enable an analysis of some features. The selected indicators may be classified into two broad categories as follows:

1. **Economic Size (GDP at market prices)**

- Reflects actual size of national economies
- Shows total economic output

2. **Economic Growth (Real GDP growth rate)**

- Measured as percentage change
- Shows economic momentum

Using these datasets, I can inspect various aspects of economic development and performance of the EU member states. Some interesting questions would be:

1. In the EU, how does the growth trajectory differ across smaller and larger economies? Are smaller economies growing relatively faster?
2. How have significant economic shocks affected the various economies of the EU?

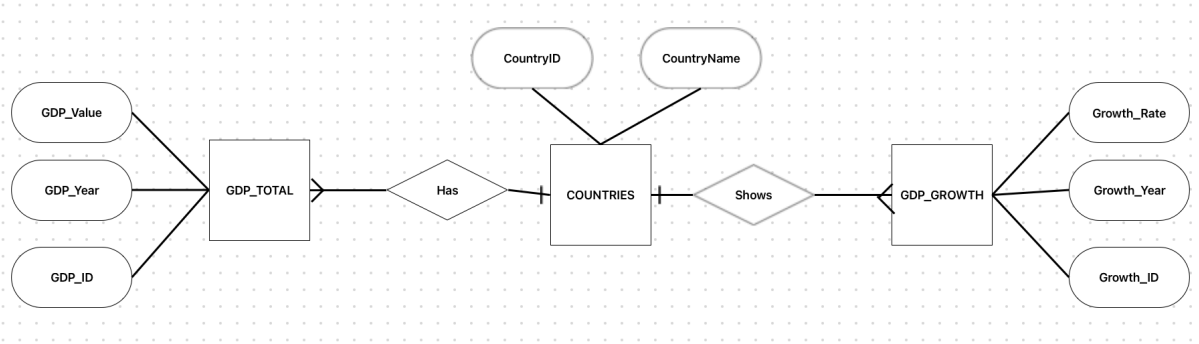
In order to answer these questions adequately, the database has to be structured in a manner that combines temporal and spatial factors while also

holding absolute measures of GDP and the rates of growth of GDP.

Stage 2 - Model your data

E/R model of the data

My E/R model diagram:



E/R Model

The structure, entities and relationships for my database are as follows:

- **Countries Table**
  - Information about each EU member country
  - Attributes:
    - CountryName
    - CountryID
- **GDP_Total**
  - Information about the total GDP values
  - Attributes:
    - GDP_Value
    - GDP_Year
    - GDP_ID
- **GDP_Growth**

    - Information about the GDP growth rates
    - Attributes:
        - Growth_Rate
        - Growth_Year
        - Growth_ID

Cardinality E/R diagram

1. COUNTRIES to GDP_TOTAL relationship ("Has"):
    - One country can hold one or many GDP total values
    - Each GDP total value can only belong to one country
2. COUNTRIES to GDP_GROWTH relationship ("Shows"):
    - One country can hold one or many GDP growth rates
    - Each GDP growth rate can only belong to one country

This cardinality structure ensures that:

1. The country has at least one record of total and growth rate of GDP
2. Multiple GDP measurements can be associated with a single country
3. Every GDP measurement must belong to exactly one country

Database tables and fields

1. 1NF：
    - Every tables has a primary keys
    - Every attributes contains values
    - No repeating groups
2. 2NF：
    - Meets 1NF requirements

- No partial dependencies
- All non-key attributes are fully dependent on their primary keys

3. 3NF：
    - Meets 2NF requirements
    - Each non-key attribute depends only on the primary key
    - No transitive dependencies

The database design is in 3NF because:

- Each table has a clear primary key
- All fields depend on their respective primary keys
- No transitive dependencies exist
- Relationships are properly established through foreign keys

Stage 3 - Create the database

Build the database structure in MySQL

Create the Database

```
CREATE DATABASE eu_gdp_db;
USE eu_gdp_db;
```

Create Countries Table

```
CREATE TABLE Countries (
    CountryID INT NOT NULL AUTO_INCREMENT,
    CountryName VARCHAR(255) NOT NULL,
    PRIMARY KEY (CountryID),
    UNIQUE (CountryName)
);
```

Create GDP_Total Table

```
CREATE TABLE GDP_Total (
    GDP_ID INT NOT NULL AUTO_INCREMENT,
    GDP_Value DECIMAL(15,2) NOT NULL,
    GDP_Year INT NOT NULL,
    CountryID INT NOT NULL,
    PRIMARY KEY (GDP_ID),
    FOREIGN KEY (CountryID) REFERENCES Countries(CountryID),
    CHECK (GDP_Year BETWEEN 2012 AND 2023)
);
```

Create GDP_Growth Table

```
CREATE TABLE GDP_Growth (
    Growth_ID INT NOT NULL AUTO_INCREMENT,
    Growth_Rate DECIMAL(5,2) NOT NULL,
    Growth_Year INT NOT NULL,
    CountryID INT NOT NULL,
    PRIMARY KEY (Growth_ID),
    FOREIGN KEY (CountryID) REFERENCES Countries(CountryID),
    CHECK (Growth_Year BETWEEN 2012 AND 2023)
);
```

Show the created tables

```
SHOW TABLES;
```

Show the structures of the tables

```
DESCRIBE Countries;
DESCRIBE GDP_Total;
DESCRIBE GDP_Growth;
```

Enter instance data

Data Collection

- Downloaded raw GDP datasets directly from Eurostat's database portal
- Selected GDP at market prices (tec00001) and Real GDP growth rate (tec00115)
- Exported data in CSV format with selected parameters for 2012-2023

Data Cleaning and Preparation
Created three distinct CSV files to match the database structure:

1. countries.csv: Extracted unique country names, removing aggregate entries such as "Euro area" and "European Union"
2. gdp_total.csv: Processed total GDP values, maintaining country-year-value structure
3. gdp_growth.csv: Organized growth rate data by country and year

Data Import Process

1. Transferred cleaned CSV files to project environment
2. Utilized MySQL's LOAD DATA INFILE command to populate database tables
3. Verified data integrity through sample queries

I loaded the data into MySQL with the `LOAD DATA INFILE` command:

Data Collection

- GDP datasets were directly downloaded from Eurostat database portal.
- Claimed GDP at Market Prices and real GDP growth rate.
- Data for 2012 to 2023 was exported in CSV format.

Data Cleaning and Preparation

Defined three separate CSV files to satisfy the requirements of the database:

1. countries.csv – Country names were populated after removing aggregates like “Euro area”, “European Union” and unique country names compiled.

2. gdp_total.csv – Total values for GDPs were entered in a country, year and value format.
3. gdp_growth.csv – Growth rates were also captured as per countries and years.

Data Import Process

1. CSV Files that were cleaned were transferred into the project area.
2. MySQL's LOAD DATA INFILE was used in this case to insert the database tables.
3. Some sample queries were executed to establish whether the data was correctly entered within the database.

I uploaded the data into MySQL and I had to call for the `LOAD DATA INFILE` command:

Countries

```
LOAD DATA LOCAL INFILE '/home/coder/project/data/countries.csv'
INTO TABLE Countries
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(CountryName);
```

GDP_Growth

```
LOAD DATA INFILE '/home/coder/project/data/gdp_growth.csv'
INTO TABLE GDP_Growth
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(Growth_Rate, Growth_Year, CountryID);
```

GDP_Total

```
LOAD DATA INFILE '/home/coder/project/data/gdp_total.csv'
INTO TABLE GDP_Total
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(GDP_Value, GDP_Year, CountryID);
```

Reflect on how well the database reflects the data

Good Points:

1. Has a clear and efficient data architecture.

- Countries, GDP_Total and GDP_Growth have been successfully defined as separate entities in the database.
- Foreign key and maintenance of referential integrity through the CountryID avoids data replication.
- Due to the normalized design, the total amount of GDP and the growth rate can be queried separately.

Areas for Improvement:

1. Coverage of Data Problems and Limitations

- For a few countries, there are some undocumented gaps within the time series data.
- The set of, for this case, data ends with 2023 meaning that to analyze the current scenario, data regularly preforming future updates will be necessary.
- This may present limits on certain critical trends even as regression models will be somewhat hinge on these construct by some countries not reported certain years.

1. Structural Issues.

- Most queries that would be meaningful to the user will need JOINs since the current one is normalized.

- Interfacing with both the GDP Total and growth rate would mean cross-queries, which, especially when dealing with larger datasets, may be a bottleneck.
- Adjusting Rational and Rounded Values (i.e Decimal(15,2) for GDP_Value and Decimal(5,2) for Growth_Rate) as required per data ranges may be important.

These reflections strongly justify the design of our database and inform on enhancement areas that could be considered in future projects.

List SQL commands

1. **How do growth patterns differ between large and small economies in the EU? Are smaller economies showing faster growth rates?**

To determine this, I classified economies into "large" and "small" based on `GDP_Value` and compare their average `Growth_Rate`:

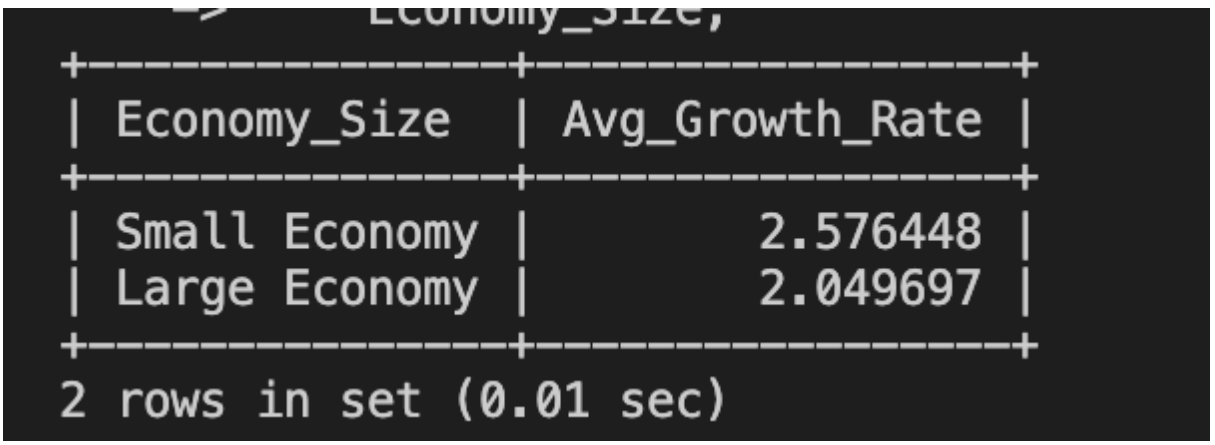
SQL:

```
SELECT
    CASE
        WHEN AVG(gdp.GDP_Value) > (SELECT AVG(GDP_Value) FROM gdp_t
otal) THEN 'Large Economy'
        ELSE 'Small Economy'
    END AS Economy_Size,
    AVG(gg.Growth_Rate) AS Avg_Growth_Rate
FROM
    gdp_growth gg
JOIN
    gdp_total gdp ON gg.CountryID = gdp.CountryID AND gg.Growth_Year =
gdp.GDP_Year
GROUP BY
    Economy_Size;
```

A error `Can't group on 'Economy_Size'` occurs.

Here's the corrected query:

```
SELECT
   Economy_Size,
   AVG(Growth_Rate) AS Avg_Growth_Rate
FROM (
   SELECT
      gdp.CountryID,
      CASE
         WHEN AVG(gdp.GDP_Value) OVER (PARTITION BY gdp.CountryID) >
            (SELECT AVG(GDP_Value) FROM GDP_Total) THEN 'Large Econo
my'
         ELSE 'Small Economy'
      END AS Economy_Size,
      gg.Growth_Rate
   FROM
      GDP_Growth gg
   JOIN
      GDP_Total gdp ON gg.CountryID = gdp.CountryID AND gg.Growth_Yea
r = gdp.GDP_Year
) subquery
GROUP BY
   Economy_Size;
```



From the query we see that small economies grow faster on average than larger economies.

2. **What impact have major economic events had on different EU economies, and how does economic size relate to resilience?**

To answer this, I focused on specific years corresponding to major economic events (e.g., 2020 covid) and analyze changes in `Growth_Rate` and `GDP_Value`:

SQL:

```
SELECT
    gdp.CountryID,
    c.CountryName,
    gdp.GDP_Year,
    gdp.GDP_Value,
    gg.Growth_Rate,
    LAG(gg.Growth_Rate) OVER (PARTITION BY gdp.CountryID ORDER BY gd
p.GDP_Year) AS Prev_Growth_Rate,
    (gg.Growth_Rate - LAG(gg.Growth_Rate) OVER (PARTITION BY gdp.Coun
tryID ORDER BY gdp.GDP_Year)) AS Growth_Change
FROM
    gdp_growth gg
JOIN
    gdp_total gdp ON gg.CountryID = gdp.CountryID AND gg.Growth_Year =
gdp.GDP_Year
JOIN
    countries c ON gg.CountryID = c.CountryID
WHERE
    gdp.GDP_Year IN (2020);
```

The data shows that 2020 was a challenging year due to the COVID crisis, with every economy experiencing negative growth rates.

Stage 4 - Create a simple web application

Web Application Development

For the web application implementation, I developed a Node.js application using Express.js framework to create an interactive interface for analyzing EU GDP data. This application was intended to resolve the major research problems

posed in Stage 1 especially economic performance benchmarking across EU member nations.

Technical Implementation

- EJS (Embedded JavaScript) for templating
- Node.js and Express.js for the server-side framework
- Body-parser for handling form submissions
- MySQL for database connectivity

The core functionality was implemented through several key components:

1. Database Connection

```
const db = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    database: 'eu_gdp_db'
});
```

2. Main Application Routes
    a. Analysis route processes selected countries and displays GDP data
    b. Both routes incorporate error handling and data validation
    c. Home route displays country selection interface

3. Data Querying
    a. Year-over-year comparisons
    b. Growth rates
    c. Total GDP values

Key Features

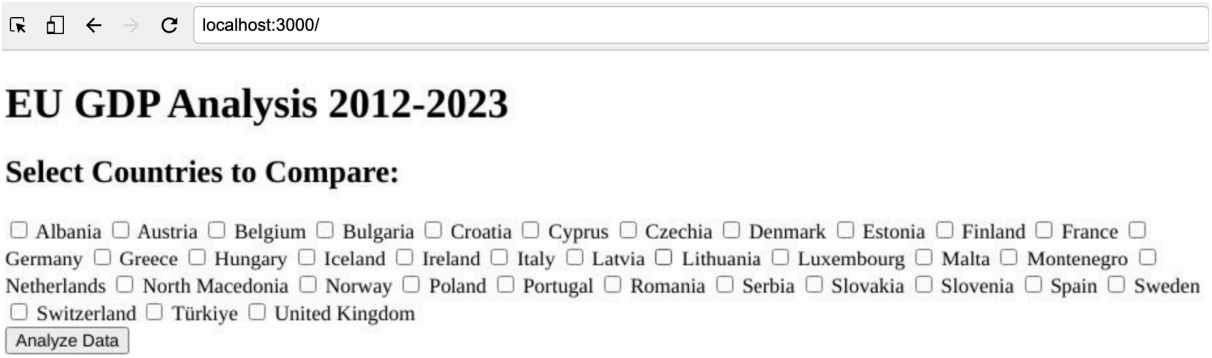
- Error handling for database operations
- Tabulated display of GDP data and growth rates
- Responsive design for better user experience
- Country selection interface for comparative analysis
- Clear separation of concerns between data access and presentation

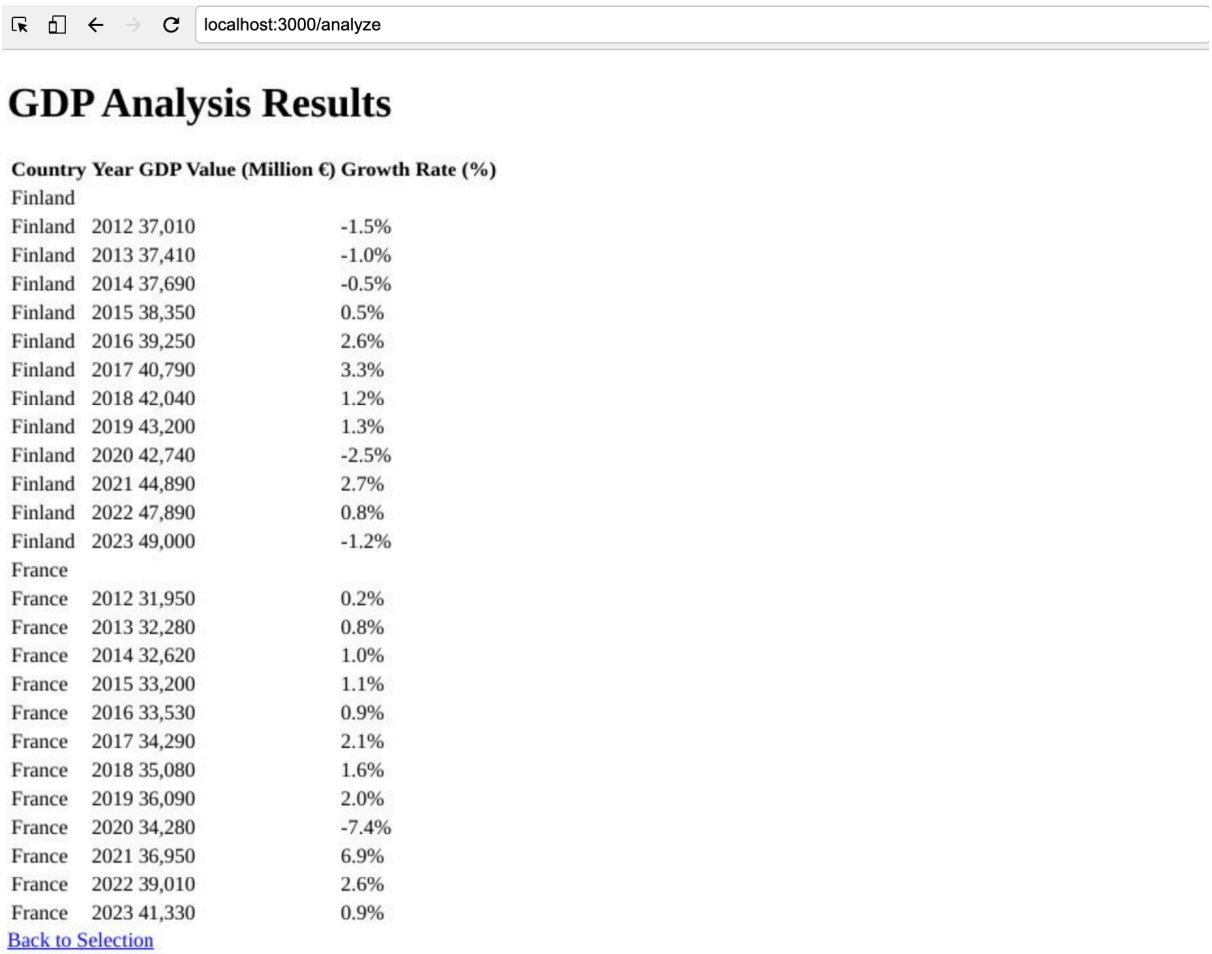
Implementation Highlights

```
app.post('/analyze', (req, res) => {
    const query = `
        SELECT
            c.CountryName,
            gt.GDP_Year,
            gt.GDP_Value,
            gg.Growth_Rate
        FROM Countries c
        JOIN GDP_Total gt ON c.CountryID = gt.CountryID
        JOIN GDP_Growth gg ON c.CountryID = gg.CountryID
            AND gt.GDP_Year = gg.Growth_Year
        WHERE c.CountryID IN (?)
        ORDER BY c.CountryName, gt.GDP_Year;
    `;
});
```

This implementation successfully meets the project requirements by providing an intuitive interface for exploring and analyzing the EU GDP data, while maintaining proper database security and data integrity.

Screenshots





References

- MySQL Node.js - https://github.com/mysqljs/mysql
- EJS Templating - https://ejs.co/
- Eurostat Database - https://ec.europa.eu/eurostat/data/database

- GDP at Market Prices (tec00001) - https://ec.europa.eu/eurostat/databrowser/view/tec00001
- Real GDP Growth Rate (tec00115) - https://ec.europa.eu/eurostat/databrowser/view/tec00115
- MySQL Documentation - https://dev.mysql.com/doc/