DATA:
https://www.kaggle.com/datasets/shushilshah/world-gdp-dataset?select=main_data.csv

---

ETL + SQL + Python (Beginner Version — 40 Questions)

1. Extract (Getting the data into Python)

1. Import `pandas` and load `main_data.csv`, `metadata_country.csv`, and `population.csv`.
2. Print the number of rows and columns for each file.
3. Show the first five rows from each DataFrame.
4. Check column names and datatypes in each DataFrame.
5. Find how many unique countries exist in `main_data.csv`.
6. Display all indicator names available in `metadata_country.csv`.
7. Identify missing values in each dataset.
8. Find all rows in `main_data.csv` where the year or value column is missing.
9. Check if every country code in `main_data.csv` appears in `population.csv`.
10. Print all indicator codes that are in `main_data.csv` but not in `metadata_country.csv`.

---

2. Clean (Fixing messy data)

11. Replace any ".." or empty strings in the value column with `NaN`.
12. Convert the "value" column to numeric type (float).
13. Remove duplicate rows from `main_data.csv`.
14. Trim extra spaces from all text columns.

15. Standardize country names (for example, change “Kenia” → “Kenya”).
16. Remove rows with invalid years (not between 1960–2023).
17. Drop any columns that are completely empty.
18. Fill missing country codes using matches from country names.
19. Save the cleaned version of each DataFrame to a new CSV file.

---

3. Transform (Making the data useful)

21. Merge `main_data.csv` with `population.csv` to add region and income level.
22. Merge the result with `metadata_country.csv` to attach indicator descriptions.
23. Create a new column showing GDP per capita change from the previous year.
24. Group the data by region and calculate the average GDP for 2020.
25. Create a column `continent` using a dictionary that maps each region to a continent.
26. Sort the data by country and year.
27. Filter and keep only data for years 2010–2020.
28. Create a small summary table showing the number of indicators per topic.
29. Round all numeric columns to two decimal places.
30. Export this final cleaned dataset to a file called `world_bank_cleaned.csv`.

---

4. Load (Into PostgreSQL using psycopg2)

31. Connect to PostgreSQL with psycopg2 and create a new database called `worldbank_data`.
32. Write SQL code in Python to create tables:

- `countries (country_code, country_name, region, income_level)`

- `facts (country_code, indicator_code, year, value)`

33. Insert rows from the cleaned DataFrames into these tables.
34. Use a transaction so that if one insert fails, everything rolls back.
35. Add a primary key to each table.
36. Write a Python function that loads data from a CSV file into a table automatically.
37. Verify that each table has the right number of rows after loading.
38. Print a confirmation message once all data is successfully loaded.
39. Create a simple log file that records when each ETL step ran.
40. Close the database connection safely at the end of the script.

What this teaches:

- Reading and exploring data in pandas
- Cleaning and merging datasets
- Running SQL commands from Python
- Loading and verifying data in PostgreSQL
- Writing clean, modular ETL code
