Contents

[Database Applications Development Final Project 2](#_Toc132623745)

[Introduction 2](#_Toc132623746)

[Conventions 2](#_Toc132623747)

[Description 2](#_Toc132623748)

[Technical Requirements 4](#_Toc132623749)

[Deliverables: 4](#_Toc132623750)

[Analysis (POC) – April 25th 4](#_Toc132623751)

# Database Applications Development Final Project

## Introduction

On this project, students will have the opportunity to demonstrate the skills developed during the session by building an application that extracts, transforms, and loads data in a Data warehouse for further use in a Data plotter application. Once the data is “production-ready” it will be used to generate personalized plots that can be visualized and stored for further use.

## Conventions

For the purpose of this proposal (project) the word “client” is used to represent GreenLab and the “consultants” is used to represent the students members of each group.

## Description

The source of data for this project comes from public sources such as [Our World in Data](https://ourworldindata.org/co2-and-greenhouse-gas-emissions). Currently the client is downloading the files required and storing them in a folder on the web server in csv format then, the client opens the files in MS Excel, performs the data transformation in excel and generate some plots that are exported as jpeg files to be later published in a static website. As they are becoming a global organization with clients all around the world the client wants to build a dynamic website where their users can generate personalized plots based on the information collected from public sources.

The following table describes the name of the files that are provided by the client with a short explanation about the information included on each file:

|  |  |
| --- | --- |
| File name | Description |
| annual-co2-emissions-per-country.csv | This file includes the annual emission (in tons) of CO2 per country until 2021. Notice that not all the countries start the data collection in the same year and some years are missing |
| nitrous-oxide-emissions.csv | This file includes the annual emission (in tons) of nitrous oxide per country. Notice that not all the countries start the data collection in the same year and some years are missing |
| methane-emissions.csv | This file includes the annual emission (in tons) of methane per country. Notice that not all the countries start the data collection in the same year and some years are missing |
| population-and-demography.csv | This file includes the population per year and country as well as other interesting demographic information |
| gdp-per-capita.csv | This file contains the GDP per capita, per country until 2021 (Not all the countries have data for all years) |

With the information included in these files, the client wants to build a “Data explorer” website that can provide a graphical view of the data (dashboards) with the purpose of create conscience on the population about the impact of our emissions in the environment. As example, the client provided these two graphs:

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

To generate the graphs a web interface like this need to be created:

<https://ourworldindata.org/explorers/co2>

Graphical user interface, chart

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Title | Gas | Count | Relative to world total |
| 1 | Annual CO2 emissions | CO2 | Per country | not checked |
| 2 | Annual share of global CO2 emissions | CO2 | Per country | checked |
| 3 | Methane emissions | Methane | Per country | not checked |
| 4 | Nitrous oxide emissions | Nitrous oxide | Per country | not checked |
| 5 | Per capita CO2 emissions | CO2 | Per capita | not available |
| 6 | Carbon emission intensity of economies | CO2 | Per $ of GDP | not available |
| 7 | As you wish | As you wish | As you wish | As you wish |
| 8 | As you wish | As you wish | As you wish | As you wish |

NOTE: Options not described in this table should not be available to the user (validation)

## Technical Requirements

1. Python must be used as the primary language of development unless strictly necessary.
2. Data should be first explored then extracted, transformed, and loaded for this purpose you can use the jupyterlab environment with pandas, NumPy and matplotlib libraries.
3. Notice that some new columns need to be calculated and added to the database table. Ex.: Per capita CO2 emissions = CO2 Emissions per country/Population
4. Once the data is “production ready” you can upload to a local database in your computer during the analysis stage (POC) but for the final presentation ideally this production data should be stored in a Data Warehouse (such as AWS Redshift)
5. Your ETL process (data pipeline) should run once per week and upload any new data to the Data warehouse.
6. To plot the requested graphs, you can use matplotlib but other eye candy tools such as [seaborn](https://seaborn.pydata.org/index.html) are highly recommended.
7. The plots do not need to be dynamic as in the web site presented as example. Instead, you can select the options required and click on a plot button to generate the graph.
8. A historical archive of all the previous graphs should be kept and included as a page on your website.
9. To test the data pipeline, we can delete in all the csv files the information relative to one of the countries and run the pipeline to update the data warehouse with the “new information”.

## Deliverables:

### Analysis (POC) – April 25th

* An analysis document that demonstrates the consultants have explored the technologies available in the market to perform the data extraction, transformation, and loading. As the conclusion of this analysis the consultants should provide a recommendation of the suite of technologies to be used in the project (for data processing). Cloud solutions are welcome!
* A proof of concept that meets the following criteria:
  + The application extracts, transforms and loads the data locally (RDBMS on PC).

1. The ETL process is automated, and it runs the pipeline once per week or when required.
   * The application can plot all the required plots locally (on jupyterlab) using production-ready data (not raw data)
2. All the functions can be used outside the jupyterlab environment (as python scripts)
   * The method that can be used to receive the requirements and return the plots is clearly identified as described.

### Final submission and presentation

* A final project presentation that includes (a template will be provided next week):
  + Team description (members, roles, and skills)
  + Description of technologies used.
  + More to come…

Pandas, matplotlib, seaborn, redshift