Part 2 Report

Below are the steps to build the data pipeline to transfer data from ClimateWatchData.org dataset to our Country Climate and Development Report (CCDR) dataset.

1. We will need to study the data and metadata schemas for both the sources. We could define the mapping between source and destination fields using bridge tables (a popular concept used in OECD while collecting, centralizing, and disseminating the data from multiple data sources). These bridge tables could be created in XML or JSON format depending on the convenience and technology stack.

2. We will need to define the scope for data pipeline (like batched pipeline for analytics which is not time sensitive vs streaming pipeline for analytics which is time sensitive, ETL (Extract, Transform, Load) vs ELT (Extract, Load, Transform) depending on the choice for additional storage and need for raw data, aggregating techniques like direct summation vs simple average vs weighted average vs moving average for relevant fields). Since the ClimateWatchData.org data is mostly alphanumeric and structured, we could use any of the existing database options to store the data. However, if data is unstructured or is in variety of formats like images, videos, etc. then we can use NoSQL databases (like MongoDB) to store it.

3. It will be also a good idea to define data classification levels (public, for office use only, confidential, etc.) at desired levels (like at country, series, dataset levels). We can handle this using the bridge tables we have created in step 1.

4. The scope from above step will help us to define and/or understand existing technology stack at destination database.

5. Once we have finalized the technology stack for destination, we will define the connections to source database. Since ClimateWatchData.org database allows an API based connection, it would be our best bet to use APIs to extract data and metadata from source database.

6. After the data is extracted, we will need to build some exploratory analysis to ensure the data quality and validations. This could be achieved through some exploratory visualizations build as part of the pipeline or a document like report to list basic statistical properties like minimum, maximum, mean, standard deviation. This will help us to identify outliers, prepare any additional footnotes, etc.

7. Using the already defined bridge tables, we will start transforming the data and metadata from source schema to destination schema.

8. One of the important data wrangling tasks would be map the country names and code from ClimateWatchData.org data dictionary to World Bank data dictionary.

9. We will add important fields (like Income Group, World Bank Regions, etc.) for additional analytics during data transformation and wrangling stage.

10. We will transform the extracted and wrangled data into destination schema so that it can be uploaded into dataset.

11. The last step will be to upload the data into destination schema using techniques like API based upload, SQL insert based upload, etc. Here the assumption is that analytics and dissemination platforms (like Tableau, Power BI, databank.worldbank.org, etc.) are already connected to destination dataset to disseminate the data to targeted users.

12. Depending on the need for periodic updates, we could either decide to schedule the pipeline job or we could run it manually.

Based on my experience, I will prefer to use Python to create the desired pipeline with bridge tables defined in JSON or XML format allowing developers to easily make changes into data mapping, data confidentiality levels, etc. Depending on the technology stack, we can either upload the data from Python using APIs or SQLs or we could export the final dataset into a file and can be uploaded into destination database easily.

I have used ETL approach (instead of ELT approach). The whole pipeline can be automated by means of any software with scheduling facility along with email support to share the status of the task after the scheduled task is completed.

Pipeline concept diagram: