**Assumptions:**

* In the table schema, \_sk field refers to the surrogate key(auto-generated primary key except for dim\_date) and \_bk refers to the business key. (Please follow this link to look at the data model. <https://github.com/nisheshk/Data-Engineering-Assessment/blob/main/data_model/data_models.png>)
* I could have stored the population metric in the dim\_location table, which would make it a type 1 SCD. That is when we get the new population metric for that city, we would replace the current population with the new population. However, I decided to store it in the fact table so that we can have the history of the population data. I am assuming here that we receive this file every year where we would receive the new population for that year. Taking this into consideration I have created the script load\_fact\_country\_demography
* For dim\_weather\_station table, I have assumed that it is a type 2 SCD. There are some attributes in this table that can change over time. For example: Station\_name could change. Also, if the weather station equipment is moved to a new place, it’s going to have new lat and long. It may be crucial to consider it as type 2 SCD considering the fact that it could have an impact in the reporting layer. So, I have introduced new fields valid\_from, valid\_to and latest\_record\_flag. To summarize, we keep the history of the changes of the station and use the surrogate key (auto-generated primary key) in the dim table which is also used in the fact table to uniquely identify the station.
* The id in the fact table points to the surrogate key in the dim table.
* Also, I am assuming we run these jobs everyday with Truncate and load technique. However, if we want to change the load technique to append(incremental) then few changes are required in the code. Like reading the data where source\_data\_date > max\_date of the fact\_table. Also, the code can be changed in a way it can identify the load technique from the config file.
* My approach on relating 2 datasets was finding the closest city for the weather station. As we already have the province field in both datasets, it narrows down our search for the city. So, the formula I have used to find the distance between city and station is as below:

AVG(abs(abs(City\_long ) – abs(Station\_long)) - abs(abs(City\_lat ) – abs(Station\_lat)))

Here, I am assuming that we will just have one city that has the least value for the above formula.

* I have exposed the GCP CloudSQL database credentials and API token in the code. Because if I set it as an env variable, it’s not going to run on your end.