Weatherapi

https://www.weatherapi.com/docs/

La API elegida para realizar el trabajo integrador del curso de Ingeniería de Datos de Coder House fue Weatherapi. Es una API que brinda información meteorológica del mundo ya sea actual como histórica (entre otras opciones). Uno puede buscar por región, ciudad, país, etc y por fecha que quiere visualizar para asi ver los distintos parámetros del clima que se presentan en la ubicación y momento deseado.

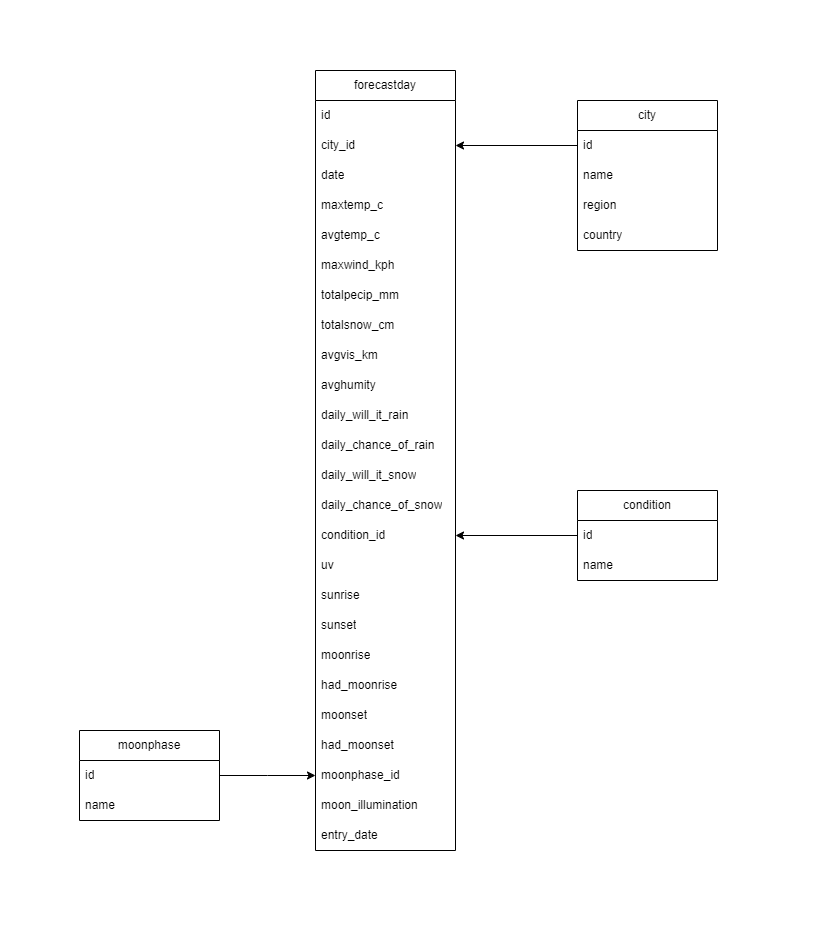
Utilizando Python, con la siguiente URL pude obtener los datos que buscaba, en un formato json, y haciendo uso de la API Key y ciertos parámetros definidos en el código:

[http://api.weatherapi.com/v1/history.json?key=d8cb833dde7a48daa8b131011242802&q={item["name"]}&q={item["region"]}&q={item["country"]}&dt={fecha}](http://api.weatherapi.com/v1/history.json?key=d8cb833dde7a48daa8b131011242802&q=%7bitem%5b%22name%22%5d%7d&q=%7bitem%5b%22region%22%5d%7d&q=%7bitem%5b%22country%22%5d%7d&dt=%7bfecha%7d)

La API tiene un prueba gratuita por unos días que permite obtener los datos del ultimo año, por lo que haciendo uso de esa opción, utilice al API Key para hacer un barrido del clima de todos los días de los últimos 365 días de un listado de ciudad que elegí (tanto de las provincias argentinas como de algunas ciudades del mundo) y lo guarde en un archivo Excel.

Mi idea cargar el archivo Excel tal cual lo tengo en una tabla en la base de datos en Redshift, para luego realizar un script que se encargue de modelar un poco mejor la información según el siguiente diseño:

(actualmente esta faltando una tabla, que permita visualizar la misma información que forecastday, pero por hora)



**Entregable Actual:**

* Weather.py: script encargado de tomar todos los registros de los 365 dias anterior y cargarlos en un Excel
* Weather-redshift.py: script que se encarga de subir los datos que están en el Excel a tabla de Weather que se encuentra en mi esquema de redshift.
* Configuración.py: datos de conexión a la base
* Weather: registro diario de los últimos 365 dias de las ciudades seleccionadas.
* DER.png: imagen del diagrama de entidad relacion actual.

**Carga de datos en la base:**

Condition:  
Texto

Descripción generada automáticamente

Moonphase:

Texto

Descripción generada automáticamente

City:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Para la tabla Weather, tanto el CREATE como el INSERT se encuentran dentro del archivo weather-redshift.py

\*\*pendiente la tabla forecastday\*\*

**Para siguientes versiones:**

* Modificar el script para tomar la información por hora del json.
* Crear una tabla en la base para contener la información diaria segmentada por hora.
* Hacer un script que se encargue de la carga de los datos de manera semanal (el máximo de días permitido por la API en su versión gratuita). Este script, a diferencia del actual, usaría las tablas auxiliares para tomar el ID de los campos moonphase y condition de manera que la carga sea directa y no haya varios pasos como ahora.
* Borrar la tabla weather ya que solo se usó para una carga inicial de forecastday.