

README

TimezoneDB API, Python Assessment - Full Stack Developer @ RCG

Creating a Python script to fetch data from TimezoneDB API, populating TZDB_TIMEZONES and TZDB_ZONE_DETAILS tables. Errors during API retrieval are logged into TZDB_ERROR_LOG. TZDB_TIMEZONES is cleared before each run. TZDB_ZONE_DETAILS is updated incrementally. Stage table approach for TZDB_ZONE_DETAILS.

Installation:

- Install vsCode for executing timezone.py file
- Install MySQL to set up the DB and create the database and respective tables for implementation
- Install Postman to manually check the functionality of the API

Explanation:

- After installing MySQL, create the timezone_DB table.
 - Use Query, CREATE DATABASE timezone_db;
- Then create four tables, TZDB_TIMEZONES, TZDB_ZONE_DETAILS, and TZDB_ZONE_DETAILS_STAGE, and TZDB_ERROR_LOG .
 - TZDB_TIMEZONES: All the records from the API are stored in this table.
 - TZDB_ZONE_DETAILS_STAGE: To receive the data from API which is filtered and is temporarily store in this table, only to further populate into TZDB_ZONE_DETAILS table
 - TZDB_ZONE_DETAILS: Only the filtered details are populated into this table
 - TZDB_ERROR_LOG: To log the errors while retrieving data from the API and to handle them.
- Filtered: Understanding from the given question, ZONEEND cannot accept NULL values. However, the data in the API contains values where ZONEEND is 'null'. To avoid inserting those values into the table, I skipped them and only entered those with proper NUMBER values for ZONEEND.
- Used timer.sleep(2) to handle data receiving from the API; else, it gives a REQ ERR 429
- After executing the timezone.py file you'll get a message saying "Data moved from staging table to main table successfully" and "Removed data from staging table successfully"
This means that the data has been moved from the staging table to the main table (TZDB_ZONE_DETAILS) and the staging table (TZDB_ZONE_DETAILS_STAGE) records have been deleted.
- Once the file is executed successfully, go to MySQL and run a query to check the TZDB_ZONE_DETAILS.
- Run a query to check if the TZDB_ZONE_DETAILS_STAGE is truncated and no records are present in it.
- Run a query to check error logs in the
- If you run the timezone.py file again and check the DB, records are updated (Check timestamp) if any and no duplicates are added.

Queries used to create tables:

- For TZDB_TIMEZONES:

```
CREATE TABLE TZDB_TIMEZONES (  
  COUNTRYCODE VARCHAR(2) NOT NULL,  
  COUNTRYNAME VARCHAR(100) NOT NULL,  
  ZONENAME VARCHAR(100) PRIMARY KEY,  
  GMTOFFSET INT,  
  IMPORT_DATE DATETIME,  
  PRIMARY KEY (ZONENAME)  
);
```

- For TZDB_ZONE_DETAILS:

```
CREATE TABLE TZDB_ZONE_DETAILS (  
  COUNTRYCODE VARCHAR(2) NOT NULL,  
  COUNTRYNAME VARCHAR(100) NOT NULL,  
  ZONENAME VARCHAR(100) NOT NULL,  
  GMTOFFSET INT NOT NULL,  
  DST INT NOT NULL,  
  ZONESTART INT NOT NULL,  
  ZONEEND INT NOT NULL,  
  IMPORT_DATE DATETIME,  
  PRIMARY KEY (ZONENAME, ZONESTART, ZONEEND)  
);
```

- For TZDB_ZONE_DETAILS_STAGE:

```
CREATE TABLE TZDB_ZONE_DETAILS_STAGE (  
  COUNTRYCODE VARCHAR(2) NOT NULL,  
  COUNTRYNAME VARCHAR(100) NOT NULL,  
  ZONENAME VARCHAR(100) NOT NULL,  
  GMTOFFSET INT NOT NULL,  
  DST INT NOT NULL,  
  ZONESTART INT NOT NULL,  
  ZONEEND INT NOT NULL,  
  IMPORT_DATE DATETIME,  
  PRIMARY KEY (ZONENAME, ZONESTART, ZONEEND)  
);
```

- For TZDB_ERROR_LOG:

```
CREATE TABLE TZDB_ERROR_LOG (  
  ERROR_DATE TIMESTAMP,  
  ERROR_MESSAGE VARCHAR(1000)  
);
```