Extract Load Process

This is the first phase of my Business Intelligence Extract-Load-Transform(ELT) Architecture for Data Analytics. The aim of this phase is the integration of data from three different sources, specificallly: Yelp, Google Drive and Hetzner Storage-box. Before detailing main components of this phase, i will briefly describe the Extract-Load-Tranform process and how ELT is different from it. I will also briefly describe the terms data integration/ingestion, data warehouse(DWH) and data pipeline.

Key Definitions

1. Data Warehouse (DWH) This is a central repository suitable for persistently storing large amounts of structural data in an agreed format. To store data in this central repository, ETL-Tools are used to implement a well designed ETL-Process.

2. Extract-Transform-Load Process: This framework is made up of three phases i.e Extraction, Transformation and Loading. In the extraction phase, heterogeneouse data is extracted or collected from several different sources(e.g social media, cloud storage etc), while the transform phase takes care of *data filtering, harmonization, aggregation and enrichment to ensure that the extracted data is converted to a homogeneous agreed format expected by the central repository. Example transformations include: setting right data types of the individual fields, eliminating duplicates, removing unwanted characters and sometimes manually correcting values. The last phase is loading where data is physically written to the target repository.

3. Extract-Load-Transform: This framework is different from ETL in that, the Loading phase is done immediately after the extraction phase and the added value of this framework lies in the possibility to transform data based on Analytical need i.e Transformation is done when data is needed. This framework is suitable for building data pipelines. In this project, python and dbt-core with postgresql adapter will be used to build ELT data pipeline.

4. Integration/Ingestion In practice, data integration is not different from data ingestion, i.e they are both implemented together. The only thing to note about data integration is that it takes care of heterogeneous data by converting these into an agreed format.

These terminologies are used in ETL and ELT, and some tools that can be used for data integration include, among others, python, power automate, Apache Airflow etc.

Data Source and Data Description

Knowing data sources is a basic requirement for data extraction and understanding incoming data is very crucial for data ingestion/integration. Beside the business need, developers require knowledge of incoming data to determine whether data will be extracted fully or incrementally. It is equally usefull when it comes to loading and scheduling. In many companies, it is common practice to store files or data in cloude storage solutions. Learning how to access cloud environments is a skill commonly required in Business Intelligence or Data Analytics. For this reason, i will use Yelp, Google Drive and Hetzner to demonstrate how they can be used as data sources for data extraction.

Yelp

Yelp is a platform that provide *crowd-sourced* reviews of businesses. It offers several endpoints that allow developers to query data about businesses. This can be done with Yelp Fusion or GraphQL API. Yelp Fusion API has several endpoints which allow developers for make upto 500 data requests on a daily using the free tier. To use Yelp's APIs, developers must create an accounton the developer portal. Once an account is created, the developer must create an application, to obtain an API key, on the developer portal.

Additionally, the application must be authenticated to allow the developer make queries programmatically using this API key. For this project i used the Yelp Fusion API, basic tier, to extract restaurants and their reviews. By default, this API decides the number of reviews it returns for a given restaurant. The extraction was done with a single request to avoid hitting the maximum number of request permited by the basic subscription. These restuarants or business are predominatly located in the state of Nord-Rhein-Westfalen(NRW) and in the city of Hannover in the state of Niedersachsen.

Google Drive API

Google Drive API is a cloud solution that enable developers to interact with files stored on google drive. With this API, developers can *upload, download, share, and manage* files stored in Drive. To use this, developers must create a developer account, create a project, and an application that will provide them with secrets necessary for programatically creating queries to interact with files on Drive.

For this project I use my private google drive account to extract production and sales records provided to me by a fictive company called **Local Baker**. The production data is stored in an excel file while sales data is stored in a csv file.**Icrosoft Sharepoint** is an alternative cloud storage solution, create for file management.

Hetzner

This is an online company and data center operator based in Bavaria, Germany. Among the services stack offered by Hetzner are cloud storage boxex which can be used to manage files. A storage box can be accessed using electronic devices like tables, computers etc, and can also be accessed via other means including: FTP/FTPS, SFTP/SCE/WebDAV etc.

Local Baker is a fictivebakery located in city of paderborn and well known for its quality flour products. However they were forced to close for some years because they lost their market shares to competitors who produced in large quantities and sold at a cheaper price although the quality of competor's product was comparably lower. Local Baker is considering reopening and would like to make this decision based on data. They *found production and sales figures* recorded in the past and have provided these records via a hetzner storage-box for analysis. The data is an excel workbook with two tabs. One for production figures and the other for sales records.For demonstration, i will use my subscribed Hetzner storage-box.

PosgreSQL

PostgreSQL is a highly available ACID(Atomicity, Consistency, Isolation, Durability) compliant and free open source object oriented relational database management system.It runs on Windows, mcOS, Linux, BSD and Solaris and supports Programming languages including Java, Python, Ruby, Perl, C++ etc. The query language in PostgreSQL is SQL with wide support for complex queries, subqueries, joins, window functions and common table expressions (CTEs). This DBMS supports custom and primitive datatypes (e.g Numeric, string, Boolean, Date/Time etc.). PostgreSQL is multi-model in structure which allows it to support other data structures e.gDocument(e.g JSON/BSONB), Key-Value(Hstore). For this Project, PostgreSQL is used as the database management system for storing the extracted and transformed data.

Data Extraction and Loading with Python

As part of the data integration process, I have written python Scripts to extract data of different formats and types from Yelp, my Google Drive Folder and my Hetzner Storage-box. Since PostgreSQL only accepts data that are in relations or tables, I perform a few data integration steps which convert the received data into a homogeneous format/schema ready for data ingestion. The data integration steps applied are:

- Transformation: Converted json data to pandas dataframes which has a table-like structure, thus suitable for ingestion in PostgreSQL.
- Standardization: All columns of the data extracted were converted to string datatype before data ingestion. This is to allow the real datatypes of the columns to be determined in the transformation phase of the ELT process.
- Privacy: Authenticated the application required to access Yelp Fusion endpoint as well as the application required access my private google drive. All my database credentials, client secrets and API key are stored in a **.env** file to avoid exposing my avoid exposing my personal identifiable information on github.

Python Scripts Descritption

- logger.py: Used to create a logging file that records all the necessary steps during the data integration/ingestion process.
- postgres_conn: Connects python to PostgreSQL
- postgresops.py: Functions for converting dataframe columns to string types as well as functions for checking schema and data table availability in PosrgreSQL are defined here. Also, functions for data ingestion set here.
- yelp_restaurants_extract.py: Functions for connecting to and retrieving data from yelp fusion endpoint. Functions for converting the retrieved files to dataframes are equally defined here.
- driveextract.py: Functions for connecting to my personal google drive to retrieve and/or relocate files from a target folder are defined here.
- convert_driveextract_to_df.py: Functions for converting google drive files to pandas dataframes are set here.
- hetznerbox_extract.py: Functions for connecting and retrieving data files from my hetzner storage-box are defined here.
- convert_hetzner_box_extract_to_df.py: Functions for converting retrieved data from files to pandas dataframes are set here.

Another important file worth noting is the **requirement.txt** which contains all the python libraries used during the implementation of the Extract-Load process. To install these libraries in the python development environment(.venv), first of all activate the development environment (on windows use cmd) then type the following command: pip install -r requirement.txt.