

B9DA100 Programming for Data Analysis, Processing and Visualization

Name: Pavan Gaikwad

Roll No: 10575449

CA2

**Technical Document for Python ETL Solution Design & Implementation**

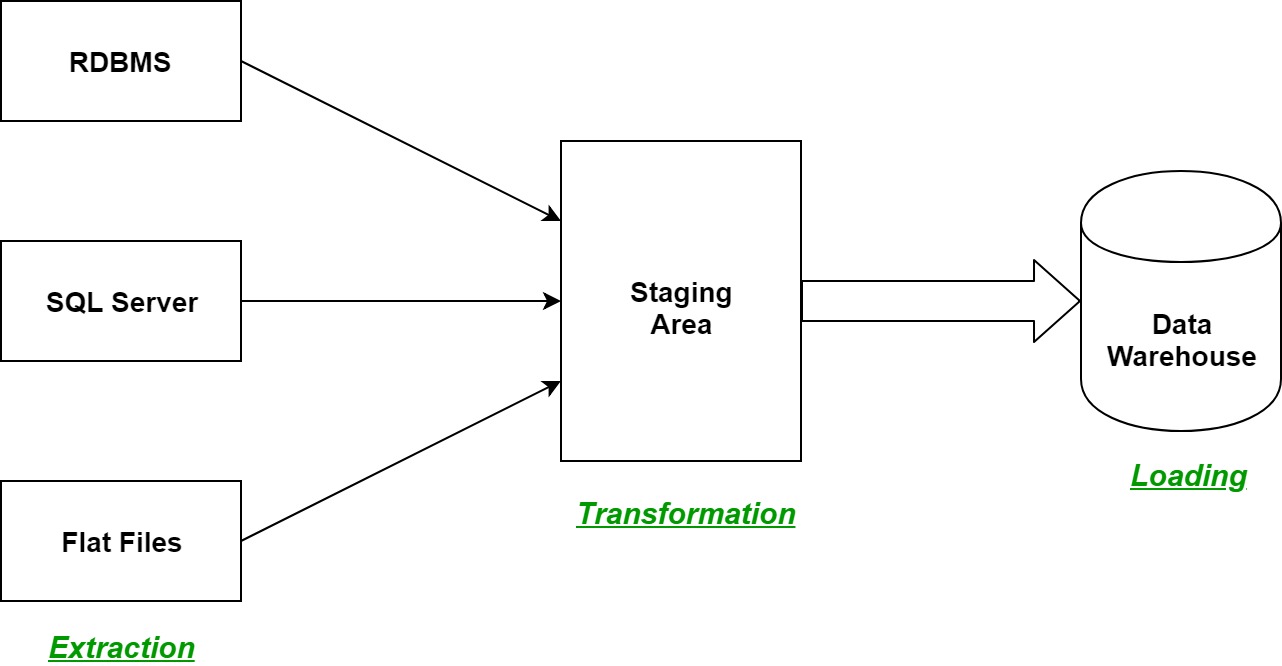
1. Scope of the document:

Python is an elegant, versatile language with an ecosystem of powerful modules and code libraries. Writing Python for ETL starts with knowledge of the relevant frameworks and libraries, such as workflow management utilities, libraries for accessing and extracting data, and fully featured ETL toolkits. pandas is an accessible, convenient, and high-performance data manipulation and analysis library.

Coding the entire ETL process from scratch is not particularly efficient, so most ETL code ends up being a mix of pure Python code and externally defined functions or objects, such as those from libraries mentioned above. For instance, users can employ pandas to filter an entire Data frame of rows containing nulls.

2. Technical Design:

ETL Architecture:



Extract:

ETL process involves extracting the data from the source system(s). Common data-source formats include [relational databases](https://en.wikipedia.org/wiki/Relational_database), [XML](https://en.wikipedia.org/wiki/XML), [JSON](https://en.wikipedia.org/wiki/JSON) and [flat files](https://en.wikipedia.org/wiki/Flat_file_database).

Transfer:

In transformation, variety of data is processes and store in same format. In data processing, filtering, cleaning, joining, splitting and sorting task involves.

Loading:

After the transformation, data is uploaded in data mart or data warehouse. The rate and period of loading is depend on business requirement.

**Pandas operations detailed for each requirement.**

#Number of non nul rows

print("The number of non null rows (by column)\n\n",data.notnull().sum())

#The number of null values for all columns

print("The number of null values for all columns are",sum(count))

#The total number of call outs by Station Area

print(data.groupby(['Station Area']).count())

#The total Number of call outs by Station Area and Date

print(data.groupby(['StationArea','Date'])['StationArea'].count())

**Data Model:**

Extracted CSV File FireBrigadeAndAmbulanceCallOuts.csv

Import Pandas

Data Cleaning

Loading to SQLite3 server

In this assignment, the extracted FireBrigadeAndAmbulanceCallOuts.csv file downloaded from data.gov.ie. This file contains details of Fire and Ambulance annual incidents between 2013 and 2019. Then by using jupyter notebook, we have imported pandas and performed operations for data cleaning which mentioned in Pandas Operations. The cleaned data have loaded on sqlite3 server and cross checked whether proper data is uploaded or not.

FireBrigadeAndAmbulanceCallOuts.db database is created in SQLite3 server and table is created by name result. The all data is stored in result table.

3. Testing

Text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Table

Description automatically generated with medium confidence

4. Reflection on Learnings:

I completed the cleaning steps that were needed for data processing and realized how important they were. The phase of integrating Python with SQL server has taught me about database concepts. The phase of integrating Python with SQL server has taught me about database concepts. Furthermore, since the source file can come from several sources, we must translate and combine the data into a single format so that we can understand all of the various sources, data formats, and how to extract the data.

5. References:

<https://www.geeksforgeeks.org/etl-process-in-data-warehouse/>

<https://www.w3schools.com/python/pandas_tutorial.asp>

<https://stackoverflow.com/questions/tagged/pandas>

<https://www.sqlitetutorial.net/sqlite-python/>

https://likegeeks.com/python-sqlite3-tutorial/