ETL and Business Analytics for given Raw data

Pruthvi Kotnani
Department of Computer Science and Engineering
R.V. College of Engineering
Bangalore, India
pruthvik.cs17@rvce.edu.in

Asst Prof. Anitha Sandeep
Department of Computer Science and Engineering
R.V. College of Engineering
Bangalore, India
anithasandeep@rvce.edu.in

-------------------------***********************--------------------

Abstract:

ETL - Extract, Transform and Load, is a process of allowing businesses to create an automated process of taking the raw data from the source database and transforming. Once transformations are done load it at the destination ODS. System tries to automatise configuration files using the target api structure which can be used to control the functionality of the system and generate the required business analysis. Airflow is used to create DAGs for flow of execution, which can be used to store and analyze the data. By the end of execution, the configuration files for the required domains are generated and stored in the database. A dashboard is created from the Database data to make a meaning sense to the organisation.

Keywords — **ETL, ODS, JSON, Airflow, BI**

-------------------------***********************--------------------

I. INTRODUCTION

This document aims to brief about the way to design and architect Data Engineering project to automate the flow of data from simple CSV files to an API compliant Operational Data Store.

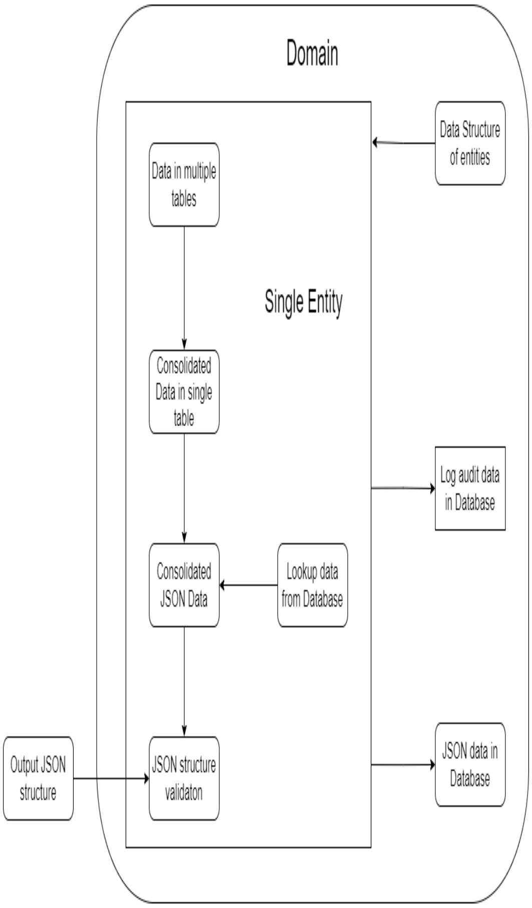
ETL and Business Analytics for given Raw data is inspired from the pain area of not able to automate the process of loading a raw data from a source to fully organized Operational Data Storage (ODS) and providing meaningful business insight for the organization to improve its overall performance.

This paper will include generation of configuration file for the data, create validations for the data to load it in the ODS using API. Create parallel programming to increase the efficiency. Create a dashboard to make meaningful understanding of the data obtained whose parameters can be changed according to need.

II. DESIGN

The process begins by taking the raw CSV files. The domain structure is such a way that there are multiple CSVs (tables) in each domain which are connected with a unique key. By using pandas in python, we consolidate the tables in to a single python dataframe. We then use the consolidated tables (dataframes). Using this dataframe we generate the JSON payload for the API. While we are generating the payload, we need to add few descriptors for which we need to do a lookup from the table on the database.

While creating the payload for the API to post it in ODS, we log the error data in a different environment (database connection). Before posting the data into the API we validate the payload with target JSON structure. Once the validation is successful, we post it to the ODS.



Once the data is settled in the ODS, we use Business Intelligence tools like tableau and Power BI to analyse the data and prepare the dashboards with proper segregation.

III. IMPLEMENTATION

The raw CSV files(data) are obtained from the AWS S3 bucket. The Python code will do create a backup for the file in the PostgreSQL database. The files then, according to the entity and domain are processed. The files processed are converted to JSON and stored in Payload Datastore. The error files which did not clear the validations are logged in error logs and Bad Data tables. The data from the table are then loaded to PostgreSQL with an API. The Loaded data is then used for various analysis. The raw CSV files(data) are obtained from the AWS S3 bucket. The Python code will do create a backup for the file in the PostgreSQL database. The files then, according to the entity and domain are processed. The files processed are converted to JSON and stored in Payload Datastore. The error files which did not clear the validations are logged in error logs and Bad Data tables. The data from the table are then loaded to PostgreSQL with an API. The Loaded data is then used for various analysis.

A. Software Configuration

1) OS: Windows/Linux/MACOS
2) Tool: Anaconda and Visual Studio Code
3) Languages: Python,SQL
4) Database: PostgreSQL
5) Airflow

B. Design Constraints

Entities are processed in a parallel manner but uploaded to the API in a hybrid manner.

The proprietary API accepts certain entities without any dependency issues, while certain entities can only be loaded if their dependencies are loaded first.

Loading criteria and uploading order is mapped in a configurable JSON file that is used to drive the code used to upload into the API.

C. Interfaces

User Interfaces of the system

The User Interface of the system has the following:

- The key graphs to analyse and quantify the end users' desire.
- Real-time data from Database to provide analysis.
- Interactive graphs which give clarity on individual entity and business needs.

Software Interfaces of the system

The Software Interfaces include the system execution in the airflow, where the data is divided into batches and the python code is run in parallel.

D. Software Testing

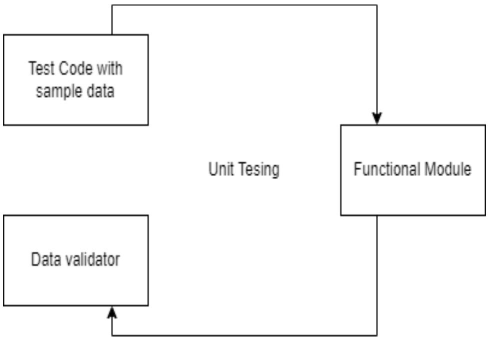
Test Environment

Testing is done in 3 phases:

1. Initial unit tests are done in local machine using local database copies.
2. Then integration testing is done in a cloud environment provided by the client.

3. In last phase, system testing is done in a production like environment.

Unit Testing



1. Unit testing is done for each helper functions and the corresponding stage that is subsequently used
2. Sample data is stored in local database copies from where they're fetched and tested against.

Integration Testing

1. Integration testing is done in a cloud environment with a sample set of data to test the data flow.
2. Each module is tested for their reliability and execution time.
3. Connections to the database and to the API is also tested in this phase.

System Testing

System testing is done on the full dataset.

Memory management and CPU throttling is monitored in this phase and a full system report is generated for the entire dataset.

Time of execution for each entity is monitored for end-to-end processing.

and the Cisco 3504 WLC (meant for smaller organizations that have a few access points). This paper discusses both on-premises and cloud-managed WLAN technologies.

Although both these approaches offer significant advantages, the choice of the best approach is determined by several factors, including the organization's structure, the proposed network design, the current network design, the wireless requirements of the organization, the budget, size of the WLAN, the ease of configuration management etc. Some of the leading on-premises and cloud-based WLAN products are also compared and analyzed in this paper, along with their advantages and disadvantages.

E. Related Work

A method for generating incremental ETL tasks (second type) from the initial load is presented in Reference [1] and is based on equational reasoning.

According to Reference [2], they are appropriate for ETL projects that already exist and want to switch to incremental mode and profit from it. Schema versioning is the focus of the study on DW evolution and modifications in the references [3], [4], [5], and [6].

The authors of this work provide a way to create a mapping between the sources and targets of the ETL process in references [12], [13], and [14]. Reference [23] outlines the commercial ETL that is being given, as well as some instances of the subject and another kind of ETL called open-source tools. In order to do research, Clinical and Translational Science Award (CTSA) grantees must develop research data marts from their clinical data warehouses.

ETL software7 is used to import data from local systems into an i2b2 project in accordance with [15] addresses. The process steps required to transfer data from one schema and representation to another are often specified in commercial and locally produced ETL systems. citation [16] The Results section more broadly provides statistics on all the significant projects in which Eureka! has been used to date. We are evaluating the adaptability of the metadata driven ETL process implemented by Eureka! in three scenarios that are either under development or completed. Each of these scenarios focuses on a single representative project within each of the three use cases.

citations [17, [18], and [19] Over 200 hospitals connected to US university medical centres are included in the UHC Clinical Database, which also includes de-identified administrative data and sparse clinical data.

Reference [20] explains how to construct an ETL process using the job submission interface of the webapp, which is partially seen in Figure 2. A data source adapter, an action to be taken with the data, a collection of concepts representing the data to load, and an optional date range are all specified by the user. Data source adapters set Eureka! up to retrieve information from a certain database or data file. Data is loaded into a target i2b2 project, database, or other data file using actions.

Model-based data transformation is offered by

Reference [21] technology and is relevant to data warehouse populating and updating. The target data warehouse model, the data models of the data sources, and the correspondence between them are all included in the metadata repository that serves as the foundation for the ETL process.

An introduction that focuses mostly on declarative metadata is given in reference [28]. Declarative and procedural metadata are necessary for the nature of the intended system behavior and may be effectively used to handle the metadata itself, for example to alter the information on the data source if monthly obtained file names change according to the date.

Technologies debate and describe the scenarios in which either of them should be chosen.

IV. Architectural Strategies

Programming Language

The entire software is built on python using various libraries like numpy, os, json, pytest and so on.

The back-end, that is, the database part is handled by PostgreSQL.

User Interface Paradigm

The power BI Model is based on ease of use. A Power BI custom visual is put into the BI report and then configured to showcase various metrics analysed in the Dashboard. The interface is interactive, that is the, it can give specific report on individuals, month. For example, the dashboard can show all, yearly analysis. This can be drilled down by clicking on a month were analysis would be changed to monthly report.

V. Conclusion

This ETL program can be integrated with any organisations, universities, etc ODS. The implementation supports creation of individual entity configuration files, backup of original data, validation of data, and loaded to ODS with respective API standards. This enables ease in logging the information of sales, profits, revenue, attrition log in case it’s a MNC or students’ information such as exams, roll number, etc. In addition to this, advanced interactive dashboards with detailed analysis of data can be created. This implementation is insightful, reliable and time saving as many of the maul Tasks are automated.

VI. Future Enhancements

This project may lead to automation of generating the dashboards. This will further reduce the human intervention. The Analysis metrics can be obtained as part of requirements gathering and can be programmed in such a way that the dashboard is created once the data is loaded to the ODS.

REFERENCES

[1] S.Sajida, Dr.S.Ramakrishna, ‘A Study of Extract–Transform–Load (ETL) Processes’, International Journal of Engineering Research & Technology (IJERT) , 2015.

[2] Martin Oberhofer, Albert Maier, Thomas Schwarz, Manfred Vodegel , ‘Metadata-driven Data Migration for SAP Projects ,’ IBM Germany - Research and Development GmbH.

[3] Petr Aubrecht,Zdenek Kouba, Metadata Driven Data Transformation, Geographical In- formation On-line Analysis (GOAL) research project, 2020.

[4] Andrew R. Post , Akshatha K. Pai ,Bradley J Metadata-driven Clinical Data Loading into i2b2 for Clinical and Translational Science Institutes ,’ PHS Grant UL1 RR025008 from the CTSA program.

[5] Petr Aubrecht. Sumatra Basics. Technical report GL–121/00 1, Czech Technical University, Depart- ment of Cybernetics, Technick ́a 2, 166 27 Prague 6, December 2000.

[6] Z. Kouba, K. Matouˇsek, P. Mikˇsovsky ́, and O. Sˇtˇep ́ankov ́a. On Updating the Data Warehouse from Multiple Data Sources. In DEXA ’98. Vi- enna, Springer-Verlag, Heidelberg, 1998.

[7] Godinez, M., Hechler, E., Koenig, K., Lockwood, S., Oberhofer, M., Schroeck, M.: The Art of Enterprise Information Architecture – A Systems-Based Approach for Unlocking Business Insight. Pearson, 1st Edition, 201Kiravuo, Timo & Särelä, Mikko & Manner, Jukka. (2013). A survey of ethernet LAN security. Communications Surveys & Tutorials, IEEE. 15. 1477-1491. 10.1109/SURV.2012.121112.00190.

[8] Leser, U., Naumann, F.: Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen. dpunkt Verlag, 1st Edition, 2006.

[9] Molina, J.-C., Pastor, O.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, 1st Edition, 2010].

[10] Collins FS, Hudson KL, Briggs JP, Lauer MS. PCORnet: turning a dream into reality. J Am Med Inform Assoc. 2014;21(4):576-7.

[11] Appel LJ. A primer on the design, conduct, and interpretation of clinical trials. Clin J Am Soc Nephrol. 2006;1(6):1360-7.

[12] Murphy SN, Weber G, Mendis M, Gainer V, Chueh HC, Churchill S, et al. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc. 2010;17(2):124-30.

[13] McMurry AJ, Murphy SN, MacFadden D, Weber G, Simons WW, Orechia J, et al. SHRINE: enabling nationally scalable multi-site disease studies. PloS one. 2013;8(3):e55811.

[14] Kimball R, Ross M. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2nd ed. New York: Wiley Computer Publishing; 2002.

[15] Skoutas and A. Simitsis. “Designing ETL processes using semantic web technologies”. Proceedings of DOLAP06, 2006.

[16] D. Skoutas and A. Simitsis. “Ontology-based conceptual design of ETL processes for both structured and semi-structured data”, International Journal on Semantic Web and Information Systems, 2007.

[17] Z, ElAkkaoui and E.Zimanyi, “Defining ETL Workflows using BPMN and BPEL”. Proceedings of DOLAP09, 2009, pp 41-48.

[18] S. Rizzi and M Golfarelli, "X-time: Schema versioning and cross-version querying in data warehouses", International Conference on Data Engineering (ICDE), 2007, pp.1471–1472.

[19] G. Papastefanatos, P. Vassiliadis, A. Simitsis and Y. Vassiliou, "Policy-Regulated Management of ETL Evolution", Journal on Data Semantics XIII, LNCS 5530, 2009, pp 146–176. [13] G. Papastefanatos, P. Vassiliadis, A. Simitsis and Y. Vassiliou, " HECATAEUS: Regulating Schema Evolution. Data Engineering", International Conference on Data Engineering (ICDE), 2010, pp 1181-1184.

[20] X. Zhang, W. Sun, W. Wang, Y. Feng, and B. Shi, “Generating Incremental ETL Processes Automatically”, Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006