Implementation and verification of a data vault

Storing and Managing Data (Module 06-32245), Autumn 2022

Sengodan Rajasekar, Karthickeyan, 2443050
kxs250@student.bham.ac.uk
December 1, 2022

***Abstract*—Implementation of the data vault model is an efficient method of designing data warehouses for an enterprise to enhance the processes of data analysis and business intelligence that result in better functioning and development of the enterprise. In this project, a data vault model has been implemented with three layers namely, a staging layer, an enterprise layer, and an information layer. This data vault has been built with the aid of two provided neuroimaging datasets. The skeleton of the data vault was created and data from both datasets were parsed and populated into the database thereby creating a data warehouse. When queried, the data vault model retrieves the data from the data warehouse and provides the user with the required results that include visualized data. This project aims to store, visualize, and retrieve neuroimaging data by implementing the data vault 2.0 methodology.**

I. Introduction

Data vault is the latest technique that is now being commonly implemented in various business enterprises for the upliftment and significant improvement in the data analysis department of the enterprises. A Data vault is an architecture for providing data analysis features to an enterprise. The data warehousing and business intelligence needs of the enterprise are addressed using the data vault method. The data vault 2.0 methodology is applied to this project and its advantages over the data vault 1.0 methodology are further discussed in this report.

For this project, two datasets were provided namely, Visuomotor functional connectivity (VM) and Pre-autism datasets. The datasets represent neuroimaging datasets that were obtained using Functional Near Infrared Spectroscopy (fNIRS). fNIRS works based on light absorption. Infrared light at varying wavelengths is passed on the scalp of the subject and the light is then captured by using photodetectors. This captured light possesses information on tissue oxygenation in the form of oxy(HbO2)- and deoxy(HbR)-haemoglobin concentrations. Hence, by using this data, fNIRS can be used to monitor brain activity. Specifically, during the experiments during which the given datasets were created, the stimulus and probe position was monitored and recorded as per requirement from the subjects while performing different exercises. A data vault has been implemented to store, visualize and retrieve neuroimages from the above experiment using the data vault 2.0 methodology.

II. State of the art of data vaults:

A. Data Vault Methodology

The data vault methodology helps in the implementation of smart data warehouses in an agile manner. Emerging problems like rapid changes, large amounts of data to handle, data complexity, flexibility and the application domain have called for the implementation of a smart solution like a data vault. It uses a data modelling methodology and was created by Dan Linstedt. The data vault enables the creation of data warehouses to carry out data analytics on an enterprise level. The data vault comprises of three different types of entities, namely,

- Hubs,
- Links,
- Satellites.

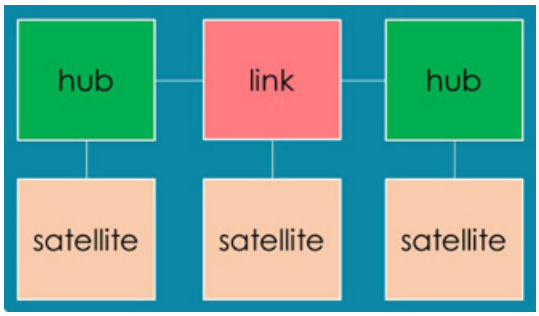

Fig. 1. Strucuture of Hubs, Links and Satellites. [1]

B. Hubs

Hubs in a data vault represent business keys or vital business concepts of the highest importance to the business. Hubs contain a unique list of business keys along with metadata. The metadata contains information about the business key that includes where and when the key was loaded [2] .

C. Links

Links in a data vault are used to connect multiple hubs. They help in recording the relationship between various hubs and hold accounts of transactions and compositions between them. The hubs are saved as foreign keys. They also contain the metadata that states when and where the link was loaded from [3].

D. Satellites

Satellites in a data vault are used to connect hubs or links. We can access the information that states when and where the data was loaded and can be used to efficiently trackback transactions. They hold information about the parent hub and parent link. They contain descriptions of their parent and vary over time based on transactions [4].

E. Data Vault 2.0 vs Data Vault 1.0

Data vault 1.0 focused on data loading and modelling techniques for data warehousing. Data vault 2.0 has increased the possibilities of data vaults and provides a complete data warehousing method [5].The various features of data vault 2.0 include agile methodology and constant improvement of the implementation to meet the rapidly changing enterprise needs. It also provides a reference architecture. It offers support for large datasets that bring forth increased complexities [6].

III. Methods:

A. Enterprise Layer

The enterprise layer is the layer for storing and managing the data. The skeleton of the data vault warehouse in created based on the Entity-Relationship diagram which contains all the hub tables, satellite tables and links along with their relationships. During the staging layer, the data is populated into the tables created in this enterprise layer accordingly. The skeleton is created using PostgreSQL to an SQL file (dataVault.sql). It consists of all the database, schema, and tables which forms the data warehouse.

B. Staging Layer

The staging layer involves the receiving of data from the given sources, transforming it and loading it into the data vault warehouse. It follows the Extract, Transform and Load (ETL). The staging layer of this project consists of a python code (staging.py), that reads and parses the datasets provided into dictionaries and lists. This completes the extraction and transformation part of ETL. The connection is established between the python code and the enterprise layer, which already contains the skeleton of created tables. This is the loading part of ETL. Once ETL is complete, the staging layer is established.

C. Information Layer

The data is made easily accessible by querying the database. The queries are passed in the python code which is connected to the data vault through PostgreSQL. The required results are visualized in the form of plots to obtain better understanding of the data and to retrieve the required information in a desirable manner. This marks the final layer of the data vault model.

IV. Results:

A. Data vault warehouse creation

The skeleton of the data vault warehouse was created by running the PostgreSQL file (dataVault.sql). The hub tables were created an satellites and links were referenced to the hub tables wherever needed. The creation of the skeleton of data vault warehouse was verified using the 'SELECT' queries for the appropriate tables under the specified schema of the database. Post running the python code, the tables were populated with the data from the datasets.

B. Python and ETL

The ETL was carried out using python. The Visuomotor functional connectivity and Pre-autism datasets were received as input in the python code using read operations. The datasets are then parsed and stored as lists an dictionaries using various defined functions for specific operations. The parsed lists are then loaded into the tables in the enterprise layer by establishing connection between python and PostgreSQL.

The Visuomotor functional connectivity (VM) dataset contains two parts, namely, metadata and data. The metadata part of the VM dataset is parsed using python into a dictionary. This dictionary is then used to insert values into the appropriate tables in the data vault warehouse created earlier.

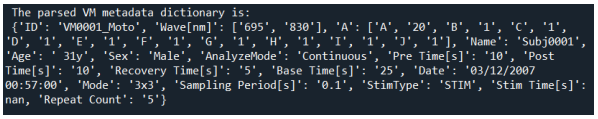

Fig. 2. Parsed Metadata of VM Dataset.

The Pre-autism dataset has five files with each file having different filetype. The file extensions of the files include .dat, .evt, .hdr, .wl1, and .wl2. The .dat, .wl1 and .wl2 files are read and parsed into lists as they are in the form of large arrays. The .hdr file which contains the metadata is read and parsed into a dictionary. The .evt file is the event timeline file which is also read and parsed into a list.

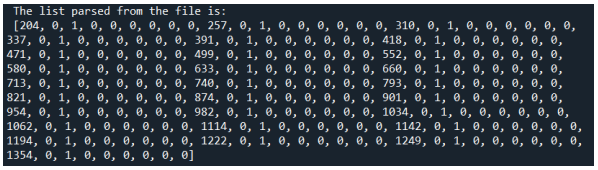

Fig. 3. Parsed Event Timeline of Pre-autism dataset.

C. Visualization using matplotlib

The results to the queries provided are obtained by carrying out operations in the python code by working with the parsed and populated data. The boxplot for comparing the distribution of either HbO2 or HbR concentrations for two intervals of time for a subject was plotted using the matplotlib python library.

Apart from the plots used to visualize in this project, there are a multitude of other ways to visualize the data and provide the user with the desired results. The following boxplot and lineplot were generated by using the parsed data that were obtained by reading the datsets and then plotting them in two axes based on the requirement.

V. Discussion:

The scope of the project is to implement and verify a data vault with data vault 2.0 methodology using the features of hubs, links and satellites to indicate the prominent business keys. This helps the enterprise to significantly advance in the processes of data analyis and business intelligence. Since

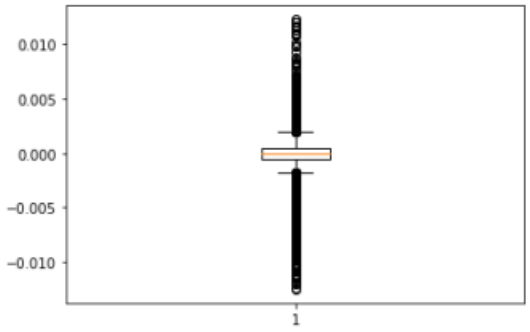

Fig. 4. Comparision Box Plot of HbO2 or HbR.

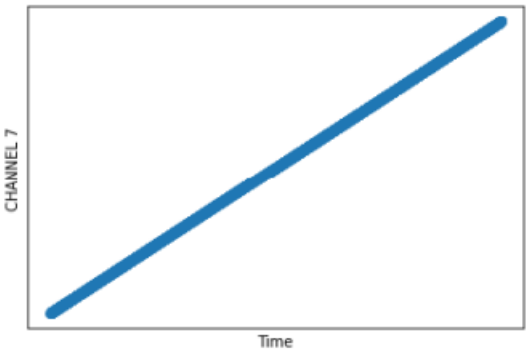

Fig. 5. Time Course of Light Raw Intensity for CH7.

the datasets were from a real-time experiment, the implementation was carried out keeping in mind the scenario of the experimental setup and conditions. But, there were some compromises to be made in the information mart layer where a GUI could have been implemented to obtain inputs from the user and return results with signficant data utilizing suitable visualization tools. Although that is for the future scope of this project implementation.

Working with large datasets that were collected in a real-time scenario requires an efficient solution that can process large datasets by running operations and queries through them in a short duration, so that the user can receive the output or the analysis at the earliest possible time. This will require application of other prominent domains like big data, machine learning and artificial intelligence. Using these domains in the project can significantly impact the time take to obtain the output, the efficiency of the data vault and the feasibility of implementation of the data vault in a real-time scenario for an enterprise. Hence, the scope of the project can be expanded.

```
Query3a: The list of names of experiments in the treatments in the database is:
['HBA_Probe1', 'MES_Probe1', 'BA_Probe1_', 'ES_Probe1.']

Query3b: The list of names of factors and treatments in the database is:
['Deoxy_ViMo', 'Oxy_Viso', 'Oxy_Moto', 'Oxy_ViMo', 'eoxy.Viso_', 'Oxy_Rest',
'Deoxy_Viso', 'xy.Viso_', 'Deoxy_Moto', 'Deoxy_Rest']
```

Fig. 6. Query output for list creation with experiment names.

VI. Conclusions:

Hence, the main features of the data vault 2.0 methodology have been implemented in this project involving the handling of real-time datasets from a medical domain and valuable information have been obtained. The hubs, links and satellites were introduced and they were structured and connected as per requirements.

Making use of hub, link and satellites lead to the usage of a list of unique business keys, a list of relationships and associations and descriptive data from the satellites in order to achieve the desirable implementation of the data vault architecture. Data vault 2.0 is much more agile, efficient and faster than the data vault 1.0 and can constantly cater to the needs of the rapidly developing enterprises and offer efficient solutons to address its data anlaysis requirements.

References

[1] https://www.data-vault.co.uk/what-is-data-vault/
[2] https://www.talend.com/resources/what-is-the-data-vault/
[3] Daniel Linstedt and Michael Olschimke, ”Building a Scalable Data Warehouse with Data Vault 2.0”, Elsevier Science, 2015, pp. 28–46.
[4] https://www.ben-morris.com/data-vault-2-modelling-the-good-the-bad-and-the-downright-confusing/
[5] https://www.sciencedirect.com/topics/computer-science/data-vault-modeling
[6] https://makingdatameaningful.com/data-vault-hubs-links-and-satellites-with-associated-loading-patterns/