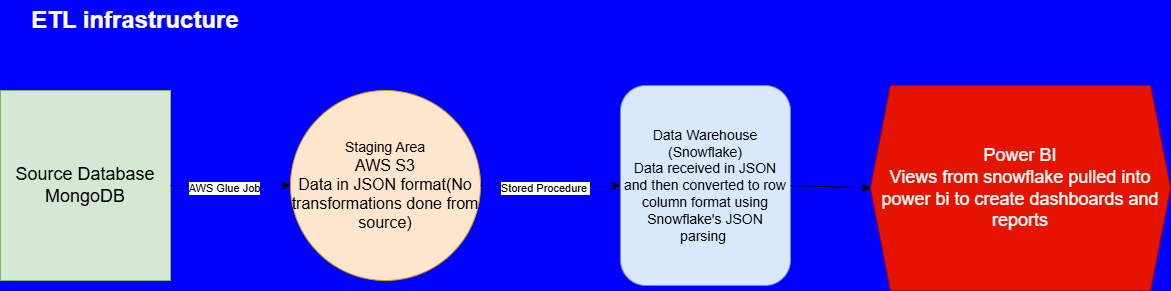
**VEOTS Data analytics documentation**

**Overview:**

In this data engineering project for VEOTS organization, the focus is on bringing in the data captured by the VEOTS applications and orchestrating the data movements to facilitate the provision of data analytics dashboards to our clients.

**Architecture:**



Components: MongoDB, AWS Glue, AWS S3, Snowflake.

Data Source: Application Data is generated in MongoDB, which is a NoSQL database and stores the data in a format called BSON (Binary JSON). BSON is a binary-encoded serialization of JSON-like documents. The data is represented in the database in the form of collections and documents.

Data Orchestration: Data extraction from MongoDB is performed via an AWS Glue job, with the resulting data stored in an AWS S3 bucket in its original JSON format. Subsequently, this data is retrieved from S3 into Snowflake through external stages. Within Snowflake, the data undergoes conversion from JSON format to a tabular format optimized for analytics purposes.

Analytics: For analytics the views are created in snowflake and these views are pulled into PowerBI for building dashboards and reports.

**Data Sources:**

The data originates solely from MongoDB, where it exists in the form of collections within the database. These collections are categorized into two types: Static and Dynamic.

**Data Processing:**

Data processing initiates with MongoDB collections being retrieved via AWS Glue job and subsequently stored in an S3 bucket as JSON files, maintaining their original format. Static and dynamic collections are organized into separate folders. At this stage, no transformations occur, aligning with our ELT methodology for data processing. The data is then loaded into the data warehouse in its native JSON format. Subsequent transformations to convert the data from JSON to relational tables are performed only after the data has been loaded into the data warehouse.

**Data Storage:**

Data Lake: AWS S3 bucket is used as a data lake to store the data that is fetched from MongoDB as the data is semi structured as it is the JSON format. From here snowflake fetches the data using external stages for doing transformations and getting the data ready for analytics.

Data Warehouse: Snowflake serves as the data warehouse, extracting data from JSON-formatted files stored in S3. It employs its JSON parsing capabilities to transform this data into relational tables, facilitating efficient analysis and querying.

**Data Quality and Governance:**

The data is extracted from its source and deposited into the data lake without alteration. From there, it is imported into Snowflake in its original JSON format before undergoing parsing into a tabular structure.

Compliance with GDPR: The data housed within the data warehouse, specifically Snowflake, adheres to GDPR compliance standards. This is achieved through the implementation of dynamic masking features provided by Snowflake, effectively obfuscating personally identifiable information (PII) within the dataset.

**Data Pipelines:**

AWS Glue Job : This job orchestrates the retrieval of data from the MongoDB source and seamlessly transfers it to AWS S3.

Snowflake(Procedure) : This task operates within Snowflake, executing a stored procedure to perform insert overwrite operations. These scripts efficiently load all tables from S3 into Snowflake.

Snowflake Scripts: These meticulously crafted scripts are pivotal in Snowflake's environment. They generate views based on the data ingested, crucial for the seamless creation of reports and dashboards within Power BI.

**Data Visualization and Reporting:**

The views crafted within Snowflake serve as the foundation for Power BI integration, enabling the creation of insightful dashboards. These dashboards, meticulously designed, offer valuable insights into the data, empowering clients to make informed decisions.

Let us understand the whole flow of the data analytics infrastructure.  
  
  
**Section 1: Data Extraction and movement to staging area.**

We extract data from source MongoDB using AWS Glue job. For detailed explanation of the glue job refer Final Optimized ETL job.docx : Here is the link to [AWS Glue job Explaination](https://ksjrudhprivatelimited-my.sharepoint.com/personal/rajat_sawant_veots_com/_layouts/15/doc.aspx?sourcedoc=%7b78278115-4cd2-4a56-aad2-38d3e4457dbd%7d&action=edit)

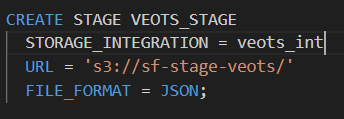
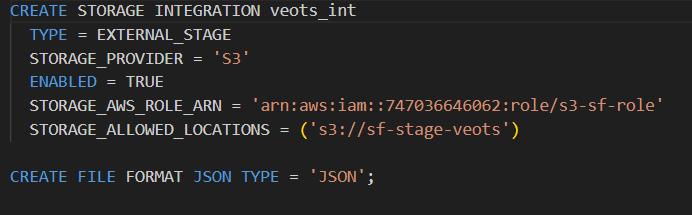
When the glue job runs successfully, all the required collections from MongoDB get stored in AWS S3 bucket in JSON format.

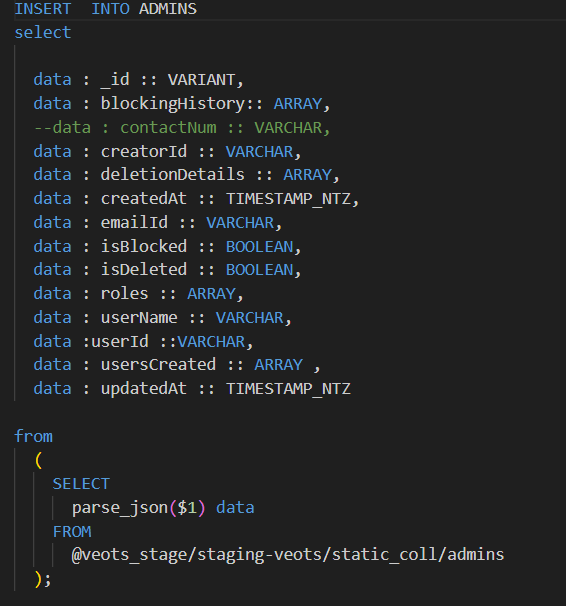
The Glue job is configured to first delete all the files related to the present database from S3 bucket and then paste the new data with new records in the bucket. Which is further pulled into snowflake.  
As we are loading data in an incremental fashion, only the records which are not loaded before will be loading in the present iteration.

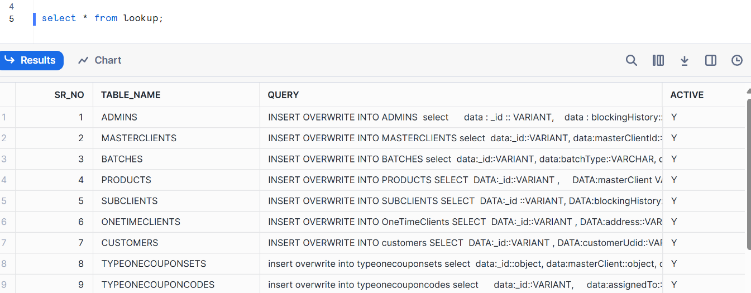
On completion of the glue job run, we are keeping the logs of the glue job in a table in snowflake.

query = f"INSERT INTO ANALYTICS.INC\_GLUE\_JOB\_LOGS VALUES ('{job\_id}','{start\_time}','{end\_time}','{status}','{error\_message}')"  
  
we are using the following query to enter the details of the glue job once the run is completed. This keeps the log details of the glue job in snowflake. Also, it enables us to trigger the loading of data from S3 to snowflake using stream and tasks.

**Section 2: Loading Data from S3 to snowflake.**

To load data from S3 into snowflake, we have configured Storage integration and External Stages in snowflake.  
  
These snowflake objects enable us to extract data from S3 bucket.

To populate all the tables in snowflake, we are leveraging the Json parsing provided by snowflake as shown below  


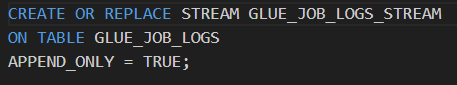
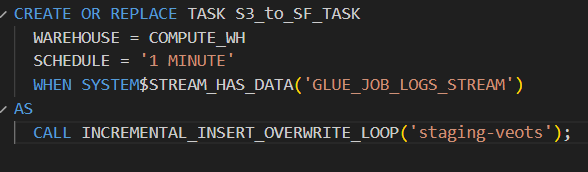
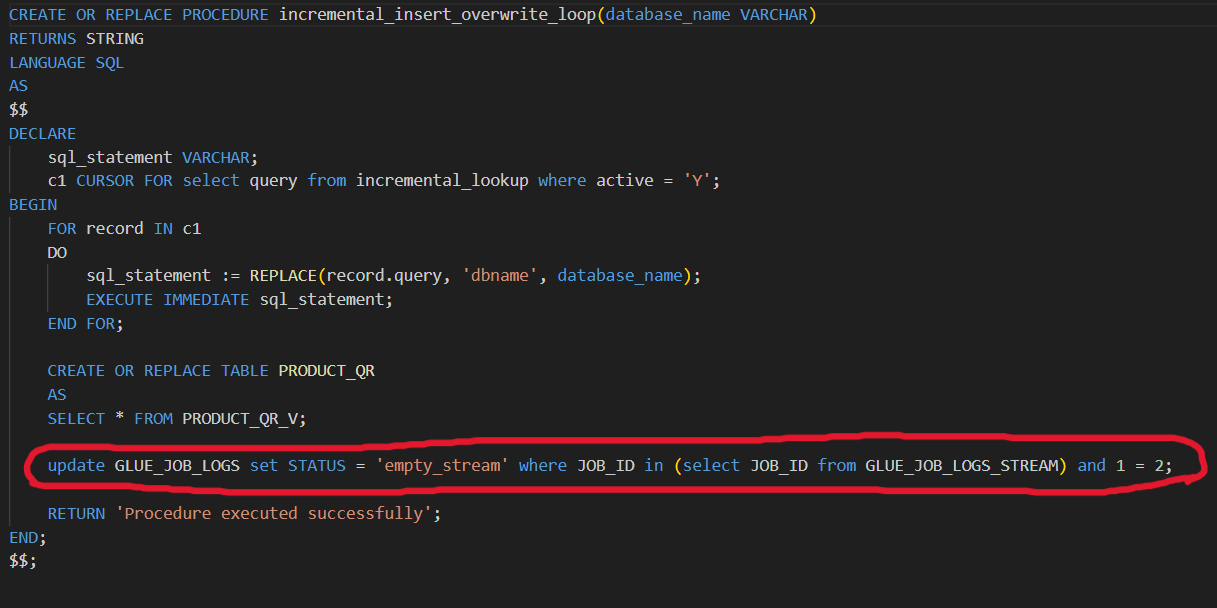
**Automation of loading all the tables in snowflake:**  
To load all the tables, we have created a procedure which uses a cursor to loop over the lookup table and load all the insert scripts.  
  
  
Lookup table

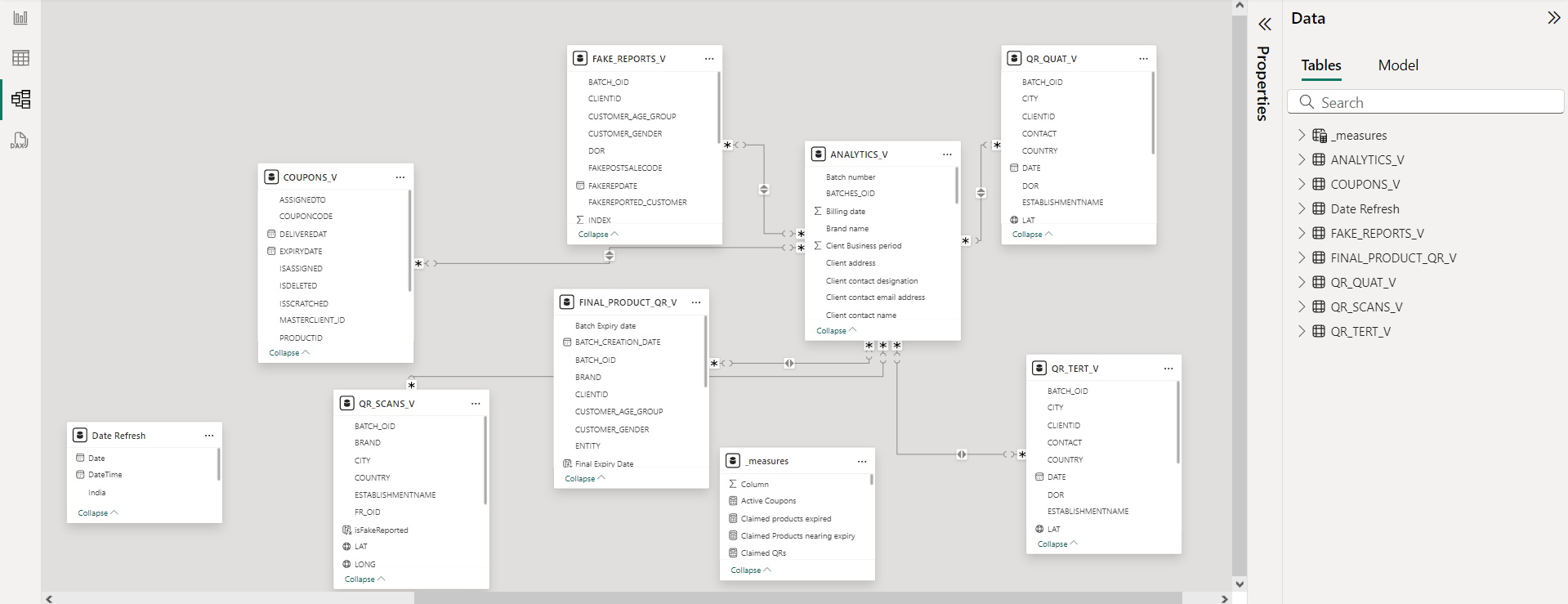
The stored procedure is configured to run all the queries in the query column which contains the insert scripts to insert the new records that are present in S3 bucket into snowflake tables.   
  
In snowflake we have created multiple views for analytics purpose to get meaningful insights from the data. These views are then pulled into Power Bi to create dashboards.

**Using stream and tasks to automate the trigger of stored procedure:**

**Streams** in Snowflake are used to track changes to data in a table, allowing you to capture insertions, updates, and deletions made to a source table over time. They act as a change data capture (CDC) mechanism, maintaining a record of changes between two points in time. Streams enable you to perform incremental data processing, such as ETL workflows, without repeatedly scanning the entire source table.

**Tasks** are used to schedule SQL queries, including DML (Data Manipulation Language) and procedural logic, to run on a recurring basis. They are typically used for automating jobs such as data transformations, maintenance, and periodic reporting.

* When the glue loads data into S3 from MongoDB, we are inserting the logs into a table (GLUE\_JOB\_LOGS) in snowflake. We have created a stream on the same table, so that whenever the GLUE\_JOB\_LOGS table gets a new record, the data is inserted into the stream.  
  
* We have created a task on condition that, if the stream has data, then run the procedure that loads all the insert scripts using a cursor.  
  
* Then at the end of the procedure we have an update statement that consumes that data from the stream, so the stream is now empty.  
  

**Section 3: Power Bi dashboards and workspace**  
The data from the views that are created for analytics are pulled into power Bi to create Dashboard reports. We have created one dataset from which multiple dashboards receive data and reflect on reports.