4 -Staging and Transformation

Background story

Now that Caladan has all the data from their cloud and on-premises data stores it is time to create a usable Operational Data Store.

It is time to conform the source data in the data lake into a more useable dataset. Downstream consumers of the data should not need to worry about negotiating between Document data and SQL Data. These downstream consumers also want a one-stop shop for all the data. This will be especially important as new source systems are brought into the lake, so that downstream consumers and the Azure Data Warehouse you will build can react to new data sources being brought online. However, the original source data must also be preserved in the Operational data store for audit and review purposes. This will enable the creation of alternative intermediate datasets at any time, and it will also enable deeper exploration for use cases such as comparing Global Covid-19 data with local Covid-19 data.

As Caladan looks toward the future, they would also like to introduce a review process such that all changes to the solution under source control must be approved by a second developer.

Technical details

- One of Caladan's most immediate needs will be to creata a normalized operational (ODS) as a intermediate data store to combine metric data with policy data.
    - The creation of this intermediate dataset is the team's opportunity to give the various downstream consumers a single location from which to load all of the data from all source systems which has been cleaned and normalized.
    - This intermediate dataset also removes cognitive overhead from downstream consumers, since the heterogenous data types and different formats are all normalized.
    - The team is completely free to choose what data types the intermediate dataset will use. This is possible because no changes are made to the original source systems, which means there is no impact to any component which consumes directly from those source systems. Additionally, no changes are being made "in place" on the raw, extracted data in the data lake.
- While some entities are common between the various source systems, it is important to understand that the data represents two seperate data types. `Metric and Policy data.` The team might consider modelling the target dataset as the following five collections:
    - Cases
    - Deaths
    - Recoveries
    - Policies
    - Geography
- When conforming this data for downstream consumption:
    - The team **is** expected to create and store unified sets of common elements which exist across all source systems (e.g., geographic names) and put in safeguard to ensure data cleanliness.
- Given the various downstream consumption patterns and user personas, e.g. business intelligence versus machine learning, the team should not apply any logical business rules at this time.

  - Creating a homogenous dataset with consistent column names and data types is common work which will benefit most, if not all, downstream consumers.
  - Various downstream consumers may have different requirements when it comes to de-duplication, resolution of conflicts between source systems, etc. The team should not make such decisions now, as these downstream requirements have not yet been specified.

Success criteria

- Policies have been implemented in the version control solution such that:
  - Developers cannot push changes directly to the main branch.
  - Changes are reviewed with at least one explicit approval from another developer before being incorporated into the main branch.
- A new operational data store has been created that encompasses both metric and policy data.
  - Data from the source systems has been transformed to use **consistent data types and formats** and an in-flow data integrity system has been put into place. for example, if source systems use different data types or formats for a date, the conformed dataset would store all dates in a single, consistent data type and format.
  - Downstream consumers have no need to load data from various systems in order to process data from all sources; this new dataset is their single location for all data from all source systems that feed into the lake today and tomorrow.
- Each record in the new dataset is marked with an identifier of the original source system.
- The original extracted data is preserved within the data lake.

Tips

- The team should concentrate on creating a Operational Data Store that is most useful for a variety of uses cases.
- The team shoud not worry about creating a star schema or DW at this time.

Resources

Ramp Up

- Extract, transform, load (ETL)
- Extract, load, and transform (ELT)
- How to teach Git

Choose Your Tools

- Ingest, prepare, and transform using Azure Databricks and Data Factory
- What is Azure HDInsight
- Load the data into Azure Synapse Analytics (formerly Azure SQL Data Warehouse) staging tables using PolyBase

Dive In

Azure Databricks

- Tutorial: Extract, transform, and load data using Azure Databricks

HDInsight

- Tutorial: Extract, transform, and load data by using Apache Hive on Azure HDInsight

Polybase

- Tutorial: Load New York Taxicab data to Azure Synapse Analytics (formerly Azure SQL Data Warehouse)

GitHub

- Enabling required reviews for pull requests