

Goal

- To have our Tableau Dashboards display a wide array of data regardless of the varying data sources.
- The need is to have all data sources be consolidated into one central repository for easy access for Tableau builders, end users and other team members to use.



Problem

- There are a multitude of data sources available but there isn't a standard method of retrieving, processing and housing them. These include:
 - Downloading external registry data
 - Often needs to be manually downloaded and placed in a shared location without a direct access feed.
 - Abstracted data
 - IT exclusive data warehouses and data from EHR
 - Leveraging IT to gain access to tables not available within our servers
 - By maintaining relationships with IT these requests get easier to fulfill.
 - We use linked server to retrieve data back to our server.
- Manual data entry
 - For processes not captured electronically, Access Databases are created so that data can be manually entered with the assistance of other team members.
 - The raw data is stored on tables that are retrieved in SQL for processing



Implementation

- Various Data import methods are used to streamline the data into a singular format
 - For external data sources
 - SSIS Visual Studios is used to perform ETL (Export, Transform and Load) to update tables with information
 - Data is imported on a cadence to ensure that data is routinely available
 - If data needs to be updated, processes are in place to refresh data.
 - SQL is the main tool to process data
 - Data is held on two main tables for all dashboards after being processed.
 - Emphasis on Query Optimization
 - When using linked servers, calculations performed on other servers if we see a delay on ours before bringing data back.
 - Temporary tables used to minimize run time and prevent deadlock
 - Instead of retrieving data multiple times on the server, we store the data once as a temp table then reuse the table locally to conserve bandwidth
 - Calculations are done in SQL to minimize strain on Tableau
 - Data is standardized and ready for visualization
 - Leads to a quick and responsive dashboard

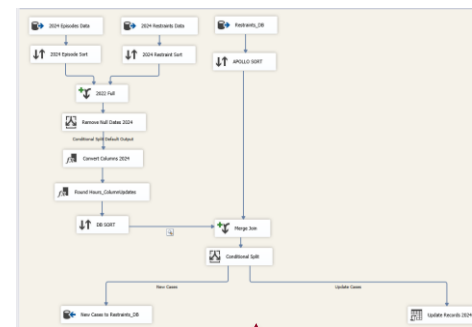
Communication

- Open and clear communication between stakeholders, business owners of data and development team
 - This ensures transparency for the project
 - The logic used is established and understood by everyone
- Quality checks are performed by those who know the data best. This is done pre and post production
 - Before a dashboard is rolled out, visualizations are validated by subject matter experts
 - After dashboard is rolled out, feedback is welcome during the secondary validation period to ensure data consistency, integrity and quality.
 - The feedback we received is then applied to future dashboards further improving the process



Results

- All the efforts leads to an interactive and responsive dashboard.
 - Having one source of truth builds confidence in the data
 - Other users can use of the data as it is readily available for other builders. This minimizes inconsistencies across multiple dashboards.
 - Standard data displayed across a multitude of dashboards ensures consistency across project
 - Data is then used to make informed health care decisions based on current data and historic data
 - Identifying gaps in processes
 - View data trends over time
 - Drill down to encounter level data
 - Team is vigilant of opportunities to improve and build on all strategies previously implemented
 - Constantly look for areas of performance improvement



```

-- Create temp table
CREATE TABLE #TempTable (
    Column1 INT,
    Column2 VARCHAR(100)
)

-- Insert data from source table
INSERT INTO #TempTable
SELECT Column1, Column2
FROM SourceTable

-- Perform calculations and update target table
UPDATE TargetTable
SET Column1 = #TempTable.Column1 * 2,
    Column2 = #TempTable.Column2 + ' - Updated'
FROM #TempTable
    
```

Example of temp tables used in SQL code

Example of SSIS Package used to import/update data