Synapse Migration

(Healthcare Provider)

Built Dedicated SQL Pool in Azure Synapse Analytics and **transitioned the existing Data Warehouse** in Azure SQL database to Synapse Analytics

DATA WAREHOUSE MIGRATION TO AZURE SYNAPSE ANALYTICS

ABOUT THE CLIENT

Client is a U.S. based Healthcare provider specializing in high quality post-acute nursing care and rehabilitation services

SITUATION

- The client was **expanding business rapidly** by adding new facilities and **integrating new systems/applications** into the existing technical environment which posed challenges & limitations when processing the workloads. Also, it was cost prohibitive to improve the configuration of technical components to accommodate for the increasing workloads.
- Merilytics partnered with the client to **design and deploy a new robust and scalable architecture** that can **handle heavy workloads** and **ensure zero downtime** of the systems while transitioning from current to the new architecture

VALUE ADDITION

- **Analyzed the current state architecture and the workloads** that the architecture can sustain in the near/short term. **Conducted a thorough capacity planning exercise** and **accounted for current growth rate and further projections** of the workloads.
- Based on the inputs, proposed a **new technical architecture** and implemented it which included **Azure Synapse Analytics, Data Lake and PolyBase** which are designed to heavy workloads and faster processing.
- **Performed Data Migration** in phases to the new environment with **zero downtime and without any business disruption. Reconfigured the dashboards** to point to the new environment and shared a **detailed metric-wise validation tracker** with client for smooth transition

IMPACT

- **Reduced overall monthly costs by ~40%** due to the improved technical architecture
- **Improved performance of the ETL process** which **reduced the run time by ~33%** to 2 hours daily.
- Efficient pipeline development **(~90% reduction in pipelines to 30 from 400**) by leveraging the "**Data-Driven Pipeline approach**" which make the pipelines more **adaptive** and **capable** of **handling varying data sources, transformations and destinations**.

TRANSITION METHODOLOGY

Data warehousing	Data Pipelines	Model and Serve	Support
Create Data Tables	Redesign the stored procedures	Rebuild dynamic pipelines	Data modelling and dashboarding	Data Validation

- Migrated the existing data tables to the SQL Dedicated Pool of Azure Synapse Analytics through **bulk load** for first time
- **Improvised the nomenclature** of older tables as per new decided standards.
- Build the Data Warehouse blueprint with separate **schemas** for staging, production, reporting, auditing and mapping tables.

- Redesigned the existing **stored procedures** with **PolyBase code for faster transfer rate** in Azure Synapse Analytics
- **Re-engineered the Stored Procedures** with updated functions which are compatible in Azure Synapse Analytics

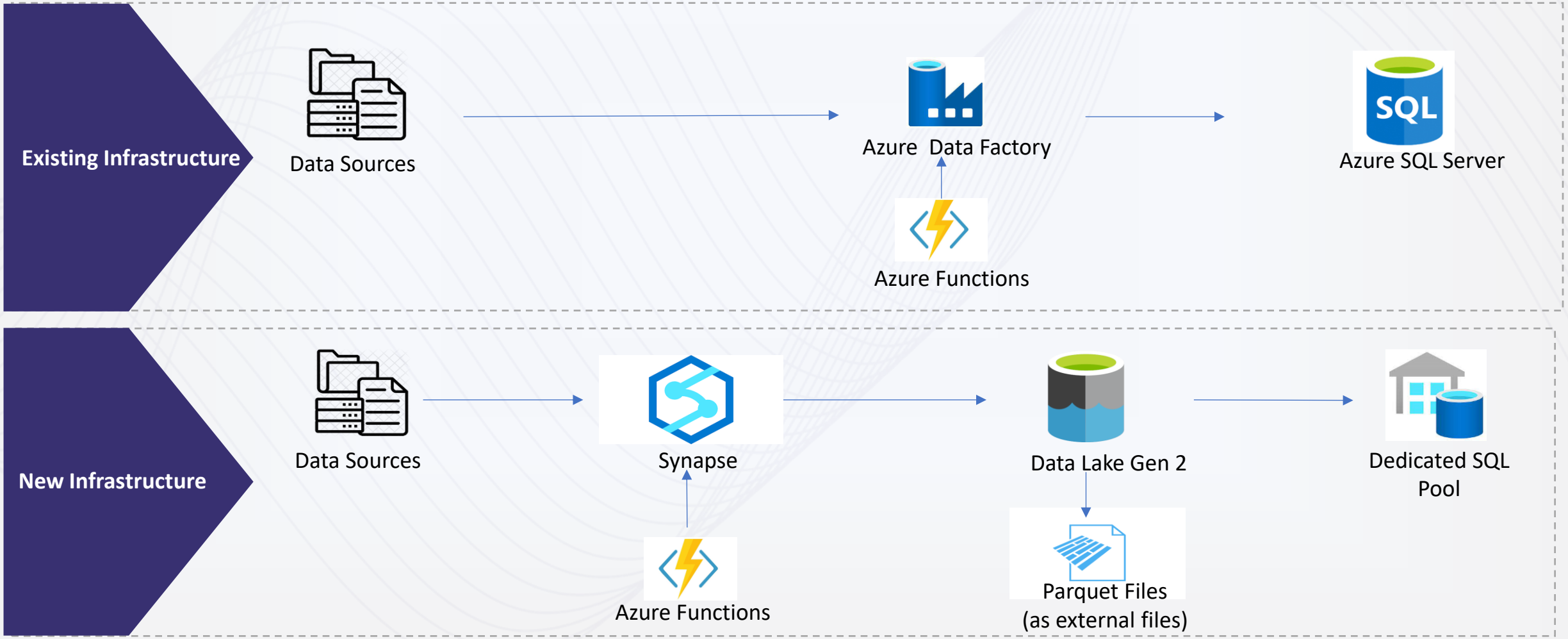
- Rebuilt all the data pipelines to load data into **dedicated SQL pool** in Azure Synapse
- New pipelines were developed based on “Data-Driven Pipelines” approach making them more efficient, adaptive and resilient

- Transitioned the current **Analysis Services data models** for different line of businesses to data fetch data from new Data Warehouse.
- **Built new Power BI dashboards and Paginated reports with source** from new Data Warehouse
- Handled the measures in the Power BI dashboards and data models with the new updated table names.

- Maintained a **transition document** and table name mapping for the changes made in the infrastructure.
- Performed the validation of data across the current dashboards and new dashboards on a daily level and maintained a **metrics/KPIs wise validation tracker**.

DATA FLOW COMPARISON BETWEEN TWO INFRASTRUCTURE

- In the earlier infrastructure, the staging was carried out directly at the production SQL database.
- After transition, the staging layer was designed in the Data Lake in the form of parquet files.
- Using Poly base, the data tables were imported to the Synapse Dedicated SQL Pool in form of external tables.



COMPARISON BETWEEN TWO INFRASTRUCTURE

POINTS OF COMPARISON	EXISTING INFRASTRUCTURE	NEW INFRASTRUCTURE
Pipelines	~400 data pipelines per table for 14 sources	~30 dynamic data pipelines for 14 sources
Schemas	Only default [dbo] schema	Separate schemas for staging, production, reporting, auditing and mapping tables
Staging	Varchar staging tables in database from pipelines	From Data Lake as external tables
Storage Size	1TB Maximum limit at 2 vCores	Auto scale and no limit as per DWU consumption
ETL Time	~3hrs	~2hrs (33% reduction)
Nomenclature	[dbo].[tbl_tabletype_Tablename]	[schema].[Source_LOB_Tablename]

DYNAMIC PIPELINES & CONTROL TABLE

1. CONTROL TABLE

Control Table was designed to contain all the details related to pipelines such as Pipeline Name, Source Directory, Columns, Staging Table Name, Stored Procedure Name, etc.

2. ONE MASTER PIPELINE

One pipeline for each source which serves as a Master pipeline, and '**Foreach'** activity which executes all sub-pipelines to fetch data from various data elements from a single source location.

3. DYNAMIC PARAMETERS

Inputs from the Control Table are passed as **dynamic parameters** to the pipelines. This would ensure minimal maintenance post development.

4. POLYBASE

Polybase technology is leveraged to access and load the staging data stored in Data Lake storage to the SQL pool in the form of **External Table**.

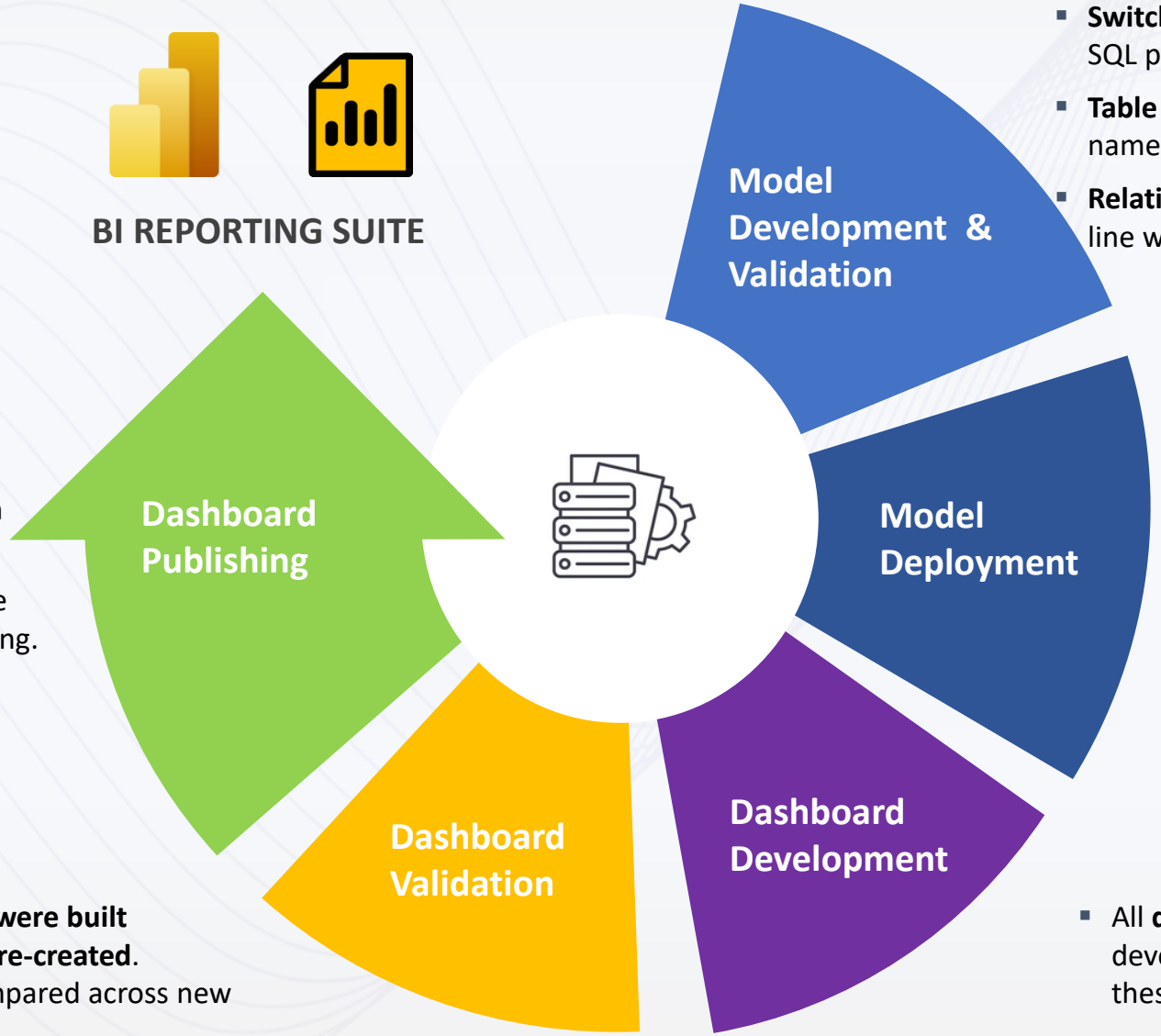
5. AUTO SCALING

ETL process can be extended to any **new data object** from the existing/new source by just adding the respective details in the **Control Table**

6. FLEXIBILITY

Any changes in the location of the source directory or the file name of a pipeline can be updated quickly in the **control table to reflect** change across ETL process

DATA MODELLING & BI REPORTING SUITE



- **Switched data source in data models** to Dedicated SQL pool.
- **Table properties** were updated with new table names.
- **Relationships** were validated and ensured to be in line with existing model.

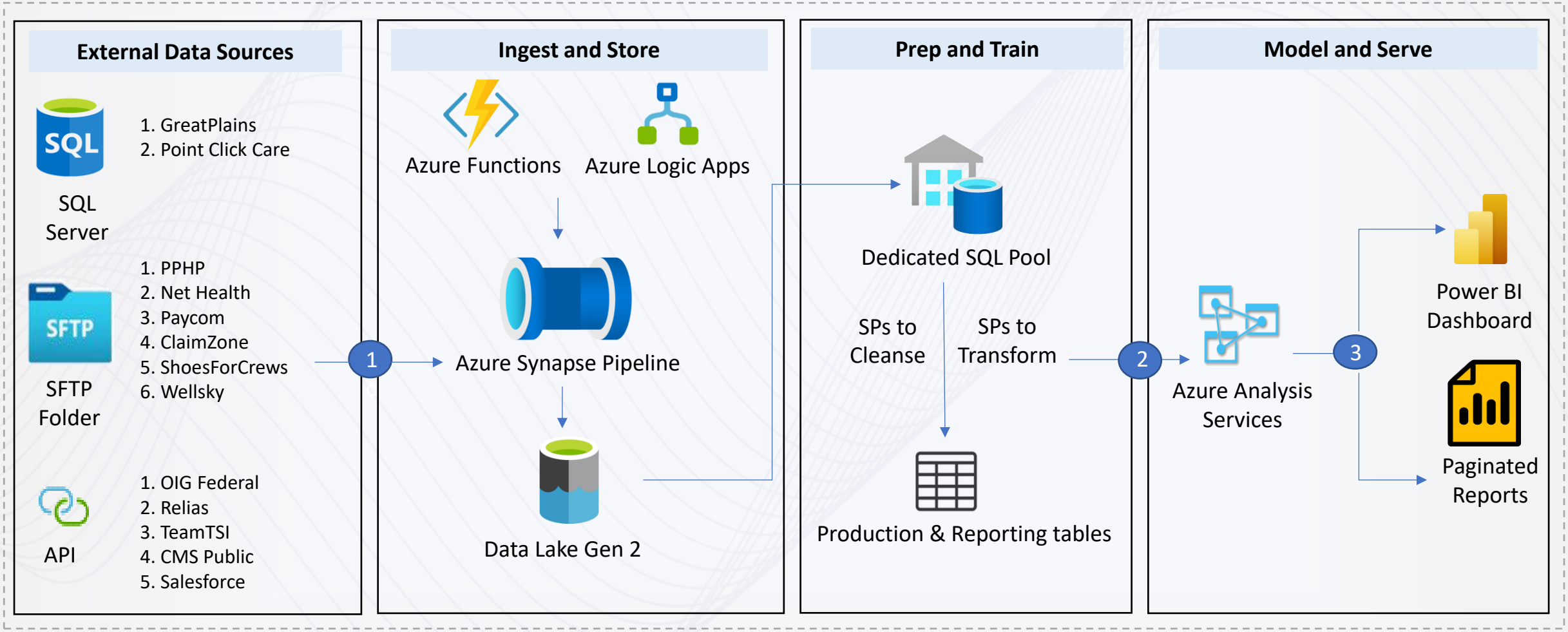
- Few **DAX measures** (such as SUMMARIZE) need to be updated manually.
- Updated data models were deployed as new data models to Azure Analysis Service (AAS).

- All **dashboards** and **paginated reports** were re-developed by establishing live connection to these new data models.

- All the **visuals** and **measures which were built locally** in the dashboard need to be **re-created**.
- The **KPI and Metric values** were compared across new and existing production dashboards

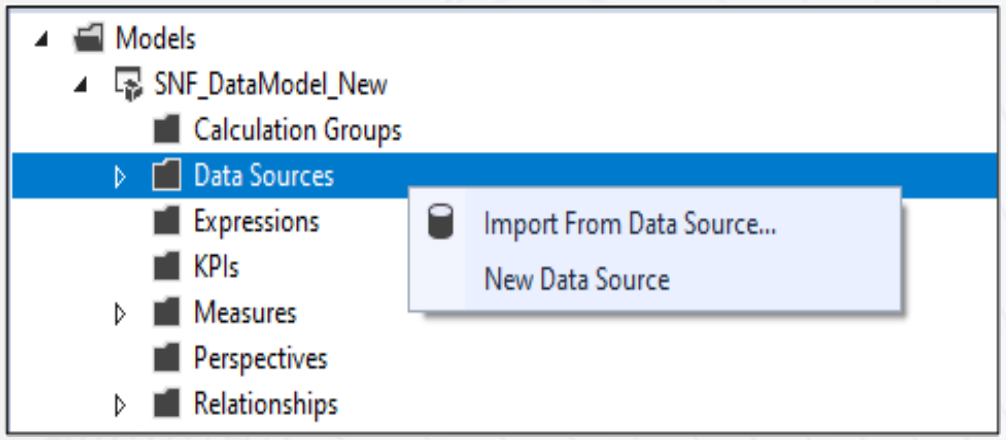
- All the **filters** were validated to be **in line** with the existing dashboard.
- The **navigation** and **bookmarks** were rebuilt and validated before publishing.

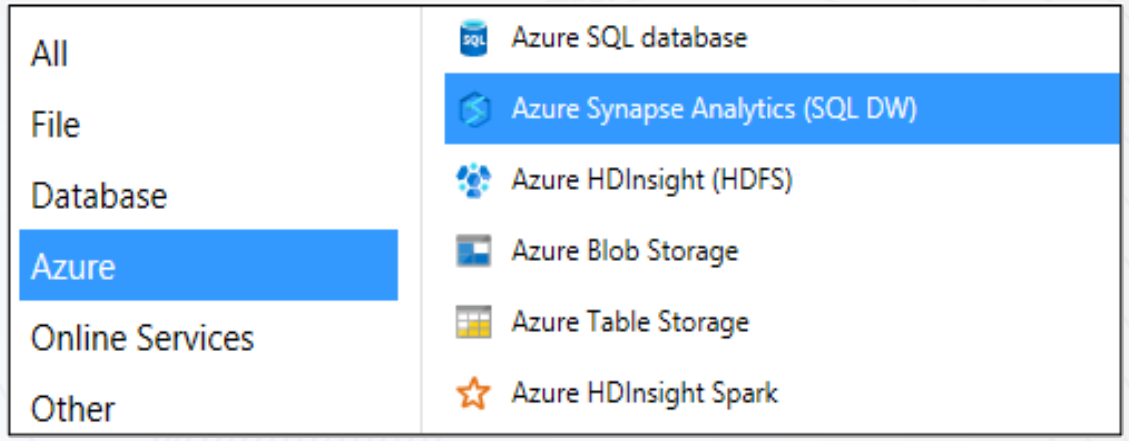
EXHIBITS #1 : HIGH LEVEL DIAGRAM OF DATAWAREHOUSE ARCHITECTURE



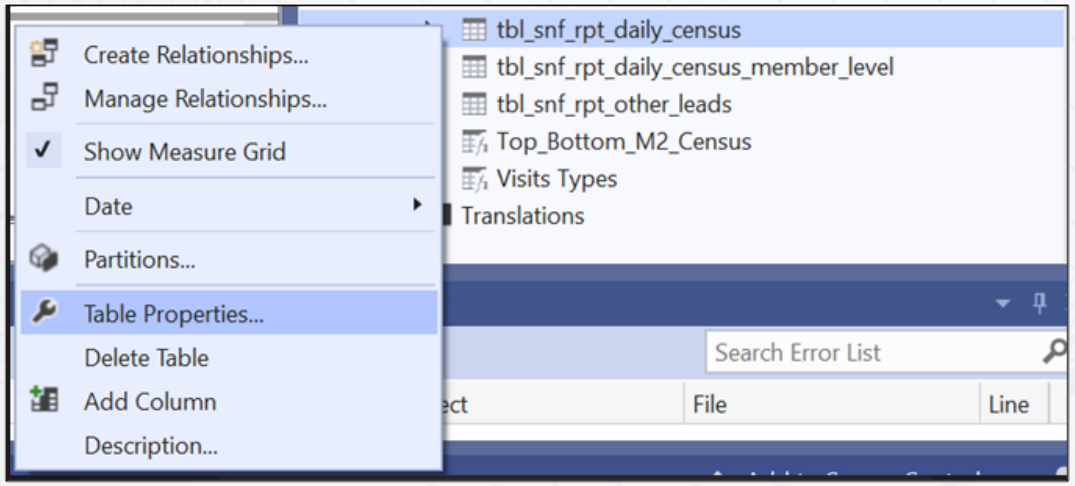
1. Data copy from different sources into Synapse Analytics environment using pipelines.
2. Data modelling and deployment using Azure Analysis Services
3. Data visualization using Power BI dashboards

EXHIBITS #2: DATA MODELLING UPDATES





Updating the data model to add new data source from Dedicated SQL Pool




```
let
    Source = #"SQL/azsqldbprod database windows net;AZRIHSDW",
    dbo_tbl_snf_rpt_daily_census = Source{[Schema="dbo",Item="tbl_snf_rpt_daily_census"]}[Data]
in
    dbo_tbl_snf_rpt_daily_census
```

Old table property

```
let
    Source = #"SQL/azrihssyndp sql azuresynapse net;AZRIHSDW",
    dbo_tbl_snf_rpt_daily_census = Source{[Schema="rpt",Item="PCC_SNF_daily_census"]}[Data]
in
    dbo_tbl_snf_rpt_daily_census
```

New table property

Updating the table properties as per new source and updated nomenclature

LEARNINGS

Sr. No.	Section	Learning Description
1	PolyBase	**COPY INTO** function can be used to import huge amounts of data in a very less time. For an instance this can import ~600M records in 10 min while the scale pool was scaled to 500 DWU.
2	Distribution	Changed the distribution of production tables that were using **SCD** to **HASH** distributed as **MERGE** statement works only with **HASH** distributed tables in Synapse.
3	Synapse	**Recursive CTE** is not supported in Synapse, and we must come up with an alternate logic based on the use case.
4	Synapse	Transactions support only **DML** operations inside the block but not **DDL**. Transactions names had also been removed as they are not supported.
5	Synapse	**CASE** statement was used as an alternative approach of **IIF()** as **IIF()** function is not supported in Polybase.
6	Synapse	**ERROR_LINE()** function is not supported, and we had removed it from everywhere as there was no alternative.
7	Synapse	**Identity(1,1)** generates a surrogate key for a table, but it does not start with 1 and increment by 1. We had used **ROW_NUMBER** function in case a key was needed where it starts with 1 and increments by 1.
8	Parquet Files	**Manual mapping** had to be used while copying data to parquet file if there are special characters or spaces in the column names from source. In case the source is a database, we can give an alias to the column name in the control table and no manual mapping is needed in such cases.
9	Modelling	During transition of data model, updated the name of the data model to build a new data model. Added a new data source and updated it for all data tables. Also ensured that none of the existing relationships and measure formulas are disturbed during transition.
10	Modelling	Some of the DAX functions such as **Summarize()** don’t update the name of underlying data table automatically after transition and had to be manually updated.
11	Dashboard	While replacing the data source of an existing dashboard with the new one, for the visuals that give error, all the related measures need to be updated and the visuals needs to be updated to point to the new measures.