Retail Sales

* Architecture of the pipeline

A diagram of a retail sales pipeline architecture

AI-generated content may be incorrect.

The above architecture represents the flow through of the project.

**Source Systems**

* **API**: Sales\_sample. json
* **CSV**: Products\_sample.csv
* **On-Prem SQL Server**: Customers\_sample.csv

Dataset for three tables:

1. **Customers** (On premises SQL Database)

**The source data** of the customers is stored in an SQL DATABASE, the below picture represents the screenshot of the table that was created and the data is configured to the table.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Products** – Stores product details- Bronze layer

Other dataset Products, we are placing this data directly to the bronze layer as shown below.



1. **Sales** – Stores sales transactions

We got the sales data as a Rest API format. (from the Git Hub)

A screenshot of a computer

AI-generated content may be incorrect.

In ADF, to execute the architecture the implementation is a Pipeline. To execute a pipeline, we have created Linked services for all 3 data sources.

**Linked Services:**

A screenshot of a computer

AI-generated content may be incorrect.

**Triggers:**

I have 2 triggers

* 1. **Scheduled**: Runs the pipeline according to the interval provided.
  2. **Tumbling-API**: It pulls the data from the API source within the time frame provided.

A screenshot of a computer

AI-generated content may be incorrect.

Along with linked and Triggers, here we have created few data sets which are used in the entire pipeline process.

**Datasets:**

A screenshot of a computer

AI-generated content may be incorrect.

**Raw to Bronze:**

Copy data activity is created for both API and SQLDB.

**For API,**

Source: Http file

Destination: Bronze layer

**for SQL,**

Source: SQLDB table

Destination: Bronze Layer

A screenshot of a computer

AI-generated content may be incorrect.

After validation and debugging the data is stored in Bronze layer as shown below,

A screenshot of a computer

AI-generated content may be incorrect.

The next step in the architecture is to take data from Bronze to silver

**Bronze to Silver:**

We are using data flow activity for the transfer.

Using the datasets which we created previously, the source to sink has been configured for all the 3 data sets.

**Products:** Source-Derived-Filter-Sink

**Sales:** Source-Derived-Select-Filter-Sink

**Customers:** Source-Derived-Filter-Sink

For all 3 data sets,

**Source:** Bronze layer

**Sink:** Silver layer

A screenshot of a computer

AI-generated content may be incorrect.

In the above data flow, we used Derived for eliminating NULL values.

I have used SQL functions for eliminating the following are the few patterns used in the derived.

**if the column is integer:**

**for 2 column mapping**

iif(!isNull(Column2), Column2, iif(!isNull(Column1), Column1, 0)

**for 3 column mapping**

iif(!isNull(Column3), Column3,

iif(!isNull(Column2), Column2,

iif(!isNull(Column1), Column1, 0)))

**if the column is date**

**for 2 column mapping**

iif(!isNull(Column2), toString(Column2, 'yyyy-MM-dd'),

iif(!isNull(Column1), toString(Column1, 'yyyy-MM-dd'), 'N/A'))

**for 3 column mapping**

iif(!isNull(Column3), toString(Column3, 'yyyy-MM-dd'),

iif(!isNull(Column2), toString(Column2, 'yyyy-MM-dd'),

iif(!isNull(Column1), toString(Column1, 'yyyy-MM-dd'), 'N/A')))

Filter is used to eliminate the rows that has all NULL values, the used expression is provided below .

!isNull(a) && !isNull(b) && !isNull(c) && !isNull(d) && !isNull(e) && !isNull(f)

Later once the configuration of the Data flow is completed, we go the pipeline where data flow activity is dragged to the pipeline and connect to the previous 2 Copy data activity

A screenshot of a computer

AI-generated content may be incorrect.

After the validation and debug are done, the data is flown to Silver layer as shown in below picture.

A screenshot of a computer

AI-generated content may be incorrect.

**Silver to SQL db:**

Once the data got to silver layer, the next step is to connect the silver layer to your o/p SQLDB.

In the SQL database we are creating the tables as per the o/p we got in Silver.

Mapping is done as required

The below 2 pictures shows the tables that got created

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Here comes the configuration of data flow for Silver to SQLDB

A screenshot of a computer

AI-generated content may be incorrect.

Another data flow activity was added and connected with previous one.

A screenshot of a computer

AI-generated content may be incorrect.

Validation and debug are performed and below 2 pictures shows the o/p data of the Silver to SQLDB

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**SQL Server to Power Bi:**

A screenshot of a computer

AI-generated content may be incorrect.