**Retailer Data Analysis**

**Introduction to Building a Data Lake for Retailer Data**

In today's data-driven retail environment, the ability to collect, store, and analyze vast amounts of data is crucial for maintaining a competitive edge. A data lake is an effective solution for managing large volumes of structured and unstructured data. This document outlines the process of building a data lake for a retailer, focusing on five key datasets: customers, orders, reviews, products, and categories.

**Overview of the Retailer Data**

1. **Customers**: Contains information about the retailer's customers, including personal details and registration data.
2. **Orders**: Stores data related to customer orders, such as order dates, shipping information, and payment details.
3. **Reviews**: Includes customer feedback on purchased products, comprising ratings and review texts.
4. **Products**: Details about the products sold by the retailer, including descriptions, prices, and stock information.
5. **Categories**: Organizes products into categories, aiding in product management and customer navigation.

**Importance of a Data Lake**

A data lake allows for the centralized storage of all retailer data, enabling advanced analytics, machine learning, and business intelligence. By consolidating data from various sources, a data lake helps in:

* **Enhanced Data Accessibility**: Provides a single repository for all data, making it accessible to different departments.
* **Scalability**: Capable of storing vast amounts of data without performance degradation.
* **Flexibility**: Supports various data formats (structured, semi-structured, and unstructured) and can be queried using multiple tools.
* **Advanced Analytics**: Facilitates the application of analytics and machine learning to gain insights into customer behavior, sales trends, and product performance.

**Components of the Data Lake**

**Customers Table:**

|  |  |  |
| --- | --- | --- |
| Column | Description | Sample Data |
| CustomerID | Unique identifier for each customer | 1 |
| FirstName | Customer's first name | John |
| LastName | Customer's last name | Doe |
| Email | Customer's email address | john.doe@example.com |
| PhoneNumber | Customer's phone number | 1234567890 |
| Address | Customer's address | 123 Main St |
| City | City where the customer resides | Anytown |
| Gender | Customer's gender | M |
|  |  |  |

**Categories**:

|  |  |  |
| --- | --- | --- |
| Column | Description | Sample Data |
| CategoryID | Unique identifier for each category | 1 |
| CategoryName | Name of the category | Category 1 |
| Description | Description of the category | First category |
| ParentCategoryID | Unique identifier for the parent category | NULL |

**Orders:**

|  |  |  |
| --- | --- | --- |
| Column | Description | Sample Data |
| OrderID | Unique identifier for each order | 1 |
| CustomerID | Unique identifier for the customer | 1 |
| OrderDate | Date when the order was placed | 44348 |
| ShippingDate | Date when the order was shipped | 44349 |
| ShippingAddress | Address where the order is to be shipped | 123 Main St |
| BillingAddress | Billing address for the order | 123 Main St |
| OrderStatus | Status of the order | Shipped |
| TotalAmount | Total amount of the order | 100 |
| PaymentMethod | Method of payment used for the order | Credit Card |
| LastmodifiedDate | Last modified record of | 2024-06-04 10:00:00 |

**Reviews:**

|  |  |  |
| --- | --- | --- |
| Column | Description | Sample Data |
| ReviewID | Unique identifier for each review | 1 |
| CustomerID | Unique identifier for the customer | 1 |
| ProductID | Unique identifier for the product | 1 |
| Rating | Rating given by the customer (e.g., 1 to 5) | 5 |
| ReviewText | Text of the review written by the customer | Great product! |
| ReviewDate | Date when the review was submitted | 44352 |

**Products:**

|  |  |  |
| --- | --- | --- |
| Column | Description | Sample Data |
| ProductID | Unique identifier for each product | 1 |
| ProductName | Name of the product | Product 1 |
| ProductPrice | Price of the product | 50 |
| CategoryID | Unique identifier for the category | 1 |
| LastModifiedDate | Last updated record | 2024-06-25 10:00:00 |

**Technical Design:**

1. Implement the control table in Azure SQL, which contains source tables information like
2. What are the tables
3. Table loading criteria

(append, scd type1, scd type2 , fullload)

1. Incremental column
2. Last ingestion Date
3. Start Implementing ADF Piepline.
4. Using lookupactivity connecting to the control table and get details.
5. Lookupactivity output is input to foreach activity
6. Inside foreach activity implement copyactivity , notebookactivity and stored proc activity
7. For copy activity source is on prem sqlserver and sink is gen2lake.
8. By using self hosted IR establish connection to on prem sqlserver.
9. Create linked service and attach selfthosed IR and then create query based parameterized dataset and attached to copy activity source side.
10. Create a linked service to establish connection to ADLS (Gen2lake)
11. Create dataset for gen2lake and attach the linked service.
12. After copy activity we will get the data into gen2lake as json file

(parquet)

1. After copy activity , using the notebook activity call the Databricks notebook
2. Using Notebookactivity we can process the data and loaded into delta tables.
3. In the notebook using mount point we connect to ADLS create a dataframe on Raw data

And loaded to bronze zone.

1. In our project we maintained 3 types of zones
2. Bronze 2) silver 3) Gold
3. From bronze to silver apply the transformations like
4. Derive new columns
5. Date Transformations
6. Case conditions
7. Join operations
8. Window functions
9. As per the business requirement aggregate the multiple silver zone tables and loaded to gold zone which is for reporting purpose
10. After notebook activity call the stored procedure activity which will update the date in the control table for the next Run.
11. For this project we used scheduling trigger and we scheduled this pipeline daily once.

**Azure Services:**

1. Azure DataFactory
2. Azure Datalake gen2l
3. Azure Databricks
4. Azure Keyvault
5. Azure Sqlserver
6. logicapps