**Project Documentation: Phase 1 - Incremental Data Loading**

**Project Name:**

Azure Enterprise Car Project

**Objective:**

Implement data incremental loading to efficiently ingest and process only new or updated data into a scalable and optimized pipeline. This phase primarily focuses on incremental loading and setting up the foundational structure for future pipeline expansions.

**Technologies Used:**

* **Azure Data Factory (ADF)**
* **Azure Data Lake (Bronze, Silver, Gold layers)**
* **Azure SQL Database**
* **Stored Procedures**
* **SQL Server**

**Architecture Overview**

The architecture for this project follows a layered approach to ensure data scalability and efficient processing:

1. **Data Lake**:
   * **Bronze Layer**: Raw, unprocessed data.
   * **Silver Layer**: Cleaned and standardized data.
   * **Gold Layer**: Data ready for reporting and analysis.
2. **Azure Data Factory (ADF)**:
   * **Pipelines**: Used for data ingestion, processing, and management.
   * **Linked Services**: Connection points to SQL Server, GitHub, and Azure Data Lake.
   * **Datasets**: Parameterized datasets that make data ingestion dynamic.
3. **SQL Server**:
   * **Source Table**: Holds the original data (source\_cars\_data).
   * **Watermark Table**: Tracks the last load date to facilitate incremental data loading.

**Phase 1: Incremental Data Loading Setup**

**1. Initial Setup**

**Step 1: SQL Source Table Creation**

The source data table (source\_cars\_data) will contain the historical and ongoing transactional data for the car project. The schema for this table is as follows:

CREATE TABLE source\_cars\_data (

Branch\_ID VARCHAR(200),

Dealer\_ID VARCHAR(200),

Model\_ID VARCHAR(200),

Revenue BIGINT,

Units\_Sold BIGINT,

Date\_ID VARCHAR(200),

Day INT,

Month INT,

BranchName VARCHAR(200),

DealerName VARCHAR(200)

);

**Step 2: Watermark Table Creation**

A watermark table (water\_table) is used to track the last loaded date so that subsequent incremental loads can start from the most recent data.

CREATE TABLE water\_table (

last\_load VARCHAR(2000)

);

INSERT INTO water\_table VALUES ('DT00000'); -- Initial load placeholder date

**Step 3: Stored Procedure for Watermark Update**

A stored procedure (UpdateWatermarkTable) is created to update the watermark table after each successful load, ensuring that future loads pick up from the latest data.

DROP PROCEDURE UpdateWatermarkTable;

CREATE PROCEDURE UpdateWatermarkTable

@lastload VARCHAR(200)

AS

BEGIN

BEGIN TRANSACTION;

UPDATE water\_table

SET last\_load = @lastload;

COMMIT TRANSACTION;

END;

**2. Azure Data Factory Pipeline Creation**

**Step 1: Create Pipeline - source\_prep**

In Azure Data Factory (ADF), create a pipeline called **source\_prep** for data preparation and ingestion. This pipeline will handle the process of copying data from SQL Server into the data lake.

**Step 2: Create Linked Services**

1. **GitHub (HTTP link service)**: To pull data from GitHub (if relevant).
2. **Azure SQL Database (SQL Server)**: For connecting to the SQL source table.
3. **Azure Data Lake**: For storing raw data in the Bronze layer.

**Step 3: Dataset Creation**

* **Parameterized Dataset for Dynamic Control**:
  + Create a parameterized dataset (ds\_git) for the data source with a load\_flag parameter.
  + Example parameter usage: @{dataset.load\_flag()}.

**Step 4: Define Copy Data Activity**

Create the **Copy Data Activity** to pull data from the source (source\_cars\_data) to Azure SQL Database or Data Lake based on the parameters defined.

**Source Query**:

SELECT \* FROM source\_cars\_data WHERE DATE\_ID > @{activity('last\_load').output.value[0].last\_load}

AND DATE\_ID <= @{activity('current\_load').output.value[0].current\_load}

**Incremental Data Loading Process**

**1. Initial Load**

The first step in incremental loading is performing a full data load. This will pull all data from the source (SQL Server) and ingest it into the Data Lake. The starting date for this first load is defined as a **known cutoff date** (e.g., 2024-01-01).

* **Initial Load Query**:  
  Pull all records from the source table.

SELECT \* FROM source\_cars\_data WHERE DATE\_ID >= '2024-01-01';

**2. Incremental Load**

Incremental loading ensures that only new or updated records are loaded after the initial load. This is achieved by tracking the last successful load date in the watermark table.

**Incremental Logic:**

* Start with a defined **Starting Date** (e.g., 2024-01-01).
* Query the source data based on the last loaded date, selecting records where DATE\_ID is greater than the last load date.

**Incremental Load Query:**

SELECT \* FROM source\_cars\_data

WHERE DATE\_ID > '@{activity('last\_load').output.value[0].last\_load}'

AND DATE\_ID <= '@{activity('current\_load').output.value[0].current\_load}'

After each successful load:

* The **Starting Date** is updated to the latest DATE\_ID processed.
* The watermark table is updated using the **stored procedure** to reflect the new last load date.

**3. Data Sink and Storage**

The final data is stored in the **Data Lake**:

* **Bronze Layer**: Raw data from the SQL source.
* **Silver Layer**: Cleaned and transformed data.
* **Gold Layer**: Data prepared for analytics and reporting.

The incremental data is inserted into the **Bronze** layer of the Data Lake initially. It can later be moved through the **Silver** and **Gold** layers for further processing and optimization.