**Project Problem Statement: S3 to Snowflake ETL Pipeline Using AWS Glue**

**Objective**

Design and implement a secure and automated data pipeline that:

* Ingests retail sales data in JSON format into AWS S3.
* Uses AWS Glue for ETL (Extract, Transform, Load) operations.
* Loads processed data into Snowflake.
* Leverages Snowflake Streams and Tasks for managing incremental data loads.

**Tools & Services**

* AWS S3
* AWS Glue (Crawlers & Jobs)
* Snowflake (Stages, Tables, Streams, Tasks)

**Problem Statement**

Your organization is launching a new data analytics platform and wants to build a scalable ETL pipeline using cloud-native tools. You have been tasked with designing and implementing a data flow that satisfies the following requirements:

**Data Description**

You will work with **retail sales transaction data** in JSON format. Each record in the dataset contains:

* transaction\_id (string): Unique ID for the transaction
* product (string): Product name
* quantity (integer): Number of units sold
* price\_per\_unit (float): Price of each unit
* total\_amount (float): Total transaction value (price\_per\_unit \* quantity)
* location (string): Store location
* transaction\_date (date): Date of the transaction

The data will be uploaded into the /raw/ folder of an S3 bucket named retail-sales-data.

**Task Breakdown**

**Step 1: Ingest Raw Data to S3**

* Upload provided sample JSON data to s3://retail-sales-data/raw/

**Step 2: Schema Discovery Using AWS Glue Crawler**

* Create a Glue crawler that scans the /raw/ folder
* Store the cataloged metadata in a database named retail\_sales\_db

**Step 3: Transformation Using AWS Glue Job**

Create a Glue ETL job to:

* Read the cataloged data
* Perform the following transformations:
  + Drop duplicate transaction\_id records
  + Calculate a new field: discounted\_total = total\_amount \* 0.95 (assuming 5% discount)
  + Convert all product names to uppercase
  + Filter out records where quantity <= 0
* Write the transformed data as JSON to s3://retail-sales-data/processed/

**Step 4: Load Transformed Data into Snowflake**

* Create an external stage in Snowflake that references the /processed/ folder
* Define a JSON file format in Snowflake
* Load the data into a raw\_sales table using COPY INTO

**Step 5: Setup Incremental Load Logic**

* Create a reporting\_sales table that holds consolidated records
* Create a Snowflake Stream on the raw\_sales table
* Create a scheduled Task that:
  + Checks for changes via the stream
  + Inserts new/changed rows into reporting\_sales

**Step 6: IAM Roles & Policies**

* Design and document IAM roles with the least privileges for:
  + Data Engineers managing Glue
  + Users uploading files to S3
  + Snowflake accessing S3 via external stage integration

**Deliverables**

* Functional ETL pipeline from S3 to Snowflake
* Transformed data loaded into Snowflake tables
* Working Stream & Task for incremental load
* Clear documentation of IAM roles and policies
* A short report or demo showing successful implementation

**Constraints**

* No BI tools should be included
* Use only S3, Glue, and Snowflake

**Suggested Folder Structure**

* S3 Bucket: retail-sales-data
  + /raw/ — for unprocessed JSON files
  + /processed/ — for Glue output files

**Notes**

* Focus on security and modularity of the pipeline
* Ensure the solution is extensible for future data sources
* Automate as much of the process as possible
* Stream and Task should be reusable and handle continuous updates