Assignment: Customer Debit Card Purchase Aggregation

Overview

In this assignment, you will develop a data processing pipeline using AWS services and Python. The goal is to aggregate customer debit card purchases on a daily basis and update the aggregated data in a MySQL table hosted on Amazon RDS. You'll work with AWS S3 for data storage, AWS Glue for data processing, and Amazon RDS for data persistence.

Objectives

- Generate mock daily transaction data and store it in CSV files.
- Upload daily transaction CSV files to an AWS S3 bucket using a Hive-style partition.
- Set up a MySQL table in Amazon RDS to store aggregated transaction data.
- Write an AWS Glue job to process daily transactions from S3, aggregate them, and update the RDS MySQL table.

Instructions

- **Part 1: Generate Mock Data**
  - Write a Python script to generate mock data for daily transactions in CSV format
  - Each record should include **customer_id, name, debit_card_number, debit_card_type, bank_name, transaction_date, and amount_spend**.
  - Generate a new CSV file for each day's data.

- **Part 2: Store Data in S3 Data Lake**
  - Create an AWS S3 bucket for storing the daily transaction CSV files.
  - Upload the daily CSV files **(Using Python or AWS CLI)** to the S3 bucket. Use a Hive-style partitioning scheme like date=yyyy-mm-dd.

- **Part 3: Setup RDS MySQL Table**
  - Create a MySQL database instance in Amazon RDS.
  - Design and create a table to store aggregated transaction data. The table should include columns for **customer_id, debit_card_number, bank_name, and total_amount_spend.**

- **Part 4: AWS Glue Job for Data Aggregation**
  - Use AWS Glue to create a job that processes the daily transaction data from the S3 bucket in incremental manner
  - The Glue job should read the existing data from the RDS MySQL table, aggregate the new daily transactions into the total amount spent, and update the MySQL table accordingly.
  - Ensure the Glue job handles incremental updates, adding new customers and updating existing ones without duplication.

Deliverables

- Python Script for Mock Data Generation: A Python script that generates daily transaction data and saves it to CSV files.

- S3 Bucket and Uploaded Files: Provide the name of the S3 bucket and screenshots showing the files organized using the Hive-style partition.
- RDS MySQL Table Schema: The SQL statements used to create the RDS MySQL table for storing aggregated data.
- AWS Glue Job Script: The script used in AWS Glue for aggregating daily transactions and updating the RDS MySQL table. Include a description of how the script handles incremental data.

Submission Guidelines

- Submit all scripts and SQL statements as text files.
- Provide screenshots for the S3 bucket contents, showing the partitioning scheme.Include a brief report (1-2 pages) explaining your solution, how you structured the data in S3, the schema of your RDS MySQL table, and the logic behind your AWS Glue job.
- Ensure all code is commented to explain the functionality and logic.
- Architecture Diagram