E-Commerce Data Pipeline

Assignment Overview

This assignment involves building a sophisticated event-driven data ingestion and transformation pipeline focusing on e-commerce transactional data. Students will design a system using AWS services such as S3, Lambda, Glue, Redshift, and SNS to ingest, transform, validate, and upsert data into Amazon Redshift for analytical purposes.

Sample Record for Daily Transactions:

```
{
  "transaction_id": "TXN123456789",
  "customer_id": "C12345",
  "product_id": "P12345",
  "quantity": 2,
  "price": 29.99,
  "transaction_date": "2023-03-15",
  "payment_type": "Credit Card",
  "status": "Completed"
}
```

Assignment Tasks

- Mock Data Generation
  - Script: Develop a Python script to generate mock data for daily transactions and dimension tables for products and customers. Ensure variability in data for comprehensive testing.
  - Transaction Data: Generate daily transaction files in CSV format, stored using the following hive-style partitioning in S3: s3://your-bucket/transactions/year=2023/month=03/day=15/transactions_2023-03-15.csv.

- Dimension Tables and Sample Records
  - Products Dimension Table (dim_products):
    - Columns: product_id, product_name, category, price, supplier_id
    - Sample Record: P12345, "Widget A", "Gadgets", 29.99, "S123"
  - Customers Dimension Table (dim_customers):
    - Columns: customer_id, first_name, last_name, email, membership_level
    - Sample Record: C12345, "John", "Doe", "john.doe@example.com", "Gold"
  - Pre-load these dimension tables into Redshift as part of the setup process.

- Data Ingestion and Transformation with AWS Glue
    - Event-Driven Ingestion: Configure an AWS Lambda function to trigger AWS Glue jobs upon detecting new files in the S3 transactions folder.

- Data Transformation and Validation:
    - Join Operations: Enrich transactional data by joining with dim_products and dim_customers based on product_id and customer_id.
    - Data Validation: Include validation logic in the Glue job to filter out transactions with invalid customer_id or product_id (e.g., missing in dimension tables).
    - Additional Transformations: Calculate the total transaction amount (quantity * price) and categorize transactions into different classes based on the amount (e.g., "Small", "Medium", "Large").
    - Upsert Operation in Amazon Redshift
    - Design the Glue job to perform an upsert operation into the fact_transactions table in Redshift, using transaction_id as the key. Consider transaction date and status when determining if an existing record should be updated.

- Notifications and Archiving
    - SNS Notifications: Configure Amazon SNS to send email notifications upon the successful completion or failure of Glue jobs, detailing the job status and any errors encountered.
    - Archiving Process: Implement a Lambda function to archive processed daily transaction files from the primary S3 transactions bucket to an S3 archive bucket, maintaining the hive-style partitioning scheme.

- Deliverables
    - Python scripts for mock data generation.
    - Source code for Lambda functions, including event handlers for S3 events, Glue job initiation, and the archiving process.
    - Glue ETL scripts for data transformation, validation, and upsert logic.
    - Comprehensive documentation detailing:
        - The pipeline architecture and data flow.
        - Data models for dimension and fact tables in Redshift.
        - Instructions for setting up AWS resources, deploying the solution, and monitoring job execution.