Retail Data Management

Overview

This project enhances ABC Retail's data management capabilities using AWS Glue, an ETL service, to streamline data processing workflows and derive actionable insights from sales and product data. It aims to integrate and analyze data from multiple sources to understand customer behavior, product performance, and market trends. This integration leads to data-driven decisions that optimize inventory management, marketing strategies, and overall business performance.

Instructions

- Review the learning materials in the ETL course
- Carefully read the situation, tasks, actions, and result sections to grasp the assignment fully
- Complete and submit your assignment via the Learning Management System (LMS)
- Follow the provided guidelines closely, ensuring your report includes all required analyses and interpretations

Situation

You are a data analyst at ABC Retail, tasked with improving data processing and analysis workflows. ABC Retail aims to leverage AWS Glue to streamline its data processing workflows and derive actionable insights from its sales and product data. Your role is crucial in unlocking insights from the company's vast stores of data to drive business growth and enhance operational efficiency.

Task

Your task is to use AWS Glue to join the Products and Orders tables based on ProductID, ensuring data integrity. Develop a script to cleanse the Sales column, converting it to a numerical format. Create a transformation to calculate net sales and remove duplicates for optimized analysis, then summarize the average sales by category and ship mode.

Action

1. **Login to the AWS Console:**
    - Open your web browser and navigate to the AWS Management Console
    - Log in with your AWS account credentials
    - Navigate to **S3**
    - Click on **Create bucket** and add the bucket name as **etl-cep-01**. Scroll down the screen and click on the **Create bucket** button.

2. **Create two subfolders inside the etl-cep-01 bucket**
    - Click on **Create folder**, add the folder name as **transaction-files** for the first folder and **product-files** for the second folder. Scroll down the screen and click on the **Create folder** button.
    - Inside **product-files** folder, click on **Upload** then click on **Add files.** Select the **product details** file from your local system and click on Open. Scroll down and click on **Upload**. Repeat the same step for transaction files and upload the transactions dataset.

3. **Create a new bucket**
- Create a new bucket named **etl-cep-output-01** just as you did above

4. **Navigate to AWS Glue and create a new database**
    - In the AWS management console, search for **AWS Glue** and select it
    - In the AWS Glue console, navigate to the **Databases** section
    - Click on **Add database and** provide the name as **abc-retail**. Scroll down and click on **Create database**

5. **Set up two classifiers to read transaction data and product data**
    - Navigate to the Data Catalog, click on **Classifiers** then click on **Add classifier**
    - Fill the first classifier details as given below, then click on **Create**
        - **Classifier name** as **cust_classifier**
        - **Classifier type and properties** as **CSV**
        - **CSV Serde – optional** as **None**
        - **Column delimiter** as **comma(,)**
        - **Quote symbol as Double-quote(“)**
        - **Column headings** as **Has headings**

    - Fill in the second classifier details as given below, then click on **Create**
        - **Classifier name** as **txnClass**

- **Classifier type and properties** as **CSV**
- **CSV Serde – optional** as **None**
- **Column delimiter** as **comma(,)**
- **Quote symbol** as **Double-quote(“)**
- **Column headings** as **Has headings** and fill in the details as given below:
  - **Order ID, Order Date, Ship Date, Aging, Ship Mode, Product ID, Sales, Quantity, Discount, Profit, Shipping Cost, Order Priority, Customer ID**

6. **Create IAM role**
   - In the AWS management console, search for **IAM** service and select it
   - Navigate to **Roles** and click on **Create role**
   - Select **AWS service** as the **Trusted entity type** and **Glue** as the **Use case** then click on **Next**
   - Select the **AdministratorAccess** policy. Scroll down and click on **Next.**
   - Enter the name as **glue-role**. Scroll down and click on **Create role.**

7. **Set up a Crawler**

   - Navigate to **AWS Glue** and click on **Databases** from the Data Catalog and select **abc-retail** database
   - Click on **Add tables using a crawler**
   - Enter the name as **retail-crawl** and click on **Next**
   - Click on **Add a data source**
   - Click on **Browse S3** and click on **etl-cep-01** then **select transaction-files/** and click on **Choose**
   - Click on **Add an S3 data source**
   - Choose classifier as **txnClass** from the drop down of **custom classifiers – optional** and click on **Next**
   - Choose **glue-role** in the **IAM role** section and click on **Next**
   - Choose **Target database** as **abc-retail** and enter the table name prefix as **txn** and click on **Next**
   - Click on **Create crawler**
   - Click on **Run crawler**

Note: Repeat above steps for other Product dataset as well. While choosing classifier choose cust_classifier.

8. **Create ETL job**
    - Navigate to **AWS Glue**, click on **ETL jobs**, and click on **Visual ETL**
    - In the **Add nodes**, double-click on **AWS Glue Data Catalog**
    - Select **Join** from the add nodes and link **Join** to both the **AWS Glue Data Catalog**
    - Click on the **Join** box, and then select **Drop Fields** from the **Add nodes**
    - Click on the **Drop Fields** box, and then select **Regex Extractor** from the **Add nodes**
    - Click on the **Regex Extractor** box, and then select **Aggregate** from the **Add nodes**
    - Click on the **Aggregate** box, and then select **Amazon S3** from the **Targets** in **Add nodes**
    - Click on the first **AWS Glue Data Catalog** box, and select **abc-retail** in the **Database** dropdown and select **txntransaction_files** under the **Table** dropdown
    - Click on the second **AWS Glue Data Catalog** box and select **abc-retail** in the **Database** dropdown and select **product_files** under **Table** dropdown
    - Select the **Join** box and add both the **AWS Glue Data Catalog** in the node parents. Select **Inner join** in the **Join type** and in the **Join conditions** box select **product id** in both the **AWS Glue Data Catalog** boxes
    - Click on the **Drop Fields** box and in the **DropFields** section select **product id** as it appears twice
    - We need to extract sales values as it has $ symbol in it. Click on **Regex Extractor** and fill the following fields:
        - **Column to extract from** as **sales**
        - **Regular expression** as **\d+**
        - **Extracted column** as **NetSales**

> **Note:** Before selecting the newly generated column in the aggregation section, **you must wait** for the **Regex Extractor data preview to load fully**.

    - Now, let's create our summary report using **Aggregate** block**.**
      Fill the following fields:
        - **Fields to group by** as **product category** and **ship mode**
        - **Field to aggregate** as **sales**
        - **Aggregation function** as **avg**

- Click on **Amazon S3** block then click on **Browse S3** and select **etl-cep-output-01**
- On the top left corner click on **Untitled job** and give the **etl-cep-job,** click on **Save** and the click on **Run**
- Check the progress in **Runs**

9. **Check the output and run the sql query**
   - Navigate to **S3** bucket to check output in **cep-etl-output-01** bucket
   - Click on **run-1715077987267-part-block-0-r-00000-snappy.parquet**
   - Navigate to **Object actions** and select **Query with S3 Select**
   - In the **Output settings** select **CSV** format and click on **Run SQL query**
   - The output appears as shown below

Result

Create a Word document with the detailed steps that you have performed with the screenshots. Upload the solution document to the Learning Management System (LMS).