AWS Athena for analysing smartphone data.

Business Overview:

Amazon Athena is a query service that allows you to analyse data directly in Amazon S3 using conventional SQL. Using a few clicks in the AWS Management Console, you can aim Athena at Amazon S3 data and start running ad-hoc searches with traditional SQL in seconds. Because Athena is serverless, you don't have to worry about setting up or maintaining infrastructure, and you just pay for the queries you perform. Even with big datasets and sophisticated queries, Athena grows automatically while processing queries in parallel, resulting in fast responses. In addition, Athena supports a variety of data formats, including CSV, JSON, ORC, Parquet, and AVRO.

Data Pipeline:

A data pipeline is a technique for transferring data from one system to another. The data may or may not be updated, and it may be handled in real-time (or streaming) rather than in batches. The data pipeline encompasses everything from harvesting or acquiring data using various methods to storing raw data, cleaning, validating, and transforming data into a query-worthy format, displaying KPIs, and managing the above process.

Dataset Description:

This dataset contains information on various smartphone models, including price, rating, specifications (SIM, processor, RAM, battery, display, camera), memory card support, and operating system. It enables analysis for consumers to compare features and make informed purchasing decisions.

➔ Languages- ● SQL, Python3 (PySpark)

➔ Services - ● AWS S3, AWS Glue, AWS Athena, Amazon CloudWatch, Power BI

Amazon S3:

Amazon S3 is an object storage service that provides manufacturing scalability, data availability, security, and performance. Users may save and retrieve any quantity of data using Amazon S3 at any time and from any location.

AWS Glue:

A serverless data integration service makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development. It runs Spark/Python code without managing Infrastructure at a nominal cost. You pay only during the run time of the job. Also, you pay storage costs for Data Catalog objects. Tables may be added to the AWS Glue Data Catalog using a crawler. The majority of AWS Glue users employ this strategy. In a single run, a crawler can crawl numerous data repositories. The crawler adds or modifies one or more tables in your Data Catalog after it's finished.

Learnings:

- there are 4 ways to create tables in Athena.
    1. Glue crawler
    2. Create table Query
    3. Ctas (create table as Query)
    4. Aws glue job With Pyspark Job
- Athena Pricing Depend on Scan of your data.
- Athena follows a pay-per-query pricing model where you only get charged for the amount of queried data. Athena costs $5/TB or 5 cents/10 GB.
- In Athena double quote is the column name and single quate is the string
- At one time in partition, it only allows 100 partition it provide multiple partition error
- To tackle with it create table “with no data” and then load data into it
- Query run time reduced for parquet format than csv
- then if we use bucketing and partition then also scan reduce and help save cost
- data repartition support on Athena but data Re-bucketing is not after bucketing or create the table.
- take care of query performance and cost reduction using compression of different file formats and algorithm utilize for compression and decompression

  like Gzip, bzip2, lzo, snappy check official documentation to read more about it.

Optimizations for query performance:

- order by: use limits in query
- joins: always write big data table on left side and small data table on right side it reduces the run time
- Group by: always write high cardinality (unique) value first and then lowest last.
- Distinct count: distinct in distinct give accurate number of distinct count but if you go with approx. distinct it gives related number but faster result 2.3% in original result, we utilize it for where we need to take random count like count million row.

Athena to Power BI connection:

- Download Athena ODBC Driver: You'll need to download and install the Athena ODBC driver from the Simba website or through the AWS documentation.
- Configure Access and Policies for Roles:
  1. In AWS IAM, create a role with the necessary permissions for accessing Athena and any other related services.
  2. Attach policies such as AmazonAthenaFullAccess, AmazonS3FullAccess, AdministrationFullAccess.

- Generate New Access Key:
  1. Navigate to IAM in the AWS Management Console.
  2. Select the user for whom you want to generate a new access key.
  3. In the Security credentials tab, click on "Create access key".
  4. Make note of the Access Key ID and Secret Access Key securely.

- Set Up ODBC Connection:
  1. Open the ODBC Data Source Administrator.
  2. Add a new System DSN (Data Source Name) using the Simba Athena ODBC driver.
  3. Configure the connection details, including authentication with the Access Key ID and Secret Access Key.
  4. Test the connection to ensure it's working properly.

- Provide S3 Path:

  1. You'll need to specify the S3 path where your Athena query results will be stored. This path should be accessible by the user/role you've set up.

- Use in Power BI:
  1. Open Power BI and go to the Get Data option.
  2. Search for ODBC and select it.
  3. Enter the ODBC connection details, including the DSN name, Access Key ID, Secret Access Key, and any other necessary information.
  4. Power BI will connect to Athena using the configured ODBC connection.

Key Takeaways:

- Understanding the project Overview and Architecture
- Table creation using Glue Crawler
- Table creation using CTAS
- Table creation using DDL
- Understanding Athena Partitioning
- Understanding Athena Bucketing
- Exploring Joins in Athena
- Optimizations in Athena
- Creating Athena Workgroup
- Understanding Athena File Formats
- Understanding Athena Pricing