**University Canada West**

**Portfolio Project( Individual Assignment )**

**Course: BUSI 653 (HBD-WINTER25-06)  
Cloud Computing Technologies**

**Professor name – Mahmood Mortazavi Dehkordi**

**Kuldeep Singh (2331953)**

**Date: March 27, 2025**

**Exploratory Data Analysis**

**Methodology**

***Data Collection and Preparation***

**Step -01-Data Ingestion**

**Why I do it** – I did this action for extract the data from operational environment of city of Vancouver to the data analytics Platform for proceed it further here .

**How I did it** – I complete this step by preparing a S3 bucket namely Inq-raw-kul in which I create a path inq-raw-kul, inquiry, animal- general -inquiry- list , year for uploading the extract data from city of Vancouver website into this path .

**Screenshot of Data ingestion –**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Step-02- Data profiling –**

**Why I do it –** I did this step for finding the data issues in the extracted data set from operational environment in data ingestion process .

**How I did it** – I complete this step by using data brew service from Aws console Platform in which I run a project namely bus-Inq-ani-list-prj-kul associated with data set bus-Inq-ani-list-ds-kul for finding the issues in the data issues but I did not any data issue .

**Screenshot of Data profiling**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Step-03- Data Cleaning**

**Why I do it –** I performed this step to clean the issues in data profiling step but I cannot found any issue therefore I just changed the invalid column names into valid column names .I Store the cleaning files in new S3 bucket inq-trf-kul in user friendly and system friendly folders.

**How I did it** – I did it applying cleaning function namely Rename which can be easily seen in added screenshot.

**Screenshot of Data Cleaning -**

A screenshot of a computer

AI-generated content may be incorrect.

**Screenshots of Storing cleaning files into inq-trf-kul S3 bucket –**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step-04- Data Cataloging**

**Why I do it –** I do it for converting the CSV file from transform bucket to data cataloguing zone to extract the table for further analysis .

**How I did it** – I complete this step by using the crawler service of AWS glue function . firstly I Crate the vacant database namely inq-ani-data-catalog-kul in which I store a table namely inq-animal-trf-system by applying crawler function of AWS Glue service .

**Screenshot of Data cataloguing –**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Descriptive analysis**

**Step-05- Data Summarization**

***Why I do it* –** I did it to Summarise the data set which I stored in catalogue part for getting some meaningful information from this data catalogue to generate descriptive analysis question after summarization .

***How I did it*** - I run the ETL job of AWS Glue service of AWS console in which apply change schema service for reduce the unnecessary columns and filter for reducing the unnecessary rows and grouping and aggregating operation to summarise the dataset properly which I would use to generate some descriptive analysis question from last summarised result . After doing the summarised part I stored the summarised files into new S3 Bucket namely inq-cur-kul in System and users folders separately and in database catalogue in table which you can see in the screenshot which I added in Data cataloguing step .

**Screenshots of Data Summarization –**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.** **A screenshot of a computer screen

AI-generated content may be incorrect.**

**Descriptive Analysis questions of final summarized information -**

***Final Summarised Table* –**

|  |  | |  | | |
| --- | --- | --- | --- | --- | --- |
| **Channel** | | **avg(numberofrecords)** | | **Reportdate** |
| Social Media | | 1.6428571428571428 | | 2025-03-02 16:24:00 |
| Chat | | 11.767123287671232 | | 2025-03-02 16:24:00 |
| E-mail | | 4.845360824742268 | | 2025-03-02 16:24:00 |
| Phone | | 42.21212121212121 | | 2025-03-02 16:24:00 |
| Mail | | 17.5 | | 2025-03-02 16:24:00 |

***Some Descriptive Analysis questions which generated from this final summarized table –***

Q1.Which communication channel has the highest average number of records for animal general inquiries?

Q2. Which channel has the lowest average number of records, and what does this suggest about its usage?

Q3. How does the average number of records compare between digital channels (Social Media, Chat, E-mail) and traditional channels (Phone, Mail)?

**Athena for creating business Questions**

***Business Q1***

A screenshot of a computer

AI-generated content may be incorrect.

***Business Q2***

A screenshot of a computer

AI-generated content may be incorrect.

***Business Q3***

A screenshot of a computer

AI-generated content may be incorrect.

***Explanation***

I constructed three business queries in AWS Athena by executing SQL queries against structured data stored in Amazon S3. I carried out registration of schemas in AWS Glue, and data were rendered analysable. I normalized SQL queries for insight extractions such as biid\_avg, min biid\_avg, and biid\_avg by channel by group on Athena. I executed the following commands on the Athena console, verified the output, and saved the data as necessary to be used at a future time. This provided me with efficient large data set processing without infrastructure that needed to be supported by me, which enabled cost-effective and scalable data-driven decision-making.

**Data Wrangling**

**Methodology**

***Step -01-Data Ingestion***

**Why I do it** – I did this action for extract the data from operational environment of city of Vancouver to the data analytics Platform for proceed it further here .

**How I did it** – I complete this step by preparing a S3 bucket namely Inq-raw-kul in which I create a path inq-raw-kul, inquiry, animal- general -inquiry- list , year for uploading the extract data from city of Vancouver website into this path .

**Screenshot of Data ingestion –**

**A screenshot of a computer

AI-generated content may be incorrect.**

***Step-02- Data profiling –***

**Why I do it –** I did this step for finding the data issues in the extracted data set from operational environment in data ingestion process .

**How I did it** – I complete this step by using data brew service from Aws console Platform in which I run a project namely bus-Inq-ani-list-prj-kul associated with data set bus-Inq-ani-list-ds-kul for finding the issues in the data issues but I did not any data issue .

**Screenshot of Data profiling**

**A screenshot of a computer

AI-generated content may be incorrect.**

***Step-03- Data Cleaning***

**Why I do it –** I performed this step to clean the issues in data profiling step but I cannot found any issue therefore I just changed the invalid column names into valid column names .I Store the cleaning files in new S3 bucket inq-trf-kul in user friendly and system friendly folders.

**How I did it** – I did it applying cleaning function namely Rename which can be easily seen in added screenshot.

**Screenshot of Data Cleaning -**

A screenshot of a computer

AI-generated content may be incorrect.

**Screenshots of Storing cleaning files into inq-trf-kul S3 bucket –**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

***Step-04- Data Cataloging***

**Why I do it –** I do it for converting the CSV file from transform bucket to data cataloguing zone to extract the table for further analysis .

**How I did it** – I complete this step by using the crawler service of AWS glue function . firstly I Crate the vacant database namely inq-ani-data-catalog-kul in which I store a table namely inq-animal-trf-system by applying crawler function of AWS Glue service .

**Screenshot of Data cataloguing –**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

***Step-05- Data Summarization***

**Why I do it –** I did it to Summarise the data set which I stored in catalogue part for getting some meaningful information from this data catalogue to generate descriptive analysis question after summarization .

**How I did it** - I run the ETL job of AWS Glue service of AWS console in which apply change schema service for reduce the unnecessary columns and filter for reducing the unnecessary rows and grouping and aggregating operation to summarise the dataset properly which I would use to generate some descriptive analysis question from last summarised result . After doing the summarised part I stored the summarised files into new S3 Bucket namely inq-cur-kul in System and users folders separately and in database catalogue in table which you can see in the screenshot which I added in Data cataloguing step .

**Screenshots of Data Summarization –**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Data Quality Control**

**Screenshots of ETL job for Quality Check**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Explanations**

I ran an ETL job on AWS Glue that performed a data quality check on the Inquiry Data in the raw zone of an S3 bucket for the City of Vancouver. The goal was to confirm if data of good enough quality to facilitate processing thereof existed. Quality data must exist to facilitate effective analysis and sound decision-making.  
 We need to validate data quality because low-quality data can lead to deceptive conclusions. I needed to make sure the data is complete, unique, and up-to-date so it will not be corrupted by missing values, duplicate records, or out-of-date information. By performing these validation activities, I was able to make data accurate and reliable for consumption by downstream processes. Aside from that, data defects were allowed to be tracked along with data governance transparency by differentiating quality check results.  
 To accomplish all of this, I developed an AWS Glue ETL job which extracted the Inquiry Data from the raw zone. The job also applied three significant data quality rules. The completeness rule searched for missing information in critical columns. The uniqueness rule searched for duplicated information in critical columns. The data freshness rule searched for fresher data and not stale data.  
 During the process of data, I maintained failed and passed rule output in a Quality Check directory of S3 bucket Transform Zone. I also used auto-balancing in a manner that there was one final output file stored rather than several copies so that the process could be cost-efficient.  
 This kept lower quality data from being passed on to downstream processing, helping ensure data integrity and enhance overall performance in data-driven decision making.

**Data monitoring and Controlling**

**Screenshots of Implementation of data monitoring and controlling in AWS Console**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Explanation** – I utilized AWS CloudWatch to monitor and track important parameters of my data infrastructure. I created a dashboard with three simple metrics on which I monitor system performance and determine likely reasons for failure. Two of them are derived from the unprocessed region of my S3 bucket: bucket size (threshold level: 40KB) and number of objects with same file (threshold level: 8 files). Monitoring these maintains my data storage within expected thresholds and avoids unwarranted copying of large files. The third one is based on AWS Glue job execution time, and 1% is taken as the reference point. This makes me aware of performance slowdowns in my ETL jobs and helps me streamline data processing.  
 In order to monitor these values, I created AWS Simple Notification Service (SNS) topics for real-time notification. I created three CloudWatch alarms that subscribe to an SNS topic so that I will receive a notification the moment any threshold is crossed. This will allow me to take action in time to avoid system slowdown, abuse of storage, or wasteful processing of data. With these controls and oversight, I can offer better data governance, cost management, and system responsiveness for my AWS infrastructure.