**Lab Exercise 1– Investigating Datasets Using SageMaker SQL**

**1. Lab Title**

**Hands-on: Investigating Datasets Using SageMaker SQL**

**2. Objective**

The objective of this lab is to:

* Understand how to use **SQL queries** inside **Amazon SageMaker** for dataset exploration.
* Perform **data inspection**, **filtering**, **aggregation**, and **statistical analysis** directly from a Jupyter notebook.
* Demonstrate how SQL can simplify dataset analysis without writing complex Python loops.

By the end of this lab, you will be able to:  
Load data from S3 or local files into SageMaker  
Query data using SQL with **DuckDB** or **PandasSQL**  
Generate useful insights such as averages, maximums, and group-based summaries

**3. Prerequisites**

Before starting this lab, ensure you have:

* An active **AWS Account** with **Amazon SageMaker Studio** access.
* A working **SageMaker domain and user profile**.
* A **Jupyter Notebook kernel** (Python 3 – Data Science).
* A dataset file (for example, house\_prices\_dataset.csv) uploaded to S3 or your local SageMaker environment.
* Basic familiarity with **SQL** and **Python Pandas**.

**4. Dataset Description**

You’ll use a sample dataset called **house\_prices\_dataset.csv**, which contains details about house listings in various cities.

| **Column** | **Description** |
| --- | --- |
| id | Unique identifier for each property |
| location | City name (e.g., Mumbai, Delhi, Bangalore) |
| area | Area of the property in square feet |
| bedrooms | Number of bedrooms |
| bathrooms | Number of bathrooms |
| price | Selling price (in INR) |

**5. Lab Setup**

**Step 1: Launch SageMaker Studio**

1. Log in to the **AWS Management Console**.
2. Navigate to **Amazon SageMaker → SageMaker Studio**.
3. Select your **User Profile** and click **Launch app → Studio**.
4. Wait for the environment to load — this opens the SageMaker Jupyter interface.

**Step 2: Create a New Notebook**

1. Click **File → New → Notebook**.
2. Select the kernel: **Python 3 (Data Science)**.
3. Rename your notebook to:
4. ***Investigating\_Datasets\_Using\_SQL.ipynb***

**6. Lab Tasks**

**Task 1: Import Required Libraries**

In your notebook’s first code cell, import the following:

!conda install -y -c conda-forge duckdb

import pandas as pd

import duckdb

import boto3

**Explanation:**

* pandas is used for handling tabular data.
* duckdb allows you to run SQL queries on Pandas DataFrames.

**Task 2: Load the Dataset**

# Load dataset from S3

dataset\_path = "s3://kedar-bankapplication/DATASETS/housing.csv"

df = pd.read\_csv(dataset\_path)

# Display first few records

df.head()

**Expected Output:**

A preview of the first five rows of your dataset should appear, confirming successful loading.

**Task 3: View Dataset Summary**

To understand your dataset’s structure:

df.info()

df.describe()

**Explanation:**

* .info() gives column names, data types, and non-null counts.
* .describe() provides basic statistical summaries such as mean, min, max, and quartiles.

**Task 4: Query the Dataset Using SQL**

Let’s explore the data using **DuckDB SQL** commands.

**Example 1: View First 10 Records**

duckdb.query("SELECT \* FROM df LIMIT 10").df()

**Example 2: Retrieve Average, Maximum, and Minimum Prices**

duckdb.query("""

SELECT

AVG(price) AS avg\_price,

MAX(price) AS max\_price,

MIN(price) AS min\_price

FROM df

""").df()

**Example 3: Count Houses by Location**

duckdb.query("""

SELECT location, COUNT(\*) AS total\_houses

FROM df

GROUP BY location

ORDER BY total\_houses DESC

""").df()

**Example 4: Filter Expensive Houses**

duckdb.query("""

SELECT \*

FROM df

WHERE price > 10000000

ORDER BY price DESC

""").df()

**Example 5: Find Average Price by Bedroom Count**

duckdb.query("""

SELECT bedrooms, ROUND(AVG(price), 2) AS avg\_price

FROM df

GROUP BY bedrooms

ORDER BY bedrooms

""").df()

**Task 6: Save Query Results**

avg\_price\_df.to\_csv("avg\_price\_by\_city.csv", index=False)

print("Saved average price data as avg\_price\_by\_city.csv")

**Explanation:**  
This saves your SQL output as a new dataset for future analysis or visualization.

**7. Validation and Observation**

After running all steps:

* Verify that all SQL queries return meaningful results.
* Check that you can visualize patterns (e.g., which cities are most expensive).
* Discuss which attributes (area, bedrooms, bathrooms) influence price the most.

**8. Cleanup**

To avoid AWS charges:

* Shut down your notebook instance from **SageMaker Studio → File → Shutdown → All kernels**.
* Optionally delete temporary files and datasets.