Lab 17: Read a CSV File, Write It, and Run SQL Queries on the DataFrame

Author: Dr. Sandeep Kumar Sharma

🎯Lab Overview

In this hands-on exercise, you will: 1. Read a CSV file using Databricks 2. Write that data into a Delta table 3. Run SQL queries on the data to extract meaningful insights

This lab is designed in a beginner-friendly, classroom-style manner.

Learning Objective

By the end of this lab, you will be able to: - Load CSV data into a DataFrame - Write the data into Delta format - Register a Delta table for SQL usage - Run SQL queries (filter, aggregation, sorting)

🛼Dataset Assumption

We assume the CSV file is stored in DBFS:

```
/FileStore/tables/employee_data.csv
```

Sample data:

```
id,name,department,salary,age
1,John,IT,65000,29
2,Asha,HR,72000,31
3,Raj,Finance,68000,28
4,Meena,IT,75000,36
5,David,HR,54000,26
```

STEP 1 — Read the CSV File

```
input_path = "/FileStore/tables/employee_data.csv"

df = spark.read.csv(input_path, header=True, inferSchema=True)
display(df)
```

This loads the CSV as a Spark DataFrame.

STEP 2 — Write Data Into Delta Format

We will write the DataFrame into a Delta location.

```
output_path = "dbfs:/mnt/employee_delta/"

df.write.mode("overwrite").format("delta").save(output_path)
```

STEP 3 — Register Delta Table for SQL Queries

Let's register the Delta path as a SQL table.

```
spark.sql("DROP TABLE IF EXISTS employee_table")
spark.sql(f"CREATE TABLE employee_table USING DELTA LOCATION '{output_path}'")
```

Check the table:

```
SELECT * FROM employee_table;
```

STEP 4 — Run SQL Queries for Insights

Now let's perform analysis.

4.1 — Find employees older than 30

```
SELECT id, name, department, age
FROM employee_table
WHERE age > 30;
```

4.2 — Find highest salary per department

```
SELECT department, MAX(salary) AS highest_salary
FROM employee_table
GROUP BY department;
```

4.3 — Count employees in each department

```
SELECT department, COUNT(*) AS employee_count
FROM employee_table
GROUP BY department;
```

4.4 — Find top 3 highest-paid employees

```
SELECT *
FROM employee_table
```

```
ORDER BY salary DESC
LIMIT 3;
```

STEP 5 — Load Delta Table Back as DataFrame (Optional)

```
delta_df = spark.read.format("delta").load(output_path)
display(delta_df)
```

What You Learned

- How to read CSV using Spark
- How to write data in Delta format
- How to create a Delta-backed SQL table
- How to perform SQL analytics inside Databricks

This is the foundation of almost all BI + analytics pipelines.

End of Lab 17

If you want, the next lab can be: - **Lab 18 – Run Advanced SQL Queries (Window functions, Joins, Aggregations)** - **Lab 19 – Build a Bronze → SQL Analytics layer using Delta**