IST769 Lab E

SQL Over Anything with Drill and Spark

In this lab, we will explore the power of Apache Drill for ad-hoc SQL query over your data. Drill data can be stored in a variety of places and in a variety of formats. We will then demonstrate SQL support in Apache Spark.

Learning Outcomes

At the end of this lab you should be able to:

- Query data from a variety of sources with Spark SQL
- Handle Structured and Semi-structured data with Drill
- Recognize the differences between Drill SQL and Spark SQL

Pre-Requisites

Before you begin:

- Open a terminal window in the lab environment
- Set the current working directory to **advanced-databases**
- Start the following services required by Drill, HDFS, Spark and Minio: **jupyter drill minio namenode datanode**

Tools Used In this Lab

The following tools will be used in this lab:

1. To access Jupyter Lab from your Windows host:
   http://localhost:8888
   The password is **SU2orange!**
2. Log-in to the drill Web UI from your windows host:
   https://localhost:8047
3. To access the Minio web client:
   http://localhost:9000
   access key: minio
   secret key: **SU2orange!**
4. To access the HDFS web client:
   http://localhost:50070
5. Drill Storage plugins can be found in the **drill-storage-plugins** folder. Run
   ```
   PS: advanced-databases> code .
   ```
   from the command line to open the code editor.

Lab Problem Set

Minio Setup:

1. Create a **labe** bucket
2. Download this file: https://raw.githubusercontent.com/mafudge/datasets/master/weather/syracuse-ny.csv ,
3. Upload it to the **labe** bucket.

QUESTIONS:

For each question, include a copy of the code required to complete the question along with a screenshot of the code and a screenshot of the output.

1. Configure a Drill storage plugin for the Minio **labe** bucket. Then write a drill query for **syracuse-ny.csv** to demonstrate you can read the file with headers.

2. Write a Drill SQL Query to get the overall average min and max temperatures by year and month. Use drill's SPLIT() function to separate Year, Month. You might need to use cast() to ensure the min and max temperatures are numeric types. Your output should include 4 columns: Year, Month, the average minimum temperature for that month, and the average maximum temperature for that month.

3. Create a view called **monthly_syracuse_weather_averages** from the query you wrote in question 2 and store it back on the **labe** bucket. (If you cannot get question 2 working, use a similar query). Provide your drill SQL code and a screenshot showing the view file is on the Minio bucket.
   NOTE: If you get an error about an immutable object, you need to change your storage config so you can write to the storage location.

4. Use the view you created in question 3 to show the weather data only the month of July.

5. Configure spark to read from Minio **labe** bucket, then load **syracuse-ny.csv** into a DataFrame and register it as the table **weather**

6. Rewrite question 2 using pure Spark SQL and the **weather** temp view. NOTE: There will be some subtle differences with how you must write the code, so be sure to **printSchema()** so you can see what the columns are.

7. Save the output from the DataFrame in question 6 to the temp view **monthly_syracuse_weather_averages**. Prove the view is there by querying it.

8. CHALLENGE YOURSELF! At the bottom of the **work/content/E-Drill-Spark.ipynb** file there is a section Called "Big Data to Small Data". Try to write a complete program that:

a. Inputs a month 1 – 12 at run-time.
b. Displays a scatter plot of min/max average monthly temperatures, where year is on the X-Axis.

IMPORTANT: When you are finished with the lab, execute:

```
PS:> docker-compose stop
```

To turn off all running services, then shut down your Azure Lab instance.