

Name: Mrunali Katta

ID: 017516785

Github link: <https://colab.research.google.com/drive/1luhq0WYKJMp26Ca5WIpPfl-bAmoHBdx4?usp=sharing>

## Homework 9

1. Set up Snowflake credentials in Google Colab secrets (+1 pt)

CO

Mrunali Katta - DATA 226 - Homework 9.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

☰

Secrets

📁 ✕

🔍

Configure your code by storing environment variables, file paths, or keys. Values stored here are private, visible only to you and the notebooks that you select.

<>

Secret name cannot contain spaces.

{x}

Notebook access

🔑

📁

	Name	Value	Actions
<input checked="" type="checkbox"/>	snowflake_accou	.....	👁 📄 🗑
<input checked="" type="checkbox"/>	snowflake_useric	.....	👁 📄 🗑
<input checked="" type="checkbox"/>	snowflake_passw	.....	👁 📄 🗑

+ Add new secret

Gemini API keys ▾

2. Download the 7 log files using the links provided in the demo Colab notebook (+1 pt)

➡ To download the sample .log.gz files using wget in Colab

CO

Mrunali Katta - DATA 226 - Homework 9.ipynb

☆ ☁

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text

☰

11s

🔍

<>

{x}

🔑

📁

▶

```
!wget -nc https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_1.log.gz
!wget -nc https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_2.log.gz
!wget -nc https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_3.log.gz
!wget -nc https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_4.log.gz
!wget -nc https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_5.log.gz
!wget -nc https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_6.log.gz
!wget -nc https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_7.log.gz
```

🔗

--2025-04-19 03:54:44-- https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample\_web\_log\_1.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10277393 (9.8M) [application/octet-stream]
Saving to: 'sample\_web\_log\_1.log.gz'

sample\_web\_log\_1.lo 100%[=====>] 9.80M 21.9MB/s in 0.4s

2025-04-19 03:54:46 (21.9 MB/s) - 'sample\_web\_log\_1.log.gz' saved [10277393/10277393]

--2025-04-19 03:54:46-- https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample\_web\_log\_2.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10277610 (9.8M) [application/octet-stream]
Saving to: 'sample\_web\_log\_2.log.gz'

sample\_web\_log\_2.lo 100%[=====>] 9.80M 22.9MB/s in 0.4s

2025-04-19 03:54:47 (22.9 MB/s) - 'sample\_web\_log\_2.log.gz' saved [10277610/10277610]

--2025-04-19 03:54:47-- https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample\_web\_log\_3.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10276732 (9.8M) [application/octet-stream]
Saving to: 'sample\_web\_log\_3.log.gz'

sample\_web\_log\_3.lo 100%[=====>] 9.80M 23.7MB/s in 0.4s

2025-04-19 03:54:49 (23.7 MB/s) - 'sample\_web\_log\_3.log.gz' saved [10276732/10276732]

--2025-04-19 03:54:49-- https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample\_web\_log\_4.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10277331 (9.8M) [application/octet-stream]
Saving to: 'sample\_web\_log\_4.log.gz'

sample\_web\_log\_4.lo 100%[=====>] 9.80M 26.6MB/s in 0.4s

🔍 Commands

+ Code

+ Text

≡

🔍

<>

{x}

🔗

📁

📄

●

```
Length: 10276732 (9.8M) [application/octet-stream]
Saving to: 'sample_web_log_3.log.gz'

sample_web_log_3.lo 100%[=====>]   9.80M  23.7MB/s   in 0.4s

2025-04-19 03:54:49 (23.7 MB/s) - 'sample_web_log_3.log.gz' saved [10276732/10276732]

--2025-04-19 03:54:49--  https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_4.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10277331 (9.8M) [application/octet-stream]
Saving to: 'sample_web_log_4.log.gz'

sample_web_log_4.lo 100%[=====>]   9.80M  26.6MB/s   in 0.4s

2025-04-19 03:54:50 (26.6 MB/s) - 'sample_web_log_4.log.gz' saved [10277331/10277331]

--2025-04-19 03:54:51--  https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_5.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10277563 (9.8M) [application/octet-stream]
Saving to: 'sample_web_log_5.log.gz'

sample_web_log_5.lo 100%[=====>]   9.80M  23.9MB/s   in 0.4s

2025-04-19 03:54:52 (23.9 MB/s) - 'sample_web_log_5.log.gz' saved [10277563/10277563]

--2025-04-19 03:54:52--  https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_6.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10276383 (9.8M) [application/octet-stream]
Saving to: 'sample_web_log_6.log.gz'

sample_web_log_6.lo 100%[=====>]   9.80M  23.9MB/s   in 0.4s

2025-04-19 03:54:54 (23.9 MB/s) - 'sample_web_log_6.log.gz' saved [10276383/10276383]

--2025-04-19 03:54:54--  https://raw.githubusercontent.com/keeyong/sjsu-data226-SP25/refs/heads/main/week13/data/sample_web_log_7.log.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10279182 (9.8M) [application/octet-stream]
Saving to: 'sample_web_log_7.log.gz'

sample_web_log_7.lo 100%[=====>]   9.80M  26.6MB/s   in 0.4s

2025-04-19 03:54:56 (26.6 MB/s) - 'sample_web_log_7.log.gz' saved [10279182/10279182]
```

3. Configure Spark environment as shown in the demo notebook (+1 pt)
1. Add the Snowflake JDBC .jar file and Initialize SparkSession
2. Create the input DataFrame (df) and parsed DataFrame (log\_df)

➔ Adding Snowflake JDBC jar and initializing SparkSession.

✓ [2] !cd /usr/local/lib/python3.11/dist-packages/pyspark/jars && wget https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.19.0/snowflake-jdbc-3.19.0.jar

🔍

```
--2025-04-19 03:55:39--  https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.19.0/snowflake-jdbc-3.19.0.jar
Resolving repo1.maven.org (repo1.maven.org)... 199.232.192.209, 199.232.196.209, 2a04:4e42:4c::209, ...
Connecting to repo1.maven.org (repo1.maven.org)|199.232.192.209|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 70986770 (68M) [application/java-archive]
Saving to: 'snowflake-jdbc-3.19.0.jar'

snowflake-jdbc-3.19 100%[=====>]  67.70M  15.1MB/s   in 5.2s

2025-04-19 03:55:45 (13.1 MB/s) - 'snowflake-jdbc-3.19.0.jar' saved [70986770/70986770]
```

🎮

```
from pyspark.sql import SparkSession
import pyspark.sql.functions as F

spark = SparkSession.builder.appName("HandleLogFiles").getOrCreate()
```

Creating input DataFrame `df` and extracting structured columns into `log\_df`.

✓ 10s

▶

```
# Load all .gz files in the directory into a DataFrame
df = spark.read.text("*.gz")

# Check the number of partitions
print(df.rdd.getNumPartitions())

df.show(truncate=False)
```

🔍

3

```
+-----+
|value|
+-----+
|123.45.67.89 - - [05/Nov/2024:02:08:16 +0000] "DELETE /cart HTTP/1.1" 500 242|
|192.168.1.1 - - [04/Nov/2024:21:23:39 +0000] "POST /checkout HTTP/1.1" 404 2781|
|234.56.78.90 - - [05/Nov/2024:07:06:19 +0000] "GET /api/data HTTP/1.1" 301 3758|
|192.168.1.1 - - [04/Nov/2024:20:03:56 +0000] "POST /home HTTP/1.1" 200 1837|
|192.168.1.1 - - [04/Nov/2024:21:25:05 +0000] "GET /products/123 HTTP/1.1" 200 3430|
|234.56.78.90 - - [04/Nov/2024:07:38:10 +0000] "GET /api/data HTTP/1.1" 404 3729|
|123.45.67.89 - - [04/Nov/2024:12:33:22 +0000] "PUT /api/data HTTP/1.1" 404 799|
|192.168.1.1 - - [04/Nov/2024:07:37:46 +0000] "GET /api/data HTTP/1.1" 500 309|
|123.45.67.89 - - [04/Nov/2024:21:52:36 +0000] "POST /checkout HTTP/1.1" 301 2375|
|123.45.67.89 - - [04/Nov/2024:08:36:44 +0000] "DELETE /api/data HTTP/1.1" 404 3449|
|192.168.1.1 - - [05/Nov/2024:03:15:43 +0000] "GET /api/data HTTP/1.1" 200 2319|
|234.56.78.90 - - [05/Nov/2024:01:26:03 +0000] "DELETE /home HTTP/1.1" 500 1168|
|234.56.78.90 - - [05/Nov/2024:03:26:33 +0000] "DELETE /cart HTTP/1.1" 500 1262|
|123.45.67.89 - - [04/Nov/2024:20:46:25 +0000] "PUT /home HTTP/1.1" 301 4401|
|123.45.67.89 - - [05/Nov/2024:08:07:51 +0000] "GET /api/data HTTP/1.1" 301 3736|
|123.45.67.89 - - [04/Nov/2024:21:01:30 +0000] "DELETE /cart HTTP/1.1" 404 2418|
|123.45.67.89 - - [04/Nov/2024:09:40:29 +0000] "POST /api/data HTTP/1.1" 301 3260|
|234.56.78.90 - - [04/Nov/2024:09:23:42 +0000] "GET /home HTTP/1.1" 200 1488|
|192.168.1.1 - - [04/Nov/2024:11:53:57 +0000] "POST /products/123 HTTP/1.1" 200 2627|
|234.56.78.90 - - [05/Nov/2024:01:26:01 +0000] "PUT /cart HTTP/1.1" 500 4406|
+-----+
only showing top 20 rows
```

```
[5] # Extract the necessary information from log data using regular expressions
pattern = r'(\d+\.\d+\.\d+\.\d+) - - \[(.*?)\] "(.*?) (.*?) HTTP.*" (\d+) (\d+)'

log_df = df.select(
    F.regexp_extract("value", pattern, 1).alias("ip"),
    F.regexp_extract("value", pattern, 2).alias("timestamp"),
    F.regexp_extract("value", pattern, 3).alias("method"),
    F.regexp_extract("value", pattern, 4).alias("url"),
    F.regexp_extract("value", pattern, 5).alias("status").cast("integer"),
    F.regexp_extract("value", pattern, 6).alias("size").cast("integer")
)

[6] log_df.show()
```

ip	timestamp	method	url	status	size
123.45.67.89	05/Nov/2024:02:08...	DELETE	/cart	500	242
192.168.1.1	04/Nov/2024:21:23...	POST	/checkout	404	2781
234.56.78.90	05/Nov/2024:07:06...	GET	/api/data	301	3758
192.168.1.1	04/Nov/2024:20:03...	POST	/home	200	1837
192.168.1.1	04/Nov/2024:21:25...	GET	/products/123	200	3430
234.56.78.90	04/Nov/2024:07:38...	GET	/api/data	404	3729
123.45.67.89	04/Nov/2024:12:33...	PUT	/api/data	404	799
192.168.1.1	04/Nov/2024:07:37...	GET	/api/data	500	309
123.45.67.89	04/Nov/2024:21:52...	POST	/checkout	301	2375
123.45.67.89	04/Nov/2024:08:36...	DELETE	/api/data	404	3449
192.168.1.1	05/Nov/2024:03:15...	GET	/api/data	200	2319
234.56.78.90	05/Nov/2024:01:26...	DELETE	/home	500	1168
234.56.78.90	05/Nov/2024:03:26...	DELETE	/cart	500	1262
123.45.67.89	04/Nov/2024:20:46...	PUT	/home	301	4401
123.45.67.89	05/Nov/2024:08:07...	GET	/api/data	301	3736
123.45.67.89	04/Nov/2024:21:01...	DELETE	/cart	404	2418
123.45.67.89	04/Nov/2024:09:40...	POST	/api/data	301	3260
234.56.78.90	04/Nov/2024:09:23...	GET	/home	200	1488
192.168.1.1	04/Nov/2024:11:53...	POST	/products/123	200	2627
234.56.78.90	05/Nov/2024:01:26...	PUT	/cart	500	4406

only showing top 20 rows

4. Compute **IP and status** combination counts using DataFrame operations (+2pt):
- Count the occurrences of each unique (ip, status) pair
  - Sort the result in **descending** order by count

➔ Computing (ip, status) combination counts using PySpark DataFrame operations and sorting them in descending order.

```
[7] ip_status_count = log_df.groupBy("ip", "status").count().orderBy(F.desc("count"))
ip_status_count.show()
```

ip	status	count
123.45.67.89	500	584162
192.168.1.1	200	584048
234.56.78.90	200	583982
234.56.78.90	500	583682
123.45.67.89	301	583534
192.168.1.1	404	583529
234.56.78.90	301	583403
234.56.78.90	404	583135
123.45.67.89	200	583091
123.45.67.89	404	582730
192.168.1.1	500	582404
192.168.1.1	301	582300

5. Repeat step 4 using Spark SQL instead of DataFrame operations (+2 pt)

➔ Computing (ip, status) combination counts using Spark SQL.

<>

35s

[8] log\_df.createOrReplaceTempView("logs")

{x}

ip\_status\_sql = spark.sql("""  
SELECT ip, status, COUNT(\*) as count  
FROM logs  
GROUP BY ip, status  
ORDER BY count DESC  
""")  
ip\_status\_sql.show()

⇌

ip	status	count
123.45.67.89	500	584162
192.168.1.1	200	584048
234.56.78.90	200	583982
234.56.78.90	500	583682
123.45.67.89	301	583534
192.168.1.1	404	583529
234.56.78.90	301	583403
234.56.78.90	404	583135
123.45.67.89	200	583091
123.45.67.89	404	582730
192.168.1.1	500	582404
192.168.1.1	301	582300

6. Write the resulting DataFrame from step 4 to a Snowflake table (+2 pt)

➔ Writing the (ip, status) counts to the Snowflake table `ip\_status\_count` using JDBC connection.

}

✓

[10] from google.colab import userdata

?

account = userdata.get('snowflake\_account')  
user = userdata.get('snowflake\_userid')  
password = userdata.get('snowflake\_password')  
database = "dev"  
schema = "analytics"

]

url = f"jdbc:snowflake://{account}.snowflakecomputing.com/?db={database}&schema={schema}&user={user}&password={password}"

✓

[11] ip\_status\_sql.write \  
.format("jdbc") \  
.option("driver", "net.snowflake.client.jdbc.SnowflakeDriver") \  
.option("url", url) \  
.mode("overwrite") \  
.option("dbtable", "ip\_status\_count") \  
.save()

7. Capture a screenshot of the Snowflake table from Snowflake Web UI (+1 pt)

➔ Snowflake Web UI preview of the `ip\_status\_count` table in the `analytics` schema, confirming successful data write from PySpark.

