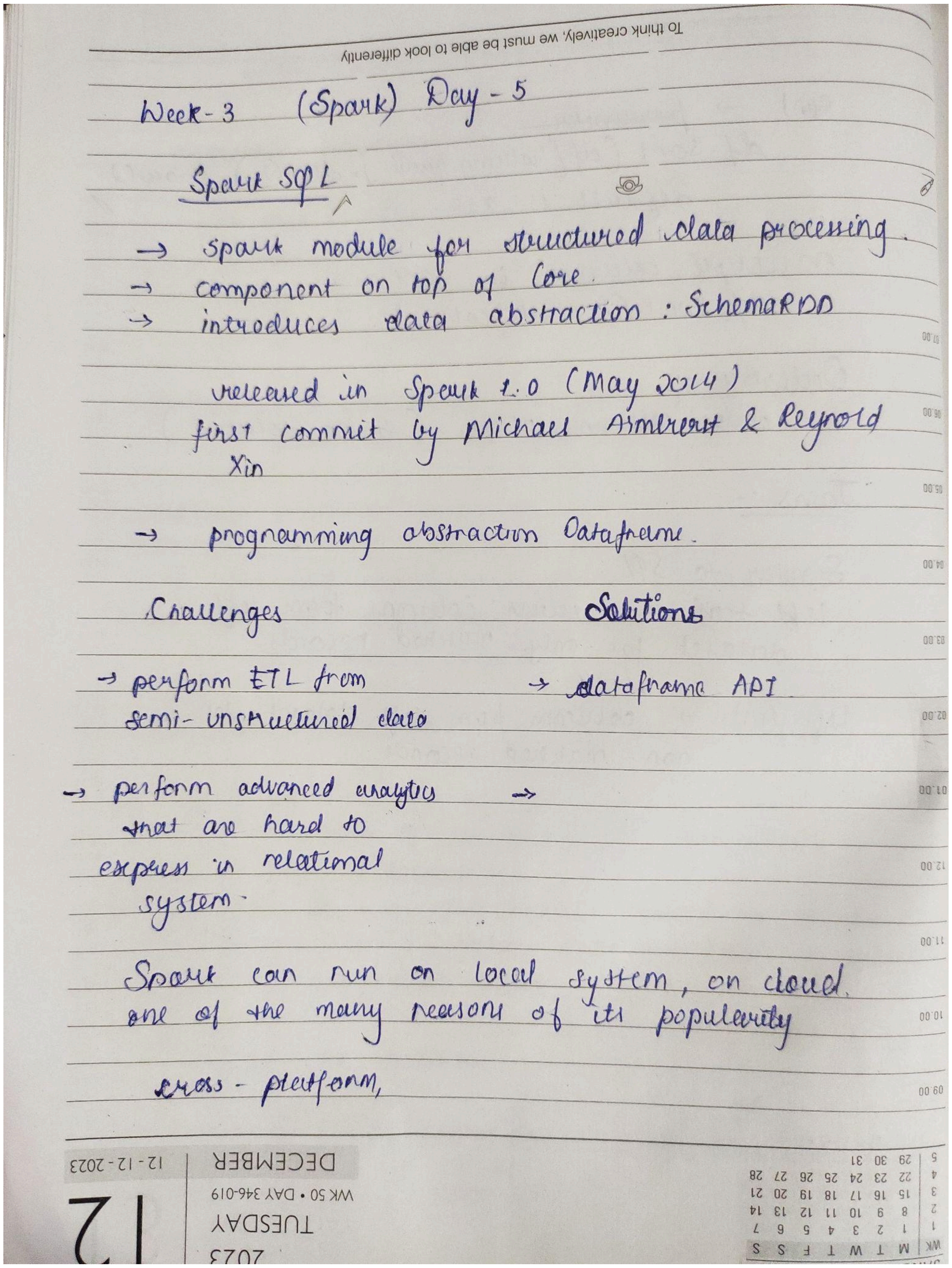
DAY-15 (Spark Assignment - 5)

Mitushi Vishwakarma

NOTES :



Week-3 (Spark) Day - 5

Spark SQL

→ spark module for structured data processing.
→ component on top of Core.
→ introduces data abstraction : SchemaRDD

released in Spark 1.0 (May 2014)
first commit by Michael Armbrust & Reynold Xin

→ programming abstraction Dataframe.

Challenges	Solutions
→ perform ETL from semi-unstructured data	→ dataframe API.
→ perform advanced analytics that are hard to express in relational system.	→

Spark can run on local system, on cloud. one of the many reasons of its popularity

cross-platform,

Spark SQL Architecture

lang API   Py   Scala   Java   Hive-QL

Spark SQL

Schema RDD

Data Frame

Data Sources   Parquet   JSON   Hive   Cassandra.

Hive → Big data Tool.
Hive QL will connect with Hive and query data.

lang API :-
→ compatible with different languages.

Schema RDD :-
→ works on schemas, tables. records.
→ called as DataFrame.

Data Sources :-
→ different (~~text~~, ~~Avro~~ Parquet, JSON, Hive, Cassandra)

Features of Spark SQL

1. Integrated
   - → seamless mixing of SQL with Spark
   - → lets us query data as RDD with APIs in Py, J, Sc.

2. Unified Data Access
   - → load and query from variety of sources
   - → provide single interface for efficient working

3. Hive compatibility
   - → run hive queries on existing hive data

4. Standard Connectivity
   - → Connect through JDBC.
   - → includes server mode with JDBC / ODBC connectivity

5. Scalability
   - → use same engine for long queries.
   - → mid-query fault tolerance
   - → query historical data.

To Aim and hit, you need one eye only, and one good finger

Spark RDD

- → RDD is fundamental data structure of spark.
- → immutable data structure, can be stored in memory or disk on cluster.
- → RDDs can contain Python, Java Object
- → fault tolerant
- → two ways to create RDDs
  - → parallelizing existing collection
  - → referencing a dataset.
- → v

Dataset and Dataframe

- → distributed collection of data organised into named columns.
- → equivalent to Relational tables with good optimization techniques.
- → DF can be created from . external databases, RDDs, Hive Tables.
- → designed for Big Data, Data Science applications.

RDD

Person
Person

Dataframe

Name	Age	Height
String	Int	Double

08 - 12 - 2023 DECEMBER

WK 49 • DAY 342-023

Errors	SQL	Dataframes	Datasets.
Syntax	Run-time	compile time	Compile
Analysis	Runtime	Run	Compile

Features :-
→ process kb to Pitabytes size data
→ supports different data formats.
→ integration with Big data tools.
→ APIs.

Read & Write

Builder methods :-
- format
- option.
- partitioning

df. = sqlContent . read . format( ). option ( ), load()

df . write . format ( ). mode().partitionBy (), save asTable

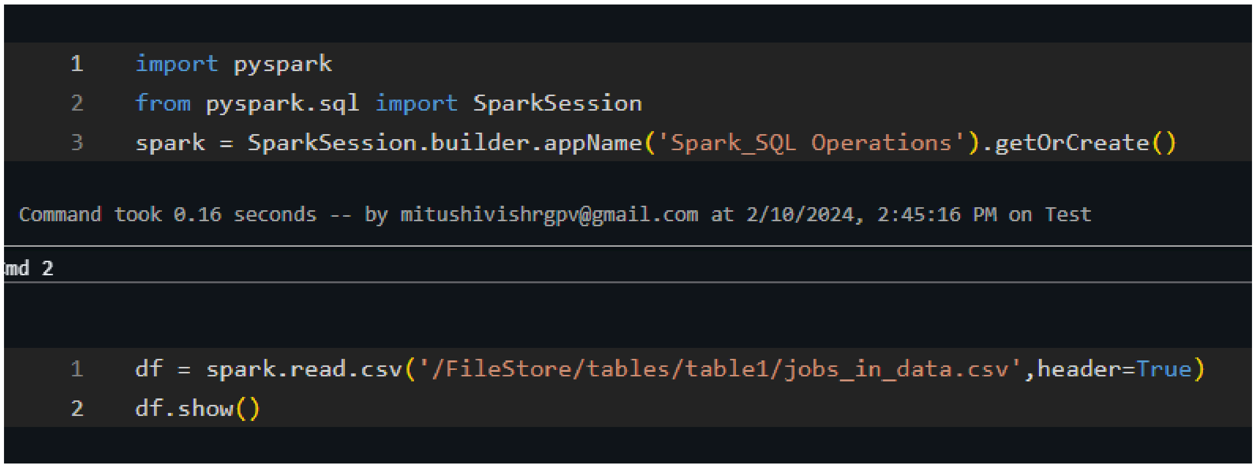
load ; save As Table create new builders

Plan Optimization & Execution

Analysis	logical Optimization	Physical Planning	Code Generation

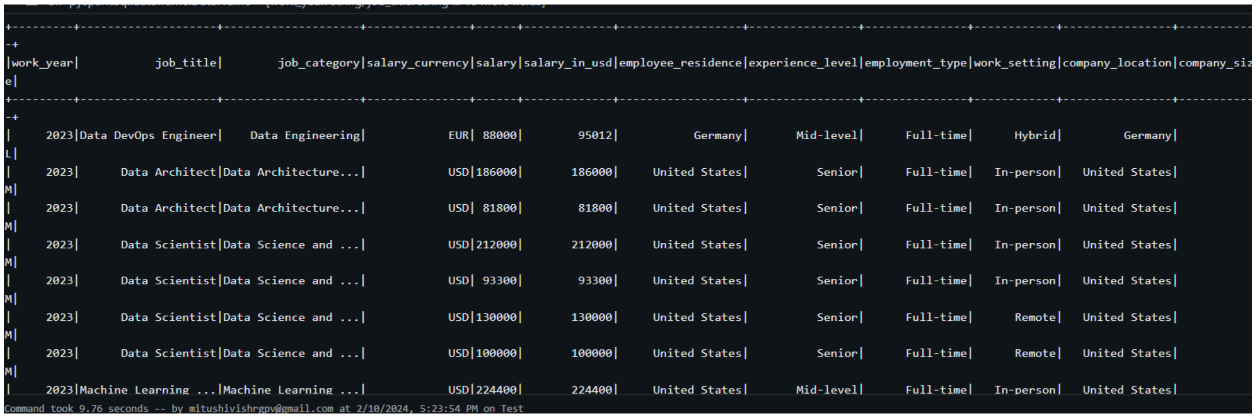
Data Frames and SqL share same optimization / execution pipeline.

Creating dataframe from csv file :



```
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('Spark_SQL Operations').getOrCreate()

Command took 0.16 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 2:45:16 PM on Test
```

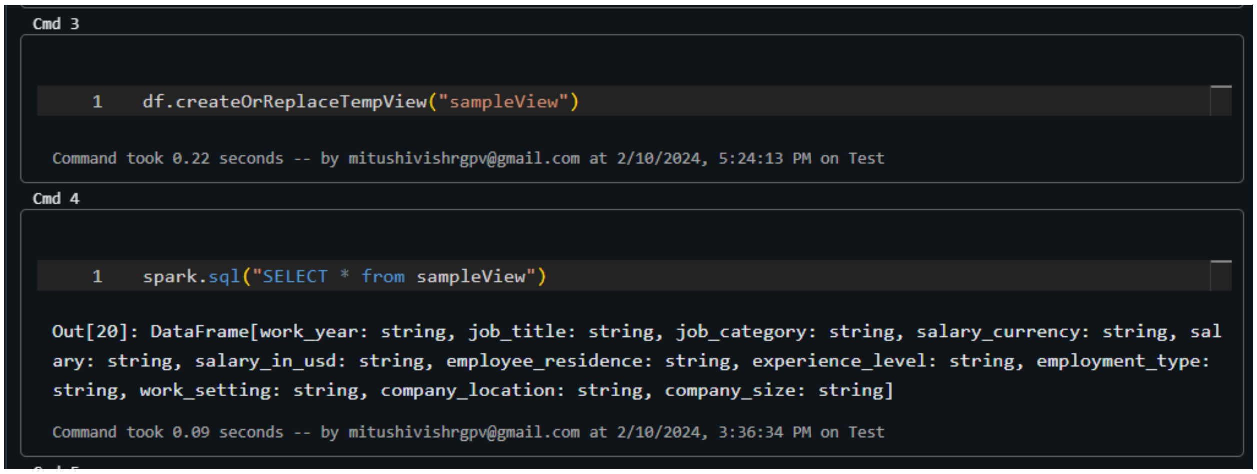
```
md 2

df = spark.read.csv('/FileStore/tables/table1/jobs_in_data.csv',header=True)
df.show()
```



```
+---------+--------------------+--------------------+---------------+------+-------------+------------------+----------------+---------------+------------+----------------+---------
-+
|work_year|           job_title|        job_category|salary_currency|salary|salary_in_usd|employee_residence|experience_level|employment_type|work_setting|company_location|company_siz
e|
+---------+--------------------+--------------------+---------------+------+-------------+------------------+----------------+---------------+------------+----------------+---------
-+
|     2023|Data DevOps Engineer|    Data Engineering|            EUR| 88000|        95012|           Germany|       Mid-level|      Full-time|      Hybrid|         Germany|
L|
|     2023|      Data Architect|Data Architecture...|            USD|186000|       186000|     United States|          Senior|      Full-time|   In-person|   United States|
M|
|     2023|      Data Architect|Data Architecture...|            USD| 81800|        81800|     United States|          Senior|      Full-time|   In-person|   United States|
M|
|     2023|      Data Scientist|Data Science and ...|            USD|212000|       212000|     United States|          Senior|      Full-time|   In-person|   United States|
M|
|     2023|      Data Scientist|Data Science and ...|            USD| 93300|        93300|     United States|          Senior|      Full-time|   In-person|   United States|
M|
|     2023|      Data Scientist|Data Science and ...|            USD|130000|       130000|     United States|          Senior|      Full-time|      Remote|   United States|
M|
|     2023|      Data Scientist|Data Science and ...|            USD|100000|       100000|     United States|          Senior|      Full-time|      Remote|   United States|
M|
|     2023|Machine Learning ...|Machine Learning ...|            USD|224400|       224400|     United States|       Mid-level|      Full-time|   In-person|   United States|
Command took 9.76 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 5:23:54 PM on Test
```

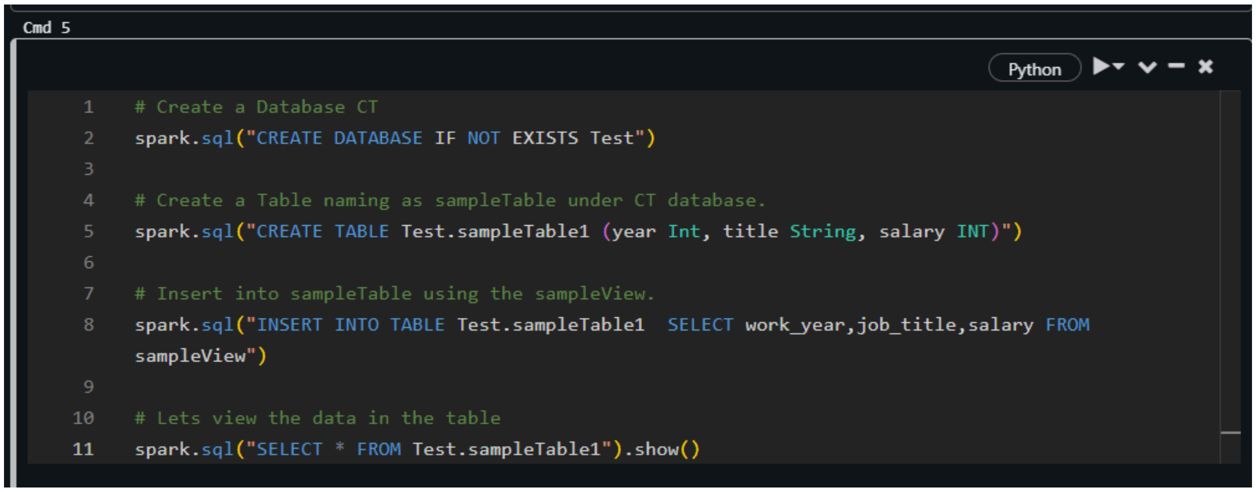
Creating temporary view from dataframe :

The view is a temporary table created from dataframe. We can run SQL queries on Temp view using spark.sql() where spark is the session created above.



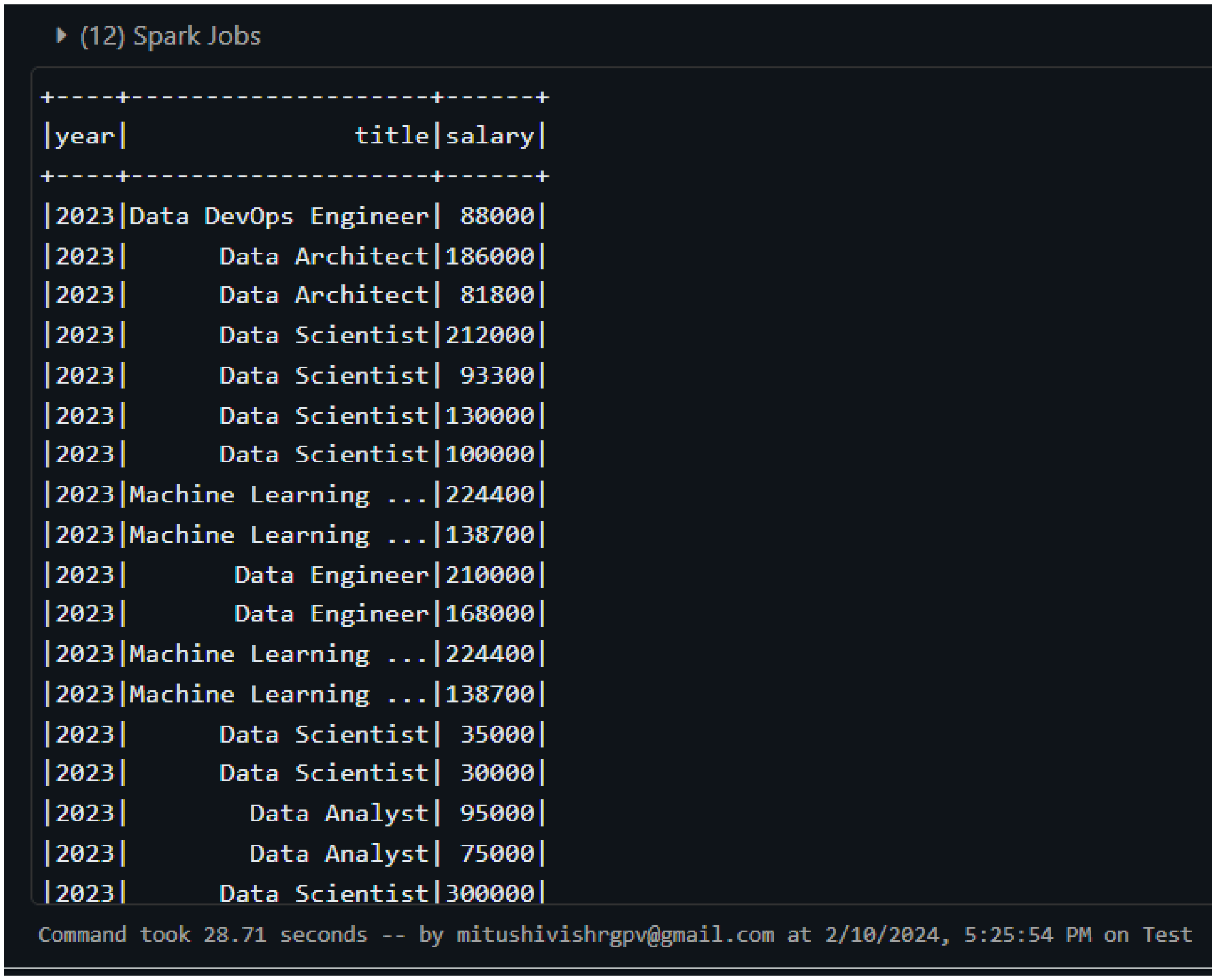
```
Cmd 3

df.createOrReplaceTempView("sampleView")

Command took 0.22 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 5:24:13 PM on Test
Cmd 4

spark.sql("SELECT * from sampleView")

Out[20]: DataFrame[work_year: string, job_title: string, job_category: string, salary_currency: string, sal
ary: string, salary_in_usd: string, employee_residence: string, experience_level: string, employment_type:
string, work_setting: string, company_location: string, company_size: string]

Command took 0.09 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 3:36:34 PM on Test
```

Creating database and table and inserting values : Here table name is given as database.tablename



Cmd 5

```python
# Create a Database CT
spark.sql("CREATE DATABASE IF NOT EXISTS Test")

# Create a Table naming as sampleTable under CT database.
spark.sql("CREATE TABLE Test.sampleTable1 (year Int, title String, salary INT)")

# Insert into sampleTable using the sampleView.
spark.sql("INSERT INTO TABLE Test.sampleTable1  SELECT work_year,job_title,salary FROM
sampleView")

# Lets view the data in the table
spark.sql("SELECT * FROM Test.sampleTable1").show()
```

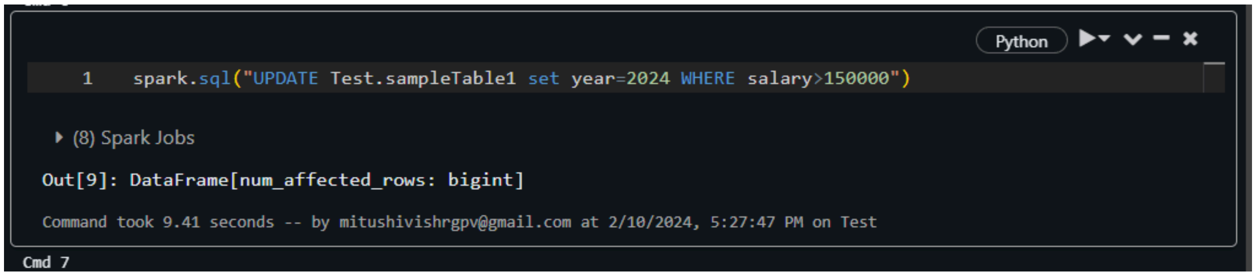


(12) Spark Jobs

```
+----+--------------------+------+
|year|               title|salary|
+----+--------------------+------+
|2023|Data DevOps Engineer| 88000|
|2023|      Data Architect|186000|
|2023|      Data Architect| 81800|
|2023|      Data Scientist|212000|
|2023|      Data Scientist| 93300|
|2023|      Data Scientist|130000|
|2023|      Data Scientist|100000|
|2023|Machine Learning ...|224400|
|2023|Machine Learning ...|138700|
|2023|       Data Engineer|210000|
|2023|       Data Engineer|168000|
|2023|Machine Learning ...|224400|
|2023|Machine Learning ...|138700|
|2023|      Data Scientist| 35000|
|2023|      Data Scientist| 30000|
|2023|        Data Analyst| 95000|
|2023|        Data Analyst| 75000|
|2023|      Data Scientist|300000|
```

Command took 28.71 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 5:25:54 PM on Test

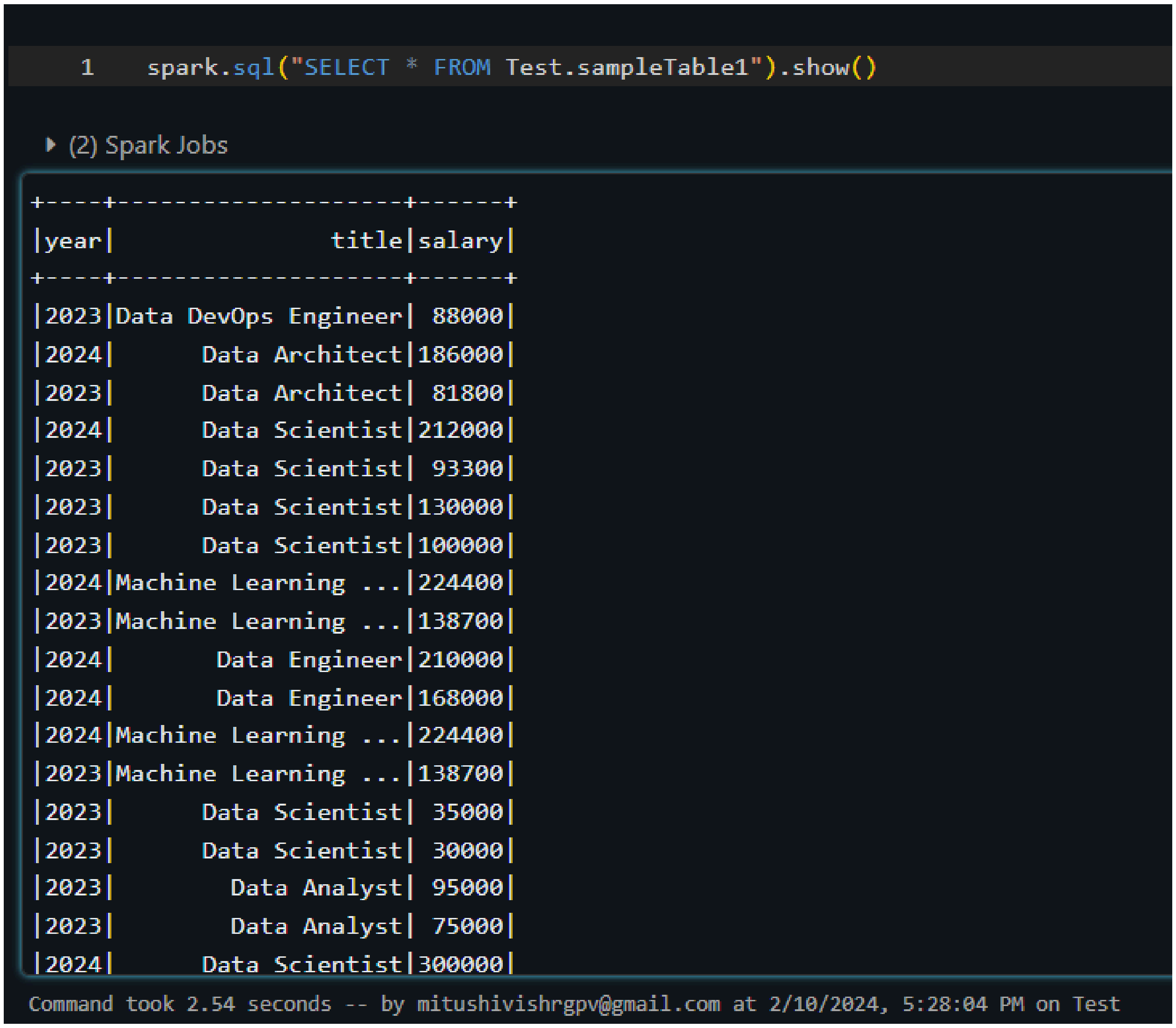
Updating Values in the table using SQL update and where clause:



```
spark.sql("UPDATE Test.sampleTable1 set year=2024 WHERE salary>150000")
```

(8) Spark Jobs

```
Out[9]: DataFrame[num_affected_rows: bigint]
Command took 9.41 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 5:27:47 PM on Test
Cmd 7
```

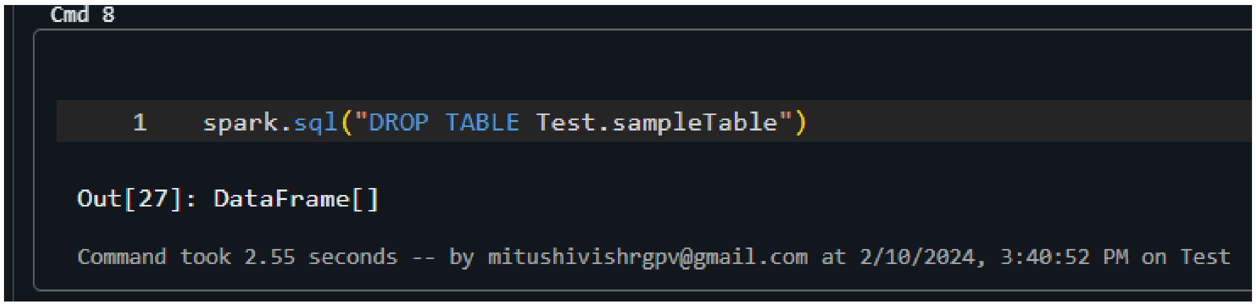


```
spark.sql("SELECT * FROM Test.sampleTable1").show()
```

(2) Spark Jobs

```
+----+--------------------+------+
|year|               title|salary|
+----+--------------------+------+
|2023|Data DevOps Engineer| 88000|
|2024|      Data Architect|186000|
|2023|      Data Architect| 81800|
|2024|      Data Scientist|212000|
|2023|      Data Scientist| 93300|
|2023|      Data Scientist|130000|
|2023|      Data Scientist|100000|
|2024|Machine Learning ...|224400|
|2023|Machine Learning ...|138700|
|2024|       Data Engineer|210000|
|2024|       Data Engineer|168000|
|2024|Machine Learning ...|224400|
|2023|Machine Learning ...|138700|
|2023|      Data Scientist| 35000|
|2023|      Data Scientist| 30000|
|2023|        Data Analyst| 95000|
|2023|        Data Analyst| 75000|
|2024|      Data Scientist|300000|
Command took 2.54 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 5:28:04 PM on Test
```

Dropping table using drop table :



```
Cmd 8
spark.sql("DROP TABLE Test.sampleTable")

Out[27]: DataFrame[]

Command took 2.55 seconds -- by mitushivishrgpv@gmail.com at 2/10/2024, 3:40:52 PM on Test
```