09/02/2024

SPARK SQL

→ Spark SQL is a component on top of the spark core that introduce a new data abstraction called schema RDD.
→ Spark introduces a programming module for structed data processing called spark SQL
→ It provides a programming abstraction called data frame and act as distributed SQL query engine.

Challenges	Solution
• Performs ETL to see from various data sources	• A Data frame API that can perform relational generations on both external resources and spark built in RDD's
• Perform advanced that are hard to express in relational systems	

Spark SQL Architectures
- Language API
- Schema RDD
- Data services

Features:-

① Integrated:
→ Integrated API's in Python, Scala, Java
→ Easy to run complex algorithms

② Unified data Analysis:
→ load and query data from various process
→ schemds RDD provide a single interface for affirimity working with struced data

③ Hive compatibility:
→ Run unmodified the queries on existing warehouses

→ SPL reuses the frontend and meta store for giving you full compatibility with Hive data, queries and UDF.

④ Standard Connectivity:
→ Connect through JDBC & ODBC
→ Spark SQL includes a server mode with industry standard JDBC & ODBC.

⑤ Scalibility:
→ Use same engine for both interactive and long queries
→ takes advantage of RDD to support mid-query fault tolerance

Spark RDD:-
→ RDD is a functional data structure of spark
→ It is an immutable distributed collection of objects that can be stored in memory.
→ Each dataset in RDD is divided into logical partitions which may be computed on different nodes to obtain.
→ parallel functional transformation (map, filters, ----)
→ Automificed rebuilt on failure
→ RDD can contain any type of Python, Java, Scala dependency including UDF
→ Formally, an RDD is read only, Partional collection of records.
→ They are 2 ways to create RDD
(i) Parallelizing & excisiting collection in your driver program.
(ii) Referming a dataset in an external storage system such as HDFs, HBase or any data source.

Dataframe:-
Data is organised into named colums, like a table in a relational databases

Dataset:- A distributed collection of data.

Feautres of Data Frame:-

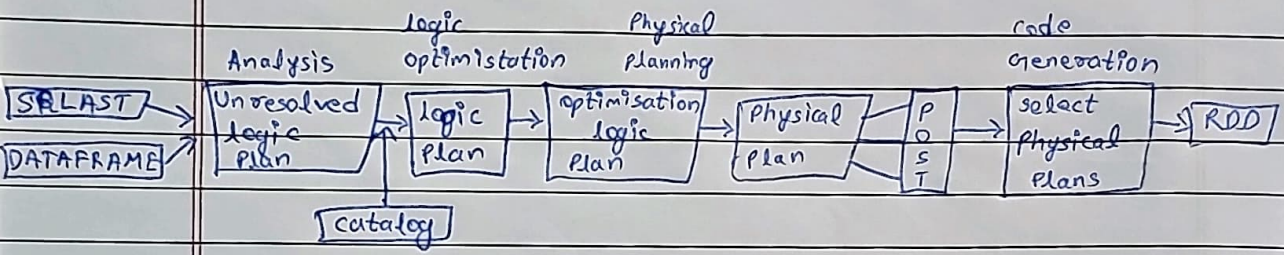
→ Ability and process the data in the site of Kilobytes to Petabytes on a single node cluster to large cluster.

→ Supports different data formats.

→ state of art optimization and code generation through the spark SQL catalyst optimiser.

→ Can be easily integrated with all Bigdata tools and frameworks via Spark-Core

Plan Optimisation and Execution:



Dataframes and SQL share the same optimization/execution pipelines.

```
1  df = spark.read.csv("/FileStore/tables/Coaching_1.csv",header=True)
2  df.show()
```

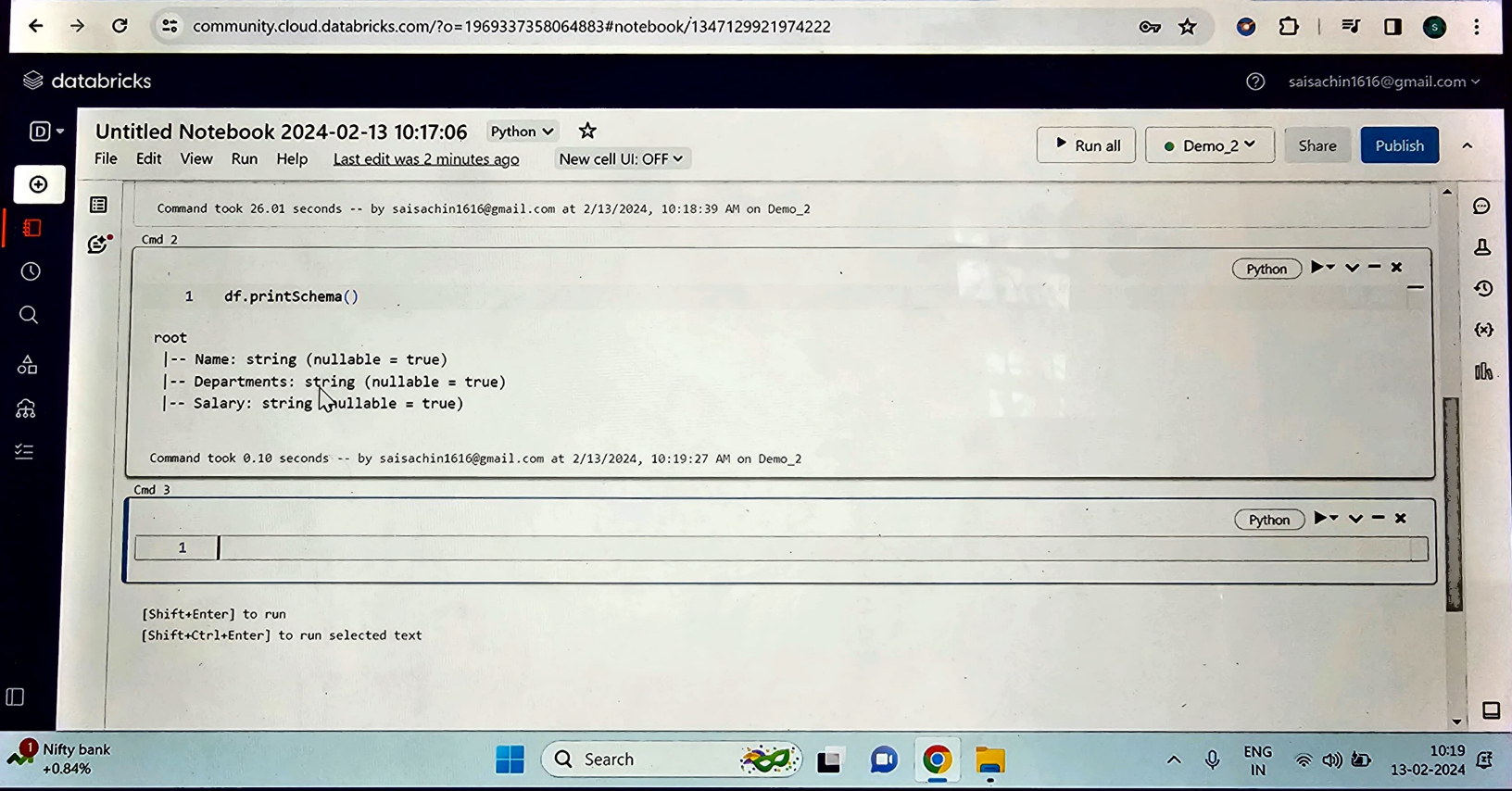
▸ (2) Spark Jobs

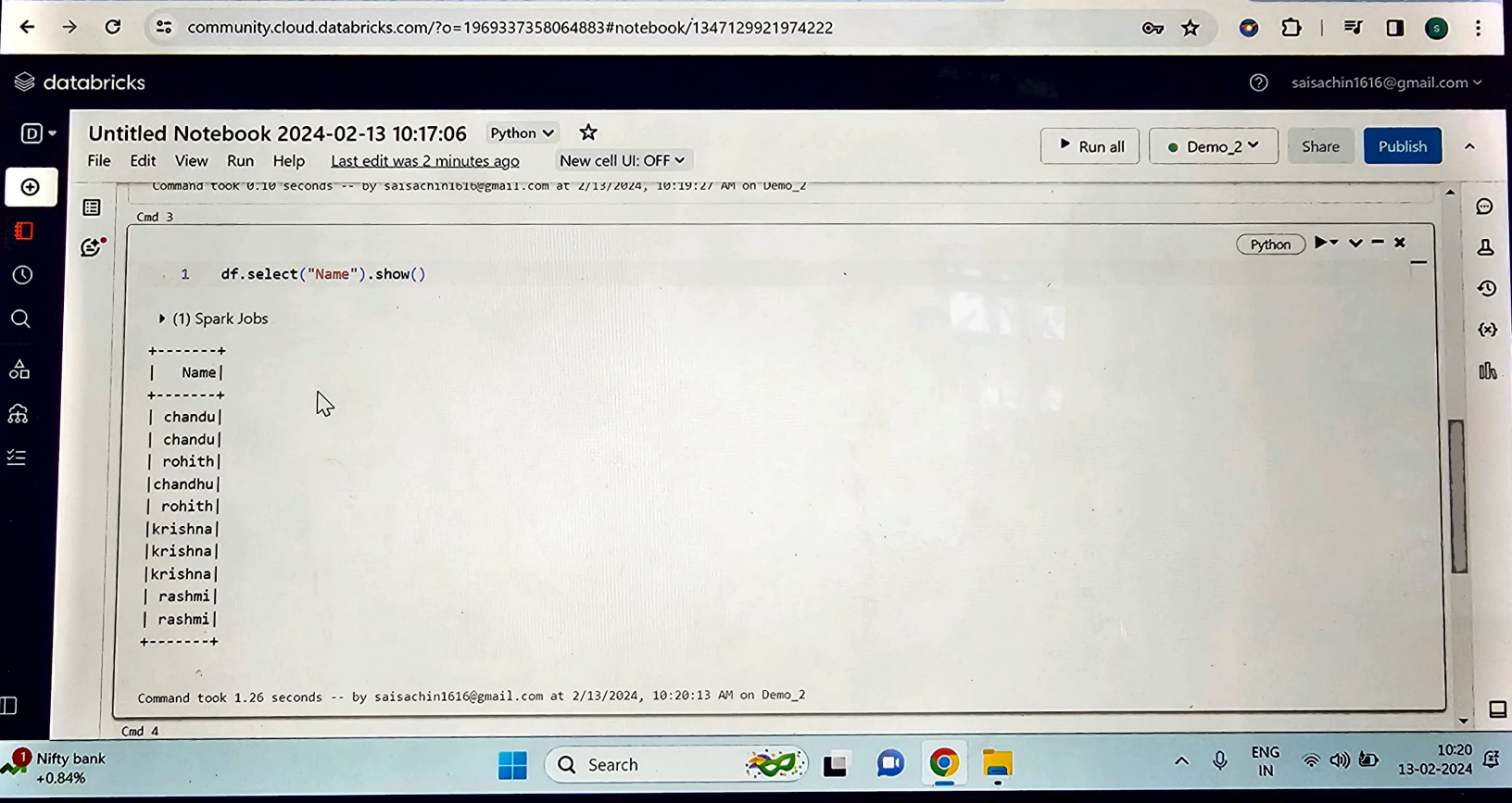
▸ df: pyspark.sql.dataframe.DataFrame = [Name: string, Departments: string ... 1 more field]

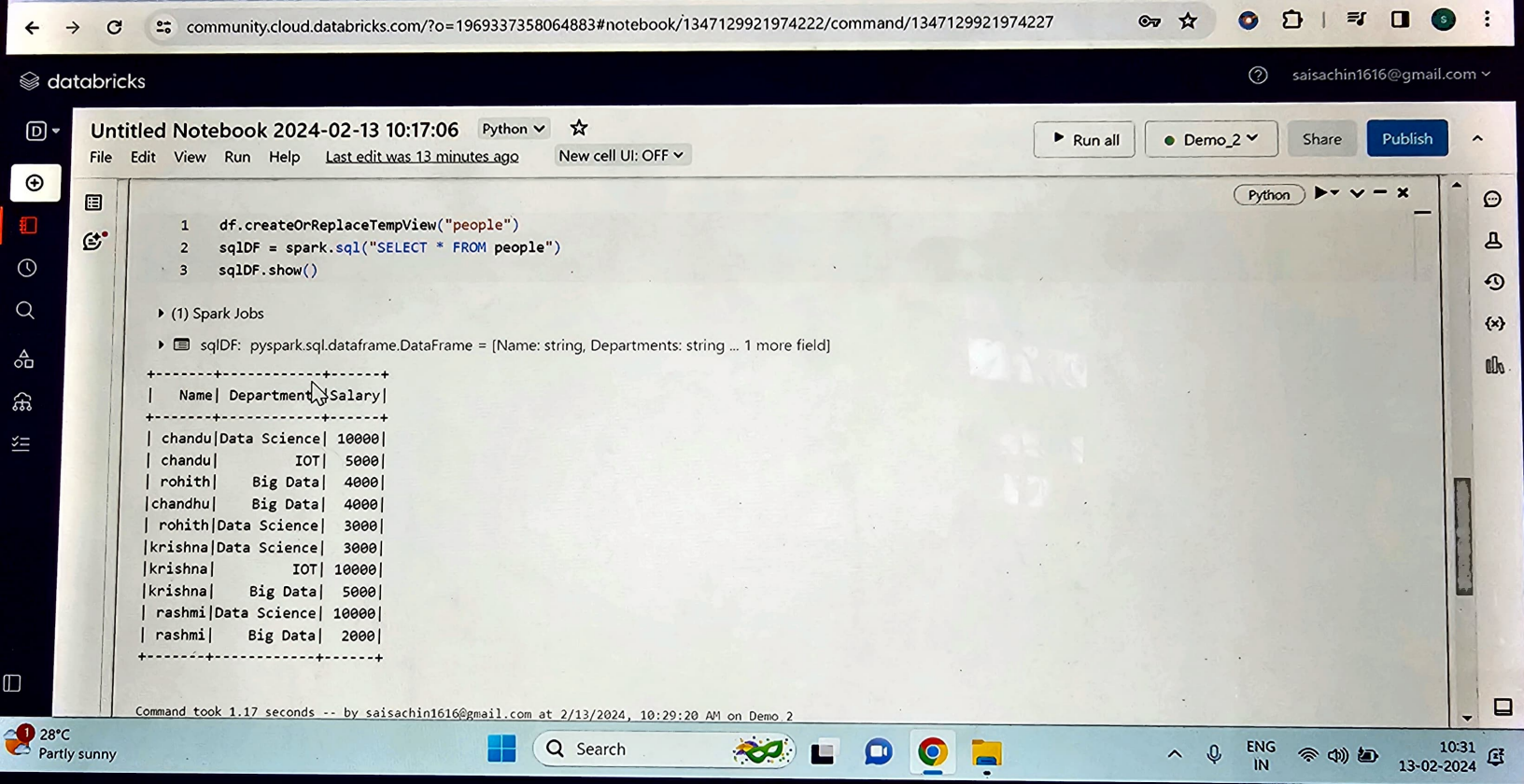
```
+-------+-----------+------+
|   Name| Departments|Salary|
+-------+-----------+------+
| chandu|Data Science| 10000|
| chandu|         IOT|  5000|
| rohith|    Big Data|  4000|
|chandhu|    Big Data|  4000|
| rohith|Data Science|  3000|
|krishna|Data Science|  3000|
|krishna|         IOT| 10000|
|krishna|    Big Data|  5000|
| rashmi|Data Science| 10000|
| rashmi|    Big Data|  2000|
+-------+-----------+------+
```

Command took 26.01 seconds -- by saisachin1616@gmail.com at 2/13/2024, 10:18:39 AM on Demo_2

Cmd 2


community.cloud.databricks.com/?o=1969337358064883#notebook/1347129921974222
databricks
saisachin1616@gmail.com
Untitled Notebook 2024-02-13 10:17:06
Python
File Edit View Run Help
Last edit was 2 minutes ago
New cell UI: OFF
Run all
Demo_2
Share
Publish
Command took 26.01 seconds -- by saisachin1616@gmail.com at 2/13/2024, 10:18:39 AM on Demo_2
Cmd 2
Python
1 df.printSchema()
root
|-- Name: string (nullable = true)
|-- Departments: string (nullable = true)
|-- Salary: string (nullable = true)
Command took 0.10 seconds -- by saisachin1616@gmail.com at 2/13/2024, 10:19:27 AM on Demo_2
Cmd 3
Python
1
[Shift+Enter] to run
[Shift+Ctrl+Enter] to run selected text
Nifty bank
+0.84%
Search
ENG
IN
10:19
13-02-2024


community.cloud.databricks.com/?o=1969337358064883#notebook/1347129921974222
databricks
saisachin1616@gmail.com
Untitled Notebook 2024-02-13 10:17:06
Python
File Edit View Run Help
Last edit was 2 minutes ago
New cell UI: OFF
Run all
Demo_2
Share
Publish
Command took 0.10 seconds -- by saisachin1616@gmail.com at 2/13/2024, 10:19:27 AM on Demo_2
Cmd 3
Python
1 df.select("Name").show()
(1) Spark Jobs
+-------+
| Name|
+-------+
| chandu|
| chandu|
| rohith|
|chandhu|
| rohith|
|krishna|
|krishna|
|krishna|
| rashmi|
| rashmi|
+-------+
Command took 1.26 seconds -- by saisachin1616@gmail.com at 2/13/2024, 10:20:13 AM on Demo_2
Cmd 4
Nifty bank
+0.84%
Search
ENG
IN
10:20
13-02-2024


community.cloud.databricks.com/?o=1969337358064883#notebook/1347129921974222/command/1347129921974227
databricks
saisachin1616@gmail.com
Untitled Notebook 2024-02-13 10:17:06
Python
File Edit View Run Help Last edit was 13 minutes ago
New cell UI: OFF
Run all
Demo_2
Share
Publish
Python
1 df.createOrReplaceTempView("people")
2 sqlDF = spark.sql("SELECT * FROM people")
3 sqlDF.show()
(1) Spark Jobs
sqlDF: pyspark.sql.dataframe.DataFrame = [Name: string, Departments: string ... 1 more field]
+-------+-----------+------+
| Name| Department|Salary|
+-------+-----------+------+
chandu	Data Science	10000
chandu	IOT	5000
rohith	Big Data	4000
chandhu	Big Data	4000
rohith	Data Science	3000
krishna	Data Science	3000
krishna	IOT	10000
krishna	Big Data	5000
rashmi	Data Science	10000
rashmi	Big Data	2000
+-------+-----------+------+
Command took 1.17 seconds -- by saisachin1616@gmail.com at 2/13/2024, 10:29:20 AM on Demo_2
28°C
Partly sunny
Search
ENG
IN
10:31
13-02-2024