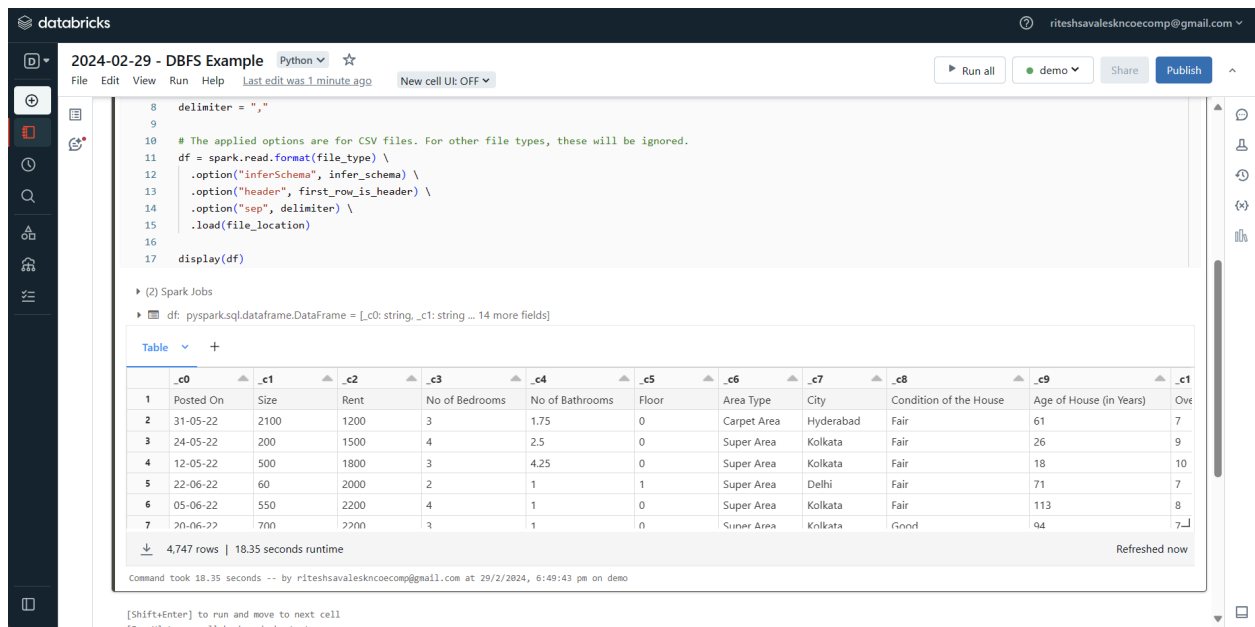


1. READ THE DATASET IN DATABRICKS COMMUNITY



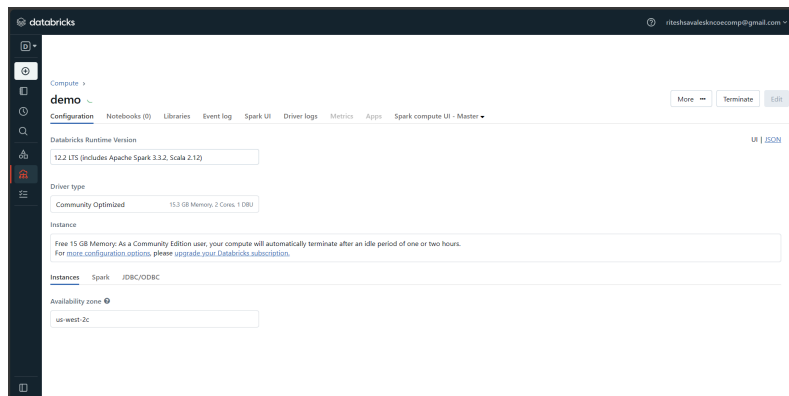
The screenshot shows a Databricks notebook titled "2024-02-29 - DBFS Example". The code in the notebook is as follows:

```
8 delimiter = ","
9
10 # The applied options are for CSV files. For other file types, these will be ignored.
11 df = spark.read.format(file_type) \
12     .option("inferSchema", infer_schema) \
13     .option("header", first_row_is_header) \
14     .option("sep", delimiter) \
15     .load(file_location)
16
17 display(df)
```

Below the code, the output shows a preview of the DataFrame with 11 columns: _c0, _c1, _c2, _c3, _c4, _c5, _c6, _c7, _c8, _c9, and _c10. The first 7 rows of data are displayed:

	_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7	_c8	_c9	_c10
1	Posted On	Size	Rent	No of Bedrooms	No of Bathrooms	Floor	Area Type	City	Condition of the House	Age of House (in Years)	Overall Rating
2	31-05-22	2100	1200	3	1.75	0	Carpet Area	Hyderabad	Fair	61	7
3	24-05-22	200	1500	4	2.5	0	Super Area	Kolkata	Fair	26	9
4	12-05-22	500	1800	3	4.25	0	Super Area	Kolkata	Fair	18	10
5	22-06-22	60	2000	2	1	1	Super Area	Delhi	Fair	71	7
6	05-06-22	550	2200	4	1	0	Super Area	Kolkata	Fair	113	8
7	20-06-22	700	2200	3	1	0	Super Area	Kolkata	Fair	64	7

The output also indicates that there are 4,747 rows and a runtime of 18.35 seconds. A command prompt at the bottom shows the execution details.



The screenshot shows the Databricks cluster configuration page for a cluster named "demo". The configuration includes:

- Databricks Runtime Version:** 12.2 LTS (includes Apache Spark 3.3.2, Scala 2.12)
- Driver type:** Community Optimized
- Instance:** Free 15 GB Memory As a Community Edition user, your compute will automatically terminate after an idle period of one or two hours. For more configuration options, please upgrade your Databricks subscription.
- Instances:** Spark, JDBC/ODBC
- Availability zone:** us-west-2c

Cluster

2. HOW MANY TYPES OF MODES WE HAVE IN SPARK?

Fail fast mode

In Apache Spark, failfast is an option that can be set to indicate whether Spark should fail immediately upon encountering a corrupted record while reading data from a file. When failfast is enabled, Spark stops the job and throws an error as soon as it encounters a malformed record, providing early detection of data issue

```
val df = spark.read
    .option("mode", "FAILFAST")
    .option("inferSchema", "true") // Optionally, infer the schema automatically
    .csv("path/to/your/file.csv")
```

```
1 df = spark.read.format(file_type)\
2   .option("mode", "FAILFAST")
3   .option("inferSchema", "true") // Optionally, infer the schema automatically
4   .csv("C:\\Users\\RITESH\\House_Rent_Dataset.csv")
5
```

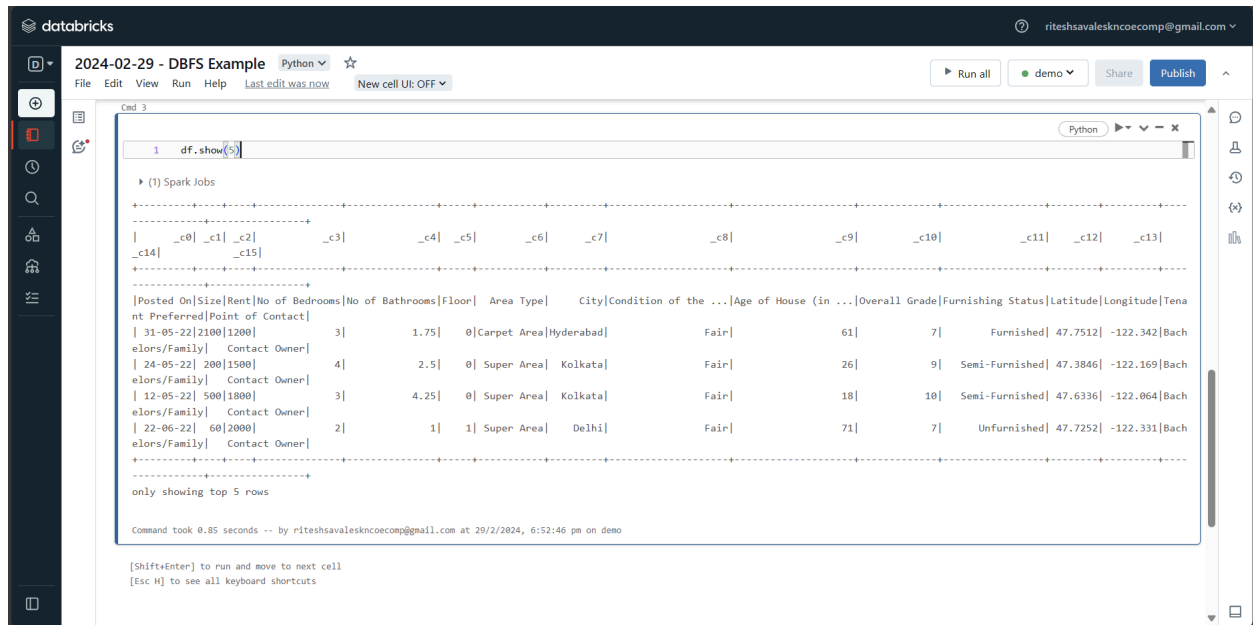
3. WHAT IS CLUSTERING SPARK?

In Spark, a "cluster" is a group of interconnected computers (nodes) that collaboratively process data and run Spark applications. It manages resources, ensures fault tolerance, enables parallel processing, and scales dynamically. A "table" in Spark refers to structured data organized in rows and columns, typically represented using DataFrames or Datasets. Tables have schemas, support querying, manipulation, and interoperability with various data sources.

4. WHAT IS TABLE IN SPARK?

In Apache Spark, a "table" refers to structured data organized into rows and columns, similar to tables in relational databases. Tables are typically represented using DataFrames or Datasets, providing an abstraction for working with structured data. They have defined schemas, enabling efficient querying, transformation, and analysis using SQL queries or DataFrame operations. Tables in Spark support interoperability with various data sources and formats.

5. WHAT WOULD YOU DO IF YOU WANT TO SHOW THE HEADER WHILE SHOWING UP RECORDS TABLE?WRITE THE CODE



The screenshot shows a Databricks notebook interface. The notebook title is "2024-02-29 - DBFS Example". The code cell contains the command `df.show(5)`. The output displays a table with 13 columns: `_c0`, `_c1`, `_c2`, `_c3`, `_c4`, `_c5`, `_c6`, `_c7`, `_c8`, `_c9`, `_c10`, `_c11`, `_c12`, and `_c13`. The first row is the header, and the following rows contain data. The output is truncated to show only the top 5 rows.

```
1 df.show(5)
```

(1) Spark Jobs

_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7	_c8	_c9	_c10	_c11	_c12	_c13
31-05-22	2100	1200	3	1.75	0	Carpet Area	Hyderabad	Fair	61	7	Furnished	47.7512	-122.342
24-05-22	200	1500	4	2.5	0	Super Area	Kolkata	Fair	26	9	Semi-Furnished	47.3846	-122.169
12-05-22	500	1800	3	4.25	0	Super Area	Kolkata	Fair	18	10	Semi-Furnished	47.6336	-122.064
22-06-22	60	2000	2	1	1	Super Area	Delhi	Fair	71	7	Unfurnished	47.7252	-122.331

only showing top 5 rows

Command took 0.85 seconds -- by riteshsavaleskncoecomp@gmail.com at 29/2/2024, 6:52:46 pm on demo

6. WHAT IS COUNT?PERFORM IN SPARK.



The screenshot shows a Databricks notebook interface. The code cell contains the following commands: `# count records in a particular column`, `# value_counts for one column let's say for rent column`, and `df.select("_c6").count()`. The output displays the count of records for the specified column, which is 4747.

```
1 # count records in a particular column
2 # value_counts for one column let's say for rent column
3 df.select("_c6").count()
4
```

(2) Spark Jobs

Out[12]: 4747

Command took 0.83 seconds -- by riteshsavaleskncoecomp@gmail.com at 29/2/2024, 7:01:17 pm on demo

7. WHAT IS GROUP BY ? PERFORM IN SPARK.



The screenshot shows a Databricks notebook interface. The code cell contains the following commands: `x=df.groupBy("_c6").count()` and `x.show()`. The output displays the result of the group by operation, showing the count of records for each value of the specified column.

```
1 x=df.groupBy("_c6").count()
2 x.show()
```

(2) Spark Jobs

x: pyspark.sql.dataframe.DataFrame = [_c6: string, count: long]

_c6	count
Area Type	1
Carpet Area	2298
Built Area	2
Super Area	2446

Command took 2.97 seconds -- by riteshsavaleskncoecomp@gmail.com at 29/2/2024, 7:10:33 pm on demo