Assignment – Day 17

-Sarthak Niranjan Kulkarni (Maverick)

- sarthakkul2311@gmail.com - (+91) 93256 02791

28/11/2024 (Thursday)

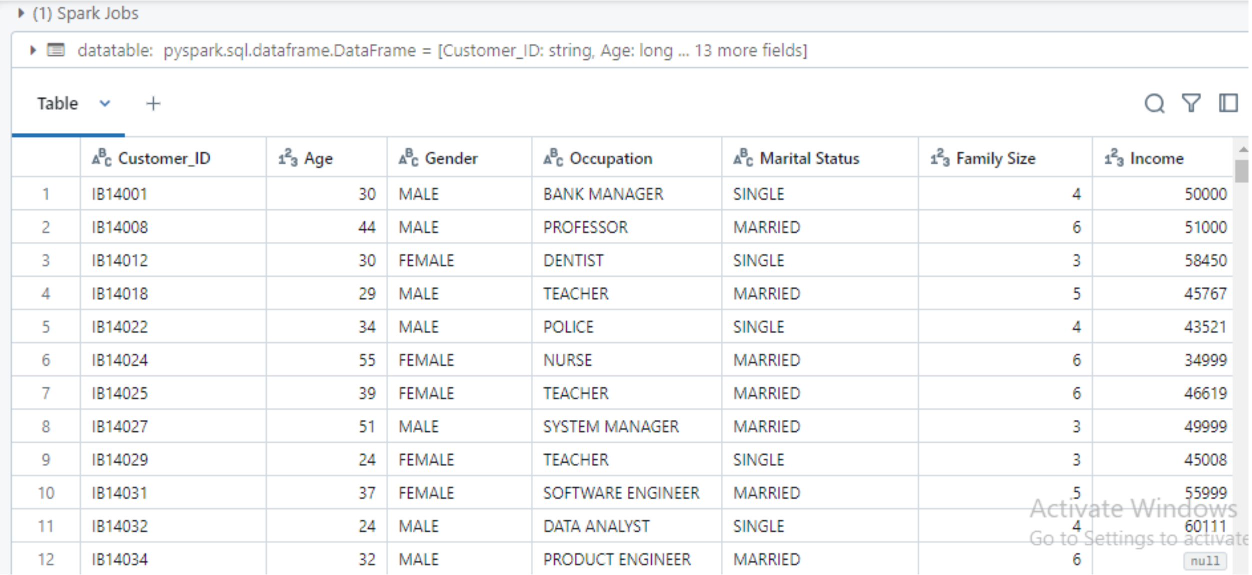
Practice of Loading Data:-

1. "Load and Display Loan Table Data"

→ # data =spark.read.table("samples.nyctaxi.trips")

datatable =spark.read.table("hive_metastore.default.loan")

datatable.display()



▸ (1) Spark Jobs

▸ datatable: pyspark.sql.dataframe.DataFrame = [Customer_ID: string, Age: long ... 13 more fields]

	Customer_ID	Age	Gender	Occupation	Marital Status	Family Size	Income
1	IB14001	30	MALE	BANK MANAGER	SINGLE	4	50000
2	IB14008	44	MALE	PROFESSOR	MARRIED	6	51000
3	IB14012	30	FEMALE	DENTIST	SINGLE	3	58450
4	IB14018	29	MALE	TEACHER	MARRIED	5	45767
5	IB14022	34	MALE	POLICE	SINGLE	4	43521
6	IB14024	55	FEMALE	NURSE	MARRIED	6	34999
7	IB14025	39	FEMALE	TEACHER	MARRIED	6	46619
8	IB14027	51	MALE	SYSTEM MANAGER	MARRIED	3	49999
9	IB14029	24	FEMALE	TEACHER	SINGLE	3	45008
10	IB14031	37	FEMALE	SOFTWARE ENGINEER	MARRIED	5	55999
11	IB14032	24	MALE	DATA ANALYST	SINGLE	4	60111
12	IB14034	32	MALE	PRODUCT ENGINEER	MARRIED	6	null

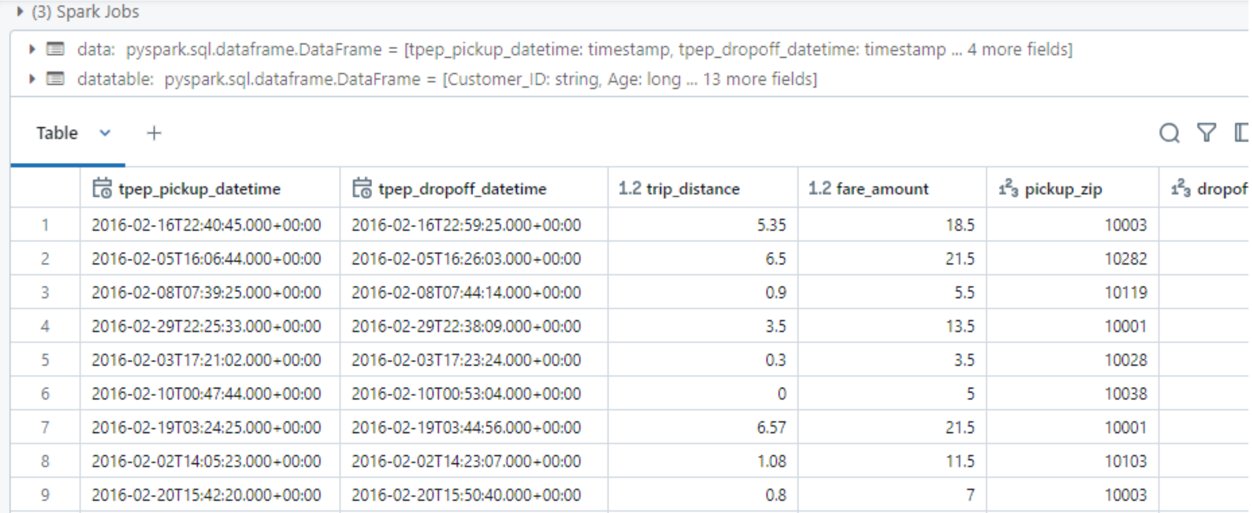
2. "Create RDDs and Load Delta Tables"

→ # to create rdds and dataframe

from pyspark import SparkContext

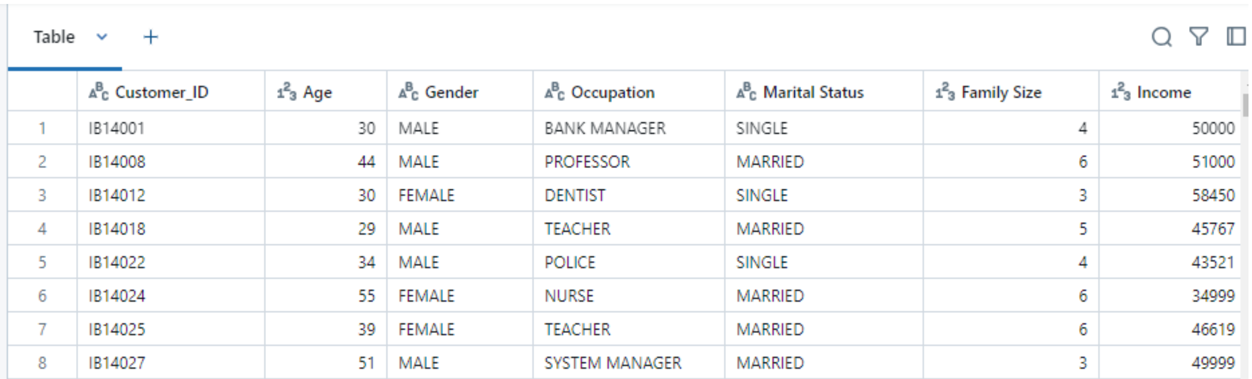
from pyspark.sql import SparkSession

```
# Initialize SparkContext and SparkSession

sc = SparkContext.getOrCreate()

spark = SparkSession.builder.appName('pyspark first program').getOrCreate()

data = spark.read.format("delta").load("dbfs:/databricks-datasets/nyctaxi-with-zipcodes/subsampled")

datatable = spark.read.format("delta").load("dbfs:/user/hive/warehouse/loan")

data.display()

datatable.display()
```



```
▸ (3) Spark Jobs
▸ data:  pyspark.sql.dataframe.DataFrame = [tpep_pickup_datetime: timestamp, tpep_dropoff_datetime: timestamp ... 4 more fields]
▸ datatable:  pyspark.sql.dataframe.DataFrame = [Customer_ID: string, Age: long ... 13 more fields]
```

	tpep_pickup_datetime	tpep_dropoff_datetime	1.2 trip_distance	1.2 fare_amount	123 pickup_zip	123 dropof
1	2016-02-16T22:40:45.000+00:00	2016-02-16T22:59:25.000+00:00	5.35	18.5	10003	
2	2016-02-05T16:06:44.000+00:00	2016-02-05T16:26:03.000+00:00	6.5	21.5	10282	
3	2016-02-08T07:39:25.000+00:00	2016-02-08T07:44:14.000+00:00	0.9	5.5	10119	
4	2016-02-29T22:25:33.000+00:00	2016-02-29T22:38:09.000+00:00	3.5	13.5	10001	
5	2016-02-03T17:21:02.000+00:00	2016-02-03T17:23:24.000+00:00	0.3	3.5	10028	
6	2016-02-10T00:47:44.000+00:00	2016-02-10T00:53:04.000+00:00	0	5	10038	
7	2016-02-19T03:24:25.000+00:00	2016-02-19T03:44:56.000+00:00	6.57	21.5	10001	
8	2016-02-02T14:05:23.000+00:00	2016-02-02T14:23:07.000+00:00	1.08	11.5	10103	
9	2016-02-20T15:42:20.000+00:00	2016-02-20T15:50:40.000+00:00	0.8	7	10003	



	ABC Customer_ID	123 Age	ABC Gender	ABC Occupation	ABC Marital Status	123 Family Size	123 Income
1	IB14001	30	MALE	BANK MANAGER	SINGLE	4	50000
2	IB14008	44	MALE	PROFESSOR	MARRIED	6	51000
3	IB14012	30	FEMALE	DENTIST	SINGLE	3	58450
4	IB14018	29	MALE	TEACHER	MARRIED	5	45767
5	IB14022	34	MALE	POLICE	SINGLE	4	43521
6	IB14024	55	FEMALE	NURSE	MARRIED	6	34999
7	IB14025	39	FEMALE	TEACHER	MARRIED	6	46619
8	IB14027	51	MALE	SYSTEM MANAGER	MARRIED	3	49999

Summary of Loading Data: -

In the first code block, I used PySpark to create a Spark session, which is essential for processing data in Databricks. I then loaded the loan data stored in a Delta format table from the Databricks File System (DBFS) into a DataFrame using spark.read.format("delta"). Delta format offers several advantages such as ACID transactions and time travel, making it a reliable choice for working with large datasets in Databricks. After loading the data, I displayed it to visually inspect the information, which allows me to quickly understand the structure of the dataset.

In the second code block, I accessed two tables from the Databricks metastore using spark.table(). This method allows me to easily query tables that have already been registered in the metastore, which is a centralized place to manage metadata for structured data. The first table, loan_table, was loaded from the default schema (hive_metastore.default), while the second table, trips_table, came from the samples.nyctaxi schema. By displaying both tables, I can examine the content and start analyzing them for insights. These two tables represent two different kinds of data: financial data in the loan_table and transportation data in the trips_table.

This entire process showcases the simplicity and flexibility of working with various data formats (like Delta) and managing data in Databricks using PySpark, which is a powerful tool for big data analysis. With this setup, I can perform various analyses, transformations, and queries on the data to derive meaningful insights.