**Name:** V Venkata Sri Prasad

**Batch:** Data Engineering

**Spark Coding Assignment**

**Question:**

Execute Pyspark -sparksql joins & Applying Functions in a Pandas DataFrame

**Procedure:**

1. **Creating a Spark Session:**

A SparkSession is the entry point to Spark that allows interaction with Spark functionality.



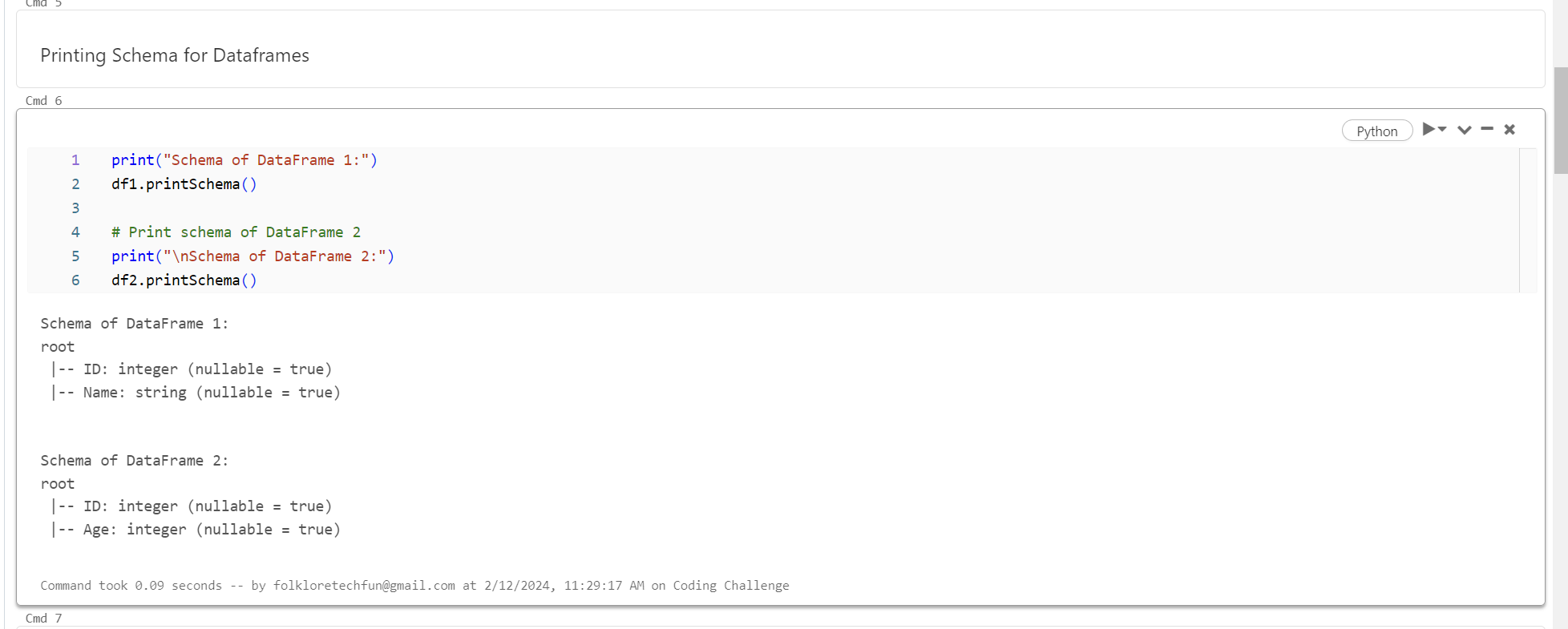
1. **Creating a Dataframe:**

A DataFrame in Spark is a distributed collection of data organized into named columns. It provides a higher-level API for working with structured and semi-structured data, enabling easy manipulation, querying, and analysis.



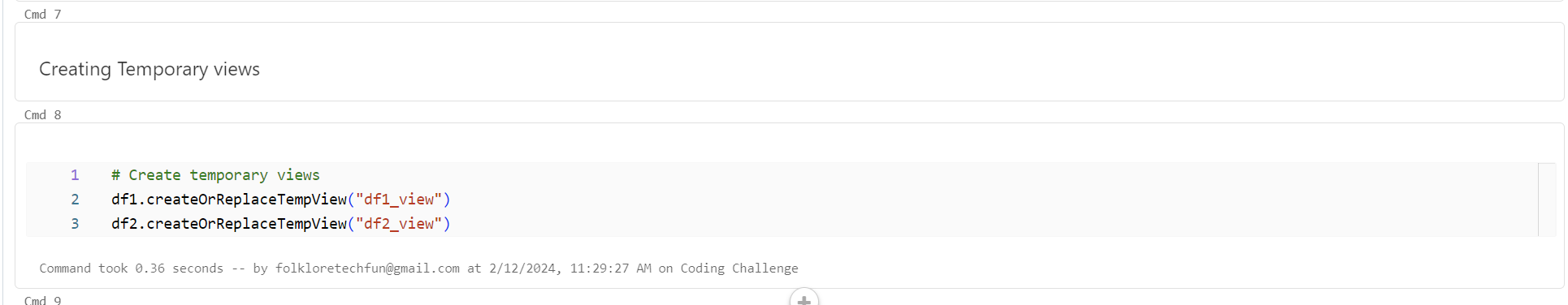
**Printing Schema:**

Printing schema in PySpark DataFrame allows you to visualize the structure of the data, including column names and their data types.



**Creating Temporary Views:**

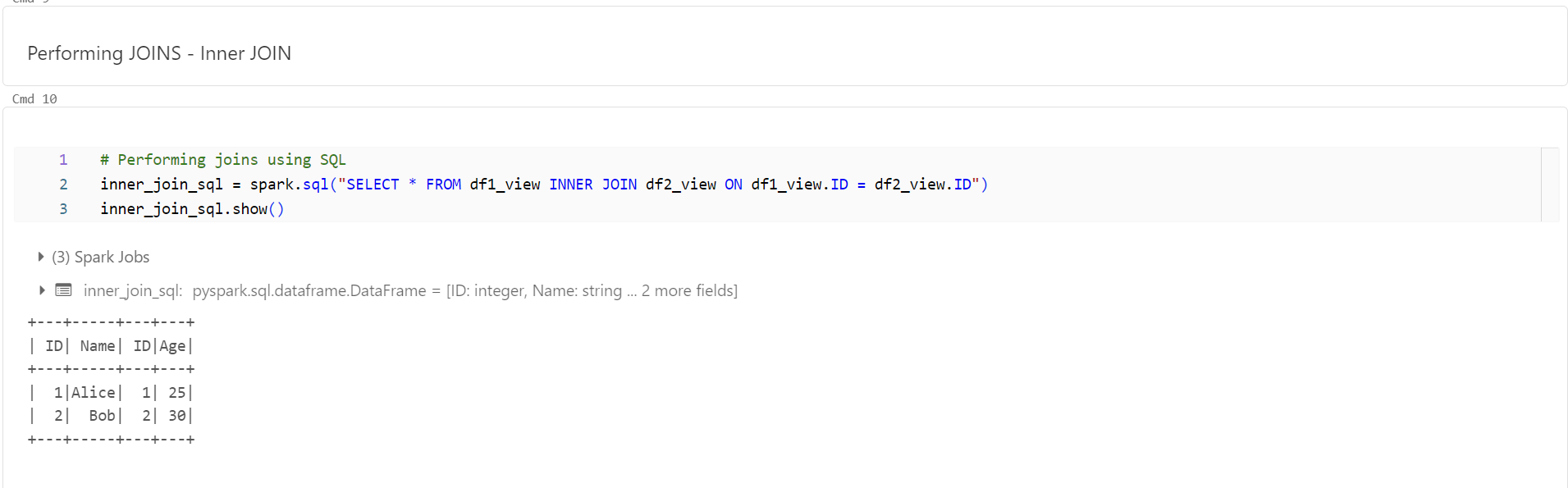
Creating temporary views in PySpark DataFrame enables you to use SQL queries on the DataFrame. It converts the DataFrame into a temporary table-like structure accessible using SQL, facilitating easier data manipulation and analysis.



**Join Operations:**

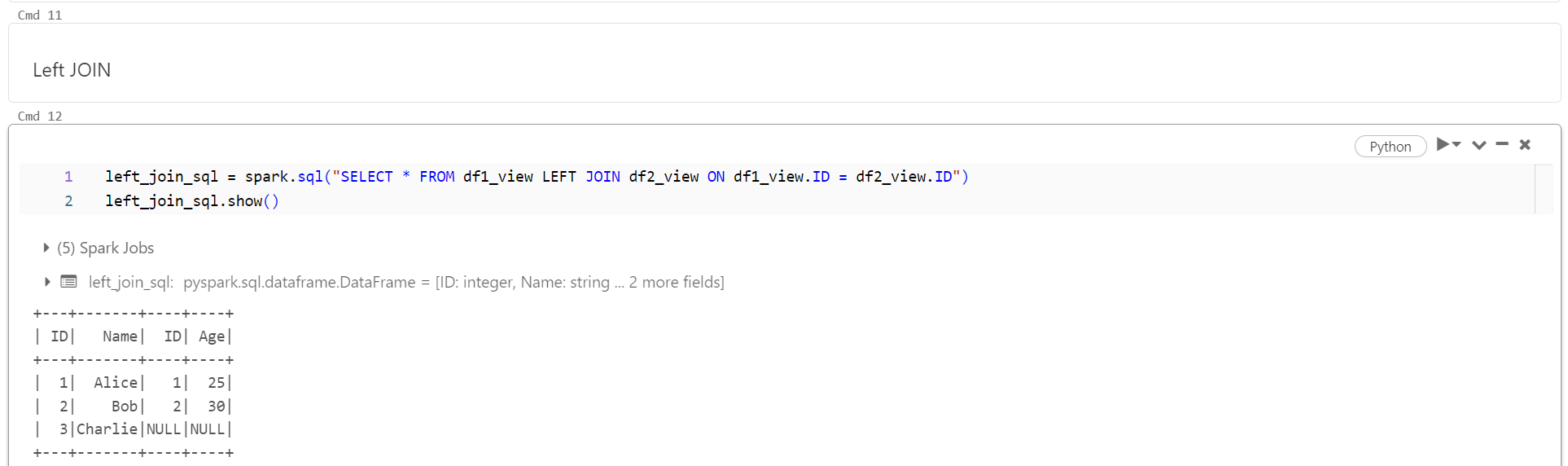
**Inner Join:**

Returns only the rows with matching keys in both DataFrames, eliminating non-matching rows. It's useful for combining related data based on common keys.



**Left Join:**

Retains all rows from the left DataFrame and includes matching rows from the right DataFrame. It ensures that all rows from the left DataFrame are preserved.



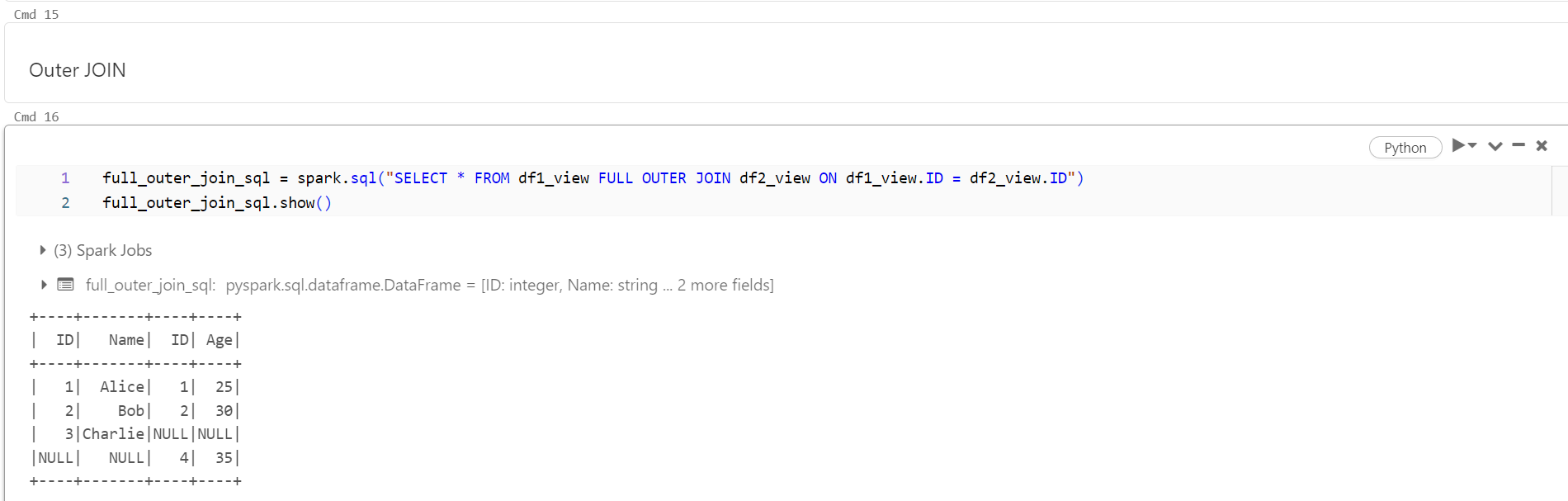
**Right Join:**

Retains all rows from the right DataFrame and includes matching rows from the left DataFrame. It ensures that all rows from the right DataFrame are preserved.



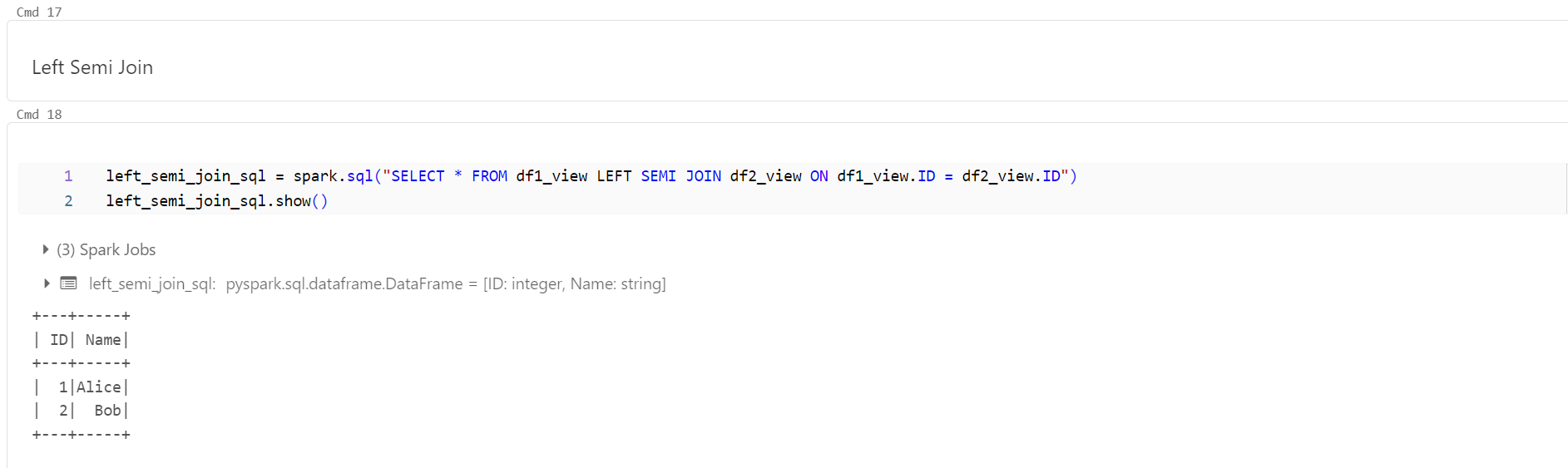
**Full Outer Join:**

Includes all rows from both DataFrames and fills in missing values with NULL where there is no match. It ensures that no data is lost during the join operation.



**Left Semi Join:**

Returns only the rows from the left DataFrame where there is a match in the right DataFrame. It's useful for filtering rows in the left DataFrame based on the existence of a match in the right DataFrame.



**Applying Functions using Pandas**

**User-defined functions (UDFs)**

User-defined functions (UDFs) offer flexibility by allowing users to define custom Python functions to extend PySpark's functionality for specific data processing requirements.

**Pandas:**

pandas\_udf allows you to apply Pandas functions directly to PySpark DataFrames, leveraging the efficiency of Pandas for element-wise operations.

