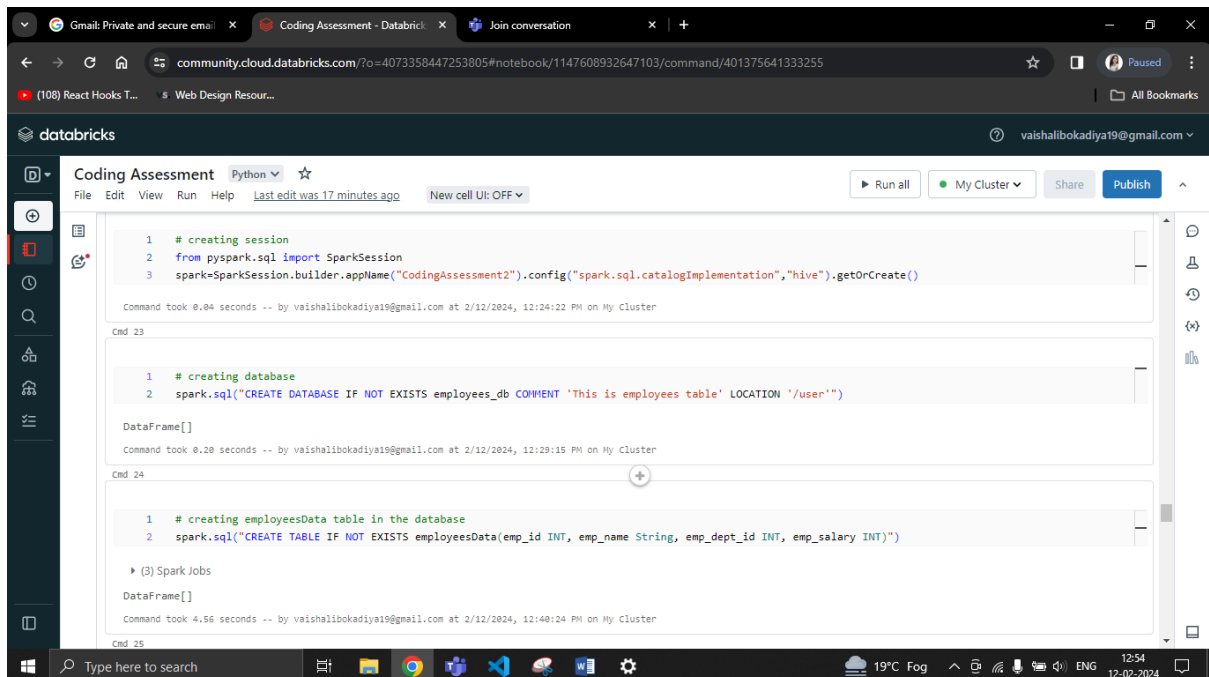


**VAISHALI BOKADIYA**  
**CODING ASSESSMENT**  
**QUESTION 2**

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

## Creating databases:



The screenshot shows a Databricks notebook titled "Coding Assessment" with the following code in the first cell:

```
1 # creating session
2 from pyspark.sql import SparkSession
3 spark=SparkSession.builder.appName("CodingAssessment2").config("spark.sql.catalogImplementation","hive").getOrCreate()
```

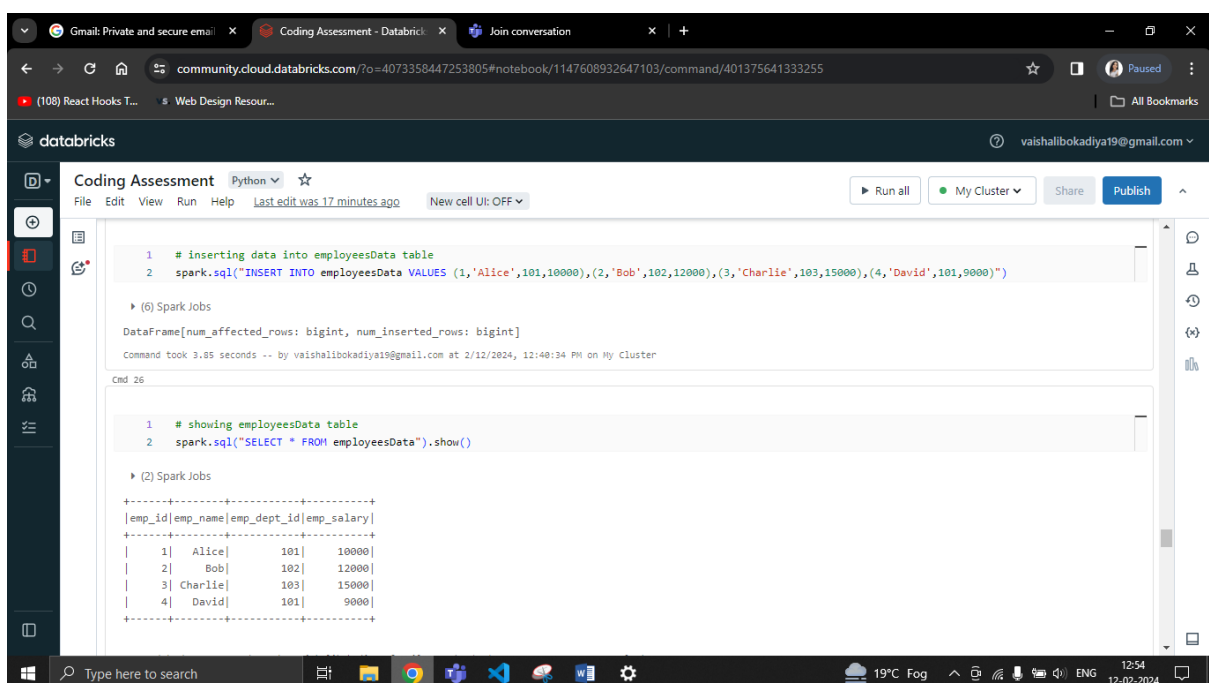
The output shows the command took 0.04 seconds. The second cell contains the following code:

```
1 # creating database
2 spark.sql("CREATE DATABASE IF NOT EXISTS employees_db COMMENT 'This is employees table' LOCATION '/user/'")
```

The output shows the command took 0.20 seconds. The third cell contains the following code:

```
1 # creating employeesData table in the database
2 spark.sql("CREATE TABLE IF NOT EXISTS employeesData(emp_id INT, emp_name String, emp_dept_id INT, emp_salary INT)")
```

The output shows the command took 4.56 seconds. The notebook interface includes a sidebar with navigation icons, a top bar with the Databricks logo and user information, and a bottom bar with a search bar and system status.



The screenshot shows a Databricks notebook titled "Coding Assessment" with the following code in the first cell:

```
1 # inserting data into employeesData table
2 spark.sql("INSERT INTO employeesData VALUES (1,'Alice',101,10000),(2,'Bob',102,12000),(3,'Charlie',103,15000),(4,'David',101,9000)")
```

The output shows the command took 3.85 seconds. The second cell contains the following code:

```
1 # showing employeesData table
2 spark.sql("SELECT * FROM employeesData").show()
```

The output shows the command took 0.20 seconds and displays the following table:

emp_id	emp_name	emp_dept_id	emp_salary
1	Alice	101	10000
2	Bob	102	12000
3	Charlie	103	15000
4	David	101	9000

The notebook interface includes a sidebar with navigation icons, a top bar with the Databricks logo and user information, and a bottom bar with a search bar and system status.

community.cloud.databricks.com/?o=4073358447253805#notebook/1147608932647103/command/401375641333255

databricks vaishalibokadiya19@gmail.com

### Coding Assessment

Python Last edit was 18 minutes ago New cell UI: OFF

Run all My Cluster Share Publish

```
1 # creating Departments table in the database
2 spark.sql("CREATE TABLE IF NOT EXISTS departments(dept_id INT, dept_name String)")
```

(3) Spark Jobs

DataFrame[]

Command took 3.31 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:40:56 PM on My Cluster

Cmd 28

```
1 # inserting data into Departments table
2 spark.sql("INSERT INTO Departments VALUES (101,'Engineering'),(102,'Sales'),(103,'Marketing'),(104,'HR')")
```

(6) Spark Jobs

DataFrame[num\_affected\_rows: bigint, num\_inserted\_rows: bigint]

Command took 6.31 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:42:33 PM on My Cluster

Cmd 29

```
1 # Showing departments table
2 spark.sql("SELECT * FROM Departments").show()
```

(2) Spark Jobs

community.cloud.databricks.com/?o=4073358447253805#notebook/1147608932647103/command/401375641333255

databricks vaishalibokadiya19@gmail.com

### Coding Assessment

Python Last edit was 18 minutes ago New cell UI: OFF

Run all My Cluster Share Publish

```
1 # inserting data into Departments table
2 spark.sql("INSERT INTO Departments VALUES (101,'Engineering'),(102,'Sales'),(103,'Marketing'),(104,'HR')")
```

(6) Spark Jobs

DataFrame[num\_affected\_rows: bigint, num\_inserted\_rows: bigint]

Command took 6.31 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:42:33 PM on My Cluster

Cmd 29

```
1 # Showing departments table
2 spark.sql("SELECT * FROM Departments").show()
```

(2) Spark Jobs

```
+-----+
|dept_id| dept_name|
+-----+
| 101 | Engineering|
| 102 | Sales|
| 103 | Marketing|
| 104 | HR|
+-----+
```

Command took 1.97 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:43:02 PM on My Cluster

Cmd 30

# Performing Joins:

The screenshot shows a Databricks notebook titled "Coding Assessment" with a Python kernel. The code in the cell performs an inner join between `employeesData` and `Departments` on the `emp_dept_id` column. The output displays a table with 4 rows and 6 columns: `emp_id`, `emp_name`, `emp_dept_id`, `emp_salary`, `dept_id`, and `dept_name`.

```
1 # Inner join
2 spark.sql("SELECT * FROM employeesData INNER JOIN Departments ON employeesData.emp_dept_id=Departments.dept_id").show()
```

emp_id	emp_name	emp_dept_id	emp_salary	dept_id	dept_name
1	Alice	101	10000	101	Engineering
2	Bob	102	12000	102	Sales
3	Charlie	103	15000	103	Marketing
4	David	101	9000	101	Engineering

The screenshot shows the same Databricks notebook, but the code now performs a left join between `employeesData` and `Departments` on the `emp_dept_id` column. The output displays a table with 4 rows and 6 columns: `emp_id`, `emp_name`, `emp_dept_id`, `emp_salary`, `dept_id`, and `dept_name`.

```
1 # left join
2 spark.sql("SELECT * FROM employeesData LEFT JOIN Departments ON employeesData.emp_dept_id=Departments.dept_id").show()
```

emp_id	emp_name	emp_dept_id	emp_salary	dept_id	dept_name
1	Alice	101	10000	101	Engineering
2	Bob	102	12000	102	Sales
3	Charlie	103	15000	103	Marketing
4	David	101	9000	101	Engineering

community.cloud.databricks.com/?o=4073358447253805#notebook/1147608932647103/command/401375641333255

databricks vaishalibokadiya19@gmail.com

Coding Assessment Python ☆

File Edit View Run Help Last edit was 18 minutes ago New cell UI: OFF

Run all My Cluster Share Publish

```
1 | 3| Charlie| 103| 15000| 103| Marketing|
2 | 4| David| 101| 9000| 101| Engineering|
+-----+-----+-----+-----+-----+-----+
Command took 1.44 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:44:50 PM on My Cluster
```

Cmd 32

```
1 # right join
2 spark.sql("SELECT * FROM employeesData RIGHT JOIN Departments ON employeesData.emp_dept_id=Departments.dept_id").show()
```

(2) Spark Jobs

emp_id	emp_name	emp_dept_id	emp_salary	dept_id	dept_name
4	David	101	9000	101	Engineering
1	Alice	101	10000	101	Engineering
2	Bob	102	12000	102	Sales
3	Charlie	103	15000	103	Marketing
NULL	NULL	NULL	NULL	104	HR

Command took 1.61 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:45:08 PM on My Cluster

Cmd 33

community.cloud.databricks.com/?o=4073358447253805#notebook/1147608932647103/command/401375641333255

databricks vaishalibokadiya19@gmail.com

Coding Assessment Python ☆

File Edit View Run Help Last edit was 18 minutes ago New cell UI: OFF

Run all My Cluster Share Publish

```
1 | 2| Bob| 102| 12000| 102| Sales|
2 | 3| Charlie| 103| 15000| 103| Marketing|
3 | NULL| NULL| NULL| NULL| 104| HR|
+-----+-----+-----+-----+-----+-----+
Command took 1.61 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:45:08 PM on My Cluster
```

Cmd 33

```
1 # full/outer join
2 spark.sql("SELECT * FROM employeesData FULL JOIN Departments ON employeesData.emp_dept_id=Departments.dept_id").show()
```

(3) Spark Jobs

emp_id	emp_name	emp_dept_id	emp_salary	dept_id	dept_name
1	Alice	101	10000	101	Engineering
4	David	101	9000	101	Engineering
2	Bob	102	12000	102	Sales
3	Charlie	103	15000	103	Marketing
NULL	NULL	NULL	NULL	104	HR

Command took 1.63 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:45:44 PM on My Cluster

# Applying functions in a pandas dataframes:

The screenshot shows a Databricks notebook titled "Coding Assessment" in Python. The code in the first cell creates a pandas DataFrame with two columns: "marks\_scored" and "negative\_marks". The DataFrame contains five rows of data. The output of the code is displayed below the code cell.

```
1 # applying functions in a pandas dataframe
2 import pyspark.pandas as pd
3 marks={"marks_scored":[90,67,98,56,87],
4        "negative_marks":[12,5,7,19,20]}
5 my_df=pd.DataFrame(marks)
6 print(my_df)
```

Command took 1.63 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 12:45:44 PM on My Cluster

Cmd 34

	marks_scored	negative_marks
0	90	12
1	67	5
2	98	7
3	56	19
4	87	20

Command took 0.61 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 1:13:10 PM on My Cluster

Cmd 35

```
1 def calculate_scores(scores):
2     return scores[0]-scores[1]
```

The screenshot shows the same Databricks notebook, but now the second cell is being executed. The code defines a function "calculate\_scores" that takes a list of two scores and returns their difference. It then applies this function to the "marks\_scored" and "negative\_marks" columns of the DataFrame. The output shows the resulting DataFrame with a new column "final\_score".

```
1 def calculate_scores(scores):
2     return scores[0]-scores[1]
3
4 final_score=my_df.apply(calculate_scores)
5 print(final_score)
```

Command took 0.61 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 1:13:10 PM on My Cluster

Cmd 35

	marks_scored	negative_marks	final_score
0	90	12	78
1	67	5	62
2	98	7	91
3	56	19	37
4	87	20	67

Command took 0.84 seconds -- by vaishalibokadiya19@gmail.com at 2/12/2024, 1:13:17 PM on My Cluster

[Shift+Enter] to run  
[Shift+Ctrl+Enter] to run selected text