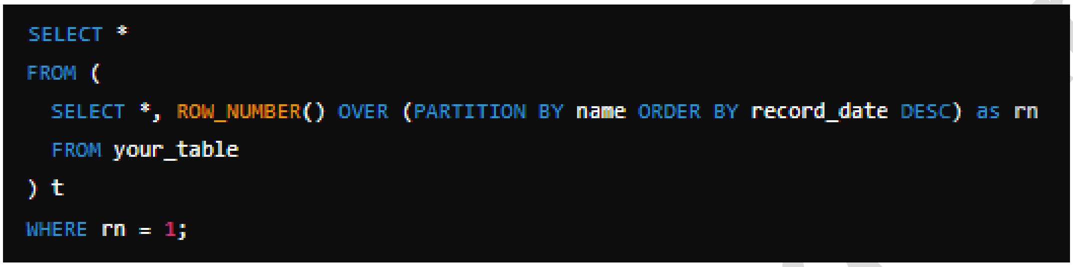
Pubmatic Data Engineer Interview Guide – Experienced 3+

Technical Round 1

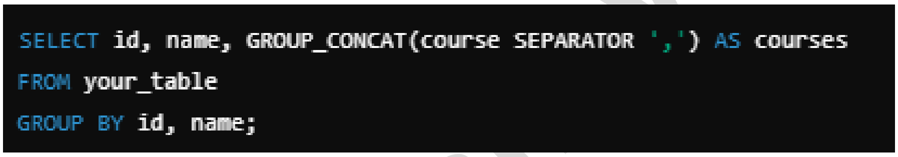
1. **Briefly introduce yourself** – Prepare a short, engaging summary highlighting your key projects, technologies, and skills relevant to the Data Engineering role.

2. **Print only the newest record for each name** – Use **SQL Window functions** (ROW_NUMBER, RANK, etc.) for this.



```
SELECT *
FROM (
  SELECT *, ROW_NUMBER() OVER (PARTITION BY name ORDER BY record_date DESC) as rn
  FROM your_table
) t
WHERE rn = 1;
```

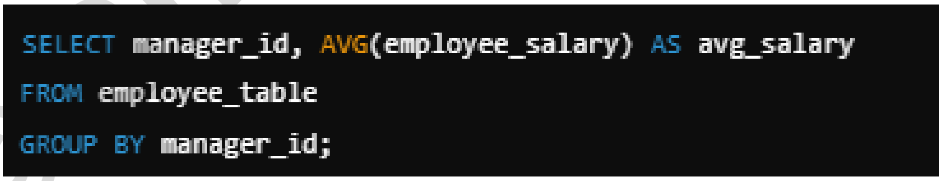
3. **Combine records by name with concatenated course values** – Aggregate values using **GROUP_CONCAT** or **STRING_AGG** depending on the database.



```
SELECT id, name, GROUP_CONCAT(course SEPARATOR ',') AS courses
FROM your_table
GROUP BY id, name;
```

4. **Reverse operation for splitting values back to original format** – Use SQL **JOINs** with **splitting functions** if needed.

5. **Find average salary for each manager** – Assume a table with **manager_id** and **employee_salary**.



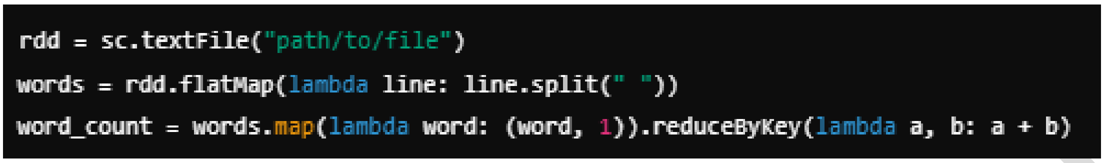
```
SELECT manager_id, AVG(employee_salary) AS avg_salary
FROM employee_table
GROUP BY manager_id;
```

6. **Count records for INNER JOIN and LEFT JOIN** – Use appropriate **JOIN** types to get counts.

   - **INNER JOIN**: Returns matching records from both tables.
   - **LEFT JOIN**: Returns all records from the left table and matching records from the right.

7. **Basic Spark commands** –
    - Create RDD: sc.parallelize([1, 2, 3])
    - Load data: spark.read.csv('path/to/file.csv')
    - Filter: data.filter(data.id > 100)

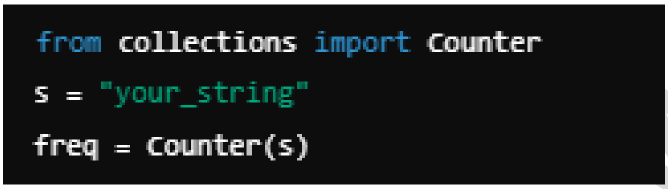
8. **Word count in Spark**:



```
rdd = sc.textFile("path/to/file")
words = rdd.flatMap(lambda line: line.split(" "))
word_count = words.map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)
```

Technical Round 2

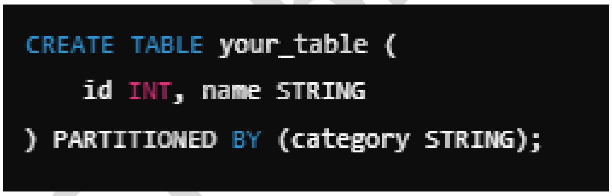
9. **Character frequency in a string (Python)**:



```
from collections import Counter
s = "your_string"
freq = Counter(s)
```

10. **Sort and merge arrays** – Use sorted and merge techniques depending on the array structure.

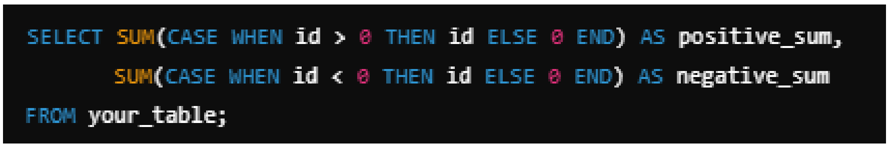
11. **Create partitioned table**:



```
CREATE TABLE your_table (
    id INT, name STRING
) PARTITIONED BY (category STRING);
```

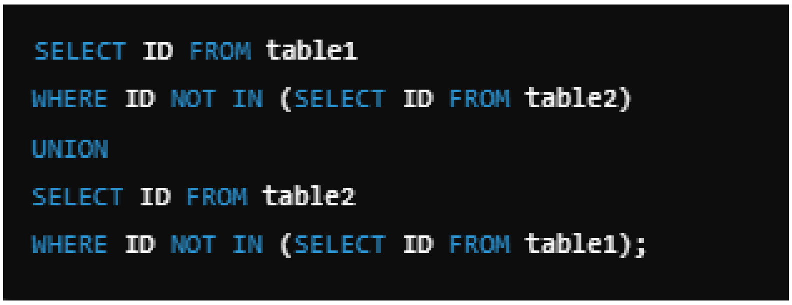
12. **Load data into Hive table from HDFS or local**:
    - **HDFS**: LOAD DATA INPATH 'hdfs_path' INTO TABLE your_table;
    - **Local**: LOAD DATA LOCAL INPATH 'local_path' INTO TABLE your_table;

13. **Sum of positive and negative numbers in a column**:



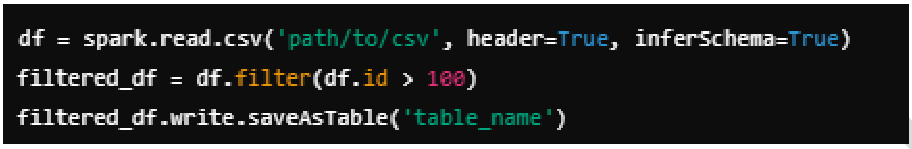
```
SELECT SUM(CASE WHEN id > 0 THEN id ELSE 0 END) AS positive_sum,
       SUM(CASE WHEN id < 0 THEN id ELSE 0 END) AS negative_sum
FROM your_table;
```

14. **Find non-common records in two tables** (SQL EXCEPT or NOT IN):



```
SELECT ID FROM table1
WHERE ID NOT IN (SELECT ID FROM table2)
UNION
SELECT ID FROM table2
WHERE ID NOT IN (SELECT ID FROM table1);
```

15. **Read CSV, filter, and write to table using PySpark**:



```
df = spark.read.csv('path/to/csv', header=True, inferSchema=True)
filtered_df = df.filter(df.id > 100)
filtered_df.write.saveAsTable('table_name')
```

Glassdoor Pubmatic Review –

https://www.glassdoor.co.in/Reviews/PubMatic-Reviews-E256835.htm

Pubmatic Careers –

https://pubmatic.com/careers/home/

Subscribe to my YouTube Channel for Free Data Engineering Content –

https://www.youtube.com/@shubhamwadekar27

Connect with me here –

https://bento.me/shubhamwadekar

Checkout more Interview Preparation Material on –

https://topmate.io/shubham_wadekar