Seekho Bigdata
Data is the New Oil

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 11 — Data Skew, Salting & Advanced Join Optimization

Karthik Kondpak
9989454737

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 11 — Data Skew, Salting & Advanced Join Optimization

Concepts Covered Today

- What is data skew and why it happens
- How to identify skew
- Join skew vs aggregation skew
- Salting technique (interview favourite)
- Broadcast vs AQE vs skew hints

Scenario

You work for an **Indian UPI / fintech platform**.

`Table: transactions`

- 2 billion rows

- 40% of transactions belong to **customer_id = 'PAYTM_MERCHANT'**

Job: Calculate **daily total amount per merchant**.

🚨 Problem: What is Data Skew?

🔹 Interview Definition

Data skew happens when **few keys contain a very large portion of data**, causing:

- Some partitions to be extremely large
- Few tasks running very long
- Other executors staying idle

✅ Question 1: How to Identify Data Skew?

🔹 Scenario

Spark job is slow and stuck at 90%.

🧪 PySpark Solution

```
transactions_df.groupBy("merchant_id")
    .count()
    .orderBy("count", ascending=False)
    .show()
```

📝 Explanation

- One key dominating output → skew confirmed
- Interviewers love this simple detection logic

✅ Question 2: Why Skewed Joins Are Dangerous?

🔹 Explanation

- Skewed key sent to **single reducer**
- Causes OOM or long-running tasks
- Cluster resources wasted

🧂 Question 3: Salting Technique (MOST ASKED)

🔹 Scenario

Join `transactions` with `merchants` table.

🧪 Step 1: Add Salt Column to Large Table

```
from pyspark.sql.functions import rand, floor

salted_txn_df = transactions_df.withColumn(
    "salt",
    floor(rand() * 10)
)
```

🧪 Step 2: Expand Small Table

```
from pyspark.sql.functions import explode, array

salted_merchants_df = merchants_df.withColumn(
    "salt",
    explode(array([0,1,2,3,4,5,6,7,8,9]))
```

```
)
```

🧪 Step 3: Perform Salted Join

```
final_df = salted_txn_df.join(
    salted_merchants_df,
    ["merchant_id", "salt"],
    "inner"
)
```

📝 Why This Works

- Skewed key distributed across multiple partitions
- Load balanced across executors

⚡ Question 4: Broadcast vs Salting — When to Use What?

✅ Broadcast

- One table is small (< few 100 MB)
- Avoids shuffle completely

✅ Salting

- Both tables are large
- Skewed keys exist

🤖 Question 5: AQE (Adaptive Query Execution)

🔹 Scenario

Spark 3.x environment

🧪 Enable AQE

```
spark.conf.set("spark.sql.adaptive.enabled", "true")
```

📝 What AQE Does

- Automatically handles skew joins
- Changes join strategy at runtime
- Splits skewed partitions

Let's build your Data Engineering journey together!

Call us directly at: 9989454737

https://seekhobigdata.com/