Seekho Bigdata
Data is the New Oil

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 25 — End-to-End Spark Optimization Case Study

Karthik Kondpak
9989454737

PySpark Scenario-Based Interview Notes (End-to-End Series)

DAY 25 — End-to-End Spark Optimization Case Study (Real Interview Simulation)

Interview Case Study Problem Statement

Scenario

You work for a **fintech company (UPI / Wallet / Credit Platform)**.

Daily pipeline:

- **transactions_fact** → 2 billion rows/day
- **customers_dim** → 20 million rows
- **merchants_dim** → 1 million rows

Current Problems

- Job takes **4–5 hours**
- Frequent **OOM errors**
- Shuffle spill to disk
- Millions of **small files** in output
- Severe **data skew** on top merchants

Interview Expectation

“How would you optimize this pipeline end-to-end?”

You are expected to explain:

Join strategy

Skew handling

Partitioning

Memory tuning

File layout optimization

STEP 1 — Understand the Data (Always Say This First)

Golden Rule: Never optimize blindly.

- Size of each table
- Join keys
- Skewed keys
- Daily growth

STEP 2 — Join Optimization Strategy

❌ Naive Join

```
fact.join(customers, "customer_id")
```

✅ Optimized Join Plan

- Broadcast `merchants_dim`
- SortMerge Joinfor `customers_dim`

```
from pyspark.sql.functions import broadcast

fact \
 .join(customers, "customer_id") \
 .join(broadcast(merchants), "merchant_id")
```

STEP 3 — Data Skew Handling (HIGH LPA SECTION)

🔥 Problem

Top merchants generate **40% of transactions**.

✅ Solution 1 — Salting

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

fact_salted = fact.withColumn(
    "salted_key",
    concat("merchant_id", rand())
)
```

✅ Solution 2 — AQE (Preferred)

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

STEP 4 — Partitioning Strategy

Wrong

`repartition(1000)`

✅ Right

```
repartition("txn_date")
```

Partition by **filter columns**, not random numbers.

STEP 5 — Memory & Shuffle Optimization

🔹 Key Configurations

```
spark.conf.set("spark.sql.shuffle.partitions", 400)
spark.conf.set("spark.memory.fraction", 0.6)
```

🔹 Avoid

- Too manypartitions
- Uncachedreused DataFrames

```
fac t.p er sis t()
```

STEP 6 — Small Files & Write Optimization

❌ Problem

Each task → one file → millions of files

✅ Solution

```
final_df \
 .coalesce(200) \
 .write \
 .format("delta") \
 .partitionBy("txn_date") \
```

```
  .save("/delta/upi_txns")
```

Delta Optimization

```
OPTIMIZE delta.`/delta/upi_txns`
ZORDER BY (merchant_id)
```

Seekho Bigdata

Data is the New Oil

Let's build your Data Engineering journey together!

Call us directly at: 9989454737

https://seekhobigdata.com/