Seekho Bigdata

Data is the New Oil

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 20 — Joins Deep Dive (Broadcast vs Sort-Merge vs Shuffle Hash)

Karthik Kondpak
9989454737

PySpark Scenario-Based Interview Questions (Complete Notes Series)

DAY 20 — Joins Deep Dive (Broadcast vs Sort-Merge vs Shuffle Hash)

Concepts Covered Today

- Join types in Spark
- Join execution strategies
- Broadcast Join
- Sort-Merge Join
- Shuffle Hash Join
- Join optimization techniques
- Real Indian production scenarios

Why Joins Are Expensive in Spark?

Joins are expensive because they usually involve:

- Data shuffle across executors
- Network transfer
- Disk spill
- Sorting or hashing

0% of Spark performance issues involve joins.

Scenario

You work for an **Indian e-commerce company**.

- Orders table → 2 billion rows
- Customers table → 5 million rows
- Products table → 50 thousand rows

Goal:

- Enrich orders with customer & product data

🔀 Join Strategies in Spark (CORE QUESTION)

Spark uses **Catalyst Optimizer** to automatically select a join strategy:

Broadcast Hash Join (BHJ)

Shuffle Hash Join (SHJ)

Sort-Merge Join (SMJ)

Broadcast Hash Join

🔹 When Spark Uses It

- One table is small enough to fit in memory
- Size < `spark.sql.autoBroadcastJoinThreshold`

```
spark.c onf.ge t("sp ark.s ql.aut oBroadc astJoi nThre shold "
)
```

(Default: 10 MB)

Example

```
from pyspark.sql.functions import broadcast

orders.join(
    broadcast(products),
    "product_id",
    "left"
)
```

✅ **Why It's Fast**

- No shuffle of large table
- Small table sent to all executors

Broadcast Join Interview Trap

Can we broadcast a 2 GB table?

Correct Answer

No. It can cause **OOM errors** on executors.

🔁 Sort-Merge Join (MOST COMMON)

🔹 When Spark Uses It

- Large tables
- Join keys are sortable
- Broadcasting not possible

Execution Steps

1. Shuffle both tables 2.

Sort data on join keys 3.

Merge matching rows

Example

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

Default join strategy for large datasets.

Shuffle Hash Join

🔹 When Spark Uses It

- One table is moderately smaller
- Enough memory to build hash table

⚠️ Less common in Spark SQL

🧠 How Spark Chooses Join Strategy?

Spark considers:

- Table size statistics
- Broadcast threshold
- Join hints
- Sortability of join keys

Join Hints

```
orders.hint("broadcast") \
      .join(products, "product_id")
```

Other hints:

- `merge`
- `shuffle_hash`

Seekho Bigdata

Data is the New Oil

Let's build your Data Engineering journey together!

📩 Call us directly at: 9989454737

🌐 https://seekhobigdata.com/