Vishal Gupta

SPARK CORE FUNDAMENTALS

UnWrap All concept in SImplest way

Adaptive Query Optimization

What is AQE?

AQE stands for Adaptive Query Execution.

AQE was introduced in spark 3.O version.

When Apache spark runs our code at that time there is lot of static planning is involved. When we runs our code spark do a lot of static planning.

But when we run this code at that time we have to take care for mutiple things to optimize performace like
--> What is condition of cluster
--> Size of data
--> Number/Size of Partitions
--> Data Skewness

But Once we execute our Query we cannot change these things.

Hence Spark will do it for us using A cool feature named as AQE

Spark will dynamically adjust our query plan based on runtime statistics Let's deep dive how spark will do this

What AQE will do ?

AQE will optimize the performace of spark job using :

Repartitioning : AQE might adjust number of partitions required to process your job

Data Skewness : AQE might adjust data skewness in your input data.

Join Strategy : AQE might update join strategy based on condition of data

All above things are going to happen at run-time dynamically by using AQE feature to improve the performance.

Once Spark complete this reoptimization. the new optimized plan is going to be executed to improve performance.

AQE Defination -

AQE is a feature in spark that aims to improve the performance of spark by dynamically optimizing and adjusting the query plan during query execution time based on runtime statistics.

AQE feature will be disabled by default. We need to enable AQE using :

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

AQE Realtime Example -

In our life we do lot of future planning but sometimes we need to replan things based on present situations.

Like this spark chnages plans at runtime baased on runtime situations

Features of AQE :

1. **Dynamically Coalescing Shuffle function**
2. **Dynamically Switching Join Strategy.**
3. **Dynamically Optimizing Skew Joins**

Lets discuss all three in detail -->

Dynamically coalescing Shuffle Partition

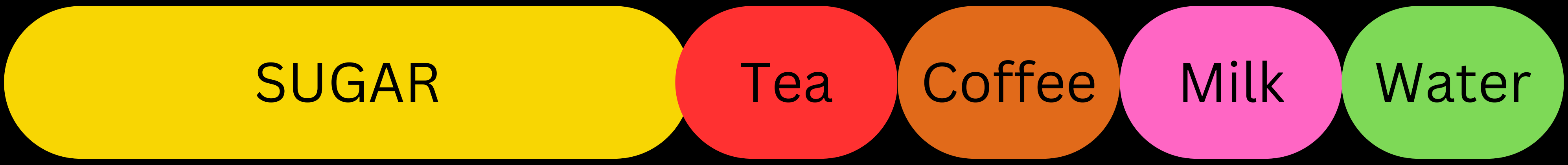
To Understand this We are assuming we have Sales data of Mutiple products.

In this data Sugar is most sold product and we also have other products as Tea, coffee etc.

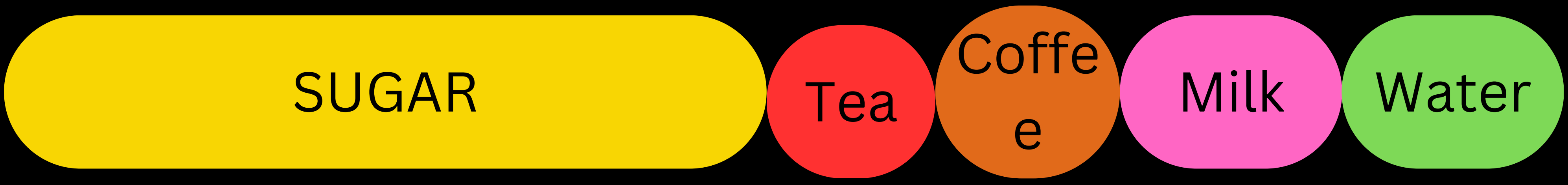
We have mutiple products and we want to **get sales numbers of each products.**

Size of data is 250MB hence we are considering two partitions P1 and P2

When data got shuffled from both partitions It looks like



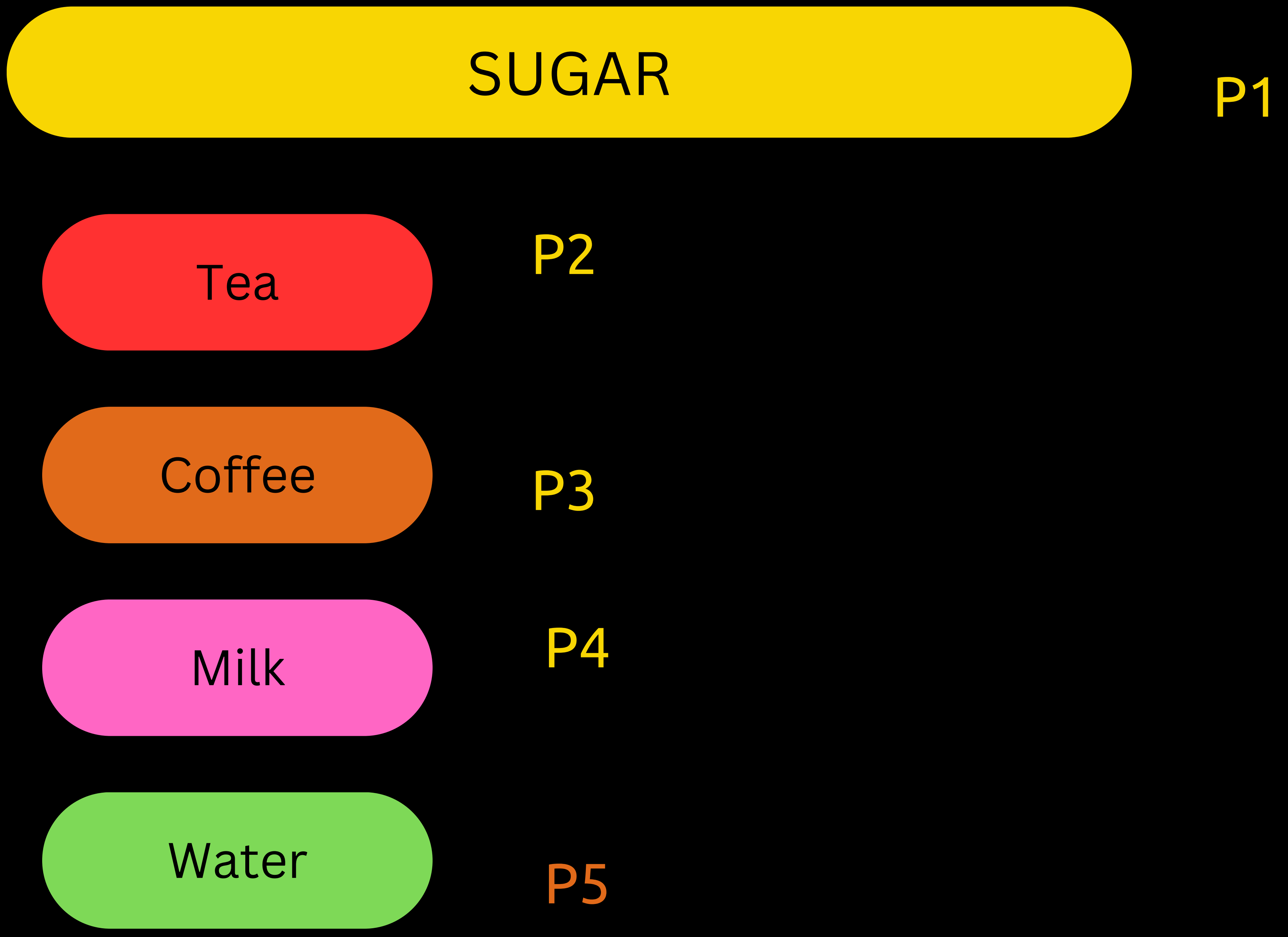
PARTITION 1 WITH DIFFERENT ITEMS



PARTITION 2 WITH DIFFERENT ITEMS

To get the sales Number we need to do grouping of items from both the partitions

Data will be grouped as



SUGAR P1

Tea P2

Coffee P3

Milk P4

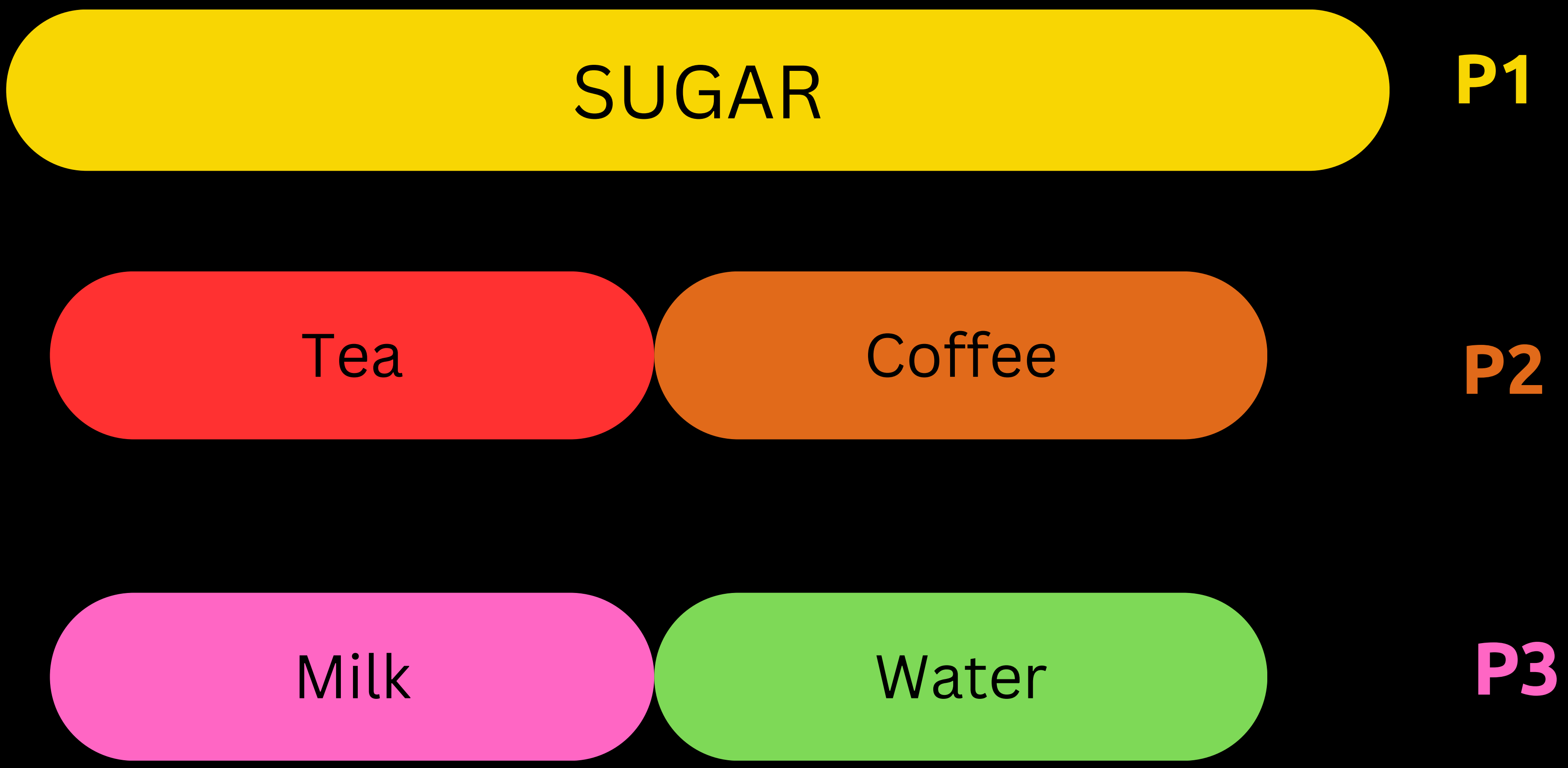
Water P5

After grouping of data from partitions P1 and P2 five partitons has been created and size of partitions are uneven.

P1 will take highest time to execute.

Here comes the work of AQE it will coalesce p2, p3 to make P2 and P3, P4 to make P3 as

AQE will coalesce partitions to optimize the performance
then partitions will look like



SUGAR P1

Tea Coffee P2

Milk Water P3

After coalesce only three partitions are left.

Hence 2 CPU Cores will be saved as Previously it was 5.

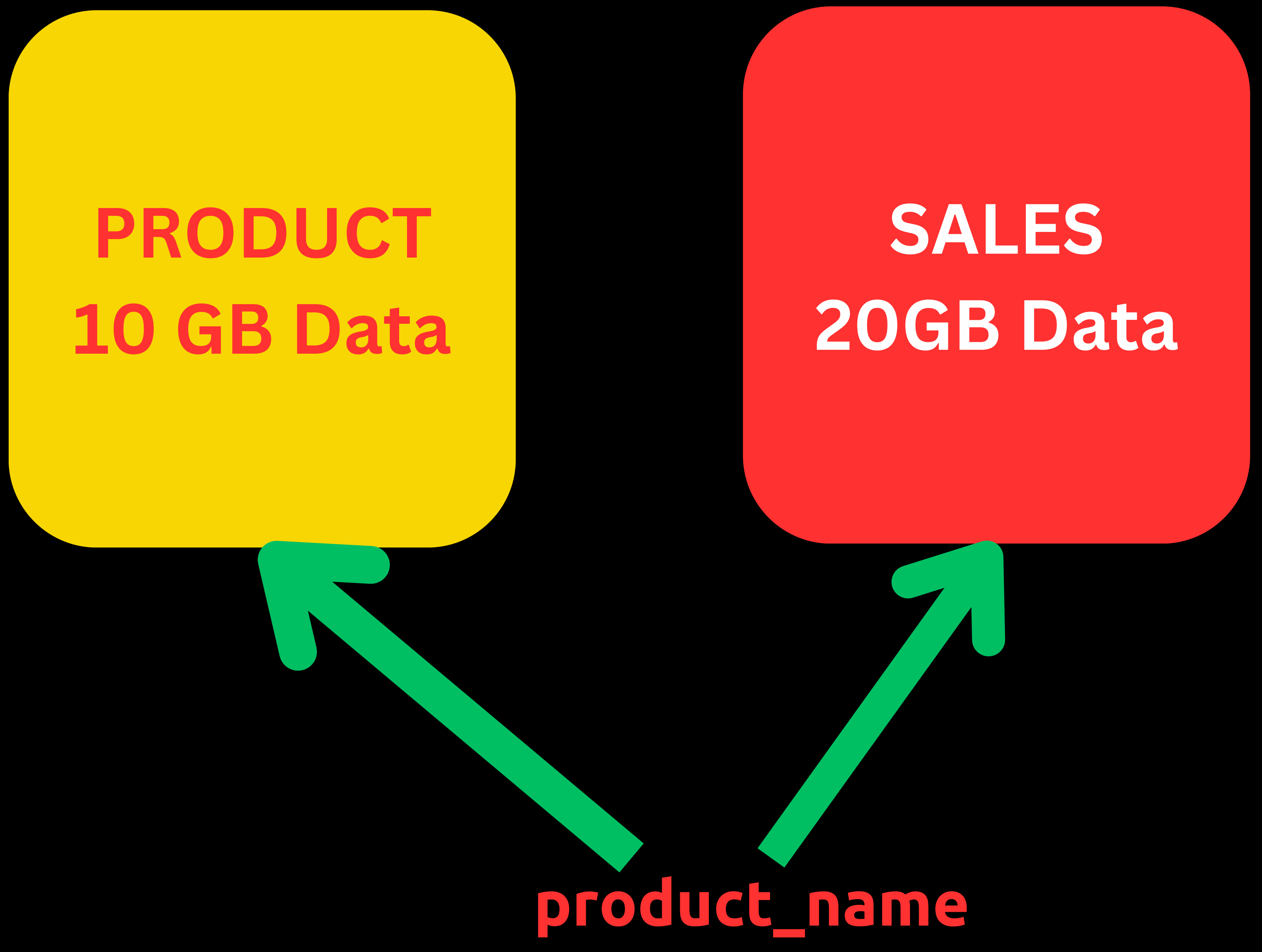
This is how AQE will optimize the performance by coalescing the shuffled partitions.

Dynamically Switching join strategy

To Understand this we will be assuming two tables

Product Table with Data 10 GB

Sales Table with Data 20 GB



PRODUCT
10 GB Data

SALES
20GB Data

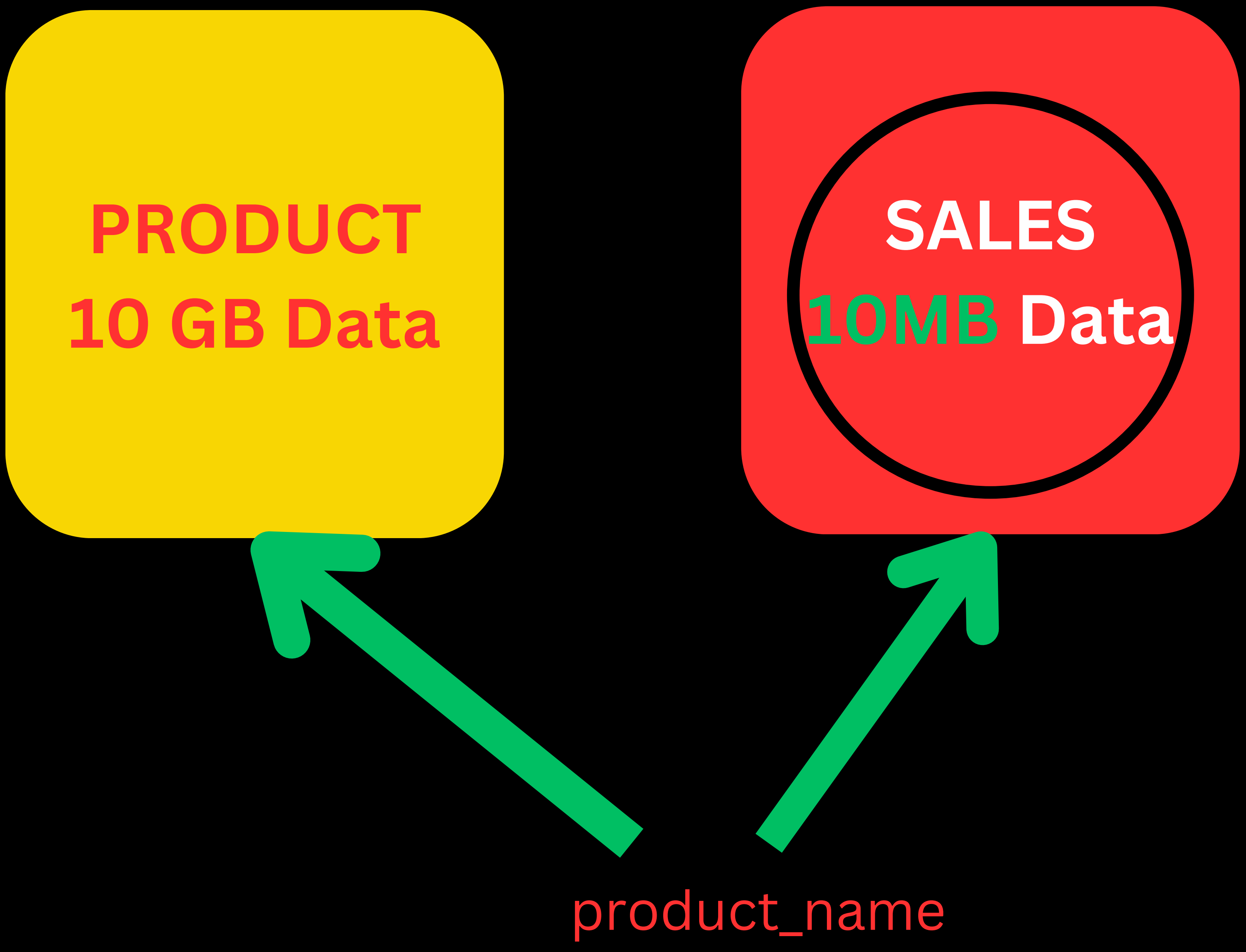
product_name

Our goal is to join both tables based on product_name

Here, By default spark will do Sort-Merge join because spark DAG engine will find size large.

But If After reading data from both tables we applied multiple transformations

& WE ARE LEFT WITH TABLES AS AFTER TRANSFORMATION



After mutiple transformations on runtime **Sales Table data size reduced to 10MB**

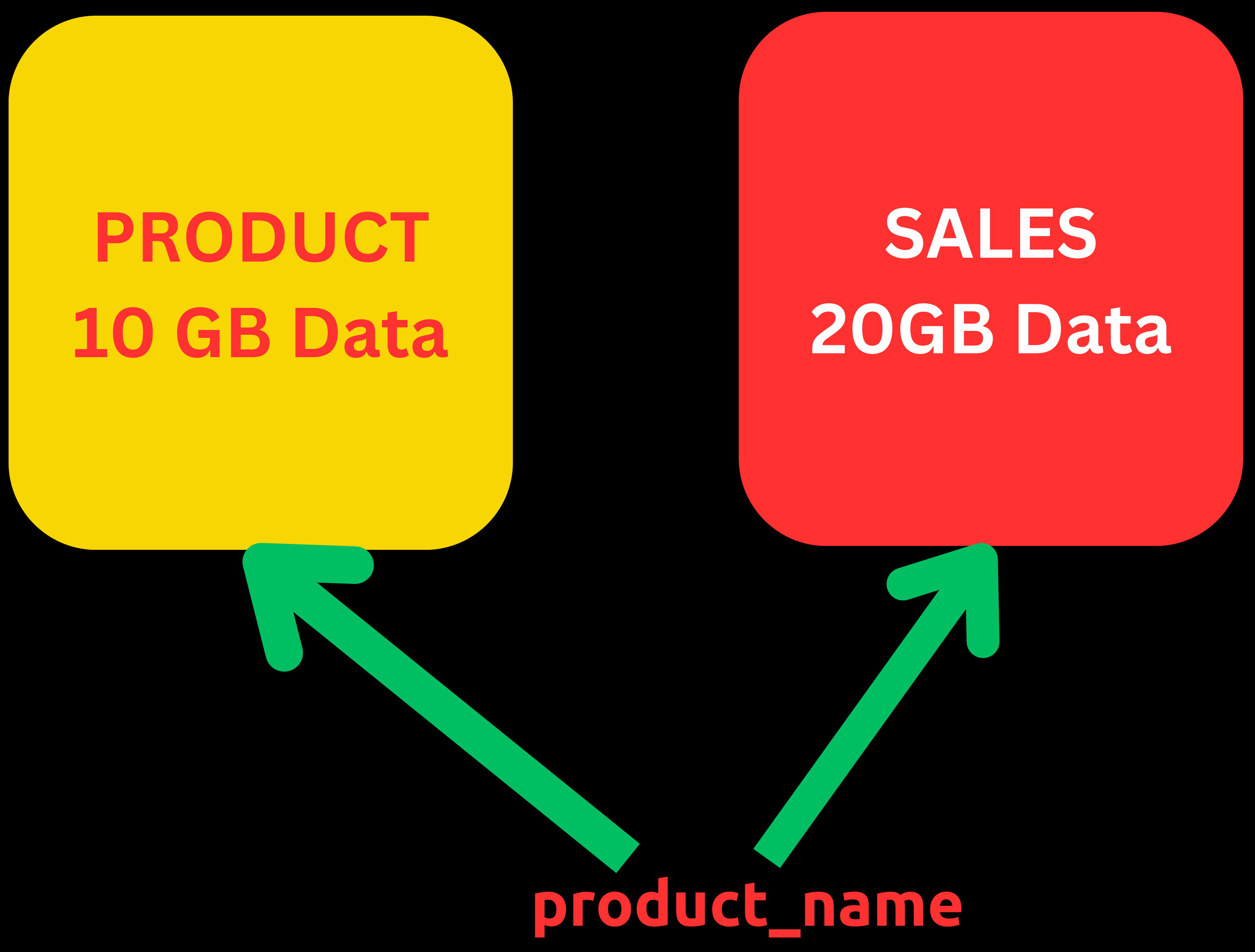
After this chnage AQE will change the join strategy from Sort-Merge to Broadcast as broadcasting of sales table is possible. This activity will happen at Runtime. Hence this is how AQE will dynamically switches join strategy.

Dynamically Switching skew join

To Understand this too we will consider same example

Product Table with Data 10 GB

Sales Table with Data 20 GB



PRODUCT
10 GB Data

SALES
20GB Data

product_name

Our goal is to join both tables based on product_name

In this case we want to get sales of mutiple products where product sugar comes around 80%. rest comes in 20%.

data is going to shuffled from both tables

After Shuffling of partitions from both tables

SUGAR with 80% of Data

Tea with 5% Data

Coffee with 5% Data

Milk with 5 % data

Water with 5 % data

In above situation Sugar will have 80% of data and it is going to take highest processing time.

Whenever we see such condition where a certain partition hold a disproportionately large amount of data compared to others. here we say Data is skewed or Data Skew Problem

How to Solve this ?

We have to split the parttions of Sugar to neutralize with other partitions.

SUGAR 20% Splitted 1

SUGAR 20% Splitted 2

SUGAR 20% Splitted 3

SUGAR 20% Splitted 4

Tea with 5% Data

Coffee with 5% Data

Milk with 5 % data

Water with 5 % data

After splitting it seems this problem is almost optimized using AQE at runtime.

Small Request-

If you find my content helpful like, comment and reshare with others
it will surely help other and mine too.

Vishal Gupta
Data Engineer
vishal.sss202@gmail.com

Vishal Gupta