```
#check that java is installed
!java -version
```

```
openjdk version "11.0.27" 2025-04-15
OpenJDK Runtime Environment (build 11.0.27+6-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.27+6-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

```
#install pyspark
!pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.11/dist-packages (3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.11/dist-packages (from pyspark) (0.10.9.7)
```

```
import os
import pandas as pd
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType, IntegerType, DoubleType, LongType, TimestampType
from pyspark.sql.functions import spark_partition_id
```

```
!curl -O https://raw.githubusercontent.com/Apoorva-888/Spark-Optimization/main/BigMart_Sales.csv
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  849k  100  849k    0     0  4829k      0 --:--:-- --:--:-- --:--:-- 4824k
```

```
spark = SparkSession.builder.appName('AutomotiveData').getOrCreate()

print(f'The Spark version is {spark.version}')
```

```
The Spark version is 3.5.1
```

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

```
'false'
```

```
spark.conf.set("spark.sql.optimizer.dynamicPartitionPruning.enabled","false")
spark.conf.get("spark.sql.optimizer.dynamicPartitionPruning.enabled")
```

```
'false'
```

```
spark.conf.set("spark.sql.autoBroadcastJoinThreshold",-1)
spark.conf.get("spark.sql.autoBroadcastJoinThreshold")
```

```
'-1'
```

```
df = spark.read.format("csv") \
    .option("inferSchema", True) \
    .option("header", True) \
    .load("BigMart_Sales.csv")

df.show(5)
```

```
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
|Item_Identifier|Item_Weight|Item_Fat_Content|Item_Visibility|           Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
|          FDA15|        9.3|         Low Fat|    0.016047301|               Dairy|249.8092|           OUT049|                     1999|
|          DRC01|       5.92|         Regular|    0.019278216|         Soft Drinks| 48.2692|           OUT018|                     2009|
|          FDN15|       17.5|         Low Fat|    0.016760075|                Meat| 141.618|           OUT049|                     1999|
|          FDX07|       19.2|         Regular|            0.0|Fruits and Vegeta...| 182.095|           OUT010|                     1998|
|          NCD19|       8.93|         Low Fat|            0.0|           Household| 53.8614|           OUT013|                     1987|
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
only showing top 5 rows
```

```
df.printSchema()
```

```
root
 |-- Item_Identifier: string (nullable = true)
 |-- Item_Weight: double (nullable = true)
 |-- Item_Fat_Content: string (nullable = true)
 |-- Item_Visibility: double (nullable = true)
 |-- Item_Type: string (nullable = true)
 |-- Item_MRP: double (nullable = true)
 |-- Outlet_Identifier: string (nullable = true)
 |-- Outlet_Establishment_Year: integer (nullable = true)
 |-- Outlet_Size: string (nullable = true)
 |-- Outlet_Location_Type: string (nullable = true)
 |-- Outlet_Type: string (nullable = true)
 |-- Item_Outlet_Sales: double (nullable = true)
```

```
!ls -lh BigMart_Sales.csv
```

```
-rw-r--r-- 1 root root 850K Jul 21 13:29 BigMart_Sales.csv
```

```
spark.conf.get("spark.sql.files.maxPartitionBytes")
```

```
'134217728b'
```

```
 df.rdd.getNumPartitions()
```

```
1
```

Peparing partitioned data

```
df_dpp_partitioned = df.write.format("parquet") \
    .partitionBy("Outlet_Type") \
    .option("path", "dpp_partitioned_BigMart_Sales") \
    .save()
```

```
df_dpp_partitioned = spark.read.parquet("dpp_partitioned_BigMart_Sales")
df_dpp_partitioned.show(5)
```

```
+---------------+-----------+----------------+---------------+------------+--------+-----------------+-------------------------+--------
|Item_Identifier|Item_Weight|Item_Fat_Content|Item_Visibility|   Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|Outlet_S
+---------------+-----------+----------------+---------------+------------+--------+-----------------+-------------------------+--------
|          FDA15|        9.3|         Low Fat|    0.016047301|       Dairy|249.8092|           OUT049|                     1999|     Med
|          FDN15|       17.5|         Low Fat|    0.016760075|        Meat| 141.618|           OUT049|                     1999|     Med
|          NCD19|       8.93|         Low Fat|            0.0|   Household| 53.8614|           OUT013|                     1987|       H
|          FDO10|      13.65|         Regular|    0.012741089| Snack Foods| 57.6588|           OUT013|                     1987|       H
|          FDH17|       16.2|         Regular|    0.016687114|Frozen Foods| 96.9726|           OUT045|                     2002|       N
+---------------+-----------+----------------+---------------+------------+--------+-----------------+-------------------------+--------
only showing top 5 rows
```

Peparing Non partitioned data

```
df_dpp_non_partitioned= df.write.format("parquet") \
    .mode("append")\
    .option("path","/content/dpp_non_partitioned_BigMart_Sales")\
    .save()
```

```
df_dpp_non_partitioned = spark.read.parquet("dpp_non_partitioned_BigMart_Sales")
df_dpp_non_partitioned.show(5)
```

```
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
|Item_Identifier|Item_Weight|Item_Fat_Content|Item_Visibility|           Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
|          FDA15|        9.3|         Low Fat|    0.016047301|               Dairy|249.8092|           OUT049|                     1999|
|          DRC01|       5.92|         Regular|    0.019278216|         Soft Drinks| 48.2692|           OUT018|                     2009|
|          FDN15|       17.5|         Low Fat|    0.016760075|                Meat| 141.618|           OUT049|                     1999|
|          FDX07|       19.2|         Regular|            0.0|Fruits and Vegeta...| 182.095|           OUT010|                     1998|
|          NCD19|       8.93|         Low Fat|            0.0|           Household| 53.8614|           OUT013|                     1987|
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
only showing top 5 rows
```

```
import os
os.listdir("/content")
!find /content -name "*BigMart*"
```

```
/content/BigMart_Sales.csv
/content/dpp_non_partitioned_BigMart_Sales
/content/dpp_partitioned_BigMart_Sales
```

```
#!rm -r dpp_partitioned_BigMart_Sales
```

```
!ls -R /content
```

```
/content:
BigMart_Sales.csv                  dpp_partitioned_BigMart_Sales
dpp_non_partitioned_BigMart_Sales  sample_data

/content/dpp_non_partitioned_BigMart_Sales:
part-00000-aa477b3a-cccc-4bc7-87ca-9f4d0e40c424-c000.snappy.parquet  _SUCCESS

/content/dpp_partitioned_BigMart_Sales:
'Outlet_Type=Grocery Store'      'Outlet_Type=Supermarket Type2'   _SUCCESS
'Outlet_Type=Supermarket Type1'  'Outlet_Type=Supermarket Type3'

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Grocery Store':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Supermarket Type1':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Supermarket Type2':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Supermarket Type3':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

/content/sample_data:
anscombe.json                  mnist_test.csv
california_housing_test.csv   mnist_train_small.csv
california_housing_train.csv  README.md
```

Double-click (or enter) to edit

```
##Non-partitioned folder
/content/dpp_non_partitioned_BigMart_Sales/
Contains: 1 .parquet file

##Partitioned folder
/content/dpp_partitioned_BigMart_Sales/
Contains: 4 folders (based on Outlet_Type), each with 1 .parquet file
```

```
#  For partitioned folder (nested subfolders):

!find /content/dpp_partitioned_BigMart_Sales -name "*.parquet" | wc -l
```

```
4
```

```
#For non-partitioned folder (flat):

!ls /content/dpp_non_partitioned_BigMart_Sales/*.parquet | wc -l
```

```
1
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
Start coding or generate with AI.
```

```
from pyspark.sql.functions import *
df_joined = df_dpp_partitioned.join(df_dpp_non_partitioned.filter(col("Outlet_Type")=="Grocery Store"), on="Item_Identifier", how="inner")
```

```
df_joined.show()
```

```
+---------------+-----------+---------------+--------------+------------+--------+-----------------+-------------------------+--------
|Item_Identifier|Item_Weight|Item_Fat_Content|Item_Visibility|   Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|Outlet_S
+---------------+-----------+---------------+--------------+------------+--------+-----------------+-------------------------+--------
|          DRA24|      19.35|        Regular|    0.040154087| Soft Drinks|164.6868|           OUT017|                     2007|       N
|          DRA24|      19.35|        Regular|    0.040154087| Soft Drinks|164.6868|           OUT017|                     2007|       N
|          DRA24|      19.35|        Regular|    0.039920687| Soft Drinks|163.3868|           OUT035|                     2004|      Sm
|          DRA24|      19.35|        Regular|    0.039920687| Soft Drinks|163.3868|           OUT035|                     2004|      Sm
|          DRA24|      19.35|        Regular|    0.039990314| Soft Drinks|165.0868|           OUT049|                     1999|     Med
|          DRA24|      19.35|        Regular|    0.039990314| Soft Drinks|165.0868|           OUT049|                     1999|     Med
|          DRA24|      19.35|        Regular|    0.039895009| Soft Drinks|162.4868|           OUT013|                     1987|       H
|          DRA24|      19.35|        Regular|    0.039895009| Soft Drinks|162.4868|           OUT013|                     1987|       H
|          DRA24|       NULL|        Regular|    0.069909188| Soft Drinks|163.2868|           OUT019|                     1985|      Sm
|          DRA24|       NULL|        Regular|    0.069909188| Soft Drinks|163.2868|           OUT019|                     1985|      Sm
|          DRA24|      19.35|        Regular|    0.066831682| Soft Drinks|163.8868|           OUT010|                     1998|       N
|          DRA24|      19.35|        Regular|    0.066831682| Soft Drinks|163.8868|           OUT010|                     1998|       N
|          DRA24|       NULL|        Regular|    0.039734882| Soft Drinks|165.7868|           OUT027|                     1985|     Med
|          DRA24|       NULL|        Regular|    0.039734882| Soft Drinks|165.7868|           OUT027|                     1985|     Med
|          FDO11|        8.0|        Regular|    0.030311951|      Breads|247.4092|           OUT049|                     1999|     Med
|          FDO11|        8.0|        Regular|    0.030264897|      Breads|250.3092|           OUT046|                     1997|      Sm
|          FDO11|        8.0|        Regular|    0.050657232|      Breads|249.9092|           OUT010|                     1998|       N
|          FDO11|       NULL|        Regular|    0.030118338|      Breads|248.8092|           OUT027|                     1985|     Med
|          FDU24|       6.78|        Regular|    0.140955857|Baking Goods|  92.212|           OUT017|                     2007|       N
|          FDU24|       6.78|        Regular|            0.0|Baking Goods|  94.012|           OUT013|                     1987|       H
+---------------+-----------+---------------+--------------+------------+--------+-----------------+-------------------------+--------
only showing top 20 rows
```

```
#Show joined result and files read

print("Files read during join:", len(df_joined.inputFiles()))
```

```
Files read during join: 5
```

```
df_joined.explain(mode="formatted")
```

```
(1) Scan parquet
Output [12]: [Item_Identifier#1041, Item_Weight#1042, Item_Fat_Content#1043, Item_Visibility#1044, Item_Type#1045, Item_MRP#1046, Outl
Batched: true
Location: InMemoryFileIndex [file:/content/dpp_partitioned_BigMart_Sales]
PushedFilters: [IsNotNull(Item_Identifier)]
ReadSchema: struct<Item_Identifier:string,Item_Weight:double,Item_Fat_Content:string,Item_Visibility:double,Item_Type:string,Item_MRP:

(2) ColumnarToRow [codegen id : 1]
Input [12]: [Item_Identifier#1041, Item_Weight#1042, Item_Fat_Content#1043, Item_Visibility#1044, Item_Type#1045, Item_MRP#1046, Outle

(3) Filter [codegen id : 1]
Input [12]: [Item_Identifier#1041, Item_Weight#1042, Item_Fat_Content#1043, Item_Visibility#1044, Item_Type#1045, Item_MRP#1046, Outle
Condition : isnotnull(Item_Identifier#1041)

(4) Exchange
Input [12]: [Item_Identifier#1041, Item_Weight#1042, Item_Fat_Content#1043, Item_Visibility#1044, Item_Type#1045, Item_MRP#1046, Outle
Arguments: hashpartitioning(Item_Identifier#1041, 200), ENSURE_REQUIREMENTS, [plan_id=564]

(5) Sort [codegen id : 2]
Input [12]: [Item_Identifier#1041, Item_Weight#1042, Item_Fat_Content#1043, Item_Visibility#1044, Item_Type#1045, Item_MRP#1046, Outle
Arguments: [Item_Identifier#1041 ASC NULLS FIRST], false, 0
```

```
(9) Exchange
Input [12]: [Item_Identifier#942, Item_Weight#943, Item_Fat_Content#944, Item_Visibility#945, Item_Type#946, Item_MRP#947, Outlet_Iden
Arguments: hashpartitioning(Item_Identifier#942, 200), ENSURE_REQUIREMENTS, [plan_id=573]

(10) Sort [codegen id : 4]
Input [12]: [Item_Identifier#942, Item_Weight#943, Item_Fat_Content#944, Item_Visibility#945, Item_Type#946, Item_MRP#947, Outlet_Iden
Arguments: [Item_Identifier#942 ASC NULLS FIRST], false, 0

(11) SortMergeJoin [codegen id : 5]
Left keys [1]: [Item_Identifier#1041]
Right keys [1]: [Item_Identifier#942]
Join type: Inner
Join condition: None

(12) Project [codegen id : 5]
Output [23]: [Item_Identifier#1041, Item_Weight#1042, Item_Fat_Content#1043, Item_Visibility#1044, Item_Type#1045, Item_MRP#1046, Outl
Input [24]: [Item_Identifier#1041, Item_Weight#1042, Item_Fat_Content#1043, Item_Visibility#1044, Item_Type#1045, Item_MRP#1046, Outle
```

```
#Show joined result and files read

print("Files read during join:", len(df2_joined.inputFiles()))
```

```
Files read during join: 5
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

DDP concept

```
# Turn off Adaptive Query Execution
spark.conf.set("spark.sql.adaptive.enabled", "false")
print("Adaptive Query Execution enabled:", spark.conf.get("spark.sql.adaptive.enabled"))

# Enable Dynamic Partition Pruning
spark.conf.set("spark.sql.optimizer.dynamicPartitionPruning.enabled", "true")
print("Dynamic Partition Pruning enabled:", spark.conf.get("spark.sql.optimizer.dynamicPartitionPruning.enabled"))

# Disable Auto Broadcast Join
spark.conf.set("spark.sql.autoBroadcastJoinThreshold", -1)
print("Auto Broadcast Join Threshold:", spark.conf.get("spark.sql.autoBroadcastJoinThreshold"))
```

```
Adaptive Query Execution enabled: false
Dynamic Partition Pruning enabled: true
Auto Broadcast Join Threshold: -1
```

```
df2 = spark.read.format("csv").option("inferSchema", True).option("header", True).load("BigMart_Sales.csv")

df2_dpp_partitioned = df2.write.format("parquet").partitionBy("Item_Identifier").option("path", "dpp2_partitioned_BigMart_Sales").save()
df2_dpp_partitioned = spark.read.parquet("dpp2_partitioned_BigMart_Sales")
df2_dpp_partitioned.show(2)

df2_dpp_non_partitioned= df2.write.format("parquet").mode("append").option("path","/content/dpp2_non_partitioned_BigMart_Sales").save()
df2_dpp_non_partitioned = spark.read.parquet("dpp2_non_partitioned_BigMart_Sales")
df2_dpp_non_partitioned.show(5)
```

```
+-----------+----------------+--------------+--------------------+--------+-----------------+-------------------------+-----------+----
|Item_Weight|Item_Fat_Content|Item_Visibility|           Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|Outlet_Size|Outl
+-----------+----------------+--------------+--------------------+--------+-----------------+-------------------------+-----------+----
|       NULL|         Regular|   0.014753811|Fruits and Vegeta...|231.7958|           OUT027|                     1985|     Medium|
|      20.35|         Regular|           0.0|Fruits and Vegeta...|234.4958|           OUT045|                     2002|       NULL|
+-----------+---------------+--------------+-------------------+--------+----------------+------------------------+-----------+----
only showing top 2 rows

+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
|Item_Identifier|Item_Weight|Item_Fat_Content|Item_Visibility|           Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
|          FDA15|        9.3|         Low Fat|    0.016047301|               Dairy|249.8092|           OUT049|                     1999|
|          DRC01|       5.92|         Regular|    0.019278216|         Soft Drinks| 48.2692|           OUT018|                     2009|
|          FDN15|       17.5|         Low Fat|    0.016760075|                Meat| 141.618|           OUT049|                     1999|
|          FDX07|       19.2|         Regular|            0.0|Fruits and Vegeta...| 182.095|           OUT010|                     1998|
|          NCD19|       8.93|         Low Fat|            0.0|           Household| 53.8614|           OUT013|                     1987|
+---------------+-----------+----------------+---------------+--------------------+--------+-----------------+-------------------------+
only showing top 5 rows
```

```
!rm -r dpp_non_partitioned_BigMart_Sales
```

```
import os
os.listdir("/content")
!find /content -name "*dpp2_*"
!rm -r dpp2_partitioned_BigMart_Sales
!ls -R /content
```

```
/content/dpp2_non_partitioned_BigMart_Sales
rm: cannot remove 'dpp2_partitioned_BigMart_Sales': No such file or directory
/content:
BigMart_Sales.csv                  dpp_partitioned_BigMart_Sales
dpp2_non_partitioned_BigMart_Sales  sample_data
dpp_non_partitioned_BigMart_Sales

/content/dpp2_non_partitioned_BigMart_Sales:
part-00000-5ea3b8e9-5090-4f50-82e4-162b6c3c7a1a-c000.snappy.parquet  _SUCCESS

/content/dpp_non_partitioned_BigMart_Sales:
part-00000-aa477b3a-cccc-4bc7-87ca-9f4d0e40c424-c000.snappy.parquet  _SUCCESS

/content/dpp_partitioned_BigMart_Sales:
'Outlet_Type=Grocery Store'      'Outlet_Type=Supermarket Type2'   _SUCCESS
'Outlet_Type=Supermarket Type1'  'Outlet_Type=Supermarket Type3'

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Grocery Store':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Supermarket Type1':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Supermarket Type2':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

'/content/dpp_partitioned_BigMart_Sales/Outlet_Type=Supermarket Type3':
part-00000-f41d5f25-07eb-4d91-9f01-8115194828b4.c000.snappy.parquet

/content/sample_data:
anscombe.json                 mnist_test.csv
california_housing_test.csv   mnist_train_small.csv
california_housing_train.csv  README.md
```

```
Start coding or generate with AI.
```

```
df2_joined = df2_dpp_partitioned.join(df2_dpp_non_partitioned.filter(col("Outlet_Type")=="Grocery Store"), on="Item_Identifier", how="inner"
df2_joined.show()
```

```
+---------------+-----------+----------------+---------------+-----------+--------+-----------------+-------------------------+--------
|Item_Identifier|Item_Weight|Item_Fat_Content|Item_Visibility|  Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|Outlet_S
+---------------+-----------+----------------+---------------+-----------+--------+-----------------+-------------------------+--------
|          DRA24|      19.35|         Regular|    0.040154087|Soft Drinks|164.6868|           OUT017|                     2007|       N
|          DRA24|      19.35|         Regular|    0.040154087|Soft Drinks|164.6868|           OUT017|                     2007|       N
|          DRA24|      19.35|         Regular|    0.039920687|Soft Drinks|163.3868|           OUT035|                     2004|      Sm
|          DRA24|      19.35|         Regular|    0.039920687|Soft Drinks|163.3868|           OUT035|                     2004|      Sm
|          DRA24|      19.35|         Regular|    0.039990314|Soft Drinks|165.0868|           OUT049|                     1999|     Med
|          DRA24|      19.35|         Regular|    0.039990314|Soft Drinks|165.0868|           OUT049|                     1999|     Med
|          DRA24|      19.35|         Regular|    0.039895009|Soft Drinks|162.4868|           OUT013|                     1987|       H
|          DRA24|      19.35|         Regular|    0.039895009|Soft Drinks|162.4868|           OUT013|                     1987|       H
|          DRA24|       NULL|         Regular|    0.069909188|Soft Drinks|163.2868|           OUT019|                     1985|      Sm
|          DRA24|       NULL|         Regular|    0.069909188|Soft Drinks|163.2868|           OUT019|                     1985|      Sm
|          DRA24|      19.35|         Regular|    0.066831682|Soft Drinks|163.8868|           OUT010|                     1998|       N
|          DRA24|      19.35|         Regular|    0.066831682|Soft Drinks|163.8868|           OUT010|                     1998|       N
|          DRA24|       NULL|         Regular|    0.039734882|Soft Drinks|165.7868|           OUT027|                     1985|     Med
|          DRA24|       NULL|         Regular|    0.039734882| Soft Drinks|165.7868|           OUT027|                    1985|     Med
|          FDO11|        8.0|         Regular|    0.030311951|      Breads|247.4092|           OUT049|                    1999|     Med
|          FDO11|        8.0|         Regular|    0.030264897|      Breads|250.3092|           OUT046|                    1997|      Sm
|          FDO11|        8.0|         Regular|    0.050657232|      Breads|249.9092|           OUT010|                    1998|       N
|          FDO11|       NULL|         Regular|    0.030118338|      Breads|248.8092|           OUT027|                    1985|     Med
|          FDU24|       6.78|         Regular|    0.140955857|Baking Goods|  92.212|           OUT017|                    2007|       N
|          FDU24|       6.78|         Regular|            0.0|Baking Goods|  94.012|           OUT013|                    1987|       H
+---------------+-----------+----------------+---------------+------------+--------+-----------------+------------------------+--------
only showing top 20 rows
```

```
df2_joined.explain(mode="formatted")
```

```
(1) Scan parquet
Output [12]: [Item_Identifier#1366, Item_Weight#1367, Item_Fat_Content#1368, Item_Visibility#1369, Item_Type#1370, Item_MRP#1371, Outl
Batched: true
Location: InMemoryFileIndex [file:/content/dpp2_partitioned_BigMart_Sales]
PushedFilters: [IsNotNull(Item_Identifier)]
ReadSchema: struct<Item_Identifier:string,Item_Weight:double,Item_Fat_Content:string,Item_Visibility:double,Item_Type:string,Item_MRP:

(2) ColumnarToRow [codegen id : 1]
Input [12]: [Item_Identifier#1366, Item_Weight#1367, Item_Fat_Content#1368, Item_Visibility#1369, Item_Type#1370, Item_MRP#1371, Outle

(3) Filter [codegen id : 1]
Input [12]: [Item_Identifier#1366, Item_Weight#1367, Item_Fat_Content#1368, Item_Visibility#1369, Item_Type#1370, Item_MRP#1371, Outle
Condition : isnotnull(Item_Identifier#1366)

(4) Exchange
Input [12]: [Item_Identifier#1366, Item_Weight#1367, Item_Fat_Content#1368, Item_Visibility#1369, Item_Type#1370, Item_MRP#1371, Outle
Arguments: hashpartitioning(Item_Identifier#1366, 200), ENSURE_REQUIREMENTS, [plan_id=881]

(5) Sort [codegen id : 2]
Input [12]: [Item_Identifier#1366, Item_Weight#1367, Item_Fat_Content#1368, Item_Visibility#1369, Item_Type#1370, Item_MRP#1371, Outle
Arguments: [Item_Identifier#1366 ASC NULLS FIRST], false, 0

(6) Scan parquet
Output [12]: [Item_Identifier#1464, Item_Weight#1465, Item_Fat_Content#1466, Item_Visibility#1467, Item_Type#1468, Item_MRP#1469, Outl
Batched: true
Location: InMemoryFileIndex [file:/content/dpp2_non_partitioned_BigMart_Sales]
PushedFilters: [IsNotNull(Outlet_Type), EqualTo(Outlet_Type,Grocery Store), IsNotNull(Item_Identifier)]
ReadSchema: struct<Item_Identifier:string,Item_Weight:double,Item_Fat_Content:string,Item_Visibility:double,Item_Type:string,Item_MRP:

(7) ColumnarToRow [codegen id : 3]
Input [12]: [Item_Identifier#1464, Item_Weight#1465, Item_Fat_Content#1466, Item_Visibility#1467, Item_Type#1468, Item_MRP#1469, Outle

(8) Filter [codegen id : 3]
Input [12]: [Item_Identifier#1464, Item_Weight#1465, Item_Fat_Content#1466, Item_Visibility#1467, Item_Type#1468, Item_MRP#1469, Outle
Condition : ((isnotnull(Outlet_Type#1474) AND (Outlet_Type#1474 = Grocery Store)) AND isnotnull(Item_Identifier#1464))

(9) Exchange
Input [12]: [Item_Identifier#1464, Item_Weight#1465, Item_Fat_Content#1466, Item_Visibility#1467, Item_Type#1468, Item_MRP#1469, Outle
Arguments: hashpartitioning(Item_Identifier#1464, 200), ENSURE_REQUIREMENTS, [plan_id=890]

(10) Sort [codegen id : 4]
Input [12]: [Item_Identifier#1464, Item_Weight#1465, Item_Fat_Content#1466, Item_Visibility#1467, Item_Type#1468, Item_MRP#1469, Outle
Arguments: [Item_Identifier#1464 ASC NULLS FIRST], false, 0

(11) SortMergeJoin [codegen id : 5]
Left keys [1]: [Item_Identifier#1366]
Right keys [1]: [Item_Identifier#1464]
Join type: Inner
Join condition: None

(12) Project [codegen id : 5]
Output [23]: [Item_Identifier#1366, Item_Weight#1367, Item_Fat_Content#1368, Item_Visibility#1369, Item_Type#1370, Item_MRP#1371, Outl
Input [24]: [Item_Identifier#1366, Item_Weight#1367, Item_Fat_Content#1368, Item_Visibility#1369, Item_Type#1370, Item_MRP#1371, Outle
```

Start coding or generate with AI.

With AQE

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

```
'true'
```

```
df = spark.read.format("csv") \
    .option("inferSchema", True) \
    .option("header", True) \
    .load("BigMart_Sales.csv")

df.show(2)
```

```
+---------------+-----------+----------------+---------------+-----------+--------+-----------------+-------------------------+---------
|Item_Identifier|Item_Weight|Item_Fat_Content|Item_Visibility|  Item_Type|Item_MRP|Outlet_Identifier|Outlet_Establishment_Year|Outlet_Si
+---------------+-----------+----------------+---------------+-----------+--------+-----------------+-------------------------+---------
|          FDA15|        9.3|         Low Fat|    0.016047301|      Dairy|249.8092|           OUT049|                     1999|     Medi
|          DRC01|       5.92|         Regular|    0.019278216|Soft Drinks| 48.2692|           OUT018|                     2009|     Medi
+---------------+-----------+----------------+---------------+-----------+--------+-----------------+-------------------------+---------
only showing top 2 rows
```

```
df.rdd.getNumPartitions()
```

```
1
```

```
df_new_with_AQE = df.groupBy("Item_Fat_Content").count()
df_new_with_AQE.show()
```

```
+----------------+-----+
|Item_Fat_Content|count|
+----------------+-----+
|         low fat|  112|
|         Low Fat| 5089|
|              LF|  316|
|         Regular| 2889|
|             reg|  117|
+----------------+-----+
```

```
df_new_with_AQE.explain(mode="formatted")
```

```
== Physical Plan ==
AdaptiveSparkPlan (5)
+- HashAggregate (4)
   +- Exchange (3)
      +- HashAggregate (2)
         +- Scan csv  (1)


(1) Scan csv
Output [1]: [Item_Fat_Content#289]
Batched: false
Location: InMemoryFileIndex [file:/content/BigMart_Sales.csv]
ReadSchema: struct<Item_Fat_Content:string>

(2) HashAggregate
Input [1]: [Item_Fat_Content#289]
Keys [1]: [Item_Fat_Content#289]
Functions [1]: [partial_count(1)]
Aggregate Attributes [1]: [count#435L]
Results [2]: [Item_Fat_Content#289, count#436L]

(3) Exchange
Input [2]: [Item_Fat_Content#289, count#436L]
Arguments: hashpartitioning(Item_Fat_Content#289, 200), ENSURE_REQUIREMENTS, [plan_id=262]

(4) HashAggregate
Input [2]: [Item_Fat_Content#289, count#436L]
Keys [1]: [Item_Fat_Content#289]
Functions [1]: [count(1)]
Aggregate Attributes [1]: [count(1)#425L]
Results [2]: [Item_Fat_Content#289, count(1)#425L AS count#426L]

(5) AdaptiveSparkPlan
Output [2]: [Item_Fat_Content#289, count#426L]
Arguments: isFinalPlan=false
```

```
Start coding or generate with AI.
```