1. Upload 2 datasets of your choice one big and one small(less than 10mb)
or
choose the existing sample datasets in the path /public/trendytech of the external lab

larger dataset -> /public/trendytech/orders/orders_1gb.csv
smaller dataset -> /public/trendytech/retail_db/customers

2. Create 2 Data Frames, one on each file and perform a join using Dataframes approach as well as spark SQL style. Do check the sparkUI to see the join strategy used

orders_schema = "order_id long , order_date string, customer_id long,order_status string"

orders_df = spark.read \
.format("csv") \
.schema(orders_schema) \
.load("/public/trendytech/orders/orders_1gb.csv")

customer_schema = """customer_id long , customer_fname string , customer_lname string , user_name string,password string ,
address string, city string, state string, pincode long """

customers_df = spark.read \
.format("csv") \
.schema(customer_schema) \
.load("/public/trendytech/retail_db/customers")

```
[2]: orders_schema = "order_id long , order_date string, customer_id long,order_status string"

[3]: orders_df = spark.read \
     .format("csv") \
     .schema(orders_schema) \
     .load("/public/trendytech/orders/orders_1gb.csv")

[4]: customer_schema = """customer_id long , customer_fname string , customer_lname string , user_name string,password string ,
                       address string, city string, state string, pincode long """

[5]: customers_df = spark.read \
     .format("csv") \
     .schema(customer_schema) \
     .load("/public/trendytech/retail_db/customers")
```

Using Dataframe approach:

orders_df.join(customers_df, orders_df.customer_id == customers_df.customer_id,"inner").write.format("noop").mode("overwrite").save()

```
[6]: orders_df.join(customers_df, orders_df.customer_id == customers_df.customer_id,"inner").write.format("noop").mode("overwrite").save()
```

Using SQL approach:

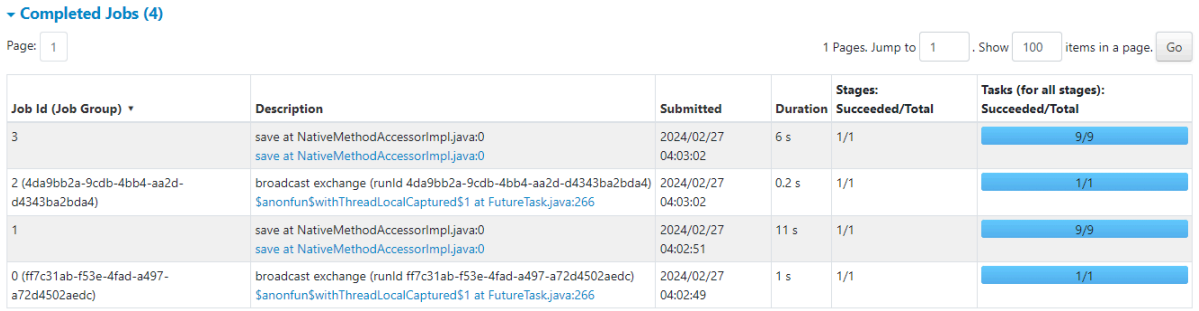
orders_df.createOrReplaceTempView("orders")

customers_df.createOrReplaceTempView("customers")

spark.sql("select * from orders inner join customers on (orders.customer_id == customers.customer_id)") \
.write.format("noop") \
.mode("overwrite").save()

```
[9]: spark.sql("select * from orders inner join customers on (orders.customer_id == customers.customer_id)") \
        .write.format("noop") \
        .mode("overwrite").save()
```

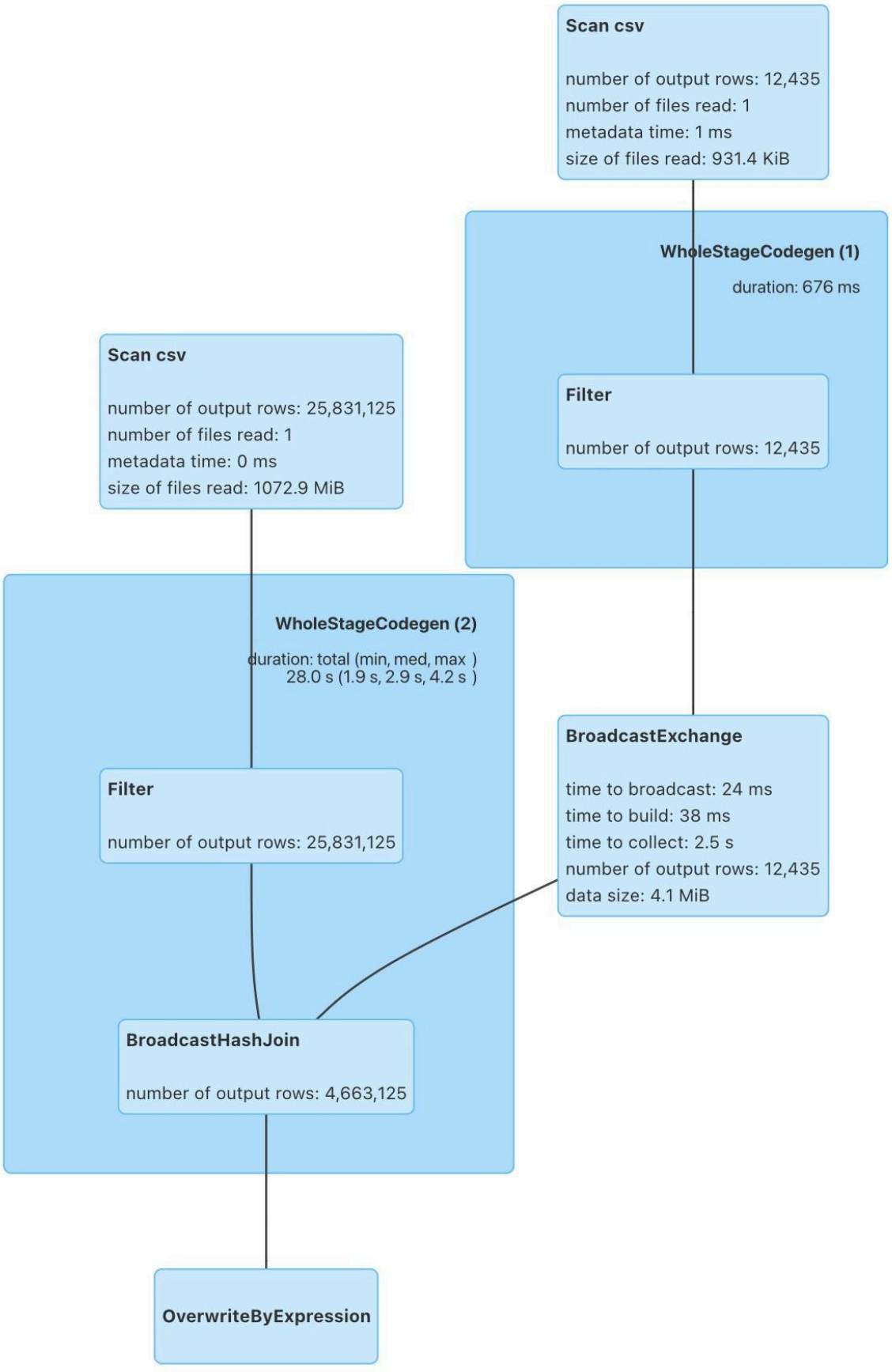
Note: There is no change with and without using SQL Style



▾ Completed Jobs (4)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id (Job Group) ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
3	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:03:02	6 s	1/1	9/9
2 (4da9bb2a-9cdb-4bb4-aa2d-d4343ba2bda4)	broadcast exchange (runId 4da9bb2a-9cdb-4bb4-aa2d-d4343ba2bda4) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2024/02/27 04:03:02	0.2 s	1/1	1/1
1	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:02:51	11 s	1/1	9/9
0 (ff7c31ab-f53e-4fad-a497-a72d4502aedc)	broadcast exchange (runId ff7c31ab-f53e-4fad-a497-a72d4502aedc) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2024/02/27 04:02:49	1 s	1/1	1/1


Scan csv
number of output rows: 12,435
number of files read: 1
metadata time: 1 ms
size of files read: 931.4 KiB
WholeStageCodegen (1)
duration: 676 ms
Filter
number of output rows: 12,435
Scan csv
number of output rows: 25,831,125
number of files read: 1
metadata time: 0 ms
size of files read: 1072.9 MiB
WholeStageCodegen (2)
duration: total (min, med, max)
28.0 s (1.9 s, 2.9 s, 4.2 s)
BroadcastExchange
time to broadcast: 24 ms
time to build: 38 ms
time to collect: 2.5 s
number of output rows: 12,435
data size: 4.1 MiB
Filter
number of output rows: 25,831,125
BroadcastHashJoin
number of output rows: 4,663,125
OverwriteByExpression

3. Disable the broadcast join by changing the threshold and perform a join again. Now check the Join strategy used.

spark.conf.get('spark.sql.autoBroadcastJoinThreshold')

spark.conf.set('spark.sql.autoBroadcastJoinThreshold',-1)

spark.conf.get('spark.sql.autoBroadcastJoinThreshold')

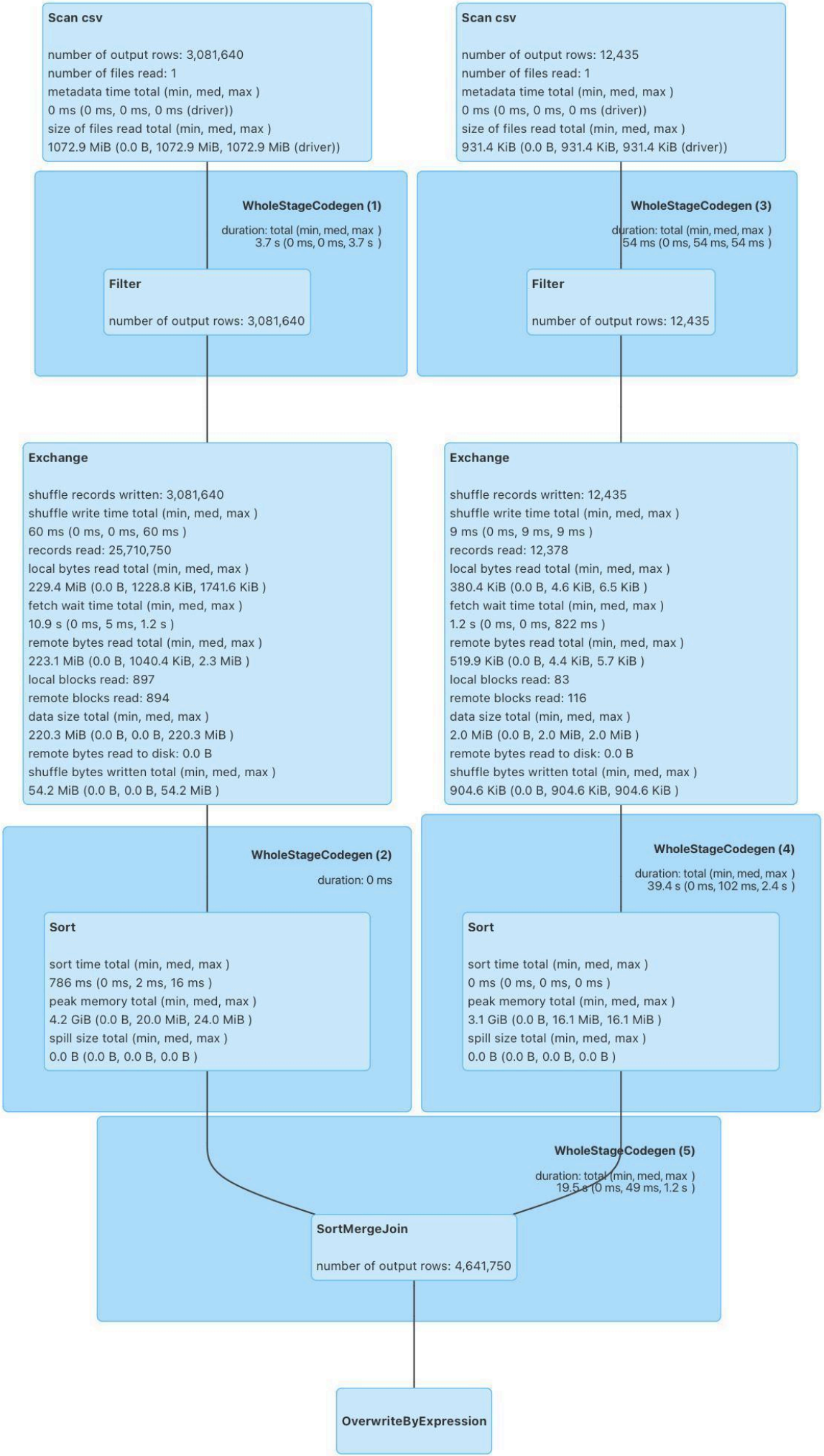
orders_df.join(customers_df, orders_df.customer_id == customers_df.customer_id,"inner").write.format("noop").mode("overwrite").save()

```
[10]: spark.conf.get('spark.sql.autoBroadcastJoinThreshold')
[10]: '10485760b'
[11]: spark.conf.set('spark.sql.autoBroadcastJoinThreshold',-1)
[12]: spark.conf.get('spark.sql.autoBroadcastJoinThreshold')
[12]: '-1'
[13]: orders_df.join(customers_df, orders_df.customer_id == customers_df.customer_id,"inner").write.format("noop").mode("overwrite").save()
```

▾ Completed Jobs (5)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:04:39	29 s	3/3 (3 failed)	210/210 (23 failed)
3	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:03:02	6 s	1/1	9/9
2 (4da9bb2a-9cdb-4bb4-aa2d-d4343ba2bda4)	broadcast exchange (runId 4da9bb2a-9cdb-4bb4-aa2d-d4343ba2bda4) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2024/02/27 04:03:02	0.2 s	1/1	1/1
1	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:02:51	11 s	1/1	9/9
0 (ff7c31ab-f53e-4fad-a497-a72d4502aedc)	broadcast exchange (runId ff7c31ab-f53e-4fad-a497-a72d4502aedc) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2024/02/27 04:02:49	1 s	1/1	1/1


Scan csv
number of output rows: 3,081,640
number of files read: 1
metadata time total (min, med, max)
0 ms (0 ms, 0 ms, 0 ms (driver))
size of files read total (min, med, max)
1072.9 MiB (0.0 B, 1072.9 MiB, 1072.9 MiB (driver))
Scan csv
number of output rows: 12,435
number of files read: 1
metadata time total (min, med, max)
0 ms (0 ms, 0 ms, 0 ms (driver))
size of files read total (min, med, max)
931.4 KiB (0.0 B, 931.4 KiB, 931.4 KiB (driver))
WholeStageCodegen (1)
duration: total (min, med, max)
3.7 s (0 ms, 0 ms, 3.7 s)
Filter
number of output rows: 3,081,640
WholeStageCodegen (3)
duration: total (min, med, max)
54 ms (0 ms, 54 ms, 54 ms)
Filter
number of output rows: 12,435
Exchange
shuffle records written: 3,081,640
shuffle write time total (min, med, max)
60 ms (0 ms, 0 ms, 60 ms)
records read: 25,710,750
local bytes read total (min, med, max)
229.4 MiB (0.0 B, 1228.8 KiB, 1741.6 KiB)
fetch wait time total (min, med, max)
10.9 s (0 ms, 5 ms, 1.2 s)
remote bytes read total (min, med, max)
223.1 MiB (0.0 B, 1040.4 KiB, 2.3 MiB)
local blocks read: 897
remote blocks read: 894
data size total (min, med, max)
220.3 MiB (0.0 B, 0.0 B, 220.3 MiB)
remote bytes read to disk: 0.0 B
shuffle bytes written total (min, med, max)
54.2 MiB (0.0 B, 0.0 B, 54.2 MiB)
Exchange
shuffle records written: 12,435
shuffle write time total (min, med, max)
9 ms (0 ms, 9 ms, 9 ms)
records read: 12,378
local bytes read total (min, med, max)
380.4 KiB (0.0 B, 4.6 KiB, 6.5 KiB)
fetch wait time total (min, med, max)
1.2 s (0 ms, 0 ms, 822 ms)
remote bytes read total (min, med, max)
519.9 KiB (0.0 B, 4.4 KiB, 5.7 KiB)
local blocks read: 83
remote blocks read: 116
data size total (min, med, max)
2.0 MiB (0.0 B, 2.0 MiB, 2.0 MiB)
remote bytes read to disk: 0.0 B
shuffle bytes written total (min, med, max)
904.6 KiB (0.0 B, 904.6 KiB, 904.6 KiB)
WholeStageCodegen (2)
duration: 0 ms
Sort
sort time total (min, med, max)
786 ms (0 ms, 2 ms, 16 ms)
peak memory total (min, med, max)
4.2 GiB (0.0 B, 20.0 MiB, 24.0 MiB)
spill size total (min, med, max)
0.0 B (0.0 B, 0.0 B, 0.0 B)
WholeStageCodegen (4)
duration: total (min, med, max)
39.4 s (0 ms, 102 ms, 2.4 s)
Sort
sort time total (min, med, max)
0 ms (0 ms, 0 ms, 0 ms)
peak memory total (min, med, max)
3.1 GiB (0.0 B, 16.1 MiB, 16.1 MiB)
spill size total (min, med, max)
0.0 B (0.0 B, 0.0 B, 0.0 B)
WholeStageCodegen (5)
duration: total (min, med, max)
19.5 s (0 ms, 49 ms, 1.2 s)
SortMergeJoin
number of output rows: 4,641,750
OverwriteByExpression

4. Give a hint for shuffle hash join and invoke the join again and check the spark UI for the join strategy used.

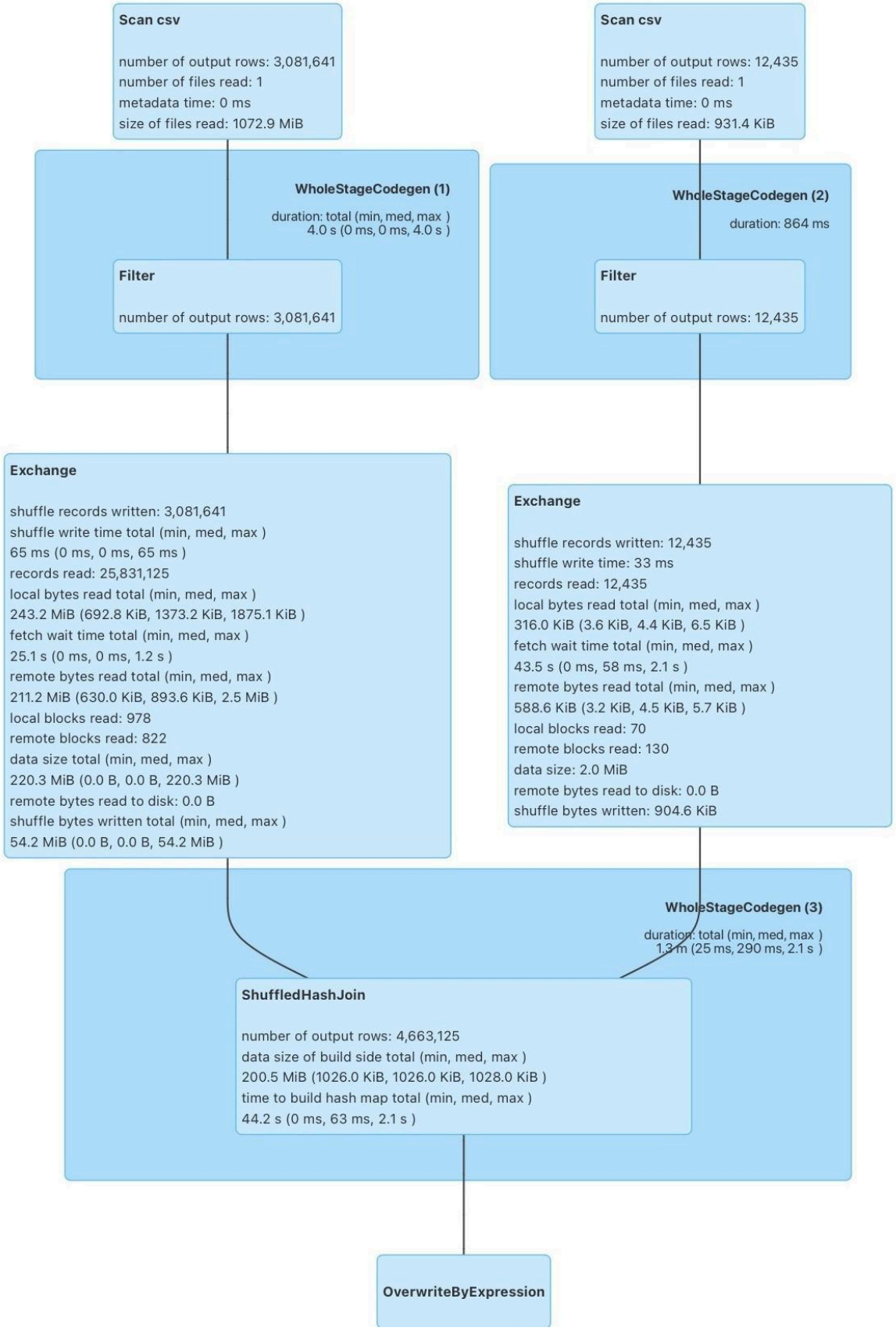
```
[14]: orders_df.join(customers_df.hint("shuffle_hash") , orders_df.customer_id == customers_df.customer_id , "inner").write.format("noop") \
.mode("overwrite").save()
```

orders_df.join(customers_df.hint("shuffle_hash") , orders_df.customer_id == customers_df.customer_id , "inner").write.format("noop") \
.mode("overwrite").save()

▾ Completed Jobs (6)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id (Job Group) ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:07:23	28 s	3/3 (2 failed)	210/210 (8 failed)
4	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:04:39	29 s	3/3 (3 failed)	210/210 (23 failed)
3	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:03:02	6 s	1/1	9/9
2 (4da9bb2a-9cdb-4bb4-aa2d-d4343ba2bda4)	broadcast exchange (runId 4da9bb2a-9cdb-4bb4-aa2d-d4343ba2bda4) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2024/02/27 04:03:02	0.2 s	1/1	1/1
1	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/27 04:02:51	11 s	1/1	9/9
0 (ff7c31ab-f53e-4fad-a497-a72d4502aedc)	broadcast exchange (runId ff7c31ab-f53e-4fad-a497-a72d4502aedc) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266	2024/02/27 04:02:49	1 s	1/1	1/1


Scan csv
number of output rows: 3,081,641
number of files read: 1
metadata time: 0 ms
size of files read: 1072.9 MiB
Scan csv
number of output rows: 12,435
number of files read: 1
metadata time: 0 ms
size of files read: 931.4 KiB
WholeStageCodegen (1)
duration: total (min, med, max)
4.0 s (0 ms, 0 ms, 4.0 s)
Filter
number of output rows: 3,081,641
WholeStageCodegen (2)
duration: 864 ms
Filter
number of output rows: 12,435
Exchange
shuffle records written: 3,081,641
shuffle write time total (min, med, max)
65 ms (0 ms, 0 ms, 65 ms)
records read: 25,831,125
local bytes read total (min, med, max)
243.2 MiB (692.8 KiB, 1373.2 KiB, 1875.1 KiB)
fetch wait time total (min, med, max)
25.1 s (0 ms, 0 ms, 1.2 s)
remote bytes read total (min, med, max)
211.2 MiB (630.0 KiB, 893.6 KiB, 2.5 MiB)
local blocks read: 978
remote blocks read: 822
data size total (min, med, max)
220.3 MiB (0.0 B, 0.0 B, 220.3 MiB)
remote bytes read to disk: 0.0 B
shuffle bytes written total (min, med, max)
54.2 MiB (0.0 B, 0.0 B, 54.2 MiB)
Exchange
shuffle records written: 12,435
shuffle write time: 33 ms
records read: 12,435
local bytes read total (min, med, max)
316.0 KiB (3.6 KiB, 4.4 KiB, 6.5 KiB)
fetch wait time total (min, med, max)
43.5 s (0 ms, 58 ms, 2.1 s)
remote bytes read total (min, med, max)
588.6 KiB (3.2 KiB, 4.5 KiB, 5.7 KiB)
local blocks read: 70
remote blocks read: 130
data size: 2.0 MiB
remote bytes read to disk: 0.0 B
shuffle bytes written: 904.6 KiB
WholeStageCodegen (3)
duration: total (min, med, max)
1.3 m (25 ms, 290 ms, 2.1 s)
ShuffledHashJoin
number of output rows: 4,663,125
data size of build side total (min, med, max)
200.5 MiB (1026.0 KiB, 1026.0 KiB, 1028.0 KiB)
time to build hash map total (min, med, max)
44.2 s (0 ms, 63 ms, 2.1 s)
OverwriteByExpression

5. By default the AQE was disabled, enable the AQE and perform a join again and see if there is a change in the number of shuffle partitions.

```
spark.conf.get("spark.sql.adaptive.enabled")

spark.conf.set("spark.sql.adaptive.enabled",True)

spark.conf.get("spark.sql.adaptive.enabled")

orders_schema = "order_id long , order_date string, customer_id long,order_status string"

orders_df = spark.read \
.format("csv") \
.schema(orders_schema) \
.load("/public/trendytech/orders/orders_1gb.csv")

customer_schema = """customer_id long , customer_fname string , customer_lname string ,
user_name string,password string ,
                address string, city string, state string, pincode long """

customers_df = spark.read \
.format("csv") \
.schema(customer_schema) \
.load("/public/trendytech/retail_db/customers")

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

orders_df.join(customers_df, orders_df.customer_id == customers_df.customer_id,"inner").write.format("noop").mode("overwrite").save()

orders_df.join(customers_df.hint("shuffle_hash") , orders_df.customer_id == customers_df.customer_id , "inner").write.format("noop") \
.mode("overwrite").save()

```
[10]: orders_df.join(customers_df, orders_df.customer_id == customers_df.customer_id,"inner").write.format("noop").mode("overwrite").save()

[11]: orders_df.join(customers_df.hint("shuffle_hash") , orders_df.customer_id == customers_df.customer_id , "inner").write.format("noop") \
      .mode("overwrite").save()
```

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/03/01 03:04:23	12 s	1/1 (2 skipped)	11/11 (11 skipped)
4	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/03/01 03:04:19	3 s	1/1	1/1
3	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/03/01 03:04:19	4 s	1/1	10/10
2	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/03/01 03:04:08	11 s	1/1 (2 skipped)	11/11 (10 skipped)
1	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/03/01 03:03:56	8 s	1/1	1/1
0	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/03/01 03:03:56	11 s	1/1	9/9

6. You need to explain left outer and semi join with relevant use cases. Demonstrate it by running in the notebook.

Ans:

```
employee_data = [(10, "Raj","1999","100","M",2000),
                 (20, "Rahul","2002","200","M",2000),
                 (30, "Raghav","2010","100","",2000),
                 (40, "Reema","2004","100","F",2000),
                 (50, "Rina","2008","400","F",2000),
                 (60, "Rasul","2014","500","M",2000)
                 ]
employee_schema = ["employee_id","name","doj","employee_dept_id","gender","salary"]

employeeDf = spark.createDataFrame(data=employee_data,schema=employee_schema)

employeeDf.show()
```

```
[2]: employee_data = [(10, "Raj","1999","100","M",2000),
                      (20, "Rahul","2002","200","M",2000),
                      (30, "Raghav","2010","100","",2000),
                      (40, "Reema","2004","100","F",2000),
                      (50, "Rina","2008","400","F",2000),
                      (60, "Rasul","2014","500","M",2000)
                      ]

[3]: employee_schema = ["employee_id","name","doj","employee_dept_id","gender","salary"]

[4]: employeeDf = spark.createDataFrame(data=employee_data,schema=employee_schema)

[5]: employeeDf.show()
```

```
+-----------+------+----+----------------+------+------+
|employee_id|  name| doj|employee_dept_id|gender|salary|
+-----------+------+----+----------------+------+------+
|         10|   Raj|1999|             100|     M|  2000|
|         20| Rahul|2002|             200|     M|  2000|
|         30|Raghav|2010|             100|      |  2000|
|         40| Reema|2004|             100|     F|  2000|
|         50|  Rina|2008|             400|     F|  2000|
|         60| Rasul|2014|             500|     M|  2000|
+-----------+------+----+----------------+------+------+
```

```
department_data = [("HR",100),
                   ("Supply",100),
                   ("Sales",100),
                   ("Stock",100),
                  ]

department_schema = ["dept_name","dept_id"]

departmentDf =
spark.createDataFrame(data=department_data,schema=department_schema)

departmentDf.show()
```

```
[6]: department_data = [("HR",100),
                       ("Supply",100),
                       ("Sales",100),
                       ("Stock",100),
                      ]

[7]: department_schema = ["dept_name","dept_id"]

[8]: departmentDf = spark.createDataFrame(data=department_data,schema=department_schema)

[9]: departmentDf.show()
```

```
+---------+-------+
|dept_name|dept_id|
+---------+-------+
|       HR|    100|
|   Supply|    100|
|    Sales|    100|
|    Stock|    100|
+---------+-------+
```

Left Outer Join:

A left outer join is a type of join operation where all rows from the left DataFrame (or table) are included in the result, along with matching rows from the right DataFrame (or table). If there is no match in the right DataFrame for a row in the left DataFrame, null values are used for the columns from the right DataFrame.

Dataframe Approach:

df_join = employeeDf.join(departmentDf,employeeDf.employee_dept_id == departmentDf.dept_id, "Left_Outer")

df_join.show()

```
[10]: df_join = employeeDf.join(departmentDf,employeeDf.employee_dept_id == departmentDf.dept_id, "Left_Outer")

[11]: df_join.show()
```

```
+-----------+------+----+----------------+------+------+---------+-------+
|employee_id|  name| doj|employee_dept_id|gender|salary|dept_name|dept_id|
+-----------+------+----+----------------+------+------+---------+-------+
|         60| Rasul|2014|             500|     M|  2000|     null|   null|
|         10|   Raj|1999|             100|     M|  2000|       HR|    100|
|         10|   Raj|1999|             100|     M|  2000|   Supply|    100|
|         10|   Raj|1999|             100|     M|  2000|    Sales|    100|
|         10|   Raj|1999|             100|     M|  2000|    Stock|    100|
|         30|Raghav|2010|             100|      |  2000|       HR|    100|
|         30|Raghav|2010|             100|      |  2000|   Supply|    100|
|         30|Raghav|2010|             100|      |  2000|    Sales|    100|
|         30|Raghav|2010|             100|      |  2000|    Stock|    100|
|         40| Reema|2004|             100|     F|  2000|       HR|    100|
|         40| Reema|2004|             100|     F|  2000|   Supply|    100|
|         40| Reema|2004|             100|     F|  2000|    Sales|    100|
|         40| Reema|2004|             100|     F|  2000|    Stock|    100|
|         20| Rahul|2002|             200|     M|  2000|     null|   null|
|         50|  Rina|2008|             400|     F|  2000|     null|   null|
+-----------+------+----+----------------+------+------+---------+-------+
```

Semi-join

Semi-join is a type of join where only the rows from the left DataFrame (or table) that have a match in the right DataFrame (or table) are returned. It does not include any columns from the right DataFrame in the result, only filtering the rows from the left DataFrame.

Dataframe Approach:

df_join = employeeDf.join(departmentDf,employeeDf.employee_dept_id == departmentDf.dept_id, "Semi")

df_join.show()

```
[12]: df_join = employeeDf.join(departmentDf,employeeDf.employee_dept_id == departmentDf.dept_id, "Semi")

[13]: df_join.show()
```

```
+-----------+------+----+----------------+------+------+
|employee_id|  name| doj|employee_dept_id|gender|salary|
+-----------+------+----+----------------+------+------+
|         10|   Raj|1999|             100|     M|  2000|
|         30|Raghav|2010|             100|      |  2000|
|         40| Reema|2004|             100|     F|  2000|
+-----------+------+----+----------------+------+------+
```

SQL Approach:

departmentDf.createOrReplaceTempView("department_table")

employeeDf.createOrReplaceTempView("employee_table")

res = spark.sql("""select * from employee_table Left Outer JOIN department_table ON \
employee_table.employee_dept_id == department_table.dept_id
""")

res = spark.sql("""select * from employee_table SEMI JOIN department_table ON \
employee_table.employee_dept_id == department_table.dept_id
""")

res.show()

```
[18]: departmentDf.createOrReplaceTempView("department_table")

[19]: employeeDf.createOrReplaceTempView("employee_table")

[20]: res = spark.sql("""select * from employee_table Left Outer JOIN department_table ON \
      employee_table.employee_dept_id == department_table.dept_id
      """)

[21]: res.show()
```

```
+-----------+------+----+----------------+------+------+---------+-------+
|employee_id|  name| doj|employee_dept_id|gender|salary|dept_name|dept_id|
+-----------+------+----+----------------+------+------+---------+-------+
|         60| Rasul|2014|             500|     M|  2000|     null|   null|
|         10|   Raj|1999|             100|     M|  2000|       HR|    100|
|         10|   Raj|1999|             100|     M|  2000|   Supply|    100|
|         10|   Raj|1999|             100|     M|  2000|    Sales|    100|
|         10|   Raj|1999|             100|     M|  2000|    Stock|    100|
|         30|Raghav|2010|             100|      |  2000|       HR|    100|
|         30|Raghav|2010|             100|      |  2000|   Supply|    100|
|         30|Raghav|2010|             100|      |  2000|    Sales|    100|
|         30|Raghav|2010|             100|      |  2000|    Stock|    100|
|         40| Reema|2004|             100|     F|  2000|       HR|    100|
|         40| Reema|2004|             100|     F|  2000|   Supply|    100|
|         40| Reema|2004|             100|     F|  2000|    Sales|    100|
|         40| Reema|2004|             100|     F|  2000|    Stock|    100|
|         20| Rahul|2002|             200|     M|  2000|     null|   null|
|         50|  Rina|2008|             400|     F|  2000|     null|   null|
+-----------+------+----+----------------+------+------+---------+-------+
```

```
result = spark.sql("""select * from employee_table SEMI JOIN department_table ON \
employee_table.employee_dept_id == department_table.dept_id
""")

result.show()
```

[22]:
```
result = spark.sql("""select * from employee_table SEMI JOIN department_table ON \
employee_table.employee_dept_id == department_table.dept_id
""")
```

[23]:
```
result.show()
```

```
+-----------+------+----+---------------+------+------+
|employee_id|  name| doj|employee_dept_id|gender|salary|
+-----------+------+----+---------------+------+------+
|         10|   Raj|1999|            100|     M|  2000|
|         30|Raghav|2010|            100|      |  2000|
|         40| Reema|2004|            100|     F|  2000|
+-----------+------+----+---------------+------+------+
```