

[Jobs \(/jobs/\)](#)
 [Stages \(/stages/\)](#)
 [Storage \(/storage/\)](#)
 [Environment \(/environment/\)](#)

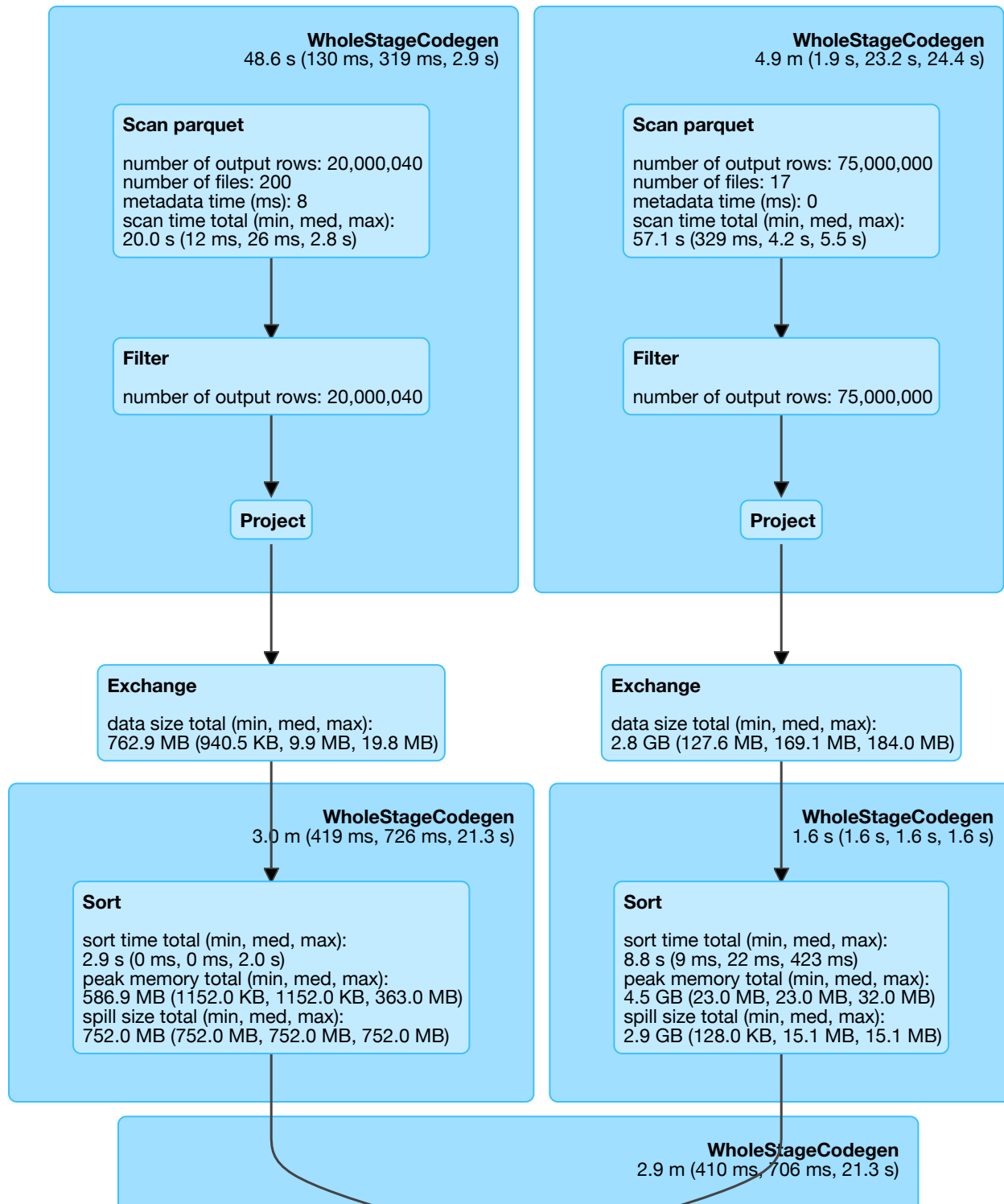
[Executors \(/executors/\)](#)
 [SQL \(/SQL/\)](#)

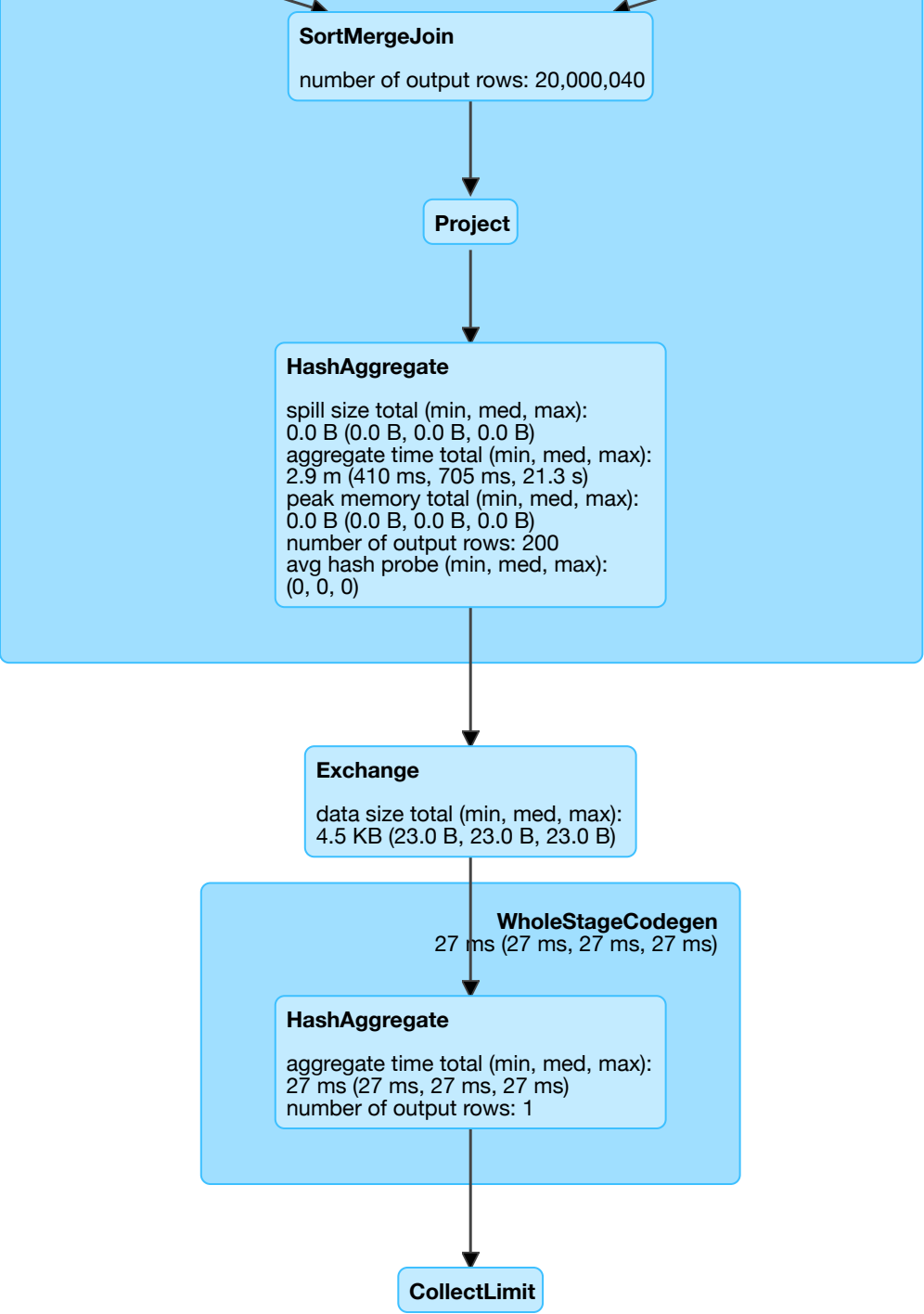
Details for Query 2

Submitted Time: 2020/05/06 21:16:52

Duration: 57 s

Succeeded Jobs: 3 (/jobs/job/?id=3)





▼ Details

```

== Parsed Logical Plan ==
GlobalLimit 21
+- LocalLimit 21
  +- Project [cast(avg_revenue#18 as string) AS avg_revenue#22]
    +- Aggregate [avg((cast(price#14 as double) * cast(num_pieces_sold#4 as double))) AS
avg_revenue#18]
      +- Join Inner, (product_id#12 = product_id#1)
        :- SubqueryAlias `o`
        : +- SubqueryAlias `orders`
        : +-
Relation[order_id#0,product_id#1,seller_id#2,date#3,num_pieces_sold#4,bill_raw_text#5] parquet
      +- SubqueryAlias `p`
        +- SubqueryAlias `products`
          +- Relation[product_id#12,product_name#13,price#14] parquet

== Analyzed Logical Plan ==
avg_revenue: string
GlobalLimit 21
+- LocalLimit 21
  +- Project [cast(avg_revenue#18 as string) AS avg_revenue#22]
    +- Aggregate [avg((cast(price#14 as double) * cast(num_pieces_sold#4 as double))) AS
avg_revenue#18]
      +- Join Inner, (product_id#12 = product_id#1)
        :- SubqueryAlias `o`
        : +- SubqueryAlias `orders`
        : +-
Relation[order_id#0,product_id#1,seller_id#2,date#3,num_pieces_sold#4,bill_raw_text#5] parquet
      +- SubqueryAlias `p`
        +- SubqueryAlias `products`
          +- Relation[product_id#12,product_name#13,price#14] parquet

== Optimized Logical Plan ==
GlobalLimit 21
+- LocalLimit 21
  +- Aggregate [cast(avg((cast(price#14 as double) * cast(num_pieces_sold#4 as double))) as string) AS
avg_revenue#22]
    +- Project [num_pieces_sold#4, price#14]
      +- Join Inner, (product_id#12 = product_id#1)
        :- Project [product_id#1, num_pieces_sold#4]
        : +- Filter isnotnull(product_id#1)
        : +-
Relation[order_id#0,product_id#1,seller_id#2,date#3,num_pieces_sold#4,bill_raw_text#5] parquet
      +- Project [product_id#12, price#14]
        +- Filter isnotnull(product_id#12)
          +- Relation[product_id#12,product_name#13,price#14] parquet

== Physical Plan ==
CollectLimit 21
+- *(6) HashAggregate(keys=[], functions=[avg((cast(price#14 as double) * cast(num_pieces_sold#4 as
double)))], output=[avg_revenue#22])
  +- Exchange SinglePartition
    +- *(5) HashAggregate(keys=[], functions=[partial_avg((cast(price#14 as double) *
cast(num_pieces_sold#4 as double)))], output=[sum#26, count#27L])
      +- *(5) Project [num_pieces_sold#4, price#14]
        +- *(5) SortMergeJoin [product_id#1], [product_id#12], Inner
          :- *(2) Sort [product_id#1 ASC NULLS FIRST], false, 0
          : +- Exchange hashpartitioning(product_id#1, 200)
          : +- *(1) Project [product_id#1, num_pieces_sold#4]

```

```
:      +- *(1) Filter isnotnull(product_id#1)
:      +- *(1) FileScan parquet [product_id#1,num_pieces_sold#4] Batched: true,
Format: Parquet, Location: InMemoryFileIndex[file:/Users/o60774/Downloads/product-
sales/sales_parquet/part-00102-e651a798-93..., PartitionFilters: [], PushedFilters:
[IsNotNull(product_id)], ReadSchema: struct<product_id:string,num_pieces_sold:string>
+- *(4) Sort [product_id#12 ASC NULLS FIRST], false, 0
+- Exchange hashpartitioning(product_id#12, 200)
+- *(3) Project [product_id#12, price#14]
+- *(3) Filter isnotnull(product_id#12)
+- *(3) FileScan parquet [product_id#12,price#14] Batched: true, Format:
Parquet, Location: InMemoryFileIndex[file:/Users/o60774/Downloads/product-sales/products_parquet/part-
00015-0f978a44..., PartitionFilters: [], PushedFilters: [IsNotNull(product_id)], ReadSchema:
struct<product_id:string,price:string>
```