Notebook:

- *Link*

Cluster:

- *N2-highmem-4, 2 - 8 autoscaling. 16.4 LTS*

Query Without Materialization:

Query:

```sql
WITH joined_files AS (
 SELECT
   f.repo_name,
   f.ref,
   f.path,
   f.mode,
   f.id,
   f.symlink_target,
   c.size,
   c.content,
   c.binary,
   c.copies
 FROM
   files f
   INNER JOIN contents c
     ON f.id = c.id
)
SELECT
 repo_name,
 path,
 size,
 RANK() OVER (PARTITION BY repo_name ORDER BY size DESC) AS size_rank,
 copies,
 binary
```

```
FROM
 joined_files
ORDER BY
 repo_name,
 size_rank;
```

Explain Plan:

```
== Physical Plan ==
AdaptiveSparkPlan (26)
+- == Initial Plan ==
   ColumnarToRow (25)
   +- PhotonResultStage (24)
      +- PhotonSort (23)
         +- PhotonShuffleExchangeSource (22)
            +- PhotonShuffleMapStage (21)
               +- PhotonShuffleExchangeSink (20)
                  +- PhotonWindow (19)
                     +- PhotonSort (18)
                        +- PhotonShuffleExchangeSource (17)
                           +- PhotonShuffleMapStage (16)
                              +- PhotonShuffleExchangeSink (15)
                                 +- PhotonProject (14)
                                    +- PhotonShuffledHashJoin Inner (13)
                                       :- PhotonShuffleExchangeSource (6)
                                       :  +- PhotonShuffleMapStage (5)
                                       :     +- PhotonShuffleExchangeSink
(4)
                                       :        +- PhotonFilter (3)
                                       :           +- PhotonRowToColumnar
(2)
                                       :              +- Scan
BigQueryV1RelationFromV2ScanWithNoMaterialization(org.apache.spark.sql.SQLC
ontext@934053c,StructType(StructField(repo_name,StringType,true),StructFiel
d(path,StringType,true),StructField(id,StringType,true)),BigQueryScanPushDo
wns(Some(StructType(StructField(repo_name,StringType,true),StructField(path
,StringType,true),StructField(id,StringType,true))),[Lorg.apache.spark.sql.
connector.expressions.filter.Predicate;@5e71c3dd,None,None,[Ljava.lang.Stri
ng;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Lor
g.apache.spark.sql.sources.Filter;@208d329b),DirectBigQueryRelation[fe-dev-
```

```
sandbox.dkushari_ds.files]) dkushari_bq_test.dkushari_ds.files (1)
                                    +- PhotonShuffleExchangeSource (12)
                                       +- PhotonShuffleMapStage (11)
                                          +- PhotonShuffleExchangeSink
(10)
                                             +- PhotonFilter (9)
                                                +- PhotonRowToColumnar
(8)
                                                   +- Scan
BigQueryV1RelationFromV2ScanWithNoMaterialization(org.apache.spark.sql.SQLC
ontext@934053c,StructType(StructField(id,StringType,true),StructField(size,
LongType,true),StructField(binary,BooleanType,true),StructField(copies,Long
Type,true)),BigQueryScanPushDowns(Some(StructType(StructField(id,StringType
,true),StructField(size,LongType,true),StructField(binary,BooleanType,true)
,StructField(copies,LongType,true))),[Lorg.apache.spark.sql.connector.expre
ssions.filter.Predicate;@1b872ece,None,None,[Ljava.lang.String;@7b8a780a,[L
java.lang.String;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Lorg.apache.spark.
sql.sources.Filter;@73415c2d),DirectBigQueryRelation[fe-dev-sandbox.dkushar
i_ds.contents]) dkushari_bq_test.dkushari_ds.contents (7)


(1) Scan
BigQueryV1RelationFromV2ScanWithNoMaterialization(org.apache.spark.sql.SQLC
ontext@934053c,StructType(StructField(repo_name,StringType,true),StructFiel
d(path,StringType,true),StructField(id,StringType,true)),BigQueryScanPushDo
wns(Some(StructType(StructField(repo_name,StringType,true),StructField(path
,StringType,true),StructField(id,StringType,true))),[Lorg.apache.spark.sql.
connector.expressions.filter.Predicate;@5e71c3dd,None,None,[Ljava.lang.Stri
ng;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Lor
g.apache.spark.sql.sources.Filter;@208d329b),DirectBigQueryRelation[fe-dev-
sandbox.dkushari_ds.files]) dkushari_bq_test.dkushari_ds.files
Output [3]: [repo_name#217, path#219, id#221]
Arguments: [repo_name#217, path#219, id#221],
[StructField(repo_name,StringType,true), StructField(path,StringType,true),
StructField(id,StringType,true)],
PushedDownOperators(None,None,None,None,List(),List()),
PreScala213BigQueryRDD[54] at RDD at PreScala213BigQueryRDD.java:40,
BigQueryV1RelationFromV2ScanWithNoMaterialization(org.apache.spark.sql.SQLC
ontext@934053c,StructType(StructField(repo_name,StringType,true),StructFiel
d(path,StringType,true),StructField(id,StringType,true)),BigQueryScanPushDo
wns(Some(StructType(StructField(repo_name,StringType,true),StructField(path
,StringType,true),StructField(id,StringType,true))),[Lorg.apache.spark.sql.
connector.expressions.filter.Predicate;@5e71c3dd,None,None,[Ljava.lang.Stri
```

```
ng;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Lor
g.apache.spark.sql.sources.Filter;@208d329b),DirectBigQueryRelation[fe-dev-
sandbox.dkushari_ds.files]), `dkushari_bq_test`.`dkushari_ds`.`files`,
Statistics(sizeInBytes=8.0 EiB, ColumnStat: N/A)

(2) PhotonRowToColumnar
Input [3]: [repo_name#217, path#219, id#221]

(3) PhotonFilter
Input [3]: [repo_name#217, path#219, id#221]
Arguments: isnotnull(id#221)

(4) PhotonShuffleExchangeSink
Input [3]: [repo_name#217, path#219, id#221]
Arguments: hashpartitioning(id#221, 200)

(5) PhotonShuffleMapStage
Input [3]: [repo_name#217, path#219, id#221]
Arguments: ENSURE_REQUIREMENTS, [id=#629]

(6) PhotonShuffleExchangeSource
Input [3]: [repo_name#217, path#219, id#221]

(7) Scan
BigQueryV1RelationFromV2ScanWithNoMaterialization(org.apache.spark.sql.SQLC
ontext@934053c,StructType(StructField(id,StringType,true),StructField(size,
LongType,true),StructField(binary,BooleanType,true),StructField(copies,Long
Type,true)),BigQueryScanPushDowns(Some(StructType(StructField(id,StringType
,true),StructField(size,LongType,true),StructField(binary,BooleanType,true)
,StructField(copies,LongType,true))),[Lorg.apache.spark.sql.connector.expre
ssions.filter.Predicate;@1b872ece,None,None,[Ljava.lang.String;@7b8a780a,[L
java.lang.String;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Lorg.apache.spark.
sql.sources.Filter;@73415c2d),DirectBigQueryRelation[fe-dev-sandbox.dkushar
i_ds.contents]) dkushari_bq_test.dkushari_ds.contents
Output [4]: [id#228, size#229L, binary#231, copies#232L]
Arguments: [id#228, size#229L, binary#231, copies#232L],
[StructField(id,StringType,true), StructField(size,LongType,true),
StructField(binary,BooleanType,true), StructField(copies,LongType,true)],
PushedDownOperators(None,None,None,None,List(),List()),
PreScala213BigQueryRDD[55] at RDD at PreScala213BigQueryRDD.java:40,
BigQueryV1RelationFromV2ScanWithNoMaterialization(org.apache.spark.sql.SQLC
ontext@934053c,StructType(StructField(id,StringType,true),StructField(size,
LongType,true),StructField(binary,BooleanType,true),StructField(copies,Long
```

```
Type,true)),BigQueryScanPushDowns(Some(StructType(StructField(id,StringType
,true),StructField(size,LongType,true),StructField(binary,BooleanType,true)
,StructField(copies,LongType,true))),[Lorg.apache.spark.sql.connector.expre
ssions.filter.Predicate;@1b872ece,None,None,[Ljava.lang.String;@7b8a780a,[L
java.lang.String;@7b8a780a,[Ljava.lang.String;@7b8a780a,[Lorg.apache.spark.
sql.sources.Filter;@73415c2d),DirectBigQueryRelation[fe-dev-sandbox.dkushar
i_ds.contents]), `dkushari_bq_test`.`dkushari_ds`.`contents`,
Statistics(sizeInBytes=8.0 EiB, ColumnStat: N/A)

(8) PhotonRowToColumnar
Input [4]: [id#228, size#229L, binary#231, copies#232L]

(9) PhotonFilter
Input [4]: [id#228, size#229L, binary#231, copies#232L]
Arguments: isnotnull(id#228)

(10) PhotonShuffleExchangeSink
Input [4]: [id#228, size#229L, binary#231, copies#232L]
Arguments: hashpartitioning(id#228, 200)

(11) PhotonShuffleMapStage
Input [4]: [id#228, size#229L, binary#231, copies#232L]
Arguments: ENSURE_REQUIREMENTS, [id=#636]

(12) PhotonShuffleExchangeSource
Input [4]: [id#228, size#229L, binary#231, copies#232L]

(13) PhotonShuffledHashJoin
Left keys [1]: [id#221]
Right keys [1]: [id#228]
Join type: Inner
Join condition: None

(14) PhotonProject
Input [7]: [repo_name#217, path#219, id#221, id#228, size#229L, binary#231,
copies#232L]
Arguments: [repo_name#217, path#219, size#229L, copies#232L, binary#231]

(15) PhotonShuffleExchangeSink
Input [5]: [repo_name#217, path#219, size#229L, copies#232L, binary#231]
Arguments: hashpartitioning(repo_name#217, 200)

(16) PhotonShuffleMapStage
```

```
Input [5]: [repo_name#217, path#219, size#229L, copies#232L, binary#231]
Arguments: ENSURE_REQUIREMENTS, [id=#644]

(17) PhotonShuffleExchangeSource
Input [5]: [repo_name#217, path#219, size#229L, copies#232L, binary#231]

(18) PhotonSort
Input [5]: [repo_name#217, path#219, size#229L, copies#232L, binary#231]
Arguments: [repo_name#217 ASC NULLS FIRST, size#229L DESC NULLS LAST]

(19) PhotonWindow
Input [5]: [repo_name#217, path#219, size#229L, copies#232L, binary#231]
Arguments: [repo_name#217, path#219, size#229L, rank(size#229L)
windowspecdefinition(repo_name#217, size#229L DESC NULLS LAST,
specifiedwindowframe(RowFrame, unboundedpreceding$(), currentrow$())) AS
size_rank#193, copies#232L, binary#231]

(20) PhotonShuffleExchangeSink
Input [6]: [repo_name#217, path#219, size#229L, size_rank#193, copies#232L,
binary#231]
Arguments: rangepartitioning(repo_name#217 ASC NULLS FIRST, size_rank#193
ASC NULLS FIRST, 200)

(21) PhotonShuffleMapStage
Input [6]: [repo_name#217, path#219, size#229L, size_rank#193, copies#232L,
binary#231]
Arguments: ENSURE_REQUIREMENTS, [id=#652]

(22) PhotonShuffleExchangeSource
Input [6]: [repo_name#217, path#219, size#229L, size_rank#193, copies#232L,
binary#231]

(23) PhotonSort
Input [6]: [repo_name#217, path#219, size#229L, size_rank#193, copies#232L,
binary#231]
Arguments: [repo_name#217 ASC NULLS FIRST, size_rank#193 ASC NULLS FIRST]

(24) PhotonResultStage
Input [6]: [repo_name#217, path#219, size#229L, size_rank#193, copies#232L,
binary#231]

(25) ColumnarToRow
Input [6]: [repo_name#217, path#219, size#229L, size_rank#193, copies#232L,
binary#231]

(26) AdaptiveSparkPlan
Output [6]: [repo_name#217, path#219, size#229L, size_rank#193,
copies#232L, binary#231]
Arguments: isFinalPlan=false


== Photon Explanation ==
The query is fully supported by Photon.
```

Observation:

- *Took around 11 minutes to run*
- *Pushdown to BQ did not happen*

Query With Materialization:

Query:

```
WITH joined_files AS (
 SELECT
   f.repo_name,
   f.ref,
   f.path,
   f.mode,
   f.id,
   f.symlink_target,
   c.size,
   c.content,
   c.binary,
   c.copies
 FROM
   files WITH ('materializationEnabled' 'true') AS  f
   INNER JOIN contents WITH ('materializationEnabled' 'true') AS c
     ON f.id = c.id
)
SELECT
```

```
 repo_name,
 path,
 size,
 RANK() OVER (PARTITION BY repo_name ORDER BY size DESC) AS size_rank,
 copies,
 binary
FROM
 joined_files
ORDER BY
 repo_name,
 size_rank;
```

Explain Plan:

```
== Physical Plan ==
AdaptiveSparkPlan (7)
+- == Initial Plan ==
   Sort (6)
   +- Exchange (5)
      +- RunningWindowFunction (4)
         +- Sort (3)
            +- Exchange (2)
               +- EdgeBigQueryPushdownPlan (1)


(1) EdgeBigQueryPushdownPlan
Arguments: [SUBQUERY_46_COL_0#273, SUBQUERY_46_COL_1#275,
SUBQUERY_46_COL_2#285L, SUBQUERY_46_COL_3#288L, SUBQUERY_46_COL_4#287],
PreScala213BigQueryRDD[7] at RDD at PreScala213BigQueryRDD.java:40
External engine query: SELECT ( SUBQUERY_45.SUBQUERY_45_COL_0 ) AS
SUBQUERY_46_COL_0 , ( SUBQUERY_45.SUBQUERY_45_COL_1 ) AS SUBQUERY_46_COL_1
, ( SUBQUERY_45.SUBQUERY_45_COL_4 ) AS SUBQUERY_46_COL_2 , (
SUBQUERY_45.SUBQUERY_45_COL_6 ) AS SUBQUERY_46_COL_3 , (
SUBQUERY_45.SUBQUERY_45_COL_5 ) AS SUBQUERY_46_COL_4 FROM ( SELECT (
SUBQUERY_42.REPO_NAME ) AS SUBQUERY_45_COL_0 , ( SUBQUERY_42.PATH ) AS
SUBQUERY_45_COL_1 , ( SUBQUERY_42.ID ) AS SUBQUERY_45_COL_2 , (
SUBQUERY_44.ID ) AS SUBQUERY_45_COL_3 , ( SUBQUERY_44.SIZE ) AS
SUBQUERY_45_COL_4 , ( SUBQUERY_44.BINARY ) AS SUBQUERY_45_COL_5 , (
SUBQUERY_44.COPIES ) AS SUBQUERY_45_COL_6 FROM ( SELECT * FROM ( SELECT *
FROM (SELECT `repo_name`, `path`, `id` FROM
`fe-dev-sandbox`.`dkushari_ds`.`files` WHERE (`id` IS NOT NULL) ) AS
BQ_CONNECTOR_QUERY_ALIAS ) AS SUBQUERY_41 WHERE ( SUBQUERY_41.ID IS NOT
NULL ) ) AS SUBQUERY_42 INNER JOIN ( SELECT * FROM ( SELECT * FROM (SELECT
`id`, `size`, `binary`, `copies` FROM
`fe-dev-sandbox`.`dkushari_ds`.`contents` WHERE (`id` IS NOT NULL) ) AS
BQ_CONNECTOR_QUERY_ALIAS ) AS SUBQUERY_43 WHERE ( SUBQUERY_43.ID IS NOT
NULL ) ) AS SUBQUERY_44 ON ( SUBQUERY_42.ID = SUBQUERY_44.ID ) ) AS
SUBQUERY_45

(2) Exchange
Input [5]: [SUBQUERY_46_COL_0#273, SUBQUERY_46_COL_1#275,
SUBQUERY_46_COL_2#285L, SUBQUERY_46_COL_3#288L, SUBQUERY_46_COL_4#287]
Arguments: hashpartitioning(repo_name#273, 200), ENSURE_REQUIREMENTS,
[plan_id=113]

(3) Sort
Input [5]: [SUBQUERY_46_COL_0#273, SUBQUERY_46_COL_1#275,
SUBQUERY_46_COL_2#285L, SUBQUERY_46_COL_3#288L, SUBQUERY_46_COL_4#287]
Arguments: [repo_name#273 ASC NULLS FIRST, size#285L DESC NULLS LAST],
false, 0

(4) RunningWindowFunction
Input [5]: [SUBQUERY_46_COL_0#273, SUBQUERY_46_COL_1#275,
SUBQUERY_46_COL_2#285L, SUBQUERY_46_COL_3#288L, SUBQUERY_46_COL_4#287]
Arguments: [repo_name#273, path#275, size#285L, rank(size#285L)
windowspecdefinition(repo_name#273, size#285L DESC NULLS LAST,
specifiedwindowframe(RowFrame, unboundedpreceding$(), currentrow$())) AS
size_rank#249, copies#288L, binary#287], [repo_name#273], [size#285L DESC
NULLS LAST], false

(5) Exchange
Input [6]: [repo_name#273, path#275, size#285L, size_rank#249, copies#288L,
binary#287]
Arguments: rangepartitioning(repo_name#273 ASC NULLS FIRST, size_rank#249
ASC NULLS FIRST, 200), ENSURE_REQUIREMENTS, [plan_id=117]

(6) Sort
Input [6]: [repo_name#273, path#275, size#285L, size_rank#249, copies#288L,
binary#287]
Arguments: [repo_name#273 ASC NULLS FIRST, size_rank#249 ASC NULLS FIRST],
true, 0
```

```
(7) AdaptiveSparkPlan
Output [6]: [repo_name#273, path#275, size#285L, size_rank#249,
copies#288L, binary#287]
Arguments: isFinalPlan=false


== Photon Explanation ==
Photon does not fully support the query because:
            Unsupported node: EdgeBigQueryPushdownPlan
[SUBQUERY_46_COL_0#273, SUBQUERY_46_COL_1#275, SUBQUERY_46_COL_2#285L,
SUBQUERY_46_COL_3#288L, SUBQUERY_46_COL_4#287], PreScala213BigQueryRDD[7]
at RDD at PreScala213BigQueryRDD.java:40.

Reference node:
      EdgeBigQueryPushdownPlan [SUBQUERY_46_COL_0#273,
SUBQUERY_46_COL_1#275, SUBQUERY_46_COL_2#285L, SUBQUERY_46_COL_3#288L,
SUBQUERY_46_COL_4#287], PreScala213BigQueryRDD[7] at RDD at
PreScala213BigQueryRDD.java:40
```

Observation:

- *Took around 23 minutes to run*
- *Pushdown to BQ did happen*

Appendix:

Things to try -

- SLA breach - is the workflow finishing within SLA?
- Try with materialization
- Check with different cluster configurations
  - Cluster DBR version (16.4 LTS+)
  - Cluser sizing