```
== Physical Plan ==
AdaptiveSparkPlan (20)
+- HashAggregate (19)
  +- Exchange (18)
    +- HashAggregate (17)
      +- Project (16)
        +- BroadcastHashJoin LeftOuter BuildRight (15)
          :- Union (10)
          :  :- Project (3)
          :  :  +- Filter (2)
          :  :     +- Scan parquet  (1)
          :  :- Project (6)
          :  :  +- Filter (5)
          :  :     +- Scan parquet  (4)
          :  +- Project (9)
          :     +- Filter (8)
          :        +- Scan parquet  (7)
          +- BroadcastExchange (14)
            +- Project (13)
              +- Filter (12)
                +- Scan csv  (11)
```

(1) Scan parquet

Output [7]: [tpep_pickup_datetime#142, passenger_count#144, trip_distance#145, PULocationID#148L, fare_amount#151, tip_amount#154, total_amount#157]

Batched: true

Location: InMemoryFileIndex [file:/home/jovyan/work/data/yellow_tripdata_2023-01.parquet]

PushedFilters: [IsNotNull(passenger_count), IsNotNull(fare_amount), IsNotNull(trip_distance), GreaterThanOrEqual(passenger_count,1.0), GreaterThan(fare_amount,0.0), GreaterThan(trip_distance,0.0)]

ReadSchema: struct<tpep_pickup_datetime:timestamp_ntz,passenger_count:double,trip_distance:double,PULocationID:bigint,fare_amount:double,tip_amount:double,total_amount:double>

(2) Filter

Input [7]: [tpep_pickup_datetime#142, passenger_count#144, trip_distance#145, PULocationID#148L, fare_amount#151, tip_amount#154, total_amount#157]

Condition : (((((isnotnull(passenger_count#144) AND isnotnull(fare_amount#151)) AND isnotnull(trip_distance#145)) AND (passenger_count#144 >= 1.0)) AND (fare_amount#151 > 0.0)) AND (trip_distance#145 > 0.0))

(3) Project

Output [5]: [trip_distance#145 AS trip_distance#200, cast(PULocationID#148L as int) AS PULocationID#220, total_amount#157 AS total_amount#280, date_format(cast(cast(tpep_pickup_datetime#142 as date) as timestamp), yyyy-MM, Some(Etc/UTC)) AS month#717, CASE WHEN (fare_amount#151 > 0.0) THEN (tip_amount#154 / fare_amount#151) ELSE 0.0 END AS tip_pct#728]

Input [7]: [tpep_pickup_datetime#142, passenger_count#144, trip_distance#145, PULocationID#148L, fare_amount#151, tip_amount#154, total_amount#157]

(4) Scan parquet

Output [7]: [tpep_pickup_datetime#309, passenger_count#311L, trip_distance#312, PULocationID#315, fare_amount#318, tip_amount#321, total_amount#324]

Batched: true

Location: InMemoryFileIndex [file:/home/jovyan/work/data/yellow_tripdata_2023-02.parquet]

PushedFilters: [IsNotNull(passenger_count), IsNotNull(fare_amount), IsNotNull(trip_distance), GreaterThan(fare_amount,0.0), GreaterThan(trip_distance,0.0)]

ReadSchema:
struct<tpep_pickup_datetime:timestamp_ntz,passenger_count:bigint,trip_distance:double,PULocationID:int,fare_amount:double,tip_amount:double,total_amount:double>

(5) Filter

Input [7]: [tpep_pickup_datetime#309, passenger_count#311L, trip_distance#312, PULocationID#315, fare_amount#318, tip_amount#321, total_amount#324]

Condition : (((((isnotnull(passenger_count#311L) AND isnotnull(fare_amount#318)) AND isnotnull(trip_distance#312)) AND (cast(passenger_count#311L as double) >= 1.0)) AND (fare_amount#318 > 0.0)) AND (trip_distance#312 > 0.0))

(6) Project

Output [5]: [trip_distance#312, PULocationID#315, total_amount#324, date_format(cast(cast(tpep_pickup_datetime#309 as date) as timestamp), yyyy-MM, Some(Etc/UTC)) AS month#1111, CASE WHEN (fare_amount#318 > 0.0) THEN (tip_amount#321 / fare_amount#318) ELSE 0.0 END AS tip_pct#1112]

Input [7]: [tpep_pickup_datetime#309, passenger_count#311L, trip_distance#312, PULocationID#315, fare_amount#318, tip_amount#321, total_amount#324]

(7) Scan parquet

Output [7]: [tpep_pickup_datetime#476, passenger_count#478L, trip_distance#479, PULocationID#482, fare_amount#485, tip_amount#488, total_amount#491]

Batched: true

Location: InMemoryFileIndex [file:/home/jovyan/work/data/yellow_tripdata_2023-03.parquet]

PushedFilters: [IsNotNull(passenger_count), IsNotNull(fare_amount), IsNotNull(trip_distance), GreaterThan(fare_amount,0.0), GreaterThan(trip_distance,0.0)]

ReadSchema:
struct<tpep_pickup_datetime:timestamp_ntz,passenger_count:bigint,trip_distance:double,PULocationID:int,fare_amount:double,tip_amount:double,total_amount:double>

(8) Filter

Input [7]: [tpep_pickup_datetime#476, passenger_count#478L, trip_distance#479, PULocationID#482, fare_amount#485, tip_amount#488, total_amount#491]

Condition : (((((isnotnull(passenger_count#478L) AND isnotnull(fare_amount#485)) AND isnotnull(trip_distance#479)) AND (cast(passenger_count#478L as double) >= 1.0)) AND (fare_amount#485 > 0.0)) AND (trip_distance#479 > 0.0))

(9) Project

Output [5]: [trip_distance#479, PULocationID#482, total_amount#491, date_format(cast(cast(tpep_pickup_datetime#476 as date) as timestamp), yyyy-MM, Some(Etc/UTC)) AS month#1113, CASE WHEN (fare_amount#485 > 0.0) THEN (tip_amount#488 / fare_amount#485) ELSE 0.0 END AS tip_pct#1114]

Input [7]: [tpep_pickup_datetime#476, passenger_count#478L, trip_distance#479, PULocationID#482, fare_amount#485, tip_amount#488, total_amount#491]

(10) Union

(11) Scan csv

Output [2]: [LocationID#55, Zone#57]

Batched: false

Location: InMemoryFileIndex [file:/home/jovyan/work/data/taxi_zone_lookup.csv]

ReadSchema: struct<LocationID:string,Zone:string>

(12) Filter

Input [2]: [LocationID#55, Zone#57]

Condition : isnotnull(cast(LocationID#55 as int))

(13) Project

Output [2]: [cast(LocationID#55 as int) AS PULocationID#796, Zone#57 AS pu_zone#797]

Input [2]: [LocationID#55, Zone#57]

(14) BroadcastExchange

Input [2]: [PULocationID#796, pu_zone#797]

Arguments: HashedRelationBroadcastMode(List(cast(input[0, int, true] as bigint)),false), [plan_id=1948]

(15) BroadcastHashJoin

Left keys [1]: [PULocationID#220]

Right keys [1]: [PULocationID#796]

Join type: LeftOuter

Join condition: None

(16) Project

Output [5]: [trip_distance#200, total_amount#280, month#717, tip_pct#728, pu_zone#797]

Input [7]: [trip_distance#200, PULocationID#220, total_amount#280, month#717, tip_pct#728, PULocationID#796, pu_zone#797]

(17) HashAggregate

Input [5]: [trip_distance#200, total_amount#280, month#717, tip_pct#728, pu_zone#797]

Keys [2]: [month#717, pu_zone#797]

Functions [4]: [partial_count(1), partial_avg(trip_distance#200), partial_avg(tip_pct#728), partial_sum(total_amount#280)]

Aggregate Attributes [6]: [count#897L, sum#898, count#899L, sum#900, count#901L, sum#902]

Results [8]: [month#717, pu_zone#797, count#903L, sum#904, count#905L, sum#906, count#907L, sum#908]

(18) Exchange

Input [8]: [month#717, pu_zone#797, count#903L, sum#904, count#905L, sum#906, count#907L, sum#908]

Arguments: hashpartitioning(month#717, pu_zone#797, 8), ENSURE_REQUIREMENTS, [plan_id=1953]

(19) HashAggregate

Input [8]: [month#717, pu_zone#797, count#903L, sum#904, count#905L, sum#906, count#907L, sum#908]

Keys [2]: [month#717, pu_zone#797]

Functions [4]: [count(1), avg(trip_distance#200), avg(tip_pct#728), sum(total_amount#280)]

Aggregate Attributes [4]: [count(1)#863L, avg(trip_distance#200)#864, avg(tip_pct#728)#865, sum(total_amount#280)#866]

Results [6]: [month#717, pu_zone#797, count(1)#863L AS trips#859L, avg(trip_distance#200)#864 AS avg_miles#860, avg(tip_pct#728)#865 AS avg_tip_pct#861, sum(total_amount#280)#866 AS revenue#862]

(20) AdaptiveSparkPlan

Output [6]: [month#717, pu_zone#797, trips#859L, avg_miles#860, avg_tip_pct#861, revenue#862]

Arguments: isFinalPlan=false

None

== Physical Plan ==

AdaptiveSparkPlan (27)

+- Filter (26)

  +- Window (25)

    +- WindowGroupLimit (24)

      +- Sort (23)

```
+- Exchange (22)
   +- WindowGroupLimit (21)
      +- Sort (20)
         +- HashAggregate (19)
            +- Exchange (18)
               +- HashAggregate (17)
                  +- Project (16)
                     +- BroadcastHashJoin LeftOuter BuildRight (15)
                        :- Union (10)
                        :  :- Project (3)
                        :  :  +- Filter (2)
                        :  :     +- Scan parquet  (1)
                        :  :- Project (6)
                        :  :  +- Filter (5)
                        :  :     +- Scan parquet  (4)
                        :  +- Project (9)
                        :     +- Filter (8)
                        :        +- Scan parquet  (7)
                        +- BroadcastExchange (14)
                           +- Project (13)
                              +- Filter (12)
                                 +- Scan csv  (11)
```

(1) Scan parquet

Output [6]: [tpep_pickup_datetime#142, passenger_count#144, trip_distance#145, PULocationID#148L, fare_amount#151, total_amount#157]

Batched: true

Location: InMemoryFileIndex [file:/home/jovyan/work/data/yellow_tripdata_2023-01.parquet]

PushedFilters: [IsNotNull(passenger_count), IsNotNull(fare_amount), IsNotNull(trip_distance), GreaterThanOrEqual(passenger_count,1.0), GreaterThan(fare_amount,0.0), GreaterThan(trip_distance,0.0)]

ReadSchema: struct<tpep_pickup_datetime:timestamp_ntz,passenger_count:double,trip_distance:double,PULocationID:bigint,fare_amount:double,total_amount:double>

(2) Filter

Input [6]: [tpep_pickup_datetime#142, passenger_count#144, trip_distance#145, PULocationID#148L, fare_amount#151, total_amount#157]

Condition : (((((isnotnull(passenger_count#144) AND isnotnull(fare_amount#151)) AND isnotnull(trip_distance#145)) AND (passenger_count#144 >= 1.0)) AND (fare_amount#151 > 0.0)) AND (trip_distance#145 > 0.0))

(3) Project

Output [3]: [cast(PULocationID#148L as int) AS PULocationID#220, total_amount#157 AS total_amount#280, date_format(cast(cast(tpep_pickup_datetime#142 as date) as timestamp), yyyy-MM, Some(Etc/UTC)) AS month#717]

Input [6]: [tpep_pickup_datetime#142, passenger_count#144, trip_distance#145, PULocationID#148L, fare_amount#151, total_amount#157]

(4) Scan parquet

Output [6]: [tpep_pickup_datetime#309, passenger_count#311L, trip_distance#312, PULocationID#315, fare_amount#318, total_amount#324]

Batched: true

Location: InMemoryFileIndex [file:/home/jovyan/work/data/yellow_tripdata_2023-02.parquet]

PushedFilters: [IsNotNull(passenger_count), IsNotNull(fare_amount), IsNotNull(trip_distance), GreaterThan(fare_amount,0.0), GreaterThan(trip_distance,0.0)]

ReadSchema:
struct<tpep_pickup_datetime:timestamp_ntz,passenger_count:bigint,trip_distance:double,PULocationID:int,fare_amount:double,total_amount:double>

(5) Filter

Input [6]: [tpep_pickup_datetime#309, passenger_count#311L, trip_distance#312, PULocationID#315, fare_amount#318, total_amount#324]

Condition : (((((isnotnull(passenger_count#311L) AND isnotnull(fare_amount#318)) AND isnotnull(trip_distance#312)) AND (cast(passenger_count#311L as double) >= 1.0)) AND (fare_amount#318 > 0.0)) AND (trip_distance#312 > 0.0))

(6) Project

Output [3]: [PULocationID#315, total_amount#324, date_format(cast(cast(tpep_pickup_datetime#309 as date) as timestamp), yyyy-MM, Some(Etc/UTC)) AS month#1117]

Input [6]: [tpep_pickup_datetime#309, passenger_count#311L, trip_distance#312, PULocationID#315, fare_amount#318, total_amount#324]

(7) Scan parquet

Output [6]: [tpep_pickup_datetime#476, passenger_count#478L, trip_distance#479, PULocationID#482, fare_amount#485, total_amount#491]

Batched: true

Location: InMemoryFileIndex [file:/home/jovyan/work/data/yellow_tripdata_2023-03.parquet]

PushedFilters: [IsNotNull(passenger_count), IsNotNull(fare_amount), IsNotNull(trip_distance), GreaterThan(fare_amount,0.0), GreaterThan(trip_distance,0.0)]

ReadSchema:
struct<tpep_pickup_datetime:timestamp_ntz,passenger_count:bigint,trip_distance:double,PULocationID:int,fare_amount:double,total_amount:double>

(8) Filter

Input [6]: [tpep_pickup_datetime#476, passenger_count#478L, trip_distance#479, PULocationID#482, fare_amount#485, total_amount#491]

Condition : (((((isnotnull(passenger_count#478L) AND isnotnull(fare_amount#485)) AND isnotnull(trip_distance#479)) AND (cast(passenger_count#478L as double) >= 1.0)) AND (fare_amount#485 > 0.0)) AND (trip_distance#479 > 0.0))

(9) Project

Output [3]: [PULocationID#482, total_amount#491, date_format(cast(cast(tpep_pickup_datetime#476 as date) as timestamp), yyyy-MM, Some(Etc/UTC)) AS month#1118]

Input [6]: [tpep_pickup_datetime#476, passenger_count#478L, trip_distance#479, PULocationID#482, fare_amount#485, total_amount#491]

(10) Union

(11) Scan csv

Output [2]: [LocationID#55, Zone#57]

Batched: false

Location: InMemoryFileIndex [file:/home/jovyan/work/data/taxi_zone_lookup.csv]

ReadSchema: struct<LocationID:string,Zone:string>

(12) Filter

Input [2]: [LocationID#55, Zone#57]

Condition : isnotnull(cast(LocationID#55 as int))

(13) Project

Output [2]: [cast(LocationID#55 as int) AS PULocationID#796, Zone#57 AS pu_zone#797]

Input [2]: [LocationID#55, Zone#57]

(14) BroadcastExchange

Input [2]: [PULocationID#796, pu_zone#797]

Arguments: HashedRelationBroadcastMode(List(cast(input[0, int, true] as bigint)),false), [plan_id=2021]

(15) BroadcastHashJoin

Left keys [1]: [PULocationID#220]

Right keys [1]: [PULocationID#796]

Join type: LeftOuter

Join condition: None

(16) Project

Output [3]: [total_amount#280, month#717, pu_zone#797]

Input [5]: [PULocationID#220, total_amount#280, month#717, PULocationID#796, pu_zone#797]

(17) HashAggregate

Input [3]: [total_amount#280, month#717, pu_zone#797]

Keys [2]: [month#717, pu_zone#797]

Functions [1]: [partial_sum(total_amount#280)]

Aggregate Attributes [1]: [sum#902]

Results [3]: [month#717, pu_zone#797, sum#908]

(18) Exchange

Input [3]: [month#717, pu_zone#797, sum#908]

Arguments: hashpartitioning(month#717, pu_zone#797, 8), ENSURE_REQUIREMENTS, [plan_id=2026]

(19) HashAggregate

Input [3]: [month#717, pu_zone#797, sum#908]

Keys [2]: [month#717, pu_zone#797]

Functions [1]: [sum(total_amount#280)]

Aggregate Attributes [1]: [sum(total_amount#280)#866]

Results [3]: [month#717, pu_zone#797, sum(total_amount#280)#866 AS revenue#862]

(20) Sort

Input [3]: [month#717, pu_zone#797, revenue#862]

Arguments: [month#717 ASC NULLS FIRST, revenue#862 DESC NULLS LAST], false, 0

(21) WindowGroupLimit

Input [3]: [month#717, pu_zone#797, revenue#862]

Arguments: [month#717], [revenue#862 DESC NULLS LAST], row_number(), 10, Partial

(22) Exchange

Input [3]: [month#717, pu_zone#797, revenue#862]

Arguments: hashpartitioning(month#717, 8), ENSURE_REQUIREMENTS, [plan_id=2032]

(23) Sort

Input [3]: [month#717, pu_zone#797, revenue#862]

Arguments: [month#717 ASC NULLS FIRST, revenue#862 DESC NULLS LAST], false, 0

(24) WindowGroupLimit

Input [3]: [month#717, pu_zone#797, revenue#862]

Arguments: [month#717], [revenue#862 DESC NULLS LAST], row_number(), 10, Final

(25) Window

Input [3]: [month#717, pu_zone#797, revenue#862]

Arguments: [row_number() windowspecdefinition(month#717, revenue#862 DESC NULLS LAST, specifiedwindowframe(RowFrame, unboundedpreceding$(), currentrow$())) AS rnk#1021], [month#717], [revenue#862 DESC NULLS LAST]

(26) Filter

Input [4]: [month#717, pu_zone#797, revenue#862, rnk#1021]

Condition : (rnk#1021 <= 10)

(27) AdaptiveSparkPlan

Output [4]: [month#717, pu_zone#797, revenue#862, rnk#1021]

Arguments: isFinalPlan=false

None