Journey Task

```
import org.apache.spark.sql.functions.col
import org.apache.spark.sql.types.IntegerType
import org.apache.spark.util._
import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
```

```
import org.apache.spark.sql.functions.col
import org.apache.spark.sql.types.IntegerType
import org.apache.spark.util._
import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
```

Read the data from Reservation Path

```
//reading reservation table
var reservation = spark.read.parquet("s3://ha-prod-omnidata-us-east-1/marketing/email/ocelot/temp/training/input/reservation")
reservation.printSchema
reservation.count
reservation.show(10,false)
reservation.select(countDistinct("traveler_uuid"))
```

```
reservation: org.apache.spark.sql.DataFrame = [traveler_uuid: string, display_brandid: int ... 4 more fields]
root
 |-- traveler_uuid: string (nullable = true)
 |-- display_brandid: integer (nullable = true)
 |-- reservation_uuid: string (nullable = true)
 |-- reservation_start_date_utc: date (nullable = true)
 |-- reservation_end_date_utc: date (nullable = true)
 |-- reservation_primary_market_search_term: string (nullable = true)
res3: Long = 1938961
+------------------------------------+--------------+------------------------------------+--------------------------+------------------------+------------------------------------+
|traveler_uuid                       |display_brandid|reservation_uuid                    |reservation_start_date_utc|reservation_end_date_utc|reservation_primary_market_search_term|
+------------------------------------+--------------+------------------------------------+--------------------------+------------------------+------------------------------------+
|00125bbe-6851-41a7-a343-6ef9c4d4abea|321           |8ff45b7c-ff55-4158-985b-0d884515f4d8|2019-12-27                |2019-12-30              |bbc62f4b-5b28-45af-9a70-6690c673ff1b  |
|002ac75a-62ee-48e7-88f4-3ff0b1d3bff5|651           |e246fa6e-71fd-4893-b09a-e9e8defd1a2e|2022-08-29                |2022-09-10              |85f3433b-f664-40af-b2cf-ffa4900cef0f  |
|002ac75a-62ee-48e7-88f4-3ff0b1d3bff5|651           |53895fd6-ffd2-48c4-8d76-fdb92c201bbb|2018-07-14                |2018-07-21              |f9bc467f-68c6-4afd-b7c1-81182a3747cf  |
|00437853-c564-421b-adeb-bb3063294216|121           |ebde35f5-d4f1-4ef6-a8f6-eb7a3bf9570b|2018-07-26                |2018-07-31              |null                                  |
|0080bc34-55f2-4503-ab09-4685c47aa37e|321           |10188438-c50f-441c-bdae-654745b4c83c|2017-09-07                |2017-09-11              |f4cbb241-9781-4fbd-9138-27a4c1bd5985  |
|00846e7d-b8c7-4056-b87c-d382c21305c9|611           |027b99ac-bd5e-499c-8d0f-560d6e25393c|2021-07-11                |2021-07-18              |null                                  |
```

```
// checking for duplicates

reservation.select("*").groupBy("traveler_uuid","reservation_start_date_utc","reservation_end_date_utc").count().show()
```

```
+--------------------+--------------------------+------------------------+-----+
|       traveler_uuid|reservation_start_date_utc|reservation_end_date_utc|count|
+--------------------+--------------------------+------------------------+-----+
|10011106-ddfb-41b...|                2016-07-23|              2016-07-30|    1|
|106b0b1b-2200-41d...|                2017-08-26|              2017-08-27|    1|
|195ad0f8-62fa-4bf...|                2022-02-11|              2022-02-13|    1|
|1b224039-a384-47b...|                2018-12-04|              2018-12-13|    1|
|1bfa2849-de87-44d...|                2018-02-08|              2018-02-11|    1|
|1ff573ce-afbe-47e...|                2022-09-11|              2022-09-17|    1|
|2f683a64-e929-48f...|                2018-08-12|              2018-08-14|    1|
|31fdcc54-944a-496...|                2015-05-28|              2015-06-06|    1|
|32c3bd3d-5f5b-44c...|                2017-06-03|              2017-06-05|    1|
|3336e5a7-4fd4-405...|                2020-07-19|              2020-07-26|    1|
|385a12c7-8a30-480...|                2020-06-04|              2020-06-07|    1|
|3ef4f94f-911e-49f...|                2022-06-09|              2022-06-13|    1|
|433eec6b-ac01-415...|                2019-12-29|              2020-01-03|    1|
|471eecbc-fcab-4ea...|                2022-03-13|              2022-03-16|    1|
|55952f4b-1528-487...|                2017-06-19|              2017-06-25|    1|
```

Select the upcoming reservation for each user based on reservation_start_date_utc and rename column

```
var temp = reservation

// filtereing all the upcomin reservations

var z = temp.filter(current_date() <= col("reservation_start_date_utc"))
z.show(2,false)
z.count()
z.select(countDistinct("traveler_uuid"),countDistinct("reservation_uuid"),countDistinct("traveler_uuid","reservation_uuid")).show()
```

```
temp: org.apache.spark.sql.DataFrame = [traveler_uuid: string, display_brandid: int ... 4 more fields]
z: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [traveler_uuid: string, display_brandid: int ... 4 more fields]
+------------------------------------+--------------+------------------------------------+--------------------------+------------------------+------------------------------------+
|traveler_uuid                       |display_brandid|reservation_uuid                    |reservation_start_date_utc|reservation_end_date_utc|reservation_primary_market_search_term|
+------------------------------------+--------------+------------------------------------+--------------------------+------------------------+------------------------------------+
|060690e9-5ebe-405c-ad51-6ca5ff9b86c0|321           |b805c3eb-b8eb-43e8-bd78-acfe9804f13a|2022-09-21                |2022-09-25              |4828cc6d-15ea-42b1-9b5f-6329c28da13c  |
|07a44152-9ab1-46db-9569-13207482fe55|321           |c4d749f2-754a-4f83-b7dd-1611d21e9703|2022-09-09                |2022-09-11              |48b5bde9-e568-45ea-99fb-cbc4497bc783  |
+------------------------------------+--------------+------------------------------------+--------------------------+------------------------+------------------------------------+
only showing top 2 rows
res11: Long = 32973
+-----------------------------+-------------------------------+--------------------------------------------+
|count(DISTINCT traveler_uuid)|count(DISTINCT reservation_uuid)|count(DISTINCT traveler_uuid, reservation_uuid)|
+-----------------------------+-------------------------------+--------------------------------------------+
|                        28217|                          32973|                                       32973|
+-----------------------------+-------------------------------+--------------------------------------------+
```

```
// Getting all the upcoming reservations
val w1 = Window.partitionBy("traveler_uuid").orderBy(col("reservation_start_date_utc"))
var filtered_reservation = z.withColumn("row",row_number.over(w1)).filter(col("row")=== 1).drop("row")

// Checking schema and count
filtered_reservation.printSchema()
filtered_reservation.count
filtered_reservation.show()
```

```
w1: org.apache.spark.sql.expressions.WindowSpec = org.apache.spark.sql.expressions.WindowSpec@31880a4a
filtered_reservation: org.apache.spark.sql.DataFrame = [traveler_uuid: string, display_brandid: int ... 4 more fields]
root
 |-- traveler_uuid: string (nullable = true)
 |-- display_brandid: integer (nullable = true)
 |-- reservation_uuid: string (nullable = true)
 |-- reservation_start_date_utc: date (nullable = true)
 |-- reservation_end_date_utc: date (nullable = true)
 |-- reservation_primary_market_search_term: string (nullable = true)
res16: Long = 28217
+--------------------+--------------+--------------------+--------------------------+------------------------+------------------------------------+
|       traveler_uuid|display_brandid|    reservation_uuid|reservation_start_date_utc|reservation_end_date_utc|reservation_primary_market_search_term|
+--------------------+--------------+--------------------+--------------------------+------------------------+------------------------------------+
|002a0336-d762-435...|           321|1d934e17-57b2-47f...|                2022-09-28|              2022-10-04|                df679837-22dc-4db...|
|00ca9bd0-fda3-43d...|           321|0979a714-43fd-45f...|                2023-02-16|              2023-02-21|                899a501b-67ee-416...|
|0238eb6d-cf17-49e...|           321|f9cd41b9-29e7-487...|                2022-10-14|              2022-10-17|                2e91c5db-5f5f-4ab...|
|035842b1-7942-479...|           321|a34122fb-7b91-494...|                2022-10-01|              2022-10-06|                f5f60e40-00b7-414...|
|0386b8e2-b575-450...|           321|719d5177-418b-49f...|                2023-07-01|              2023-07-08|                4025d802-f9fa-45f...|
```

```
//changing column names as per the required names
filtered_reservation=filtered_reservation.withColumnRenamed("traveler_uuid", "public_uuid")
filtered_reservation=filtered_reservation.withColumnRenamed("display_brandid", "brand_id")
filtered_reservation=filtered_reservation.withColumnRenamed("reservation_uuid", "next_reservation_uuid")
filtered_reservation=filtered_reservation.withColumnRenamed("reservation_start_date_utc", "next_reservation_arrival_date")
filtered_reservation=filtered_reservation.withColumnRenamed("reservation_end_date_utc", "next_reservation_departure_date")
filtered_reservation=filtered_reservation.withColumnRenamed("reservation_primary_market_search_term", "next_reservation_primary_market_search_term_uuid")
filtered_reservation.printSchema()
filtered_reservation.show(2,false)
```

```
filtered_reservation: org.apache.spark.sql.DataFrame = [public_uuid: string, display_brandid: int ... 4 more fields]
filtered_reservation: org.apache.spark.sql.DataFrame = [public_uuid: string, brand_id: int ... 4 more fields]
filtered_reservation: org.apache.spark.sql.DataFrame = [public_uuid: string, brand_id: int ... 4 more fields]
filtered_reservation: org.apache.spark.sql.DataFrame = [public_uuid: string, brand_id: int ... 4 more fields]
filtered_reservation: org.apache.spark.sql.DataFrame = [public_uuid: string, brand_id: int ... 4 more fields]
filtered_reservation: org.apache.spark.sql.DataFrame = [public_uuid: string, brand_id: int ... 4 more fields]
root
 |-- public_uuid: string (nullable = true)
 |-- brand_id: integer (nullable = true)
 |-- next_reservation_uuid: string (nullable = true)
 |-- next_reservation_arrival_date: date (nullable = true)
 |-- next_reservation_departure_date: date (nullable = true)
 |-- next_reservation_primary_market_search_term_uuid: string (nullable = true)
+------------------------------------+--------+------------------------------------+-----------------------------+-------------------------------+---------------------------------------------
---+
|public_uuid                         |brand_id|next_reservation_uuid               |next_reservation_arrival_date|next_reservation_departure_date|next_reservation_primary_market_search_term_u
uid|
+------------------------------------+--------+------------------------------------+-----------------------------+-------------------------------+---------------------------------------------
```

```
//reading the destination recommendation table
var destination = spark.read.parquet("s3://ha-prod-omnidata-us-east-1/marketing/email/ocelot/temp/training/input/destination_reco")
destination.printSchema()
destination.count()
destination.show(2,false)
```

```
destination: org.apache.spark.sql.DataFrame = [key_type: string, destination: string ... 1 more field]
root
 |-- key_type: string (nullable = true)
 |-- destination: string (nullable = true)
 |-- input: map (nullable = true)
 |    |-- key: string
 |    |-- value: array (valueContainsNull = true)
 |    |    |-- element: string (containsNull = true)
res25: Long = 11350
+-----------+------------------------------------+---------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|key_type   |destination                         |input
|
+-----------+------------------------------------+---------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|[0e88f173-7e43-4eb0-8ac8-7bf88b5ef77c -> [348687.0], 8727754c-4162-4245-aefb-eff4eae93a1d -> [274809.0], 6f6477b6-e070-4757-9312-fb03151d407f
```

```
// Exploding array input to key and value columns
var desti_exp = destination.select($"key_type",$"destination",explode($"input"))
desti_exp.printSchema
desti_exp.count
```

```
desti_exp: org.apache.spark.sql.DataFrame = [key_type: string, destination: string ... 2 more fields]
root
 |-- key_type: string (nullable = true)
 |-- destination: string (nullable = true)
 |-- key: string (nullable = false)
 |-- value: array (nullable = true)
 |    |-- element: string (containsNull = true)
res30: Long = 99333
```

```
var desti_ext=desti_exp.select($"key_type",$"destination",$"key",explode($"value"))

desti_ext.printSchema
desti_ext.count
desti_ext.show(5,false)
```

```
desti_ext: org.apache.spark.sql.DataFrame = [key_type: string, destination: string ... 2 more fields]
root
 |-- key_type: string (nullable = true)
 |-- destination: string (nullable = true)
 |-- key: string (nullable = false)
 |-- col: string (nullable = true)
res33: Long = 99333
+-----------+------------------------------------+------------------------------------+--------+
|key_type   |destination                         |key                                 |col     |
+-----------+------------------------------------+------------------------------------+--------+
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|0e88f173-7e43-4eb0-8ac8-7bf88b5ef77c|348687.0|
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|8727754c-4162-4245-aefb-eff4eae93a1d|274809.0|
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|6f6477b6-e070-4757-9312-fb03151d407f|270846.0|
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|f0a5aa39-d0b6-4f3c-b6fd-9e3777eb5997|249444.0|
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|e4205acb-44ac-430a-afe1-deaa2b690f3d|222204.0|
+-----------+------------------------------------+------------------------------------+--------+
only showing top 5 rows
```

```
// Changing the datatype and renaming column col
val destination_ext= desti_ext.withColumn("Rating",col("col").cast(IntegerType)).drop("col")

//checking the schema and count
destination_ext.printSchema()
destination_ext.count
destination_ext.show(3,false)
destination_ext.select(countDistinct("destination"),countDistinct("key"),countDistinct("destination","key")).show()
```

```
destination_ext: org.apache.spark.sql.DataFrame = [key_type: string, destination: string ... 2 more fields]
root
 |-- key_type: string (nullable = true)
 |-- destination: string (nullable = true)
 |-- key: string (nullable = false)
 |-- Rating: integer (nullable = true)
res38: Long = 99333
+-----------+------------------------------------+------------------------------------+------+
|key_type   |destination                         |key                                 |Rating|
+-----------+------------------------------------+------------------------------------+------+
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|0e88f173-7e43-4eb0-8ac8-7bf88b5ef77c|348687|
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|8727754c-4162-4245-aefb-eff4eae93a1d|274809|
|destination|0206b892-e9fe-4c7d-9d2e-3db3dad27bb1|6f6477b6-e070-4757-9312-fb03151d407f|270846|
+-----------+------------------------------------+------------------------------------+------+
only showing top 3 rows
+---------------------------+-------------------+--------------------------------+
|count(DISTINCT destination)|count(DISTINCT key)|count(DISTINCT destination, key)|
+---------------------------+-------------------+--------------------------------+
```

Enrich the reservation with the destination recommendation file

```
// enriching the df with destination recommendation file
var reservation_join = filtered_reservation.join(destination_ext,filtered_reservation("next_reservation_primary_market_search_term_uuid")===destination_ext("destination"),"left")
```

```
//checking the schema and count
reservation_join.printSchema()
reservation_join.count
reservation_join.show(2,false)
```

```
reservation_join: org.apache.spark.sql.DataFrame = [public_uuid: string, brand_id: int ... 8 more fields]
root
 |-- public_uuid: string (nullable = true)
 |-- brand_id: integer (nullable = true)
 |-- next_reservation_uuid: string (nullable = true)
 |-- next_reservation_arrival_date: date (nullable = true)
 |-- next_reservation_departure_date: date (nullable = true)
 |-- next_reservation_primary_market_search_term_uuid: string (nullable = true)
 |-- key_type: string (nullable = true)
 |-- destination: string (nullable = true)
 |-- key: string (nullable = true)
 |-- Rating: integer (nullable = true)
res44: Long = 256810
+------------------------------------+--------+------------------------------------+----------------------------+----------------------------+-----------------------------------------
---+-----------+-----------------------------------+------------------------------------+------+
|public_uuid                         |brand_id|next_reservation_uuid               |next_reservation_arrival_date|next_reservation_departure_date|next_reservation_primary_market_search_term_u
uid|key_type   |destination                        |key                                 |Rating|
+------------------------------------+--------+------------------------------------+----------------------------+----------------------------+-----------------------------------------
```

```
// getting the top 5 recommendations
val w2 = Window.partitionBy("public_uuid").orderBy(col("Rating").desc)
var reservation_top5 = reservation_join.withColumn("row",row_number.over(w2)).filter(col("row") <= 5)

//checking the schema and count
reservation_top5.printSchema()
reservation_top5.show(10,false)
reservation_top5.count()
```

```
w2: org.apache.spark.sql.expressions.WindowSpec = org.apache.spark.sql.expressions.WindowSpec@3c7314ee
reservation_top5: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [public_uuid: string, brand_id: int ... 9 more fields]
root
 |-- public_uuid: string (nullable = true)
 |-- brand_id: integer (nullable = true)
 |-- next_reservation_uuid: string (nullable = true)
 |-- next_reservation_arrival_date: date (nullable = true)
 |-- next_reservation_departure_date: date (nullable = true)
 |-- next_reservation_primary_market_search_term_uuid: string (nullable = true)
 |-- key_type: string (nullable = true)
 |-- destination: string (nullable = true)
 |-- key: string (nullable = true)
 |-- Rating: integer (nullable = true)
 |-- row: integer (nullable = true)
+------------------------------------+--------+------------------------------------+----------------------------+----------------------------+-----------------------------------------
---+-----------+-----------------------------------+------------------------------------+------+---+
|public_uuid                         |brand_id|next_reservation_uuid               |next_reservation_arrival_date|next_reservation_departure_date|next_reservation_primary_market_search_term_u
uid|key_type   |destination                        |key                                 |Rating|row|
```

Do the pivoting for recommended destinations

```
//pivoting the table to get the top 5 recommendations
var reservation_top5_pivot = reservation_top5.groupBy("public_uuid").pivot("row").agg(first("key"))
//checking the schema and count
reservation_top5_pivot.printSchema
reservation_top5_pivot.count
reservation_top5_pivot.show(2)
```

```
reservation_top5_pivot: org.apache.spark.sql.DataFrame = [public_uuid: string, 1: string ... 4 more fields]
root
 |-- public_uuid: string (nullable = true)
 |-- 1: string (nullable = true)
 |-- 2: string (nullable = true)
 |-- 3: string (nullable = true)
 |-- 4: string (nullable = true)
 |-- 5: string (nullable = true)
res54: Long = 28217
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|         public_uuid|                   1|                   2|                   3|                   4|                   5|
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|002a0336-d762-435...|8c188197-3698-437...|a1bba0e9-9d39-499...|abd2745f-a2ed-465...|77277805-bf81-474...|cef296bb-2cee-49f...|
|00ca9bd0-fda3-43d...|5bd07004-1ea8-4c1...|2f1b39c2-9473-4b5...|0eb179f8-347b-4e0...|f4cbb241-9781-4fb...|403dd646-ce67-4e6...|
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
only showing top 2 rows
```

Rename columns as destination_reco_1, destination_reco_2, destination_reco_3, destination_reco_4 and destination_reco_5

```
// renaming all the columns
reservation_top5_pivot=reservation_top5_pivot.withColumnRenamed("1", "destination_reco_1")
reservation_top5_pivot=reservation_top5_pivot.withColumnRenamed("2", "destination_reco_2")
reservation_top5_pivot=reservation_top5_pivot.withColumnRenamed("3", "destination_reco_3")
reservation_top5_pivot=reservation_top5_pivot.withColumnRenamed("4", "destination_reco_4")
reservation_top5_pivot=reservation_top5_pivot.withColumnRenamed("5", "destination_reco_5")
reservation_top5_pivot.show(2)
reservation_top5_pivot.count
reservation_top5_pivot.select(countDistinct("public_uuid")).show()
```

```
reservation_top5_pivot: org.apache.spark.sql.DataFrame = [public_uuid: string, destination_reco_1: string ... 4 more fields]
reservation_top5_pivot: org.apache.spark.sql.DataFrame = [public_uuid: string, destination_reco_1: string ... 4 more fields]
reservation_top5_pivot: org.apache.spark.sql.DataFrame = [public_uuid: string, destination_reco_1: string ... 4 more fields]
reservation_top5_pivot: org.apache.spark.sql.DataFrame = [public_uuid: string, destination_reco_1: string ... 4 more fields]
reservation_top5_pivot: org.apache.spark.sql.DataFrame = [public_uuid: string, destination_reco_1: string ... 4 more fields]
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|         public_uuid|  destination_reco_1|  destination_reco_2|  destination_reco_3|  destination_reco_4|  destination_reco_5|
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|002a0336-d762-435...|8c188197-3698-437...|a1bba0e9-9d39-499...|abd2745f-a2ed-465...|77277805-bf81-474...|cef296bb-2cee-49f...|
|00ca9bd0-fda3-43d...|5bd07004-1ea8-4c1...|2f1b39c2-9473-4b5...|0eb179f8-347b-4e0...|f4cbb241-9781-4fb...|403dd646-ce67-4e6...|
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
only showing top 2 rows
res59: Long = 28217
+---------------------------+
|count(DISTINCT public_uuid)|
+---------------------------+
|                      28217|
+---------------------------+
```

```
// Joining the top 5 recommendations to the final table
var final_data = filtered_reservation.join(reservation_top5_pivot,filtered_reservation("public_uuid")===reservation_top5_pivot("public_uuid"),"left").drop(reservation_top5_pivot("public_uui
d"))

// selecting the required columns

var final_df = final_data.select("public_uuid","brand_id","next_reservation_uuid","next_reservation_arrival_date","next_reservation_departure_date","next_reservation_primary_market_search_t
erm_uuid","destination_reco_1","destination_reco_2","destination_reco_3","destination_reco_4","destination_reco_5")
final_data.printSchema
final_data.count

//checking the schema and count
final_df.printSchema
final_df.count
final_df.show(2)
```

```
final_data: org.apache.spark.sql.DataFrame = [brand_id: int, next_reservation_uuid: string ... 9 more fields]
```

```
final_df: org.apache.spark.sql.DataFrame = [public_uuid: string, brand_id: int ... 9 more fields]
root
 |-- brand_id: integer (nullable = true)
 |-- next_reservation_uuid: string (nullable = true)
 |-- next_reservation_arrival_date: date (nullable = true)
 |-- next_reservation_departure_date: date (nullable = true)
 |-- next_reservation_primary_market_search_term_uuid: string (nullable = true)
 |-- public_uuid: string (nullable = true)
 |-- destination_reco_1: string (nullable = true)
 |-- destination_reco_2: string (nullable = true)
 |-- destination_reco_3: string (nullable = true)
 |-- destination_reco_4: string (nullable = true)
 |-- destination_reco_5: string (nullable = true)
res64: Long = 28217
root
 |-- public_uuid: string (nullable = true)
```

Write the data into Journey Path

```
//writing the final df to the given path
final_data.write.parquet("s3://ha-prod-omnidata-us-east-1/marketing/email/ocelot/temp/training/output/premnivas/journey")
```

```
//checking th written file

var journey = spark.read.parquet("s3://ha-prod-omnidata-us-east-1/marketing/email/ocelot/temp/training/output/premnivas/journey")

journey.printSchema()
journey.count()
journey.show(2,false)
```

```
journey: org.apache.spark.sql.DataFrame = [brand_id: int, next_reservation_uuid: string ... 9 more fields]
root
 |-- brand_id: integer (nullable = true)
 |-- next_reservation_uuid: string (nullable = true)
 |-- next_reservation_arrival_date: date (nullable = true)
 |-- next_reservation_departure_date: date (nullable = true)
 |-- next_reservation_primary_market_search_term_uuid: string (nullable = true)
 |-- public_uuid: string (nullable = true)
 |-- destination_reco_1: string (nullable = true)
 |-- destination_reco_2: string (nullable = true)
 |-- destination_reco_3: string (nullable = true)
 |-- destination_reco_4: string (nullable = true)
 |-- destination_reco_5: string (nullable = true)
res84: Long = 28217
+--------+------------------------------------+-----------------------------+-------------------------------+----------------------------------------------------+------------------------------------
---+------------------------------------+------------------------------------+------------------------------------+------------------------------------+------------------------------------+
|brand_id|next_reservation_uuid               |next_reservation_arrival_date|next_reservation_departure_date|next_reservation_primary_market_search_term_uuid|public_uuid
|destination_reco_1                  |destination_reco_2                  |destination_reco_3                  |destination_reco_4                  |destination_reco_5                  |
```