Marketing Activity

```
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
import org.apache.spark.sql.SaveMode
```

```
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.Window
import org.apache.spark.sql.SaveMode
```

Select email_hash, brand_id, activity_type, activity_timestamp from Message_Gear Data

```
var msg_gear = spark.read.parquet("s3://ha-prod-omnidata-us-east-1/marketing/email/ocelot/temp/training/input/message_gear")
```

```
msg_gear: org.apache.spark.sql.DataFrame = [email_hash: string, activity_type: string ... 2 more fields]
```

```
//checking the schema and count
msg_gear.printSchema
msg_gear.count
```

```
root
 |-- email_hash: string (nullable = true)
 |-- activity_type: string (nullable = true)
 |-- activity_timestamp: timestamp (nullable = true)
 |-- category_code: integer (nullable = true)
res87: Long = 1000000
```

Filter the data for activity_type in (click, open)

```
var filter_msg_gear= msg_gear.filter(col("activity_type") === "click"|| col("activity_type") === "open")
```

```
filter_msg_gear: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, activity_type: string ... 2 more fields]
```

```
//checking the schema and count
filter_msg_gear.printSchema
filter_msg_gear.count
filter_msg_gear.groupBy("activity_type").count().show()
```

```
root
 |-- email_hash: string (nullable = true)
 |-- activity_type: string (nullable = true)
 |-- activity_timestamp: timestamp (nullable = true)
 |-- category_code: integer (nullable = true)
res92: Long = 320874
+-------------+------+
|activity_type| count|
+-------------+------+
|        click| 24158|
|         open|296716|
+-------------+------+
```

```
//checking data
filter_msg_gear.select(count("email_hash")).show()
filter_msg_gear.select(count("category_code")).show()
filter_msg_gear.select(countDistinct("email_hash")).show()
filter_msg_gear.select(countDistinct("category_code")).show()
filter_msg_gear.select(countDistinct("email_hash","activity_type")).show()
filter_msg_gear.select(countDistinct("email_hash","category_code")).show()

// Extracting the rows with respect to distinct email_hash
val distinct_msg_gear = filter_msg_gear.select("email_hash","category_code").distinct()
distinct_msg_gear.printSchema()
distinct_msg_gear.count()
distinct_msg_gear.show(5,false)
```

```
+-----------------+
|count(email_hash)|
+-----------------+
|           320874|
+-----------------+
+--------------------+
|count(category_code)|
+--------------------+
|              320874|
+--------------------+
+-------------------------+
|count(DISTINCT email_hash)|
+-------------------------+
|                   261620|
+-------------------------+
+----------------------------+
|count(DISTINCT category_code)|
+----------------------------+
```

last_marketing_open – maximum value of activity_timestamp where activity_type = open

```
//getting the recent timestamp activity for open activity type

var temp_filter_msg_gear = filter_msg_gear

val w1 = Window.partitionBy("email_hash").orderBy(col("activity_timestamp").desc)
temp_filter_msg_gear = temp_filter_msg_gear.filter(col("activity_type") ==="open").withColumn("row",row_number.over(w1)).filter(col("row")=== 1).drop("row")
temp_filter_msg_gear=temp_filter_msg_gear.drop("activity_type").drop("category_code")

//renaming the column

temp_filter_msg_gear=temp_filter_msg_gear.withColumnRenamed("activity_timestamp", "last_marketing_open")
temp_filter_msg_gear.printSchema()
temp_filter_msg_gear.count()
temp_filter_msg_gear.show(5, false)
```

```
temp_filter_msg_gear: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, activity_type: string ... 2 more fields]
w1: org.apache.spark.sql.expressions.WindowSpec = org.apache.spark.sql.expressions.WindowSpec@2163c319
temp_filter_msg_gear: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, activity_type: string ... 2 more fields]
temp_filter_msg_gear: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, activity_timestamp: timestamp]
temp_filter_msg_gear: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, last_marketing_open: timestamp]
root
 |-- email_hash: string (nullable = true)
 |-- last_marketing_open: timestamp (nullable = true)
res108: Long = 245257
+----------------------------------------------------------------+-------------------+
|email_hash                                                      |last_marketing_open|
+----------------------------------------------------------------+-------------------+
|0001d47f537d205149e3bf4a1b2e819c194feec0f23b6062e34cdc95a8e538d8|2022-05-25 04:55:33|
|00026b34df9c37b068be3fb1a579316178a067a4f203a571693b1331ba80358b|2022-05-15 11:39:19|
|00088915a7dba05ee1856e30e25c58e5e54d27dc91aa15097299f3a1d4ea3c9d|2022-02-20 19:15:45|
|000ac9c72742a063af4ae687b1a97a5b833ce8563a38242623bf4aa17aeb5fda|2021-02-04 18:09:51|
|001055f933b991e974bfa415e9b8620cd7370fa8e40a45293b9c4bd2f8a8e7eb|2021-07-15 12:26:53|
+----------------------------------------------------------------+-------------------+
```

Marketing Activity

```
//left joining the df to include the last marketing open column
var filter_msg_gear_1 = distinct_msg_gear.join(temp_filter_msg_gear,distinct_msg_gear("email_hash")===temp_filter_msg_gear("email_hash"),"left").drop(temp_filter_msg_gear("email_hash"))

//checking the schema and count
filter_msg_gear_1.show(5,false)
filter_msg_gear_1.printSchema()
filter_msg_gear_1.count
```

```
filter_msg_gear_1: org.apache.spark.sql.DataFrame = [category_code: int, email_hash: string ... 1 more field]
+-------------+--------------------------------------------------------------+------------------+
|category_code|email_hash                                                    |last_marketing_open|
+-------------+--------------------------------------------------------------+------------------+
|0            |0001d47f537d205149e3bf4a1b2e819c194feec0f23b6062e34cdc95a8e538d8|2022-05-25 04:55:33|
|0            |00026b34df9c37b068be3fb1a579316178a067a4f203a571693b1331ba80358b|2022-05-15 11:39:19|
|0            |00088915a7dba05ee1856e30e25c58e5e54d27dc91aa15097299f3a1d4ea3c9d|2022-02-20 19:15:45|
|0            |000ac9c72742a063af4ae687b1a97a5b833ce8563a38242623bf4aa17aeb5fda|2021-02-04 18:09:51|
|0            |001055f933b991e974bfa415e9b8620cd7370fa8e40a45293b9c4bd2f8a8e7eb|2021-07-15 12:26:53|
+-------------+--------------------------------------------------------------+------------------+
only showing top 5 rows
root
 |-- category_code: integer (nullable = true)
 |-- email_hash: string (nullable = true)
 |-- last_marketing_open: timestamp (nullable = true)
res114: Long = 261620
```

last_marketing_click – maximum value of activity_timestamp where activity_type = click

```
//getting the recent timestamp activity for open activity type

var temp_filter_msg_gear_two = filter_msg_gear
val w2 = Window.partitionBy("email_hash").orderBy(col("activity_timestamp").desc)
temp_filter_msg_gear_two = temp_filter_msg_gear_two.filter(col("activity_type") === "click").withColumn("row",row_number.over(w2)).filter(col("row")=== 1).drop("row")
temp_filter_msg_gear_two=temp_filter_msg_gear_two.drop("activity_type").drop("category_code").drop("last_marketing_open")

//renaming the column

temp_filter_msg_gear_two=temp_filter_msg_gear_two.withColumnRenamed("activity_timestamp", "last_marketing_click")
temp_filter_msg_gear_two.printSchema()
temp_filter_msg_gear_two.count()
temp_filter_msg_gear_two.show(5,false)
temp_filter_msg_gear_two.select(countDistinct("email_hash")).show()
```

```
temp_filter_msg_gear_two: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, activity_type: string ... 2 more fields]
w2: org.apache.spark.sql.expressions.WindowSpec = org.apache.spark.sql.expressions.WindowSpec@3360b86d
temp_filter_msg_gear_two: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, activity_type: string ... 2 more fields]
temp_filter_msg_gear_two: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, activity_timestamp: timestamp]
temp_filter_msg_gear_two: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [email_hash: string, last_marketing_click: timestamp]
root
 |-- email_hash: string (nullable = true)
 |-- last_marketing_click: timestamp (nullable = true)
res118: Long = 22480
+--------------------------------------------------------------+--------------------+
|email_hash                                                    |last_marketing_click|
+--------------------------------------------------------------+--------------------+
|00144b54ce7f70aad92de38f7a249142f846f89b67b35056b0a12a93815ffb44|2021-04-08 17:47:24 |
|001f2eb48cbb798b2f92926ca04efa6104c0dde96f7fc00d72f8666fafe149a1|2022-04-10 12:53:32 |
|00371492dd8accca5583991d9239f1bbfde569bb09037204e57e7b3830527321|2021-06-24 13:02:16 |
|0041f579480b0651ca83e2cee4ee584fa0db5d8a575b19d7054bf9012a970338|2021-05-20 04:14:34 |
|004a0fff8d9a87cc8d78bd99bc3a4bb7f530f4751302ef6c9f9fe364aa985af8|2020-09-21 19:12:48 |
+--------------------------------------------------------------+--------------------+
```

```
//left joining the df to include the last_marketing_click column
var filter_msg_gear_2 = filter_msg_gear_1.join(temp_filter_msg_gear_two, filter_msg_gear_1("email_hash")=== temp_filter_msg_gear_two("email_hash"),"left").drop(temp_filter_msg_gear_two
("email_hash"))

//checking the schema and count
filter_msg_gear_2.printSchema()
filter_msg_gear_2.count
filter_msg_gear_2.select(countDistinct("email_hash")).show()
filter_msg_gear_2.show(false)
```

```
filter_msg_gear_2: org.apache.spark.sql.DataFrame = [category_code: int, email_hash: string ... 2 more fields]
root
 |-- category_code: integer (nullable = true)
 |-- email_hash: string (nullable = true)
 |-- last_marketing_open: timestamp (nullable = true)
 |-- last_marketing_click: timestamp (nullable = true)
res146: Long = 261620
+-------------------------+
|count(DISTINCT email_hash)|
+-------------------------+
|                   245257|
+-------------------------+
+-------------+--------------------------------------------------------------+-------------------+--------------------+
|category_code|email_hash                                                    |last_marketing_open|last_marketing_click|
+-------------+--------------------------------------------------------------+-------------------+--------------------+
|0            |0001d47f537d205149e3bf4a1b2e819c194feec0f23b6062e34cdc95a8e538d8|2022-05-25 04:55:33|null                |
|0            |00026b34df9c37b068be3fb1a579316178a067a4f203a571693b1331ba80358b|2022-05-15 11:39:19|null                |
|0            |00088915a7dba05ee1856e30e25c58e5e54d27dc91aa15097299f3a1d4ea3c9d|2022-02-20 19:15:45|null                |
```

last_activity_timestamp – maximum of last_marketing_open or last_marketing_click

```
// getting the last activity timestamp of the two "last_marketing_open" and "last_marketing_click"

filter_msg_gear_2 = filter_msg_gear_2.withColumn("last_activity_timestamp", greatest(col("last_marketing_open"),col("last_marketing_click")))

//checking the schema and count

filter_msg_gear_2.printSchema()
filter_msg_gear_2.count
filter_msg_gear_2.select(countDistinct("email_hash")).show()
filter_msg_gear_2.show(50,false)
```

```
filter_msg_gear_2: org.apache.spark.sql.DataFrame = [category_code: int, email_hash: string ... 3 more fields]
root
 |-- category_code: integer (nullable = true)
 |-- email_hash: string (nullable = true)
 |-- last_marketing_open: timestamp (nullable = true)
 |-- last_marketing_click: timestamp (nullable = true)
 |-- last_activity_timestamp: timestamp (nullable = true)
res161: Long = 261620
+-------------------------+
|count(DISTINCT email_hash)|
+-------------------------+
|                   245257|
+-------------------------+
+-------------+--------------------------------------------------------------+-------------------+--------------------+----------------------+
|category_code|email_hash                                                    |last_marketing_open|last_marketing_click|last_activity_timestamp|
+-------------+--------------------------------------------------------------+-------------------+--------------------+----------------------+
|0            |0001d47f537d205149e3bf4a1b2e819c194feec0f23b6062e34cdc95a8e538d8|2022-05-25 04:55:33|null                |2022-05-25 04:55:33   |
|0            |00026b34df9c37b068be3fb1a579316178a067a4f203a571693b1331ba80358b|2022-05-15 11:39:19|null                |2022-05-15 11:39:19   |
```

Select the column based on the output Schema and write it to the Marketing Activity Path

```
// writing the file to the destination
filter_msg_gear_2.write.mode("overwrite").parquet("s3://ha-prod-omnidata-us-east-1/marketing/email/ocelot/temp/training/output/premnivas/marketing")
```

Marketing Activity

```
//checking the schema and count of the uploaded file

var gear = spark.read.parquet("s3://ha-prod-omnidata-us-east-1/marketing/email/ocelot/temp/training/output/premnivas/marketing")
gear.printSchema
gear.count
gear.show(10, false)
```

```
gear: org.apache.spark.sql.DataFrame = [category_code: int, email_hash: string ... 3 more fields]
root
 |-- category_code: integer (nullable = true)
 |-- email_hash: string (nullable = true)
 |-- last_marketing_open: timestamp (nullable = true)
 |-- last_marketing_click: timestamp (nullable = true)
 |-- last_activity_timestamp: timestamp (nullable = true)
res170: Long = 261620
+-------------+----------------------------------------------------------------+-------------------+-------------------+----------------------+
|category_code|email_hash                                                      |last_marketing_open|last_marketing_click|last_activity_timestamp|
+-------------+----------------------------------------------------------------+-------------------+-------------------+----------------------+
|0            |0000d293ede29ebbbbd4e316bdde82824ce3c72622088be4988c030e36d020ff|2021-07-28 22:30:33|null               |2021-07-28 22:30:33    |
|0            |0003bc21a25a44e804fd164a7df91d383e5fdebcb2a8b585e0505e1aa73a2604|2022-04-21 02:29:35|null               |2022-04-21 02:29:35    |
|0            |000a61d4d59dcedff74dae31fbdf08d09e6fb890de1c14697a543bf7660a1e8a|2022-04-19 15:38:53|null               |2022-04-19 15:38:53    |
|0            |000d4f208666bd3387a14701d655bf6efadeb495a03a5f806b1e78c6dbae8a6e|2020-11-11 19:02:40|null               |2020-11-11 19:02:40    |
|0            |000f7e8fe969bab6d6d4b4ba47d935751eaf66f399a43ab0c7441485d12ce2d6|2022-06-01 22:20:35|null               |2022-06-01 22:20:35    |
|0            |001c4dbe676f56732eb51c6404b44c43fca0c1eea9bb0b32bd8d06f3a5edf8b5|2022-02-16 22:02:37|null               |2022-02-16 22:02:37    |
|0            |0021637ad6cb946458d3e2b8f65c31c09320486e8cc1208ee30cf16e8ebf37eb|2022-04-28 22:23:42|null               |2022-04-28 22:23:42    |
```