Module 5 - Data Serving

March 7, 2025

0.0.1 Objective:

- Save and retrieve processed data efficiently inside Dataproc.
- Serve data in a structured way for analysis.
- Use Parquet, Hive, and CSV

[18]:
```
spark = SparkSession.builder \
.appName('Olist Ecommerce Performance Optmization') \
.config('spark.executor.memory','6g') \
.config('spark.executor.cores','4') \
.config('spark.executor.instances','2') \
.config('spark.driver.memory','4g') \
.config('spark.driver.maxResultSize','2g') \
.config('spark.sql.shuffle.partitions','64') \
.config('spark.default.parallelism','64') \
.config('spark.sql.adaptive.enabled','true') \
.config('spark.sql.adaptive.coalescePartition.enabled','true') \
.config('spark.sql.autoBroadcastJoinThreshold',20*1024*1024) \
.config('spark.sql.files.maxPartitionBytes','64MB') \
.config('spark.sql.files.openCostInBytes','2MB') \
.config('spark.memory.fraction',0.8) \
.config('spark.memory.storageFraction',0.2) \
.getOrCreate()
```

[20]:
```
full_orders_df = spark.read.parquet('/olist/processed/')
```

[22]:
```
full_orders_df.printSchema()
```

```
root
 |-- customer_id: string (nullable = true)
 |-- order_id: string (nullable = true)
 |-- seller_id: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- order_status: string (nullable = true)
 |-- order_purchase_timestamp: timestamp (nullable = true)
 |-- order_approved_at: timestamp (nullable = true)
 |-- order_delivered_carrier_date: timestamp (nullable = true)
 |-- order_delivered_customer_date: timestamp (nullable = true)
 |-- order_estimated_delivery_date: timestamp (nullable = true)
```

```
|-- order_item_id: integer (nullable = true)
|-- shipping_limit_date: timestamp (nullable = true)
|-- price: double (nullable = true)
|-- freight_value: double (nullable = true)
|-- product_category_name: string (nullable = true)
|-- product_name_lenght: integer (nullable = true)
|-- product_description_lenght: integer (nullable = true)
|-- product_photos_qty: integer (nullable = true)
|-- product_weight_g: integer (nullable = true)
|-- product_length_cm: integer (nullable = true)
|-- product_height_cm: integer (nullable = true)
|-- product_width_cm: integer (nullable = true)
|-- seller_zip_code_prefix: integer (nullable = true)
|-- seller_city: string (nullable = true)
|-- seller_state: string (nullable = true)
|-- customer_unique_id: string (nullable = true)
|-- customer_zip_code_prefix: integer (nullable = true)
|-- customer_city: string (nullable = true)
|-- customer_state: string (nullable = true)
|-- geolocation_zip_code_prefix: integer (nullable = true)
|-- geolocation_lat: double (nullable = true)
|-- geolocation_lng: double (nullable = true)
|-- geolocation_city: string (nullable = true)
|-- geolocation_state: string (nullable = true)
|-- review_id: string (nullable = true)
|-- review_score: string (nullable = true)
|-- review_comment_title: string (nullable = true)
|-- review_comment_message: string (nullable = true)
|-- review_creation_date: string (nullable = true)
|-- review_answer_timestamp: string (nullable = true)
|-- payment_sequential: integer (nullable = true)
|-- payment_type: string (nullable = true)
|-- payment_installments: integer (nullable = true)
|-- payment_value: double (nullable = true)
|-- is_delivered: integer (nullable = true)
|-- is_canceled: integer (nullable = true)
|-- order_revenue: double (nullable = true)
|-- customer_segment: string (nullable = true)
|-- hour_of_day: integer (nullable = true)
|-- order_day_type: string (nullable = true)
```

[23]:
```
# save as Parquet in hdfs

full_orders_df.write.mode('overwrite').parquet('/olist/proc')
```

[24]:
```
# Save is as a parquet in Google cloud storage

full_orders_df.write.mode('overwrite').parquet('gs://
↪dataproc-staging-us-central1-458263062208-tw36mmqt/temp_data')
```

[ ]:
```

```

[25]:
```
full_orders_df.write.mode('overwrite').saveAsTable('full_order_detail')
```

```
25/02/28 17:13:50 WARN SessionState: METASTORE_FILTER_HOOK will be ignored,
since hive.security.authorization.manager is set to instance of
HiveAuthorizerFactory.
```

[ ]:
```
spark.sql('show tables')
```

[ ]:
```

```

[26]:
```
full_orders_df.write.mode('overwrite').option('header','true').csv('/olist/proc/
↪')
```