SAMSUNG INNOVATION CAMPUS

BIG DATA

CAPSTONE PROJECT -1

LOGISTICS & DELIVERY PERFORMANCE ANALYSIS

BD702 - Group 8

TEAM MEMBERS

1. MALAK OSAMA
2. MENNA TAREK
3. MENNA ALLAH MAHMOUD

TABLE OF CONTENT

DATA SOURCE & STRUCTURAL OVERVIEW

LOGISTICS SUPPLY CHAIN REAL WORLD DATA

Originates from real-world logistics and supply chain operations.

For the purpose of analysis, the dataset has been divided into two main components:

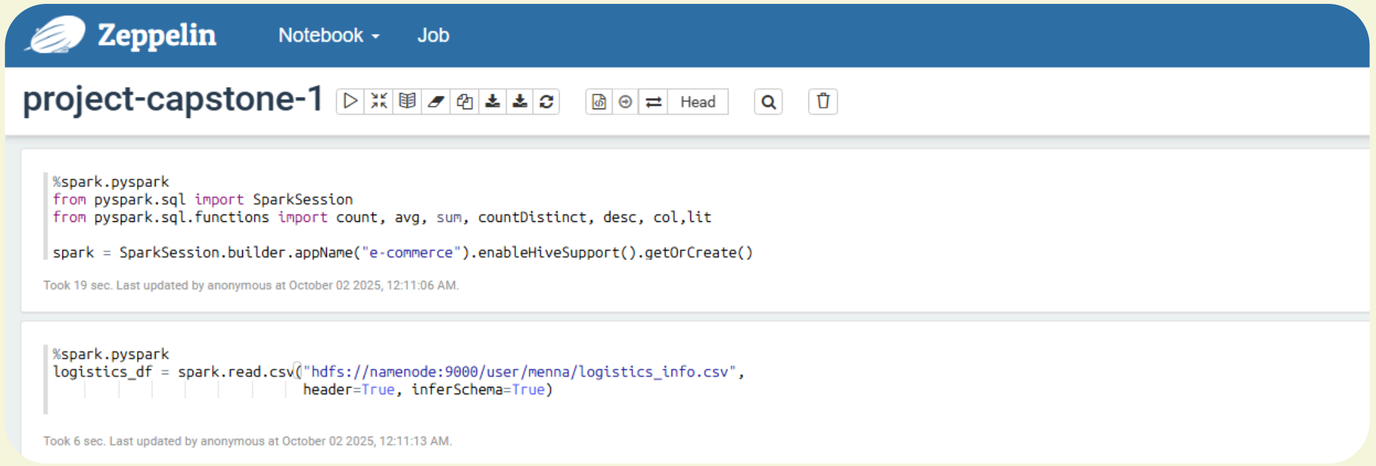
- **Logistics Information**: Contains details related to shipments, delivery tracking, warehouse handling, and transit times.
- **Order Information**: Contains details related to individual orders, including order IDs, order_date, category_name, ,department_name, order_item quantity, expected_delivery_date.

DATA EXTRACTION

1. Uploading Data to HDFS

- hdfs dfs -put /home/menna/projects/Big-Data-Cluster/logistics_info.csv /user/menna/

2. Reading Data from HDFS in PySpark



DATA TRANSFORMATION

PREPROCESSING

FILTERING OF UNRELTED DATA

ADJUSTING ATTRIBUTES SO THAT EACH COUNTRY HAS A WAREHOUSE ID

ADJUSTING THE ATTRIBUTES DATA TYPES

TRANSFORMATION

1. JOINING DATASETS

The Logistics Information and Order Information datasets were joined on the order_id column.

2. CALCULATING DELIVERY TIME DELTA

A new column, delivery_time_delta_days, was derived using the Spark datediff function

3. DERIVING KEY PERFORMANCE INDICATOR (KPI) – IS_LATE_DELIVERY

```
# Calculate delivery delta and late delivery
joined_df = joined_df.withColumn(
    "delivery_time_delta_days",
    F.datediff(F.col("actual_delivery_date"), F.col("expected_delivery_date"))
).withColumn(
    "is_late_delivery",
    F.when(F.col("delivery_time_delta_days") > 0, True).otherwise(False)
)
```

Took 0 sec. Last updated by anonymous at October 02 2025, 12:11:49 AM.

TRANSFORMATION

4. CREATING THE FINAL OUTPUT TABLE: STAGING_LOGISTICS_FACT

We selected only the relevant columns from the joined DataFrame to create a concise table.
The table captures

- Operational metrics (delivery time delta, late delivery flag) a
- Financial/logistics metrics (shipping cost, warehouse_id) for downstream analysis.

```
%spark.pyspark
# Select the columns that exist in joined_df
Staging_Logistics_Fact = joined_df.select(
    "order_id",
    "delivery_time_delta_days",
    "is_late_delivery",
    "shipping_cost",
    "warehouse_id"
)

Staging_Logistics_Fact.show()
```

order_id	delivery_time_delta_days	is_late_delivery	shipping_cost	warehouse_id
15081.289	0	false	84.99157	29.0
56444.684	0	false	181.99	25.0
7508.5713	-1	false	93.81015	2.0
56196.926	0	false	99.8906	14.0
5565.5796	-1	false	171.07587	1.0
32955.824	0	false	145.46329	0.0
35385.855	-1	false	167.99	0.0
36338.4	2	true	116.99	0.0
27692.854	-1	false	113.15623	21.0
56918.418	0	false	127.39	1.0
67071.39	0	false	111.575935	2.0
58629.56	-1	false	299.98	5.0
15836.077	2	true	290.95166	4.0
45317.816	-1	false	189.0	20.0
39779.484	-1	false	123.49	0.0

Took 1 sec. Last updated by anonymous at October 02 2025, 12:11:52 AM.

TRANSFORMATION

5. CALCULATING AVERAGE DELIVERY TIME DELTA PER WAREHOUSE

To evaluate the performance of each warehouse, we calculated the average delivery time delta using Spark SQL on the Staging_Logistics_Fact table.

This metric provides insight into how early or late orders are delivered on average for each warehouse.

```
%spark.pyspark
Staging_Logistics_Fact.createOrReplaceTempView("staging_logistics_fact")

avgDeliveryTimeDelta = spark.sql("""
    SELECT
        warehouse_id,
        AVG(delivery_time_delta_days) AS avg_delivery_time_delta
    FROM staging_logistics_fact
    GROUP BY warehouse_id
""")
avgDeliveryTimeDelta.show()
```

warehouse_id	avg_delivery_time_delta
147.0	-1.0
8.0	0.06310679611650485
70.0	0.058823529411764705
67.0	0.21052631578947367
0.0	0.07115198451113262
69.0	-0.16666666666666666
7.0	0.09925093632958802
142.0	2.0
112.0	-0.3333333333333333
124.0	1.0
128.0	-0.5
108.0	0.0
133.0	1.0
88.0	0.1111111111111111

Took 1 sec. Last updated by anonymous at October 02 2025, 12:16:38 AM.

DATA LOADING & INTEGRATION

Step 1: HDFS Directory Creation

Step 2: Write Processed Data to HDFS

Step 3: Create Hive Table Schema pointing to the Parquet files in HDFS

```
menna@DESKTOP-3PNO10O:~/projects/Big-Data-Cluster$ docker exec -it namenode bash
root@126a36e89378:/# hdfs dfs -ls /user/bigdata
Found 2 items
drwxr-xr-x   - root supergroup          0 2025-10-01 21:17 /user/bigdata/avgDeliveryTimeDelta
drwxr-xr-x   - root supergroup          0 2025-10-01 21:12 /user/bigdata/staging_logistics_fact
```

```
%spark.pyspark
from pyspark.sql.functions import col

Staging_Logistics_Fact_fixed = Staging_Logistics_Fact.select(
    col("order_id").cast("int"),
    col("delivery_time_delta_days").cast("int"),
    col("is_late_delivery").cast("boolean"),
    col("shipping_cost").cast("float"),
    col("warehouse_id").cast("int")
)

Staging_Logistics_Fact_fixed.write.mode("overwrite").parquet("hdfs://namenode:9000/user/bigdata/staging_logistics_fact")
```

Took 1 sec. Last updated by anonymous at October 02 2025, 12:12:09 AM.

```
menna@DESKTOP-3PNO10O:~/projects/Big-Data-Cluster$ docker exec -it hive-server bash
root@b35050373558:/# beeline -u "jdbc:hive2://localhost:10000/default"
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000/default
Connected to: Apache Hive (version 2.3.2)
Driver: Hive JDBC (version 2.3.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.3.2 by Apache Hive
0: jdbc:hive2://localhost:10000/default>
0: jdbc:hive2://localhost:10000/default> CREATE EXTERNAL TABLE staging_logistics_fact (
. . . . . .>   order_id STRING,
. . . . . .>   delivery_time_delta_days INT,
. . . . . .>   is_late_delivery BOOLEAN,
. . . . . .>   shipping_cost DOUBLE,
. . . . . .>   warehouse_id STRING
. . . . . .> )
. . . . . .> STORED AS PARQUET
. . . . . .> LOCATION 'hdfs:///user/bigdata/staging_logistics_fact';
No rows affected (1.449 seconds)
0: jdbc:hive2://localhost:10000/default> SHOW TABLES;
+------------------------+
|        tab_name        |
+------------------------+
| staging_logistics_fact |
```

```
%spark.pyspark
avgDeliveryTimeDelta.write.mode("overwrite").parquet("hdfs://namenode:9000/user/bigdata/avgDeliveryTimeDelta")
```

Took 0 sec. Last updated by anonymous at October 02 2025, 12:17:01 AM.

```
0: jdbc:hive2://localhost:10000/default> CREATE TABLE avgDeliveryTimeDelta(
. . . . . .>   warehouse_id INT,
. . . . . .>   avg_delivery_time_delta INT
. . . . . .> )
. . . . . .> STORED AS PARQUET
. . . . . .> LOCATION 'hdfs://namenode:9000/user/bigdata/avgDeliveryTimeDelta';
No rows affected (1.472 seconds)
```

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

//user/bigdata/staging_logistics_fact Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	0 B	Oct 01 20:40	3	128 MB	_SUCCESS
-rw-r--r--	root	supergroup	223.35 KB	Oct 01 20:40	3	128 MB	part-00000-c0018834-cd9e-4fe5-a06a-18b5793b3e1c-c000.snappy.parquet

Showing 1 to 2 of 2 entries Previous 1 Next

Hive Add a name... Add a description...

0.74s Database default Type text

```
SELECT * FROM staging_logistics_fact
```

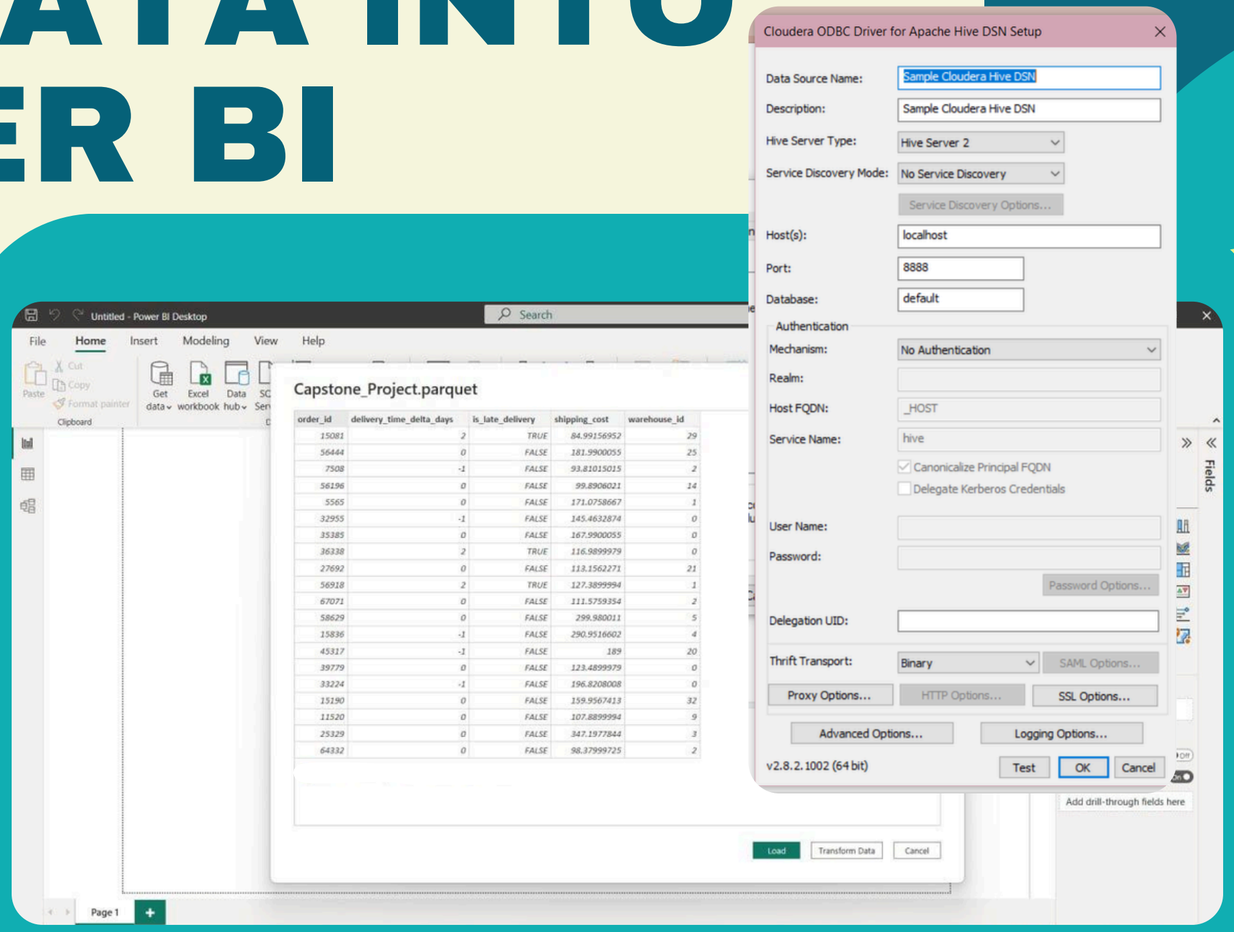
Query History Saved Queries Query Builder Results (100+)

	staging_logistics_fact.order_id	staging_logistics_fact.delivery_time_delta_days	staging_logistics_fact.is_late_delivery	stagin
1	15081	2	true	84.991
2	56444	0	false	181.99
3	7508	-1	false	93.810
4	56196	0	false	99.890
5	5565	0	false	171.07

LOADING DATA INTO POWER BI

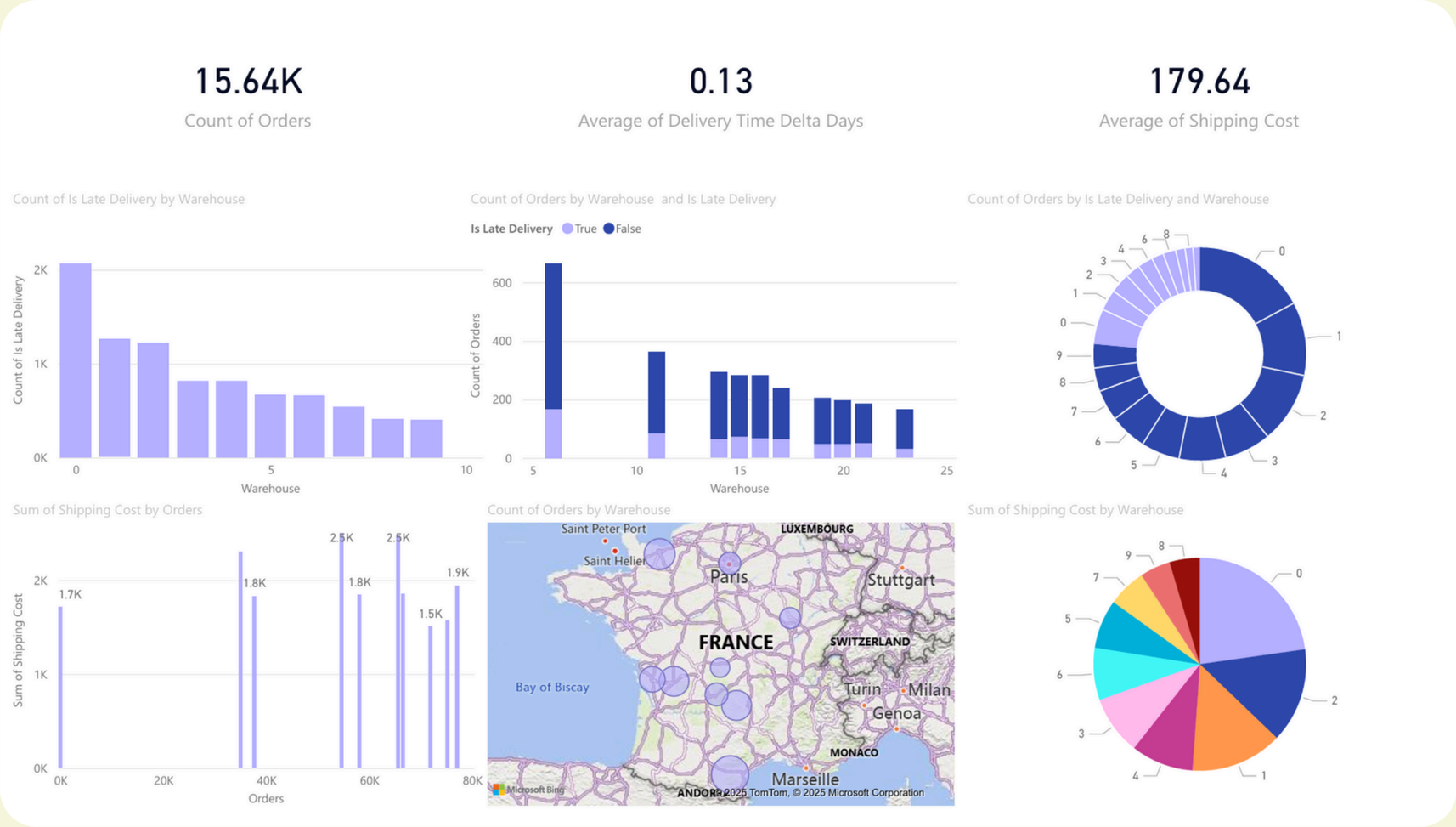
After the previous steps integrated the processed datasets into the Hive ecosystem for analysis, reporting, and visualization.

Data is loaded to powerBi to visualize it using ODBC to connect between Hive server and Power BI



DATA VISUALIZATION

VISUALIZING IN POWER BI



Made a dashboard of statistics from the data through Power BI to visualize the data used

THANK YOU