

## **DAY 1 – DATA ANALYST JOURNEY**

**Topic: Data Understanding, Cleaning & SQL Analysis**

**Dataset: Superstore Sales Dataset**

**Tools Used: LibreOffice Calc, MySQL Workbench**

### **1. Objective of Day 1**

**The objective of Day 1 was to:**

- **Understand a real-world dataset**
- **Clean and standardize messy data (dates)**
- **Import data into MySQL**
- **Perform basic SQL analysis**
- **Think like a data analyst, not just write queries**

### **2. Dataset Description**

**The dataset used is a Superstore Sales Dataset, which represents sales transactions of a retail business.**

**Each row represents:**

- **One product sold in an order**

**Important columns used:**

- **Order ID**
- **Order Date**
- **Ship Date**
- **Region**
- **Category**
- **Sub-Category**

- **Sales**
- **Profit**
- **Quantit**

### **3. Initial Problem Faced**

**When exploring the dataset, I observed that:**

- **The Order Date and Ship Date columns had mixed formats, such as:**
  - **11-08-2016**
  - **4/15/2017**
  - **2016-09-12**

**Because of mixed date formats:**

- **MySQL failed to import dates correctly**
- **Spreadsheet auto-formatting caused errors**

**This is a real-world data issue, not a beginner mistake.**

### **4. Data Cleaning in Spreadsheet (LibreOffice)**

**Steps followed:**

- 1. Opened the CSV file in LibreOffice Calc**
- 2. Carefully analyzed date patterns**
- 3. Split date components (day, month, year) into separate columns**
- 4. Reconstructed dates manually using logic**

## **5. Converted all dates into a single standard format**

**Final standardized date format:**

**YYYY-MM-DD**

**Key learning:**

**Automatic tools fail on messy data. Logical thinking works.**

## **5. Final Clean Dataset**

**After cleaning:**

- **All date values were consistent**
- **No slashes (/) or mixed formats remained**
- **Dataset was safe for database import**

**The cleaned file was saved as:**

**superstore\_final.csv**

## **6. Creating Database and Table in MySQL**

**Database creation:**

```
CREATE DATABASE data_analyst;
```

```
USE data_analyst;
```

**Table creation:**

```
CREATE TABLE orders (  
    order_id VARCHAR(50),  
    order_date DATE,
```

**ship\_date DATE,**  
**region VARCHAR(50),**  
**category VARCHAR(50),**  
**sub\_category VARCHAR(50),**  
**sales DECIMAL(10,2),**  
**profit DECIMAL(10,2),**  
**quantity INT**  
**);**

## **7. Importing Data into MySQL**

### **Steps:**

- 1. Open MySQL Workbench**
- 2. Select the data\_analyst database**
- 3. Use Table Data Import Wizard**
- 4. Import superstore\_final.csv**
- 5. Fix column mapping issues (duplicate category error)**
- 6. Successfully imported all rows**

### **Verification:**

**SELECT COUNT(\*) FROM orders;**  
**SELECT \* FROM orders LIMIT 5;**

	order_id	order_date	ship_date	region	category	sub_category	sales	profit	quantity
▶	CA-2016-152156	2016-08-11	2016-11-11	South	Furniture	Bookcases	261.96	41.91	2
	CA-2016-152156	2016-08-11	2016-11-11	South	Furniture	Chairs	731.94	219.58	3
	CA-2016-138688	2016-12-06	2016-06-16	West	Office Supplies	Labels	14.62	6.87	2
	US-2015-108966	2015-11-10	2015-10-18	South	Furniture	Tables	957.58	-383.03	5
	US-2015-108966	2015-11-10	2015-10-18	South	Office Supplies	Storage	22.37	2.52	2

	COUNT(*)
▶	9994

## 8. SQL Analysis Performed

### 8.1 Total Sales and Profit

**SELECT**

**ROUND(SUM(sales),2) AS total\_sales,**

**ROUND(SUM(profit),2) AS total\_profit**

**FROM orders;**

	total_sales	total_profit
▶	2297201.07	286397.79

### 8.2 Date Range of Business

**SELECT**

**MIN(order\_date) AS first\_order,**

**MAX(order\_date) AS last\_order**

**FROM orders;**

	first_order	last_order
▶	2014-01-02	2017-12-30

### 8.3 Region-wise Performance

**SELECT**

```

region,

ROUND(SUM(sales),2) AS total_sales,

ROUND(SUM(profit),2) AS total_profit

FROM orders

GROUP BY region;

```

	region	total_sales	total_profit
▶	South	391721.90	46749.71
	West	725457.93	108418.79
	Central	501239.88	39706.45
	East	678781.36	91522.84

## 8.4 Category-wise Profit

```

SELECT

category,

ROUND(SUM(profit),2) AS total_profit

FROM orders

GROUP BY category

ORDER BY total_profit DESC;

```

	category	total_profit
▶	Technology	145455.66
	Office Supplies	122490.88
	Furniture	18451.25

## 8.5 Profit Margin by Region

```

SELECT

region,

```

**ROUND((SUM(profit)/SUM(sales))\*100,2) AS  
profit\_margin  
FROM orders  
GROUP BY region  
ORDER BY profit\_margin DESC;**

	region	profit_margin
▶	West	14.94
	East	13.48
	South	11.93
	Central	7.92

## **9. Key Insights from Day 1**

- **The business operates across multiple years**
- **Some regions generate high sales but low profit**
- **Certain categories have weak profit margins**
- **Profit margin is a better metric than sales alone**

## **10. Business Recommendation**

**Based on Day 1 analysis:**

- **Focus on improving low-margin categories**
- **Investigate regions with high sales but low profit**
- **Optimize discount and pricing strategies**

## **11. Key Learnings from Day 1**

- **Real datasets are messy**
- **Data cleaning is more important than modeling**

- **SQL is not just for queries, but for insights**
- **Logical thinking beats tool dependency**