# Activity: "Fix the Dataset!" – Data Cleaning and Manipulation in MySQL

**Objective:** Please screenshot the results of each number.

Use SQL queries to clean, manipulate, and extract insights from a messy sales dataset.

1. **Create the Table and Insert Data**

```
CREATE TABLE sales (
  OrderID INT PRIMARY KEY,
  CustomerName VARCHAR(100),
  OrderDate VARCHAR(20), -- Stored as text to simulate messy dates
  Product VARCHAR(50),
  Quantity INT,
  Price DECIMAL(10,2),
  Region VARCHAR(20)
);

INSERT INTO sales VALUES
(1, 'John Smith', '03-14-25', 'T-shirt', 2, 20.00, 'North'),
(2, 'jane smith', '14/03/2025', 'Jeans', -1, 40.00, 'south'),
(3, 'Robert', '2025.03.15', 'Hoodie', 1, NULL, 'EAST'),
(4, NULL, '15-03-2025', 'Jacket', 1, 60.00, 'west');
```

2. **Tasks to Perform**

- ***Standardize Dates to `YYYY-MM-DD`***
- Use `STR_TO_DATE()` to parse and standardize the formats.

```
SELECT
  OrderID,
  STR_TO_DATE(OrderDate, '%m-%d-%y') AS FixedDate1,
  STR_TO_DATE(OrderDate, '%d/%m/%Y') AS FixedDate2,
  STR_TO_DATE(OrderDate, '%Y.%m.%d') AS FixedDate3,
  STR_TO_DATE(OrderDate, '%d-%m-%Y') AS FixedDate4
FROM sales;
```

3. **Normalize Region Capitalization**

```
SELECT
  OrderID,
```

```
  Region,
  CONCAT(UCASE(LEFT(Region, 1)), LCASE(SUBSTRING(Region, 2))) AS NormalizedRegion
FROM sales;
```

4. **Remove or Flag Invalid Records**

   - Negative quantity
   - NULL price or customer name

```
SELECT * FROM sales

WHERE Quantity < 0 OR Price IS NULL OR CustomerName IS NULL;
```

5. **Add a `Total` Column in a Query**

```
SELECT
  *,
  Quantity * Price AS Total
FROM sales
WHERE Quantity > 0 AND Price IS NOT NULL AND CustomerName IS NOT NULL;
```

6. **Group Sales by Region and Calculate Revenue**

```
SELECT
  CONCAT(UCASE(LEFT(Region, 1)), LCASE(SUBSTRING(Region, 2))) AS Region,
  SUM(Quantity * Price) AS Revenue
FROM sales
WHERE Quantity > 0 AND Price IS NOT NULL AND CustomerName IS NOT NULL
GROUP BY Region;
```

7. **Sort by Most Recent Orders**

```
SELECT *,

  STR_TO_DATE(OrderDate, '%d-%m-%Y') AS CleanDate

FROM sales

ORDER BY CleanDate DESC;
```

8. **Find Most Sold Product**

```
SELECT Product, SUM(Quantity) AS TotalSold
FROM sales
GROUP BY Product
```

```
ORDER BY TotalSold DESC
LIMIT 1;
```

9. **Filter Orders Above $50**

```
SELECT *, Quantity * Price AS Total
FROM sales
WHERE Quantity * Price > 50;
```