INTRODUCTION-

This café sales dataset contains 10,000 rows and 8 columns, and is intentionally dirty for Exploratory Data Analysis (EDA). This dataset is from Kaggle. The cleaning is done with PostgreSQL and the visualization with Power BI.

DATA CLEANING-

➔ Identify the distinct values in each column.

```
SELECT DISTINCT(item) FROM cafe_sales_2
```

```
OUTPUT-

Cake, Salad, Tea, Coffee, Juice, Smoothie, Cookie, Sandwich, Null, ERROR, UNKNOWN
```

Doing this for rest of the columns

➔ As the missing values are in nulls, ERROR and UNKNOWN. Identifying how many of these are there in columns.

```
SELECT        COUNT(*) FILTER(WHERE item IS NULL OR item = 'ERROR' OR item = 'UNKNOWN') AS
item_null,

              COUNT(*) FILTER(WHERE quantity IS NULL OR quantity = 'ERROR' OR quantity =
'UNKNOWN') AS quantity_null,

              COUNT(*) FILTER(WHERE price_per_unit IS NULL OR price_per_unit = 'ERROR' OR
price_per_unit = 'UNKNOWN') AS price_null,

              COUNT(*) FILTER(WHERE total_spent IS NULL OR total_spent = 'ERROR' OR
total_spent = 'UNKNOWN') AS total_null,

              COUNT(*) FILTER(WHERE payment_method IS NULL OR payment_method = 'ERROR' OR
payment_method = 'UNKNOWN') AS payment_null,

              COUNT(*) FILTER(WHERE location IS NULL OR location = 'ERROR' OR location =
'UNKNOWN') AS location_null,

              COUNT(*) FILTER(WHERE transaction_date IS NULL OR transaction_date = 'ERROR' OR
transaction_date = 'UNKNOWN') AS date_null

FROM cafe_sales_2.
```

```
OUTPUT-

item_null - 969

quantity_null - 479

price_null - 533

total_null - 502

payment_null - 3178

location_null - 3961

date_null - 460
```

➔ **location –** filling the missing values in location column with ‘In-store’.

```
UPDATE cafe_sales_2
SET location = 'In-store'
WHERE location = 'ERROR'
OR location = 'UNKNOWN'
OR location IS NULL
```

➔ **payment_method -** filling the missing values with ‘Cash’.

```
UPDATE cafe_sales_2
SET payment_method = 'Cash'
WHERE payment_method = 'UNKNOWN'
OR payment_method = 'ERROR'
OR payment_method IS NULL
```

➔ **Item –** filling ‘item’ column with respective prices in ‘price_per_unit’.

```
UPDATE cafe_sales_2
SET item =
	CASE
		WHEN price_per_unit = '4.0' THEN 'Smoothie'
		WHEN price_per_unit = '1.0' THEN 'Cookie'
		WHEN price_per_unit = '3.0' THEN 'Cake'
		WHEN price_per_unit = '5.0' THEN 'Salad'
		WHEN price_per_unit = '1.5' THEN 'Tea'
		WHEN price_per_unit = '2.0' THEN 'Coffee'
		WHEN price_per_unit = '3.0' THEN 'Juice'
	END
WHERE item = 'ERROR'
OR item = 'UNKNOWN'
OR item IS NULL
```

➔ **price_per_unit –** price_per_unit filling is done using 2 other columns. 'total_spent/quantity'.

```
UPDATE cafe_sales_2
SET price_per_unit = total_spent::REAL/quantity::REAL
WHERE price_per_unit IS NULL
OR price_per_unit = 'ERROR'
OR price_per_unit = 'UNKNOWN'
AND total_spent <> 'ERROR'
AND total_spent <> 'UNKNOWN'
AND total_spent IS NOT NULL
AND quantity <> 'ERROR'
AND quantity <> 'UNKNOWN'
AND quantity IS NOT NULL
```

➔ **quantity –** Same as price_per_unit, using 2 other columns. 'total_spent/price_per_unit'

```
UPDATE cafe_sales_2
SET quantity = total_spent::REAL/price_per_unit::REAL
WHERE quantity IS NULL
OR quantity = 'ERROR'
OR quantity = 'UNKNOWN'
AND total_spent <> 'ERROR'
AND total_spent <> 'UNKNOWN'
AND total_spent IS NOT NULL
AND price_per_unit <> 'ERROR'
AND price_per_unit <> 'UNKNOWN'
AND price_per_unit IS NOT NULL
```

➔ **total_spent –** Using 2 other columns. ‘quantity * price_per_unit’.

```
UPDATE cafe_sales_2
SET total_spent = quantity::REAL * price_per_unit::REAL
OR total_spent IS NULL
OR total_spent = 'ERROR'
WHERE total_spent = 'UNKNOWN'
AND quantity <> 'ERROR'
AND quantity <> 'UNKNOWN'
AND quantity IS NOT NULL
AND price_per_unit <> 'ERROR'
AND price_per_unit <> 'UNKNOWN'
AND price_per_unit IS NOT NULL
```

➔ **price_per_unit –** Updating the column with filled values in ‘item’

```
UPDATE cafe_sales_2
SET price_per_unit =
	CASE
		WHEN item = 'Sandwich' THEN '4.0'
		WHEN item = 'Cookie' THEN '1.0'
		WHEN item = 'Cake' THEN '3.0'
		WHEN item = 'Salad' THEN '5.0'
		WHEN item = 'Tea' THEN '1.5'
		WHEN item = 'Coffee' THEN '2.0'
		WHEN item = 'Juice' THEN '3.0'
		WHEN item = 'Smoothie' THEN '4.0'
	END
WHERE price_per_unit = 'UNKNOWN'
OR price_per_unit = 'ERROR'
OR price_per_unit IS NULL
```

➔ Check for any missing values.

```
SELECT        COUNT(*) FILTER(WHERE item IS NULL OR item = 'ERROR' OR item = 'UNKNOWN') AS
item_null,

              COUNT(*) FILTER(WHERE quantity IS NULL OR quantity = 'ERROR' OR quantity =
'UNKNOWN') AS quantity_null,

              COUNT(*) FILTER(WHERE price_per_unit IS NULL OR price_per_unit = 'ERROR' OR
price_per_unit = 'UNKNOWN') AS price_null,

              COUNT(*) FILTER(WHERE total_spent IS NULL OR total_spent = 'ERROR' OR
total_spent = 'UNKNOWN') AS total_null,

              COUNT(*) FILTER(WHERE payment_method IS NULL OR payment_method = 'ERROR' OR
payment_method = 'UNKNOWN') AS payment_null,

              COUNT(*) FILTER(WHERE location IS NULL OR location = 'ERROR' OR location =
'UNKNOWN') AS location_null,

              COUNT(*) FILTER(WHERE transaction_date IS NULL OR transaction_date = 'ERROR' OR
transaction_date = 'UNKNOWN') AS date_null

FROM cafe_sales_2.
```

➔ If any more of those values are still there in the columns, then update them. Update total_spent, quantity, price_per_unit with '0', item with 'UNAVAILABLE'.

quantity -

```
UPDATE cafe_sales_2

SET quantity = '0'

WHERE quantity = 'ERROR'

OR quantity = 'UNKNOWN'

OR quantity IS NULL
```

total_spent -

```
UPDATE cafe_sales_2

SET total_spent = '0'

WHERE total_spent = 'ERROR'

OR total_spent = 'UNKNOWN'

OR total_spent IS NULL
```

price_per_unit -

```
UPDATE cafe_sales_2

SET price_per_unit = '0'

WHERE price_per_unit = 'ERROR'

OR price_per_unit = 'UNKNOWN'

OR price_per_unit IS NULL
```

item –

```
UPDATE cafe_sales_2
SET item = 'UNAVAILABLE'
WHERE item = 'ERROR'
OR item = 'UNKNOWN'
OR item IS NULL
```

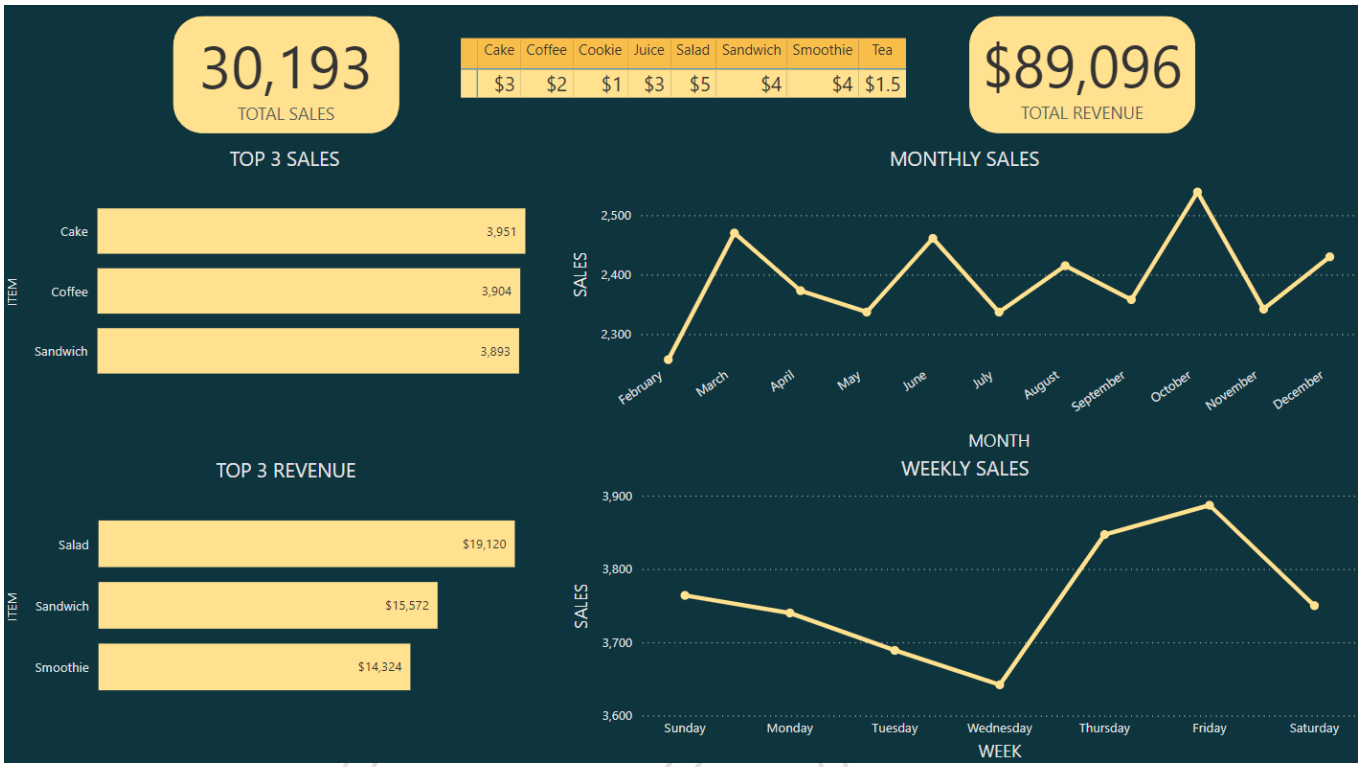
➔ If total_spent is filled, but price_per_unit and quantity are missing then fill them manually.

```
UPDATE cafe_sales_2
SET price_per_unit = '5.0',
quantity = '5',
item = 'Salad'
WHERE transaction_id = 'TXN_7376255'
```

➔ **transaction_date –** Filling this with a default date of '2023-01-01'.

```
UPDATE cafe_sales_2
SET transaction_date = '2023-01-01'
WHERE transaction_date IS NULL
OR transaction_date = 'ERROR'
OR transaction_date = 'UNKNOWN'
```

VISUALIZATION-



This visualization was created using Power BI.

- At the top, it presents total sales and total revenue, with the price per item displayed in between.
- Next, it highlights the top three selling products and the top three revenue-generating products.
- Lastly, the monthly and weekly sales graphs provide an overview of sales trends over time.

Key insights from the visualization:

- Cake is the top-selling product, but Salad generates the highest revenue due to its higher price point.
- Sandwich ranks in the top three for both sales volume and revenue.
- Monthly and weekly sales patterns remain fairly stable, with little fluctuation.
- Monthly sales range between 2,300 – 2,500 units.
- Weekly sales vary between 3,600 – 3,900 units.