SQL Query Document

Coffee Shop Sales Analysis Project

Query 1 : Data Cleaning

- **Standardize Formats**: Ensure consistency in date, time, and text formats across the dataset.
- **Correct Data Types**: Modify column data types to match the expected data (e.g., dates, numbers, text).
- **Rename Columns**: Clarify column names for better understanding and remove any encoding issues.
- **Remove Duplicates**: Identify and eliminate duplicate records to avoid skewed analysis.
- **Handle Missing Values**: Address null or missing data by either filling, averaging, or removing them.
- **Enforce Consistency**: Apply constraints like `NOT NULL` and proper data lengths to maintain data integrity.
- **Eliminate Anomalies**: Identify and correct any outliers or inconsistencies that might distort analysis.
- **Normalize Data**: Standardize numerical scales or formats across similar columns to enable consistent analysis.
- **Removing Anomalies**: Any special characters or encoding issues (like `ï»¿`)

```
Create database Project_Coffee_Shop_Sale;

use Project_Coffee_Shop_Sale;

select Count(*) from coffee_shop_sale;

update coffee_shop_sale
set transaction_date = str_to_date(transaction_date, "%d-%m-%Y");
```

```sql
alter table coffee_shop_sale
modify column transaction_date Date;

alter table coffee_shop_sale
change column ï»¿transaction_id transaction_id int;

-- Modify column data types and names as part of data cleaning
alter table coffee_shop_sale
modify column transaction_id int not null,
modify column transaction_date date not null,
modify column transaction_time time not null,
modify column transaction_qty int not null,
modify column store_id int not null,
modify column store_location varchar(100) not null,
modify column product_id int not null,
modify column unit_price decimal(10, 2) not null,
modify column product_category varchar(50) not null,
modify column product_type varchar(50) not null,
modify column product_detail text not null;

describe coffee_shop_sale;
```

Field	Type	Null	Key	Default	Extra
transaction_id	int	NO		NULL	
transaction_date	date	NO		NULL	
transaction_time	time	NO		NULL	
transaction_qty	int	NO		NULL	
store_id	int	NO		NULL	
store_location	varchar(100)	NO		NULL	
product_id	int	NO		NULL	
unit_price	decimal(10,2)	NO		NULL	
product_category	varchar(50)	NO		NULL	
product_type	varchar(50)	NO		NULL	
product_detail	text	NO		NULL	

Query 2 : MoM Growth And Difference For Sales

```sql
-- MoM Growth and Difference For Sales
SELECT
    Sale_Month,
    Total_Sales,
    LAG(Total_Sales) OVER (ORDER BY Sale_Month) AS Previous_Month_Sales,
    (Total_Sales - LAG(Total_Sales) OVER (ORDER BY Sale_Month)) AS MoM_Difference,
    ROUND(
        ((Total_Sales - LAG(Total_Sales) OVER (ORDER BY Sale_Month)) /
        LAG(Total_Sales) OVER (ORDER BY Sale_Month)) * 100, 2
    ) AS MoM_Growth_Percentage,
    CONCAT(
        IF(ROUND(
            ((Total_Sales - LAG(Total_Sales) OVER (ORDER BY Sale_Month)) /
            LAG(Total_Sales) OVER (ORDER BY Sale_Month)) * 100, 2) >= 0, '+ ', '- '),
        ROUND(
            ABS(
                (Total_Sales - LAG(Total_Sales) OVER (ORDER BY Sale_Month)) /
                LAG(Total_Sales) OVER (ORDER BY Sale_Month) * 100
            ), 2
        ),
        '% | ',
        ROUND(
            ABS(Total_Sales - LAG(Total_Sales) OVER (ORDER BY Sale_Month)) / 1000, 2
        ),
        'K Vs LM'
    ) AS MoM_Formatted
FROM (
    SELECT
        MONTH(transaction_date) AS Sale_Month,
        SUM(transaction_qty * unit_price) AS Total_Sales
    FROM
        coffee_shop_sale
    GROUP BY
        Sale_Month
) AS Monthly_Sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Sale_Month	Total_Sales	Previous_Month_Sales	MoM_Difference	MoM_Growth_Percentage	MoM_Formatted
1	81677.74	NULL	NULL	NULL	NULL
2	76145.19	81677.74	-5532.55	-6.77	- 6.77% | 5.53K Vs LM
3	98834.68	76145.19	22689.49	29.80	+ 29.80% | 22.69K Vs LM
4	118941.08	98834.68	20106.40	20.34	+ 20.34% | 20.11K Vs LM
5	156727.76	118941.08	37786.68	31.77	+ 31.77% | 37.79K Vs LM
6	166485.88	156727.76	9758.12	6.23	+ 6.23% | 9.76K Vs LM

Query 3 : Total Sales Monthly

```
80    -- Total Sales Monthly
81 •  select month(transaction_date) AS Monthly_sale,
82           concat("$ ",round(sum(transaction_qty*unit_price), 0)) As Total_Sales
83    from coffee_shop_sale
84    group by Monthly_sale;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Monthly_sale	Total_Sales
1	$ 81678
2	$ 76145
3	$ 98835
4	$ 118941
5	$ 156728
6	$ 166486

Query 4 : Total Orders Monthly

```
122    -- Total Orders Monthly
123 •  select month(transaction_date) AS Monthly_sale,
124           round(count(transaction_qty)) As Total_Orders
125    from coffee_shop_sale
126    group by Monthly_sale;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Monthly_sale	Total_Orders
1	17314
2	16359
3	21229
4	25335
5	33527
6	35352

Query 5 : Total Quantity Monthly

```
164    -- Total Quantity Monthly
165 •  select month(transaction_date) AS Monthly_Qty,
166           concat(round(sum(transaction_qty)/1000, 2), "K  Qty") As Total_Qty
167    from coffee_shop_sale
168    group by Monthly_Qty;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Monthly_sale	Total_Orders
1	17314
2	16359
3	21229
4	25335
5	33527
6	35352

Query 6 : MoM Growth and Difference For Orders

```sql
-- MoM Growth and Difference For Orders
SELECT
    Monthly_sale,
    Total_Orders, LAG(Total_Orders) OVER (ORDER BY Monthly_sale) As Previous_Month_Orders,
    (Total_Orders-LAG(Total_Orders) OVER (ORDER BY Monthly_sale)) as MoM_Difference,
        ROUND(
        ((Total_Orders-LAG(Total_Orders) OVER (ORDER BY Monthly_sale)) /
        LAG(Total_Orders) OVER (ORDER BY Monthly_sale)) * 100, 2
    ) AS MoM_Growth_Percentage_Orders,
    CONCAT(
        IFNULL(CONCAT(
            IF((Total_Orders - LAG(Total_Orders) OVER (ORDER BY Monthly_sale)) >= 0, '+ ', '- '),
            ROUND(
                ABS(
                    ((Total_Orders - LAG(Total_Orders) OVER (ORDER BY Monthly_sale)) /
                    LAG(Total_Orders) OVER (ORDER BY Monthly_sale)) * 100
                ), 2
            ), '%'
        ), 'N/A'),
        ' | ',
        IFNULL(CONCAT(
            IF((Total_Orders - LAG(Total_Orders) OVER (ORDER BY Monthly_sale)) >= 0, '+ ', '- '),
            ROUND(ABS((Total_Orders - LAG(Total_Orders) OVER (ORDER BY Monthly_sale))), 2)
        ), 'N/A'),
        ' Orders Vs LM'
    ) AS MoM_Comparison
FROM (
    SELECT
        MONTH(transaction_date) AS Monthly_sale,
        ROUND(COUNT(transaction_qty)) AS Total_Orders
    FROM
        coffee_shop_sale
    GROUP BY
        Monthly_sale
) AS Monthly_Orders;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Monthly_sale	Total_Orders	Previous_Month_Orders	MoM_Difference	MoM_Growth_Percentage_Orders	MoM_Comparison
1	17314	NULL	NULL	NULL	N/A | N/A Orders Vs LM
2	16359	17314	-955	-5.52	- 5.52% | - 955 Orders Vs LM
3	21229	16359	4870	29.77	+ 29.77% | + 4870 Orders Vs LM
4	25335	21229	4106	19.34	+ 19.34% | + 4106 Orders Vs LM
5	33527	25335	8192	32.33	+ 32.33% | + 8192 Orders Vs LM
6	35352	33527	1825	5.44	+ 5.44%

+ 32.33% | + 8192 Orders Vs LM

Query 7 : MoM Growth and Difference For Orders

```sql
-- MoM Growth and Difference For Quantity
SELECT
    MONTH(transaction_date) AS Monthly_Qty,
    CONCAT(ROUND(SUM(transaction_qty) / 1000, 2), 'K Qty') AS Total_Qty,
    LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date)) AS Previous_Month_Qty,
    (SUM(transaction_qty) - LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date))) AS MoM_Difference,
    ROUND(
        ((SUM(transaction_qty) - LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date))) /
        LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date))) * 100, 2
    ) AS MoM_Growth_Percentage,
    CONCAT(
        IFNULL(CONCAT(
            IF((SUM(transaction_qty) - LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date))) >= 0, '+ ', '- '),
            ROUND(
                ABS(
                    ((SUM(transaction_qty) - LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date))) /
                    LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date))) * 100
                ), 2
            ), '%'
        ), 'N/A'),
        ' | ',
        IFNULL(CONCAT(
            IF((SUM(transaction_qty) - LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date))) >= 0, '+ ', '- '),
            ROUND(ABS((SUM(transaction_qty) - LAG(SUM(transaction_qty)) OVER (ORDER BY MONTH(transaction_date)))) / 1000, 2)
        ), 'N/A'),
        'K Qty Vs LM'
    ) AS Mom_Formatted
FROM
    coffee_shop_sale
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Monthly_Qty	Total_Qty	Previous_Month_Qty	MoM_Difference	MoM_Growth_Percentage	Mom_Formatted
1	24.87K Qty	NULL	NULL	NULL	N/A | N/AK Qty Vs LM
2	23.55K Qty	24870	-1320	-5.31	- 5.31% | - 1.32K Qty Vs LM
3	30.41K Qty	23550	6856	29.11	+ 29.11% | + 6.86K Qty Vs LM
4	36.47K Qty	30406	6063	19.94	+ 19.94% | + 6.06K Qty Vs LM
5	48.23K Qty	36469	11764	32.26	+ 32.26% | + 11.76K Qty Vs LM
6	50.94K Qty	48233	2709	5.62	+ 5.62% | + 2.71K Qty Vs LM

Query 8 : Sale By Store Location For March

```sql
-- Sale By Store Location for March
select store_location, sum(transaction_qty*unit_price) as Total_Sale
from coffee_shop_sale
where month(transaction_date) = 3
group by store_location;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

store_location	Total_Sale
Lower Manhattan	32888.68
Hell's Kitchen	33110.57
Astoria	32835.43

Query 8 : Calculate Daily Sales Matrix and Include the average Daily Sale for the month

```sql
-- Calculate daily sales metrics and include the average daily sale for the month
WITH DailySales AS (
    SELECT
        DAY(transaction_date) AS Day,
        SUM(transaction_qty * unit_price) AS Total_Sale_Per_Day,
        SUM(transaction_qty) AS Total_Qty,
        COUNT(transaction_id) AS Total_Orders
    FROM coffee_shop_sale
    WHERE MONTH(transaction_date) = 3
    GROUP BY DAY(transaction_date)
),
MonthlyAvg AS (
    SELECT
        AVG(Total_Sale_Per_Day) AS Avg_Sale_Per_Day
    FROM DailySales
)
-- Main query to include daily metrics, average sales, and classification
SELECT
    Day,
    Total_Sale_Per_Day,
    Total_Qty,
    Total_Orders,
    (SELECT Avg_Sale_Per_Day FROM MonthlyAvg) AS Avg_Sale_Per_Day,
    CASE
        WHEN Total_Sale_Per_Day > (SELECT Avg_Sale_Per_Day FROM MonthlyAvg) THEN 'Above Average'
        WHEN Total_Sale_Per_Day < (SELECT Avg_Sale_Per_Day FROM MonthlyAvg) THEN 'Below Average'
        ELSE 'Average'
    END AS Sales_Comparison
FROM DailySales
ORDER BY Day;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Day	Total_Sale_Per_Day	Total_Qty	Total_Orders	Avg_Sale_Per_Day	Sales_Comparison
1	3040.25	968	661	3188.215484	Below Average
2	2996.05	963	673	3188.215484	Below Average
3	3155.15	1010	710	3188.215484	Below Average
4	2781.90	897	624	3188.215484	Below Average
5	2945.30	952	675	3188.215484	Below Average
6	2618.05	836	587	3188.215484	Below Average
7	2803.50	883	629	3188.215484	Below Average
8	3523.26	1039	742	3188.215484	Above Average
9	3459.97	983	722	3188.215484	Above Average
10	3441.58	1051	750	3188.215484	Above Average
11	3211.65	1002	689	3188.215484	Above Average
12	3088.33	947	667	3188.215484	Below Average
13	3627.65	1110	730	3188.215484	Above Average
14	3312.66	918	691	3188.215484	Above Average
15	3338.03	1000	721	3188.215484	Above Average
16	3386.11	1037	758	3188.215484	Above Average
17	3181.75	942	693	3188.215484	Below Average
18	3408.36	1001	731	3188.215484	Above Average
19	3340.03	988	725	3188.215484	Above Average
20	3262.28	988	735	3188.215484	Above Average
21	3209.80	949	650	3188.215484	Above Average
22	3284.11	1014	669	3188.215484	Above Average

Query 9 : Weekend And Weekday Sales

```sql
-- Define the CTE to classify days as Weekend or Weekday
WITH cte AS (
    SELECT
        *,
        CASE
            WHEN WEEKDAY(transaction_date) IN (5, 6) THEN 'Weekend'
            ELSE 'Weekday'
        END AS Weekend_Weekday,
        MONTHNAME(transaction_date) AS Month_Name
    FROM coffee_shop_sale
),
-- Calculate total sales for Weekday and Weekend by Month
sales_summary AS (
    SELECT
        Month_Name,
        Weekend_Weekday,
        SUM(transaction_qty * unit_price) AS Total_Sales
    FROM cte
    GROUP BY
        Month_Name,
        Weekend_Weekday
),
-- Pivot the sales data to get Weekday and Weekend sales in separate columns
pivot_sales AS (
    SELECT
        Month_Name,
        COALESCE(SUM(CASE WHEN Weekend_Weekday = 'Weekday' THEN Total_Sales END), 0) AS Weekday_Sale,
        COALESCE(SUM(CASE WHEN Weekend_Weekday = 'Weekend' THEN Total_Sales END), 0) AS Weekend_Sale
    FROM sales_summary
    GROUP BY Month_Name
),
-- Calculate Total Sales for each Month
total_sales AS (
    SELECT
        Month_Name,
        SUM(Weekday_Sale + Weekend_Sale) AS Total_Sale
    FROM pivot_sales
    GROUP BY Month_Name
)
-- Final output combining Weekday Sale, Weekend Sale, and Total Sale
SELECT
    ps.Month_Name AS Month_Name,
    ps.Weekday_Sale,
    ps.Weekend_Sale,
    ts.Total_Sale
FROM pivot_sales ps
JOIN total_sales ts
    ON ps.Month_Name = ts.Month_Name
ORDER BY
    STR_TO_DATE(CONCAT('01 ', ps.Month_Name), '%d %M')  -- Ordering by the chronological month
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month_Name	Weekday_Sale	Weekend_Sale	Total_Sale
April	79592.51	39348.57	118941.08
February	54002.67	22142.52	76145.19
January	58513.11	23164.63	81677.74
June	121484.08	45001.80	166485.88
March	73367.33	25467.35	98834.68
May	116627.84	40099.92	156727.76

Query 10 : Sales With Respective Product Category

```
294    -- Sales With Respective Product Category
295 •  select product_category, Sum(transaction_qty*unit_price) as Total_Sale
296    from coffee_shop_sale
297    where month(transaction_date) = 3
298    group by product_category
299    order by Total_Sale desc
300    limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_category	Total_Sale
Coffee	38303.60
Tea	27910.65
Bakery	11902.58
Drinking Chocolate	10253.50
Coffee beans	5256.20

Query 11 : Sale With Respective Product Type

```
302    -- Sales With Respective Product Type
303 •  select product_type, Sum(transaction_qty*unit_price) as Total_Sale
304    from coffee_shop_sale
305    where month(transaction_date) = 3
306    group by product_type
307    order by Total_Sale desc
308    limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_type	Total_Sale
Barista Espresso	13078.20
Brewed Chai tea	11029.65
Hot chocolate	10253.50
Gourmet brewed coffee	9789.10
Brewed Black tea	6875.00