MY SQL QUERIES

CONVERT DATE (transaction_date) COLUMN TO PROPER DATE FORMAT

```
UPDATE coffee_shop_sales

SET transaction_date = STR_TO_DATE(transaction_date, '%d-%m-%Y');
```

ALTER DATE (transaction_date) COLUMN TO DATE DATA TYPE

```
ALTER TABLE coffee_shop_sales

MODIFY COLUMN transaction_date DATE;
```

CONVERT TIME (transaction_time) COLUMN TO PROPER DATE FORMAT

```
UPDATE coffee_shop_sales

SET transaction_time = STR_TO_DATE(transaction_time, '%H:%i:%s');
```

ALTER TIME (transaction_time) COLUMN TO DATE DATA TYPE

```
ALTER TABLE coffee_shop_sales

MODIFY COLUMN transaction_time TIME;
```

DATA TYPES OF DIFFERENT COLUMNS

```
DESCRIBE coffee_shop_sales;
```

CHANGE COLUMN NAME `ï»¿transaction_id` to transaction_id

```
ALTER TABLE coffee_shop_sales

CHANGE COLUMN `ï»¿transaction_id` transaction_id INT;
```

TOTAL SALES

```
SELECT ROUND(SUM(unit_price * transaction_qty)) as Total_Sales

FROM coffee_shop_sales

WHERE MONTH(transaction_date) = 5 -- for month of (CM-May)
```

TOTAL SALES KPI - MOM DIFFERENCE AND MOM GROWTH

```
SELECT
    MONTH(transaction_date) AS month,
    ROUND(SUM(unit_price * transaction_qty)) AS total_sales,
    (SUM(unit_price * transaction_qty) - LAG(SUM(unit_price * transaction_qty), 1)
    OVER (ORDER BY MONTH(transaction_date))) / LAG(SUM(unit_price * transaction_qty), 1)
    OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) IN (4, 5) -- for months of April and May
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);
```

TOTAL ORDERS

```
SELECT COUNT(transaction_id) as Total_Orders
FROM coffee_shop_sales
WHERE MONTH (transaction_date)= 5 -- for month of (CM-May)
```

TOTAL ORDERS KPI - MOM DIFFERENCE AND MOM GROWTH

```
SELECT
    MONTH(transaction_date) AS month,
    ROUND(COUNT(transaction_id)) AS total_orders,
    (COUNT(transaction_id) - LAG(COUNT(transaction_id), 1)
    OVER (ORDER BY MONTH(transaction_date))) / LAG(COUNT(transaction_id), 1)
    OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
FROM
    coffee_shop_sales
WHERE
```

```
    MONTH(transaction_date) IN (4, 5) -- for April and May
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);
```

TOTAL QUANTITY SOLD

```
SELECT SUM(transaction_qty) as Total_Quantity_Sold
FROM coffee_shop_sales
WHERE MONTH(transaction_date) = 5 -- for month of (CM-May)
```

TOTAL QUANTITY SOLD KPI - MOM DIFFERENCE AND MOM GROWTH

```
SELECT
    MONTH(transaction_date) AS month,
    ROUND(SUM(transaction_qty)) AS total_quantity_sold,
    (SUM(transaction_qty) - LAG(SUM(transaction_qty), 1)
    OVER (ORDER BY MONTH(transaction_date))) / LAG(SUM(transaction_qty), 1)
    OVER (ORDER BY MONTH(transaction_date)) * 100 AS mom_increase_percentage
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) IN (4, 5)   -- for April and May
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);
```

CALENDAR TABLE – DAILY SALES, QUANTITY and TOTAL ORDERS

```
SELECT
    SUM(unit_price * transaction_qty) AS total_sales,
    SUM(transaction_qty) AS total_quantity_sold,
    COUNT(transaction_id) AS total_orders
FROM
    coffee_shop_sales
WHERE
    transaction_date = '2023-05-18'; --For 18 May 2023
```

SALES TREND OVER PERIOD

```
SELECT AVG(total_sales) AS average_sales
FROM (
    SELECT
        SUM(unit_price * transaction_qty) AS total_sales
    FROM
        coffee_shop_sales
                WHERE
        MONTH(transaction_date) = 5  -- Filter for May
    GROUP BY
        transaction_date
) AS internal_query;
```

DAILY SALES FOR MONTH SELECTED

```
SELECT
    DAY(transaction_date) AS day_of_month,
    ROUND(SUM(unit_price * transaction_qty),1) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5  -- Filter for May
GROUP BY
    DAY(transaction_date)
ORDER BY
```

```
    DAY(transaction_date);
```

COMPARING DAILY SALES WITH AVERAGE SALES – IF GREATER THAN “ABOVE AVERAGE” and LESSER THAN “BELOW AVERAGE”

```
SELECT
    day_of_month,
    CASE
        WHEN total_sales > avg_sales THEN 'Above Average'
        WHEN total_sales < avg_sales THEN 'Below Average'
        ELSE 'Average'
    END AS sales_status,
    total_sales
FROM (
    SELECT
        DAY(transaction_date) AS day_of_month,
        SUM(unit_price * transaction_qty) AS total_sales,
        AVG(SUM(unit_price * transaction_qty)) OVER () AS avg_sales
    FROM
        coffee_shop_sales
    WHERE
        MONTH(transaction_date) = 5  -- Filter for May
    GROUP BY
        DAY(transaction_date)
) AS sales_data
ORDER BY
    day_of_month;
```

SALES BY WEEKDAY / WEEKEND:

```
SELECT
    CASE
        WHEN DAYOFWEEK(transaction_date) IN (1, 7) THEN 'Weekends'
        ELSE 'Weekdays'
    END AS day_type,
    ROUND(SUM(unit_price * transaction_qty),2) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5  -- Filter for May
GROUP BY
    CASE
        WHEN DAYOFWEEK(transaction_date) IN (1, 7) THEN 'Weekends'
        ELSE 'Weekdays'
    END;
```

SALES BY STORE LOCATION

```
SELECT
                store_location,
                SUM(unit_price * transaction_qty) as Total_Sales
FROM coffee_shop_sales
WHERE
                MONTH(transaction_date) =5
GROUP BY store_location
ORDER BY        SUM(unit_price * transaction_qty) DESC
```

SALES BY PRODUCT CATEGORY

```
SELECT
                product_category,
                ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
FROM coffee_shop_sales
WHERE
                MONTH(transaction_date) = 5
GROUP BY product_category
```

```
ORDER BY SUM(unit_price * transaction_qty) DESC
```

SALES BY PRODUCTS (TOP 10)

```
SELECT
        product_type,
        ROUND(SUM(unit_price * transaction_qty),1) as Total_Sales
FROM coffee_shop_sales
WHERE
        MONTH(transaction_date) = 5
GROUP BY product_type
ORDER BY SUM(unit_price * transaction_qty) DESC
LIMIT 10
```

SALES BY DAY | HOUR

```
SELECT
  ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales,
  SUM(transaction_qty) AS Total_Quantity,
  COUNT(*) AS Total_Orders
FROM
  coffee_shop_sales
WHERE
  DAYOFWEEK(transaction_date) = 3 -- Filter for Tuesday (1 is Sunday, 2 is Monday, ..., 7 is Saturday)
  AND HOUR(transaction_time) = 8 -- Filter for hour number 8
  AND MONTH(transaction_date) = 5; -- Filter for May (month number 5)
```

TO GET SALES FROM MONDAY TO SUNDAY FOR MONTH OF MAY

```
SELECT
  CASE
    WHEN DAYOFWEEK(transaction_date) = 2 THEN 'Monday'
    WHEN DAYOFWEEK(transaction_date) = 3 THEN 'Tuesday'
    WHEN DAYOFWEEK(transaction_date) = 4 THEN 'Wednesday'
```

```
        WHEN DAYOFWEEK(transaction_date) = 5 THEN 'Thursday'
        WHEN DAYOFWEEK(transaction_date) = 6 THEN 'Friday'
        WHEN DAYOFWEEK(transaction_date) = 7 THEN 'Saturday'
        ELSE 'Sunday'
    END AS Day_of_Week,
    ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5 -- Filter for May (month number 5)
GROUP BY
    CASE
        WHEN DAYOFWEEK(transaction_date) = 2 THEN 'Monday'
        WHEN DAYOFWEEK(transaction_date) = 3 THEN 'Tuesday'
        WHEN DAYOFWEEK(transaction_date) = 4 THEN 'Wednesday'
        WHEN DAYOFWEEK(transaction_date) = 5 THEN 'Thursday'
        WHEN DAYOFWEEK(transaction_date) = 6 THEN 'Friday'
        WHEN DAYOFWEEK(transaction_date) = 7 THEN 'Saturday'
        ELSE 'Sunday'
    END;
```

TO GET SALES FOR ALL HOURS FOR MONTH OF MAY

```
SELECT
    HOUR(transaction_time) AS Hour_of_Day,
    ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5 -- Filter for May (month number 5)
GROUP BY
    HOUR(transaction_time)
```

```
ORDER BY
    HOUR(transaction_time);
```