SQL PROJECT

(Coffee shop sales)

[By Krish Kumar Mishra]

"Hello, I’m Krish Mishra. In this SQL project, I used moderate to complex queries to solve sales problems for a coffee café. The results help the café owner track sales issues and find solutions."

1. Calculate the total sales for each respective month ?

```sql
SELECT
    SUM(unit_price * transaction_qty) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 1;
```

You can select particular month to see total sales

Result Grid | Filter Rows:

total_sales
81677.73999999928

2.Calculate the total sales increase or decrease month on month?

```sql
SELECT
    MONTH(transaction_date) AS month,
    ROUND(SUM(transaction_qty * unit_price)) AS total_sales,
    round((SUM(transaction_qty * unit_price) -
        LAG(SUM(transaction_qty * unit_price), 1) OVER (ORDER BY MONTH(transaction_date))
    ) / LAG(SUM(transaction_qty * unit_price), 1) OVER (ORDER BY MONTH(transaction_date)) * 100 ,2
    )AS mom_increase
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) IN (2,3)
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);
```

You can change month to know sales status

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

month	total_sales	mom_increase
2	76145	NULL
3	98835	29.8

3. Calculate the total order of respective month?

```
SELECT
    count(transaction_id) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5;
```

Result Grid | Filter Rows:

	total_sales
▶	33527

4. Calculate the month on month increase or decrease in number of sales

```sql
SELECT
    MONTH(transaction_date) AS month,
    COUNT(transaction_id) AS number_of_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_orders
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) IN (4, 5)  -- Replace with the desired months
GROUP BY
    MONTH(transaction_date)
ORDER BY
    MONTH(transaction_date);
```

Result Grid | Filter Rows: | Export:

month	number_of_orders	mom_increase_orders
4	25335	NULL
5	33527	32.3347

5. Calculate the total quantity sold for respective month?

```sql
SELECT
    sum(transaction_qty) AS total_qty_sold
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5;
```

Result Grid

total_qty_sold
48233

6. Segment sales data into weekdays and weekends ?

```sql
SELECT
    CASE
        WHEN DAYOFWEEK(transaction_date) IN (1 , 7) THEN 'weekends'
        ELSE 'weekday'
    END AS day_type,
    SUM(unit_price * transaction_qty) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 1
GROUP BY (day_type);
```

Result Grid | Filter Rows:

day_type	total_sales
weekends	23164.63000000004
weekday	58513.10999999929

7. Sales data by different store locations?

```sql
SELECT
    store_location,
    CONCAT(ROUND(SUM(unit_price * transaction_qty) / 1000,
                2),
            ' k') AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 6
GROUP BY store_location
ORDER BY total_sales DESC;
```

Result Grid | Filter Rows:

store_location	total_sales
Hell's Kitchen	56.96 k
Astoria	55.08 k
Lower Manhattan	54.45 k

8. MoM sales increase or decrease for each store locations to identify trends?

```sql
SELECT
    store_location,  -- Assuming there is a column for store location
    MONTH(transaction_date) AS month,
    SUM(transaction_qty) AS total_qty_sold,
    (SUM(transaction_qty) -
        LAG(SUM(transaction_qty), 1) OVER (PARTITION BY store_location ORDER BY MONTH(transaction_date))
    ) / NULLIF(LAG(SUM(transaction_qty), 1) OVER (PARTITION BY store_location ORDER BY MONTH(transaction_date)), 0) * 100
    AS mom_increase_qty
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) IN (4, 5)  -- Replace with the desired months
GROUP BY
    store_location, MONTH(transaction_date)
ORDER BY
    store_location, MONTH(transaction_date);
```

Result Grid | Filter Rows: | Export:

store_location	month	total_qty_sold	mom_increase_qty
Astoria	4	12026	NULL
Astoria	5	16114	33.9930
Hell's Kitchen	4	12194	NULL
Hell's Kitchen	5	15944	30.7528
Lower Manhattan	4	12249	NULL
Lower Manhattan	5	16175	32.0516

9. Sales of each day in selected month?

```sql
SELECT
    SUM(transaction_qty * unit_price) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 3
GROUP BY transaction_date;
```

Result Grid | Filter Rows:

total_sales
3040.2500000000005
2996.049999999998
3155.1499999999983
2781.8999999999983
2945.3000000000006
2618.0499999999993
2803.5000000000005
3523.2599999999966
3459.9699999999975
3441.5799999999977
3211.649999999996
3088.3299999999986
3627.6499999999983
3312.6599999999967
3338.0299999999957
3386.1099999999988
3181.749999999999
3408.359999999999
3340.029999999998
3262.279999999998
3209.799999999998
3284.1099999999974
3361.1299999999987
3586.1999999999975
3380.949999999998
3310.8299999999967
3674.3499999999976
2792.549999999998
2492.000000000001
2932.819999999997
2888.0799999999977

10. Avg sales of each selected month?

```
SELECT
    AVG(total_sales)
FROM
    (SELECT
        SUM(transaction_qty * unit_price) AS total_sales
    FROM
        coffee_shop_sales
    WHERE
        MONTH(transaction_date) = 4
    GROUP BY transaction_date) AS inner_query;
```

Result Grid | Filter Rows:

	AVG(total_sales)
▶	3964.7026666666643

11. Sales performance across different product category?

```sql
select product_category, round(sum(unit_price*transaction_qty)) as total_sales
from coffee_shop_sales
group by product_category;
```

Result Grid | Filter Rows:

product_category	total_sales
Coffee	269952
Tea	196406
Drinking Chocolate	72416
Bakery	82316
Flavours	8409
Loose Tea	11214
Coffee beans	40085
Packaged Chocolate	4408
Branded	13607

12. Total sales and total quantity order at particular month ,day , time ?

```sql
SELECT
    SUM(unit_price * transaction_qty) AS total_sales,
    SUM(transaction_qty) AS total_qty_sold
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5
        AND HOUR(transaction_time) = 7
        AND DAYOFWEEK(transaction_date) = 2;
```

Result Grid | Filter Rows:

total_sales	total_qty_sold
2266.209999999999	692

13. Which top 5 time(hour) is more busy respect to sales ?

```sql
SELECT
    HOUR(transaction_time),
    SUM(unit_price * transaction_qty) AS total_sales
FROM
    coffee_shop_sales
WHERE
    MONTH(transaction_date) = 5
GROUP BY HOUR(transaction_time)
ORDER BY total_sales desc
limit 5;
```

Result Grid | Filter Rows:

HOUR(transaction_time)	total_sales
10	19639.13000000001
9	19145.270000000022
8	18822.31000000003
7	14350.680000000037
11	10312.160000000014