SQL PROJECT

(Coffee shop sales)



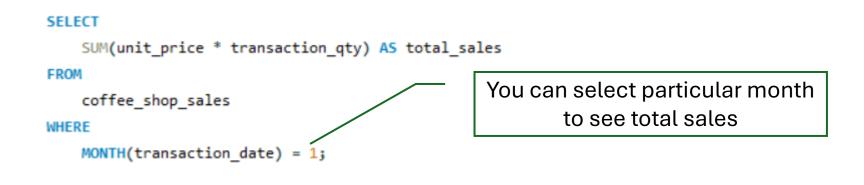
[By Krish Kumar Mishra]

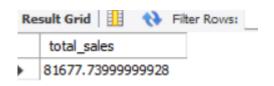




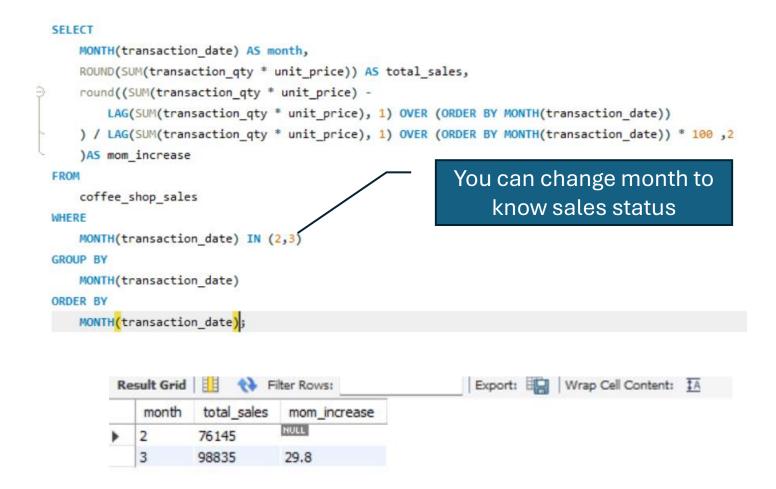
"Hello, I'm Krish Mishra. In this SQL project, I used moderate to complex queries to solve sales problems for a coffee café. The results help the café owner track sales issues and find solutions."

1. Calculate the total sales for each respective month?





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



3. Calculate the total order of respective month?

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

```
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);
```

Re	sult Grid	Filter Rows	5: E	Export:
	month	number_of_orders	mom_increase_orders	
•	4	25335	NULL	,
	5	33527	32.3347	

5. Calculate the total quantity sold for respective month?

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



6. Segment sales data into weekdays and weekends?

```
SELECT

CASE

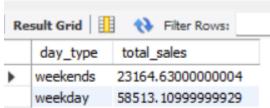
WHEN DAYOFWEEK(transaction_date) IN (1 , 7) THEN 'weekends'
ELSE 'weekday'
END AS day_type,
SUM(unit_price * transaction_qty) AS total_sales

FROM

coffee_shop_sales
WHERE

MONTH(transaction_date) = 1

GROUP BY (day_type);
```



7. Sales data by different store locations?

```
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		
	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?

```
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);
```

R	Result Grid 11 🛟 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?

SELECT

SUM(transaction_qty * unit_price) AS total_sales

FROM

coffee_shop_sales

WHERE

MONTH(transaction_date) = 3

GROUP BY transaction_date;

Re	sult Grid 🔠 🙌 Filter Rows
	total_sales
•	3040.2500000000005
	2996.04999999998
	3155.149999999983
	2781.899999999983
	2945.300000000006
	2618.049999999993
	2803.5000000000005
	3523.259999999966
	3459.969999999975
	3441.579999999977
	3211.64999999999
	3088.329999999986
	3627.649999999983
	3312.659999999967
	3338.029999999957
	3386.109999999988
	3181.74999999999
	3408.35999999999
	3340.02999999998
	3262.279999999998
	3209.79999999998
	3284, 1099999999974
	3361.1299999999987
	3586.199999999975
	3380.94999999998
	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;
             AVG(total_sales)
               3964.7026666666643
```

11. Sales performance across different product category?

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

	product_category	total_sales
٠	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?

```
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_qty_sold

692

total_sales

2266.209999999999

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

limit 5;

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

Re	Result Grid Filter Rows:		
	HOUR(transaction_time)	total_sales	
•	10	19639.13000000001	
	9	19145.270000000022	
	8	18822.31000000003	
	7	14350.680000000037	
	11	10312.160000000014	