

# Coffee Shop Sales




## KPI Requirements:

### 1.Total Sales Analysis

Calculate the sale for each month and mom difference and mom diff pct.

```
select month(transaction_date) as month,
       round(sum(unit_price * transaction_qty)) as total_sales,
       Round (sum(unit_price * transaction_qty)- lag(sum(unit_price *
transaction_qty),1)
       over(order by month(transaction_date))) as MoM_diff,
       (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_pct
from
  coffee_shop_sales
where
  month(transaction_date) in (4,5)---for may month
group by
  month(transaction_date)
order by
```

month(transaction\_date)

Result Grid    Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 				
	month	total_sales	MoM_diff	MoM_pct
▶	4	118941	NULL	NULL
	5	156728	37787	31.769242384551315

## 2. Total Order Analysis

Calculate the total orders for each month and Mom diff and Mom pct from May.

SELECT

month(transaction\_date) AS month,

COUNT(transaction\_id) AS total\_orders,

COUNT(transaction\_id) - LAG(COUNT(transaction\_id), 1) OVER  
(ORDER BY month(transaction\_date)) AS MoM\_diff,

ROUND(

(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, 2

) AS MoM\_pct

FROM

coffee\_shop\_sales

WHERE




month(transaction\_date) IN (4, 5)

GROUP BY

month(transaction\_date)

ORDER BY

month(transaction\_date);

Result Grid    Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 				
	month	total_orders	MoM_diff	MoM_pct
▶	4	25335	NULL	NULL
	5	33527	8192	32.33

### 3.Total quantity sold

Calculate the quantity for each month, Mom diff and Mom pct.

SELECT

month(transaction\_date) AS month,

ROUND(SUM(transaction\_qty)) AS total\_qty,

SUM(transaction\_qty) - LAG(SUM(transaction\_qty), 1) OVER  
(ORDER BY month(transaction\_date)) AS MoM\_diff,

ROUND(

((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, 2

) AS MoM\_pct

FROM

coffee\_shop\_sales

WHERE




month(transaction\_date) IN (4, 5)

GROUP BY

month(transaction\_date)

ORDER BY

month(transaction\_date);

Result Grid    Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 				
	month	total_qty	MoM_diff	MoM_pct
▶	4	36469	NULL	NULL
	5	48233	11764	32.26

#### 4. Sales analysis for weekdays and weekends.

Select

case when dayofweek(transaction\_date) in (1,7) then 'Weekend'

Else 'Weekdays'

end as Day\_type,

concat(round(sum(unit\_price \* transaction\_qty)/1000,1),'k') as  
total\_sales

from coffee\_shop\_sales

where

month(transaction\_date) = 5

group by

case when dayofweek(transaction\_date) in (1,7) then  
'Weekend'

Else 'Weekdays'

End

Result Grid		Filter Rows:	
	Day_type	total_sales	
▶	Weekdays	116.6k	
	Weekend	40.1k	

## 5. Total sales, quantity, orders in sales location

Select store\_location,

concat(round(Sum(unit\_price \* transaction\_qty)/1000,1),'k') as total\_sales,

concat(round(sum(transaction\_qty)/1000,1),'k') as total\_qty,

concat(round(count(transaction\_id)/1000,1), 'k') as total\_orders

from coffee\_shop\_sales

where month(transaction\_date) = 5

group by

store\_location;

Result Grid		Filter Rows:		Export:	Wrap Cell Content:
	store_location	total_sales	total_qty	total_orders	
▶	Lower Manhattan	51.7k	16.2k	10.8k	
	Hell's Kitchen	52.6k	15.9k	11.3k	
	Astoria	52.4k	16.1k	11.5k	

## 6.Daily sales analysis with average line

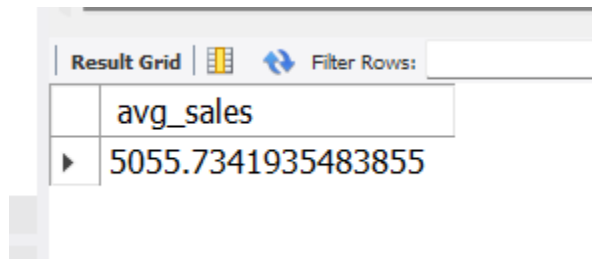
select avg(total\_sales) as avg\_sales

from

(

select

```
(sum(unit_price * transaction_qty)) as total_sales  
from coffee_shop_sales  
where month(transaction_date) = 5  
group by transaction_date
```






The screenshot shows a database interface with a 'Result Grid' tab. It contains a single row with the column name 'avg\_sales' and the value '5055.7341935483855'. There is a 'Filter Rows' button and a search input field at the top.

avg_sales
5055.7341935483855

### 7.Daily sales for a month

```
select day(transaction_date) as date_no,  
       round(sum(unit_price * transaction_qty),2) as total_sales  
from coffee_shop_sales  
where month(transaction_date) = 5  
group by transaction_date  
order by transaction_date
```

Result Grid     Filter Rows: <input type="text"/>		
	day_no	total_sales
▶	1	4731.45
	2	4625.5
	3	4714.6
	4	4589.7
	5	4701
	6	4205.15
	7	4542.7
	8	5604.21
	9	5100.97
	10	5256.33
	11	4850.06
	12	4681.13

Result 18 × 

## 8. Daily sales which is above average and below average

With cte as

(

SELECT DAY(transaction\_date) AS date\_no,

ROUND(SUM(unit\_price \* transaction\_qty), 2) AS total\_sales,

AVG(SUM(unit\_price \* transaction\_qty)) OVER () AS avg\_sales

FROM coffee\_shop\_sales

WHERE MONTH(transaction\_date) = 5

GROUP BY DAY(transaction\_date)

)

select date\_no, total\_sales,

case when total\_sales > avg\_sales then 'Above average' else 'Below average' end as Average\_line

from cte

ORDER BY date\_no;

date_no	total_sales	Average_line
1	4731.45	Below average
2	4625.5	Below average
3	4714.6	Below average
4	4589.7	Below average
5	4701	Below average
6	4205.15	Below average
7	4542.7	Below average
8	5604.21	Above average
9	5100.97	Above average
10	5256.33	Above average
11	4850.06	Below average
12	4681.13	Below average

## 9. Sales analysis by product category

select product\_category,

round(sum(unit\_price \* transaction\_qty),2) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date) =5

group by product\_category



Result Grid			Filter Rows:
	product_category	total_sales	
▶	Coffee	60362.85	
	Drinking Chocolate	16319.75	
	Tea	44539.85	
	Bakery	18565.52	
	Packaged Chocolate	981.09	
	Loose Tea	2395.15	
	Flavours	1905.6	
	Coffee beans	8768.95	
	Branded	2889	

## 10.Top 10 products by sales

```

select product_type,
       round(sum(unit_price * transaction_qty),2) as total_sales
from coffee_shop_sales
where month(transaction_date) =5
group by product_type
order by round(sum(unit_price * transaction_qty),2) desc
limit 10;

```

Result Grid		Filter Rows:	Ex
product_type	total_sales		
Barista Espresso	20423.75		
Brewed Chai tea	17427.35		
Hot chocolate	16319.75		
Gourmet brewed coffee	15559.2		
Brewed herbal tea	10930		
Brewed Black tea	10778		
Premium brewed coffee	8739.2		
Organic brewed coffee	8350.2		
Scone	8305.28		
Drip coffee	7290.5		

## 11.Sales Analysis by sales for hours

select hour(transaction\_time) as hours,



round(sum(unit\_price \* transaction\_qty),2) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date) =5

group by hour(transaction\_time)

order by hour(transaction\_time)

result Grid     Filter Rows:	
hours	total_sales
6	4912.93
7	14350.68
8	18822.31
9	19145.27
10	19639.13
11	10312.16
12	8869.79
13	9379.21
14	9057.66
15	9525.15
16	9154.31
17	8966.85

## 12. Sales in each day name

select

case when dayofweek(transaction\_date) = 1 then 'Sunday'  
 case when dayofweek(transaction\_date) = 2 then 'Monday'  
 case when dayofweek(transaction\_date) = 3 then 'Tuesday'  
 case when dayofweek(transaction\_date) = 4 then 'Wednesday'  
 case when dayofweek(transaction\_date) = 5 then 'Thursday'  
 case when dayofweek(transaction\_date) = 6 then 'Friday'  
 case when dayofweek(transaction\_date) = 7 then 'Saturday'  
 end as Days,

sum(unit\_price \* transaction\_qty) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date)

```

group by case when dayofweek(transaction_date) = 1 then 'Sunday'
          case when dayofweek(transaction_date) = 2 then 'Monday'
          case when dayofweek(transaction_date) = 3 then 'Tuesday'
          case when dayofweek(transaction_date) = 4 then 'Wednesday'
          case when dayofweek(transaction_date) = 5 then 'Thursday'
          case when dayofweek(transaction_date) = 6 then 'Friday'
          case when dayofweek(transaction_date) = 7 then 'Saturday'
end

```

Days	total_sales
Monday	25221.3
Tuesday	25346.99
Wednesday	25464.51
Thursday	20254.08
Friday	20340.96
Saturday	20795.11
Sunday	19304.81

### Functionalities used in this:

- STR\_TO\_DATE
- SUM
- ROUND
- AVG
- LAG
- MONTH()
- DAYOFWEEK()
- DAY()

- CONCAT()
- CASE WHEN
- GROUP BY
- ORDER BY
- MIN/MAX
- SUBQUERY, CTE
- WHERE