



DAILY TREND & DESCRIPTIVE ANALYSIS

WELCOME!

THANK YOU FOR VISITING TO SEE MY FIRST REAL-TIME DATA ANALYSIS PROJECT. I'M EXCITED TO SHARE THE INSIGHTS AND TRENDS I DISCOVERED WHILE ANALYZING CAFE LAZA'S DATA OF MARCH- 2024. YOUR FEEDBACK AND SUPPORT ARE GREATLY APPRECIATED AS I EMBARK ON THIS DATA ANALYSIS JOURNEY!



CONTENT

- 01** **WHAT IS THE OBJECTIVE OF THIS PROJECT.**
- 02** **QUESTION OR STEPS TO COMPLETE THIS PROJECT SUCCESSFULLY.**
- 03** **EXPERIENCE AND LEARNING FROM THE COMPLETED PROJECT.**



OBJECTIVE OF PROJECT

Objective: Descriptive Analysis for Cafe Laza (March Data)

Cafe Laza in Indore wants a descriptive analysis of its data collected in March. The goals are to:

- Understand the performance and trends of the cafe in March.
- Compare March data with the previous month to highlight changes and trends.

This analysis will help Cafe Laza make informed decisions based on the performance and trends observed in March.





QUERIE_1

WHAT WERE THE TOP 10 REVENUE DATES?

```
with common as(  
    select sum(my_amount) total_revenue,date(date) as date,rank()over(order by sum(my_amount)desc)  
    from orders_view  
    inner join amounts_view on orders_view.`invoice no.`=amounts_view.`invoice no.`  
    group by date  
    order by total_revenue desc  
    limit 10)  
select date from common;
```

These the top 10 revenue dates

date
2024-03-18
2024-03-21
2024-03-15
2024-03-12
2024-03-16
2024-03-01
2024-03-28
2024-03-27
2024-03-14
2024-03-22

QUERIE_2

WHAT ARE THE TOP 5 DATES BY ORDERS_QUANTITY



```
with common as(
  select date(date) as date,
  count(date(date)) as day_wise_quantity,dense_rank()over(order by count(date(date)) desc) as ranks
  from orders_view
  group by date(date)
  order by day_wise_quantity desc
  limit 5)
  select date from common;
```

	date
▶	2024-03-18
	2024-03-16
	2024-03-15
	2024-03-21
	2024-03-11

These are the top 5 dates by orders quantity

QUERIE_3

AVERAGE TIME_HOUR OF TOP 10 ORDER QUANTITIES



```
with common_table as(  
    select *, date_format(date,"%Y %M %d") dt,date(date) as Dates, hour(time(date)) as hours  
    from orders_view)  
,common_2 as(  
    select dates,count(dates) quantity,round(avg(hours)) as Avg_Time_Hour  
    from common_table  
    group by Dates  
    order by count(dates) desc  
    limit 10)  
select round(avg(Avg_time_hour)) as Avg_Time  
from common_2;
```

Average Time Hour

Result Grid	
	Avg_Time
17	17

QUERIE_4

AT WHICH WEEK DID THE CAFE HAVE THE MOST SALES RETRIEVE WITH THE TOTAL SALES?



SELECT

```
SUM(my_amount) AS total_per_week,  
CEIL(DAY(date) / 7) AS week
```

FROM

```
orders_view
```

JOIN

```
amounts_view USING (`invoice no.`)
```

GROUP BY CEIL(DAY(date) / 7)

ORDER BY total_per_week **DESC**

LIMIT 5;

Total sales per week

total_per_week	week
23454	3
15986	2
14655	1
14480	4
4310	5

QUERIE_5

AT WHICH WEEK DID THE CAFE HAVE THE MOST ORDER QUANTITIES?



SELECT

```
CEIL(DAY(date) / 7) AS week,  
COUNT(`invoice no.`) AS weekly_quantity_ordered
```

FROM

```
orders_view
```

JOIN

```
amounts_view USING (`invoice no.`)
```

```
GROUP BY CEIL(DAY(date) / 7)
```

```
ORDER BY weekly_quantity_ordered DESC
```

```
LIMIT 5;
```

Total_Quantity per week

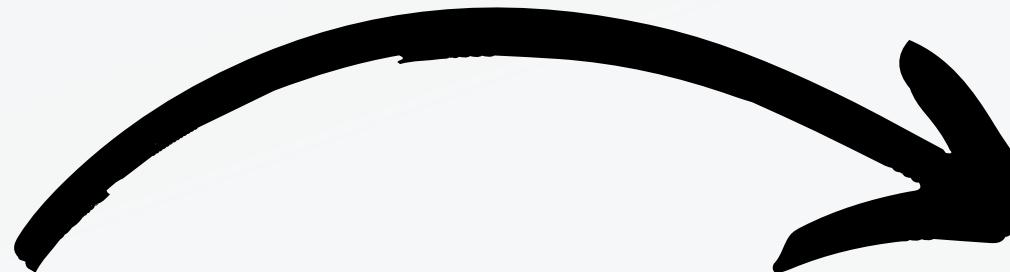
week	weekly_quantity_ordered
3	280
2	207
1	202
4	198
5	71

QUERIE_6

RETRIVE THE TOTAL SALES AND RUNNING TOTAL SALES WEEKLY.



```
select ceil(day(date)/7) as week,sum(my_amount) as total,  
lead(sum(my_amount))over(rows between unbounded preceding and current row )  
as running_total  
from amounts_view join orders_view using(`invoice no.`)  
group by ceil(day(date)/7)  
having total != 10;
```



Result Grid | Filter Rows:

	week	total	running_total
▶	5	4310	14480
	4	14480	23454
	3	23454	15986
	2	15986	14655
	1	14655	NULL

QUERIE_7

HOW DOES THE TREND OF ORDER_QUANTITY AND TOTAL OF EACH DAY LOOK LIKE IN WHOLE MONTH.



SELECT

```
DATE(date) date,  
COUNT(DISTINCT `invoice no.`) AS Trend_eachday_quantity  
SUM(my_amount) AS total_trend
```

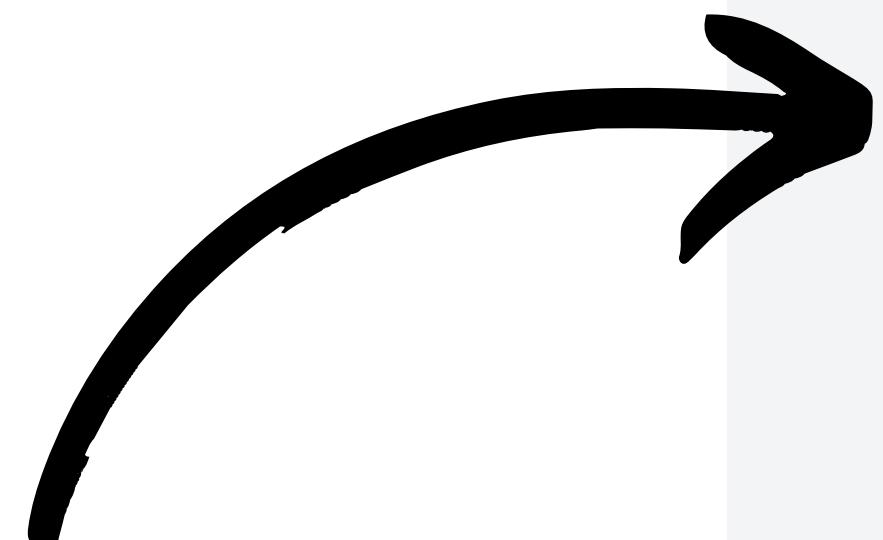
FROM

```
orders_view  
INNER JOIN  
amounts_view USING (`invoice no.`)
```

GROUP BY DATE(date)

having date is not null

ORDER BY DATE(date);



date	Trend_eachday_quantity	total_trend
2024-03-05	25	1885
2024-03-06	32	2220
2024-03-07	27	1690
2024-03-08	26	2360
2024-03-09	23	1330
2024-03-10	20	1670
2024-03-11	39	2375
2024-03-12	36	3291
2024-03-13	30	2200
2024-03-14	33	2760
2024-03-15	42	3910
2024-03-16	47	3220
2024-03-17	36	2368
2024-03-18	52	4704
2024-03-19	36	2517
2024-03-20	26	2080
2024-03-21	41	4655
2024-03-22	38	2550
2024-03-23	34	2395
2024-03-24	20	1610
2024-03-26	31	2150
2024-03-27	38	2805
2024-03-28	37	2970
2024-03-29	38	2465
2024-03-31	33	1845

QUERIE_8

DATE OF MINIMUM TOTAL_AMOUNT AND TOTAL_AMOUNT FROM EVERY WEEK.



```
with common as(
  select date(date) as date, sum(my_amount) as total
  from amounts_view inner join orders_view using(`invoice no.`)
  group by date(date)
  order by date(date))
, common_2 as(
  select *,ceil(day(date)/7) as weeks from common)
,common_3 as(
  select * from common_2
  where total in (select min(total) from common_2
  group by weeks))
,common_4 as(
  select *, dayname(date)as days
  from common_3)
select TOTAL,ifnull(date,0)as date from common_4
WHERE TOTAL != 10;
```

	total	date
	1575	2024-03-03
	1330	2024-03-09
	2080	2024-03-20
	1610	2024-03-24
	1845	2024-03-31



Conclusion

- Week 3 and 2 Went well for sales meanwhile 5 went the lowest.
- On the date 18,16,15 cafe achieved the most sales.
- Cafe Achieved the most orders in week 3 (280) and lowest on week 5 (71).
- On the date 18,16,15,21 got the most orders these 4 four increased the trend overall.





PERSONAL EXPERIENCE

- *USING COMMON TABLE EXPRESSION IS JUST LIKE BREAKING THE PROBLEM INTO PIACES.*
- *ALL THE QUERY AND TECHNICAL PART IS WASTE UNLESS ANLALYST TAKE INTEREST IN THE PROBLEM.*
- *KNEW ABOUT THE CEIL METHOD IN SQL AND HOW TO CALCULATE WEEKS FROM DATE.*
- *SPENDING TIME IN SELECTING USEFULL VARIABLES IS ACTUALLY WORTHY.*