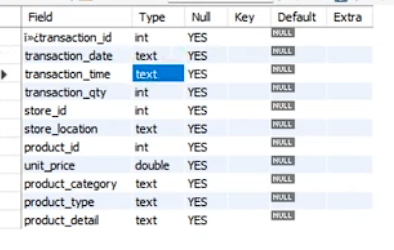
**Coffee Shop Sales**

**Data Cleaning:-**

describe coffee\_shop\_sales;



**Q1 Change transaction\_date & transaction\_time into date and time datatype also rename the ï»¿transaction\_id into transaction\_id ?**

update coffee\_shop\_sales

set transaction\_date = str\_to\_date(transaction\_date, '%d-%m-%Y');

Alter table coffee\_shop\_sales

modify column transaction\_date date;

update coffee\_shop\_sales

set transaction\_time = str\_to\_date(transaction\_time, '%H:%i:%s');

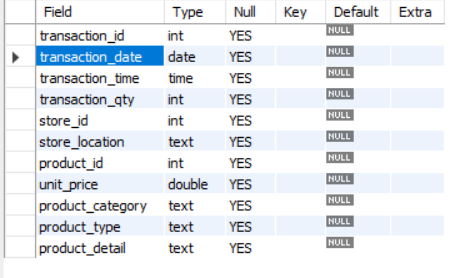
Alter table coffee\_shop\_sales

modify column transaction\_time time;

Alter table coffee\_shop\_sales

change ï»¿transaction\_id transaction\_id int;

describe coffee\_shop\_sales;



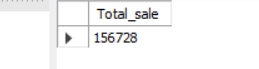
**KPI:-**

**Total Sales**

select round(sum(unit\_price\*transaction\_qty),0) as Total\_sale

from coffee\_shop\_sales

where month(transaction\_date) = 5; -- 5 = may month



**-- % MOM increase and decrease in sales**

**MOM increase and decrease in sales =** (Current Month sales – Previous month sales / Previous Month sales) \* 100

select

month(transaction\_date) as Month\_Number,

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

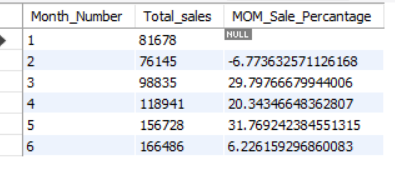
(sum(unit\_price \* transaction\_qty) - lag(sum(unit\_price \* transaction\_qty),1) over (order by month(transaction\_date)))

/ lag(sum(unit\_price \* transaction\_qty),1) over (order by month(transaction\_date)) \*100 MOM\_Sale\_Percantage

from coffee\_shop\_sales

group by month(transaction\_date)

order by month(transaction\_date);

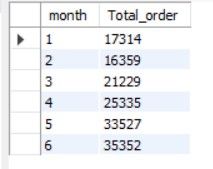


**Q4 total no. of order**

select month(transaction\_date) as month,count(transaction\_id) as Total\_order

from coffee\_shop\_sales

group by month(transaction\_date);



**Q5 display mom increase in order count?**

select

month(transaction\_date) as month,

count(transaction\_id) as Total\_order,

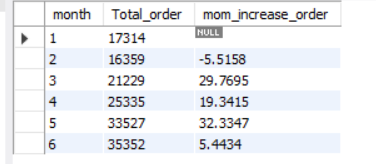
(count(transaction\_id) - lag(count(transaction\_id),1) over (order by month(transaction\_date))) /

lag(count(transaction\_id),1) over (order by month(transaction\_date)) \*100 as mom\_increase\_order

from coffee\_shop\_sales

group by month(transaction\_date)

order by month(transaction\_date);

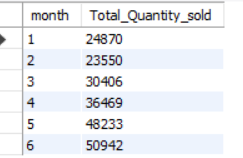


**Q6 total Quantity sold?**

select month(transaction\_date) as month,sum(transaction\_qty) as Total\_Quantity\_sold

from coffee\_shop\_sales

group by month(transaction\_date);



**Q7 MOM increase in order quantity?**

select

month(transaction\_date) as month,

sum(transaction\_qty) as Total\_Quantity\_Sold,

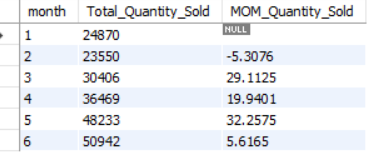
(sum (transaction\_qty) - lag(sum(transaction\_qty),1) over (order by month(transaction\_date))) /

lag(sum (transaction\_qty),1) over (order by month(transaction\_date)) \*100 as MOM\_Quantity\_Sold

from coffee\_shop\_sales

group by month(transaction\_date)

order by month(transaction\_date);



**Q8 find out total sale,total Quantity, total orders for 24-05-2023 date also use "K" thousand behind the result?**

select concat(round(sum(transaction\_qty \* unit\_price)/1000,1), 'K') as Total\_sale,

concat(round(sum(transaction\_qty)/1000,1), 'K') as Total\_Quantity\_Sold,

concat(round(Count(distinct transaction\_id)/1000,1), 'K') as Total\_order

from coffee\_shop\_sales

where transaction\_date = '2023-05-18';



Weekends = Saturday and Sunday also Remember that Sunday = 1 & Saturday =7 on week Number

Weekday = Monday to Friday

**Q9 categerioes the weekday and weekend sales data?**

select

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

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

else 'Weekday'

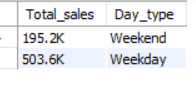
end Day\_type

from coffee\_shop\_sales

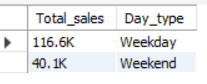
group by Case when dayofweek(transaction\_date) in (1,7) then 'Weekend'

else 'Weekday'

end;



-- in case we need only may month weekday & weekend sale so after from we use where condition(where month(transaction\_date) = 5 here 5 repersent may month number.



**Q10 sales data for different store location?**

select

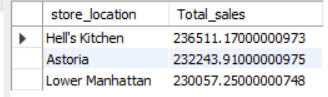
store\_location,

concat(round(sum(transaction\_qty \* unit\_price)/1000,1),’K’) as Total\_sales

from coffee\_shop\_sales

group by store\_location

order by sum(transaction\_qty \* unit\_price) desc;



-- find out where transaction\_date = '2023-05-18' and store\_location = 'Hell''s Kitchen' this condition total\_sales



**Q11 Average sales based on month?**

select

avg(transaction\_qty \* unit\_price) as Avg\_sale

from coffee\_shop\_sales

where month(transaction\_date) = 4;



-- Here for april month output is wrong because Avgerage formula is = sum of total order / count of order

-- but (transaction\_qty \* unit\_price) we do multiplication so no sum for accurate answer firstly we do sum of transaction\_qty \* unit\_price

-- then average.

-- for this we use subquery

select

concat(round(avg(total\_sale)/1000,1),'k') as Avg\_Total\_sale

from

(select sum(transaction\_qty \* unit\_price) as total\_sale

from coffee\_shop\_sales

where month(transaction\_date) = 4

group by transaction\_date) as internal\_query



**Here I notice that if I not alise the subquery name so query not execute**

**Q12- Show dailey sale for may month?**

select

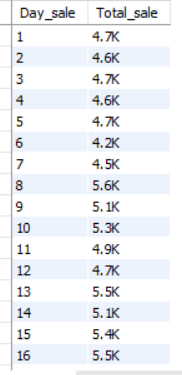
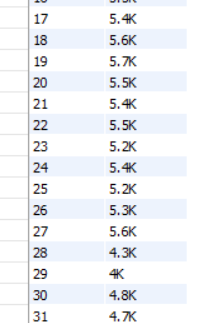
day(transaction\_date) as Day\_sale,

concat(round(sum(transaction\_qty \* unit\_price)/1000,1),'K') as Total\_sale

from coffee\_shop\_sales

where month(transaction\_date) = 5

group by day(transaction\_date);

**Q13- Based on dailey sale and avg sale for may month also if sale id greater then avg sale then above sale and wise versa?**

SELECT day\_of\_month,avg\_sales,total\_sales,

CASE

WHEN total\_sales > avg\_sales THEN 'Above Average'

WHEN total\_sales < avg\_sales THEN 'Below Average'

ELSE 'Average'

END AS sales\_status

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)

ORDER BY day\_of\_month;



Here I got a error msg in avg\_sales if I not use OVER () function error msg is Error Code: 1111. Invalid use of group function

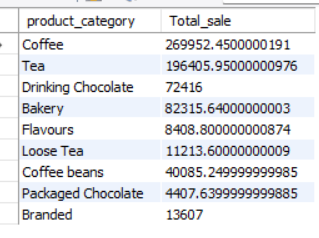
This error because of

**Q14 find out total sale under product\_category?**

select product\_category,sum(unit\_price \* transaction\_qty) as Total\_sale

from coffee\_shop\_sales

group by product\_category;



We also find out the value for specified month eg for may month category by sale

select product\_category,sum(unit\_price \* transaction\_qty) as Total\_sale

from coffee\_shop\_sales

where month(transaction\_date) = 5

group by product\_category

order by sum(unit\_price \* transaction\_qty);



**Q14 find out top 10 total sale under product\_type?**

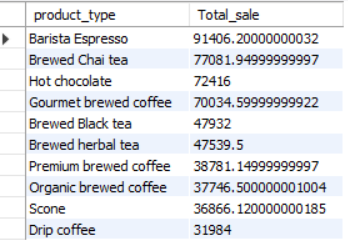
select product\_type,sum(unit\_price \* transaction\_qty) as Total\_sale

from coffee\_shop\_sales

group by product\_type

order by sum(unit\_price \* transaction\_qty) desc

limit 10;



**Q14 Find out the total sale done in each hour for 6 may month?**

select hour(transaction\_time) As Transaction\_Hour,sum(transaction\_qty \* unit\_price) as Total\_sale

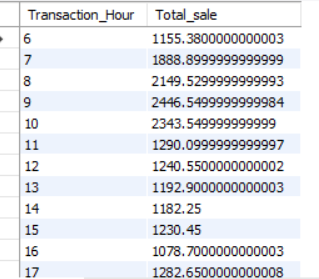
from coffee\_shop\_sales

where month(transaction\_date) = 5

and dayofweek(transaction\_date) = 6

group by hour(transaction\_time)

order by hour(transaction\_time);



**Q15 find out total sale,total quanitity,total order for 3 may and till 8 hours?**

select

concat(round(sum(transaction\_qty \* unit\_price)/1000,1), 'K') as Total\_sale,

concat(round(sum(transaction\_qty)/1000,1), 'K') as Total\_Quantity\_Sold,

concat(round(Count(distinct transaction\_id)/1000,1), 'K') as Total\_order

from coffee\_shop\_sales

where

DAYOFWEEK(transaction\_date) = 3 -- Tuesday (1 is Sunday, 2 is Monday, ..., 7 is Saturday)

AND HOUR(transaction\_time) = 8 -- hour number 8

AND MONTH(transaction\_date) = 5; -- May (month number 5)



**Q16 find out april month all week sale with week day name?**

select dayname(transaction\_date) As Day\_name,sum(transaction\_qty \* unit\_price) as Total\_sale

from coffee\_shop\_sales

where month(transaction\_date) = 4

group by dayname(transaction\_date);

