* Change column name

alter table coffee\_shop\_sales change column ï»¿transaction\_id transaction\_id int;

* TOTAL SALES

select round(sum(unit\_price\*transaction\_qty)) as total\_sales from coffee\_shop\_sales

where month(transaction\_date)=3 -- march month

* month on month increase and decrease in sales

SELECT

MONTH(transaction\_date) AS month, -- number of month

ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales, -- total sales column

(SUM(unit\_price \* transaction\_qty) - LAG(SUM(unit\_price \* transaction\_qty), 1) -- month sales difference

OVER (ORDER BY MONTH(transaction\_date))) / LAG(SUM(unit\_price \* transaction\_qty), 1) -- divided by pre.month sales

OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage -- percentage

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

* month on month increase and decrease in sales

SELECT

MONTH(transaction\_date) AS month, -- number of month

ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales, -- total sales column

(SUM(unit\_price \* transaction\_qty) - LAG(SUM(unit\_price \* transaction\_qty), 1) -- month sales difference

OVER (ORDER BY MONTH(transaction\_date))) / LAG(SUM(unit\_price \* transaction\_qty), 1) -- divided by pre.month sales

OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage -- percentage

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

* month on month total no.in orders increase or decrease

SELECT

MONTH(transaction\_date) AS month, -- number of month

ROUND(count(transaction\_id)) AS total\_orders, -- total orders column

(count(transaction\_id) - LAG(count(transaction\_id), 1) -- month orders difference

OVER (ORDER BY MONTH(transaction\_date))) / LAG(count(transaction\_id), 1) -- divided by pre.month sales

OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage -- percentage

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

-- total\_orders inn repective month

select sum(transaction\_qty) as total\_quantity\_sold

from coffee\_shop\_sales

where

month (transaction\_date)= 6 -- june month

* month on month total quantity increase or decrease

SELECT

MONTH(transaction\_date) AS month, -- number of month

ROUND(sum(transaction\_qty)) AS total\_quantity\_sold, -- total quantity sold column

(sum(transaction\_qty) - LAG(sum(transaction\_qty), 1) -- month total quantity difference

OVER (ORDER BY MONTH(transaction\_date))) / LAG(sum(transaction\_qty), 1) -- divided by pre.month sales

OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage -- percentage

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);

* implement tootips to display detailed matrix

select

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

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

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

from coffee\_shop\_sales

where

transaction\_date='2023-05-18'

-- weekdays and weekends sales

-- weekdays - sat and sun

-- weekdays - mon to fri

sun = 1

mon = 2

.

.

sat = 7

select

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

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 -- may month

group by

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

else 'weekdays'

end

* sales by store location

select

store\_location,

concat(round(sum(unit\_price\*transaction\_qty)/1000,2),'k')as total\_sales

from coffee\_shop\_sales

where month(transaction\_date) = 3 -- march

group by store\_location

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

* daily sales with avg line

select concat(round(avg(total\_sales)/1000,1),'k' )as avg\_sales

from (select sum(transaction\_qty\*unit\_price) as total\_sales from coffee\_shop\_sales

where month(transaction\_date) = 4 -- april month

group by transaction\_date) as internal\_query

select day(transaction\_date) as day\_of\_month,

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

from coffee\_shop\_sales

where month(transaction\_date) = 5

group by day(transaction\_date)

order by day(transaction\_date)

* COMPARING DAILY SALES WITH AVERAGE SALES – IF GREATER THAN “ABOVE AVERAGE” and LESSER THAN “BELOW AVERAGE”

select

day\_of\_month,

CASE

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

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

ELSE 'Average'

END AS sales\_status,

total\_sales

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)

) AS sales\_data

ORDER BY

day\_of\_month;

* sales anlysis by product category

select product\_category,sum(unit\_price\*transaction\_qty) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date)=5

group by product\_category

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

* top 10 product by sales

select product\_type,sum(unit\_price\*transaction\_qty) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date)=5

group by product\_type

order by sum(unit\_price\*transaction\_qty) desc limit 10;

* SALES BY DAY | HOUR

SELECT

ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales,

SUM(transaction\_qty) AS Total\_Quantity,

COUNT(\*) AS Total\_Orders

FROM

coffee\_shop\_sales

WHERE

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

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

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

-- TO GET SALES FOR ALL HOURS FOR MONTH OF MAY

SELECT

HOUR(transaction\_time) AS Hour\_of\_Day,

ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5 -- Filter for May (month number 5)

GROUP BY

HOUR(transaction\_time)

ORDER BY

HOUR(transaction\_time);

* TO GET SALES FROM MONDAY TO SUNDAY FOR MONTH OF MAY

SELECT

CASE

WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'

WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'

WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'

WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'

WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'

WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'

ELSE 'Sunday'

END AS Day\_of\_Week,

ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5 -- Filter for May (month number 5)

GROUP BY

CASE

WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'

WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'

WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'

WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'

WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'

WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'

ELSE 'Sunday'

END;