**MSSQL Script to verify the data**

-- addding a new column to transform text format date into date format

alter table coffee\_shop\_sales add transaction\_date\_new date;

-- extracting dates into new column as as date format yyyy-mm-dd

update coffee\_shop\_sales

set transaction\_date\_new = convert(date,cast(transaction\_date as varchar(20)),103);

-- addding a new column to transform text format time into time format

alter table coffee\_shop\_sales

add transaction\_time\_new time;

--extracting dates into a new column as a time format hh:mm:ss

update coffee\_shop\_sales

set transaction\_time\_new = convert(time, cast(transaction\_time as varchar(20)));

--dropping old columns of time and date

alter table coffee\_shop\_sales

drop column transaction\_date, transaction\_time;

--renaming the newer columns of date and time to older column names

exec sp\_rename 'coffee\_shop\_sales.transaction\_date\_new', 'transaction\_date', 'column'

exec sp\_rename 'coffee\_shop\_sales.transaction\_time\_new', 'transaction\_time', 'column';

--finding the sales for all months

select

MONTH(transaction\_date) as Month\_Num,

ceiling(sum(unit\_price\*transaction\_qty)) as Total\_Sales

from

coffee\_shop\_sales

group by

MONTH(transaction\_date)

order by

MONTH(transaction\_date);

--finding percentage increase in sales for every month comparing to previous month

--type 1

Select

MONTH(transaction\_date) as Month\_Num,

ceiling(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))) \* 100

/

(LAG(sum(unit\_price\*transaction\_qty),1)

over (order by month(transaction\_date))) as Mom\_Inrease\_Percentage

from

coffee\_shop\_sales

group by

MONTH(transaction\_date)

order by

MONTH(transaction\_date);

--type 2

WITH MonthlySales AS (

SELECT

MONTH(transaction\_date) AS Month\_Num,

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

FROM coffee\_shop\_sales

GROUP BY MONTH(transaction\_date)

)

SELECT

Month\_Num,

Total\_Sales,

ROUND(

(Total\_Sales - LAG(Total\_Sales, 1) OVER (ORDER BY Month\_Num)) \* 100.0 /

NULLIF(LAG(Total\_Sales, 1) OVER (ORDER BY Month\_Num), 0),

2

) AS Mom\_Increase\_Percentage

FROM MonthlySales

ORDER BY Month\_Num;

--finding no.of orders for every month

select

month(transaction\_date) as Month\_Num,

count(transaction\_id) as Total\_No\_of\_Orders

from

coffee\_shop\_sales

group by

month(transaction\_date)

order by

month(transaction\_date);

--finding percentage increase in orders for every month comparing to previous month

--type 1

select

MONTH(transaction\_date) as Month\_Num,

count(transaction\_id) as Total\_No\_of\_Orders,

(count(transaction\_id) - lag(count(transaction\_id),1)

over (order by month(transaction\_date))) \* 100

/

(lag(count(transaction\_id),1)

over (order by month(transaction\_date))) as Mom\_Increase\_Percentage

from

coffee\_shop\_sales

group by

MONTH(transaction\_date)

order by

MONTH(transaction\_date);

--type 2

with monthly\_orders as

(

select

MONTH(transaction\_date) as Month\_Num,

cast(count(transaction\_id) as decimal) as Total\_No\_of\_Orders

from coffee\_shop\_sales

group by

MONTH(transaction\_date)

)

select

Month\_Num,

Total\_No\_of\_Orders,

(Total\_No\_of\_Orders-lag(Total\_No\_of\_Orders,1)

over (order by Month\_Num)) \* 100

/

(lag(Total\_No\_of\_Orders,1)

over (order by Month\_Num)) as Mom\_Increase\_Percentage

from

monthly\_orders

order by

Month\_Num;

--finding quantity sold for every month

select

month(transaction\_date) as Month\_Num,

sum(transaction\_qty) as Quantity\_Sold

from

coffee\_shop\_sales

group by

month(transaction\_date)

order by

month(transaction\_date);

--finding increase in quantity sold every month compared to previous month

with monthly\_quantity as

(

select

month(transaction\_date) as Month\_Num,

sum(transaction\_qty) as Quantity\_Sold

from

coffee\_shop\_sales

group by

month(transaction\_date)

)

select

Month\_Num,

Quantity\_Sold,

(Quantity\_Sold-lag(Quantity\_Sold,1)

over (order by Month\_Num)) \* 100.0

/

(lag(Quantity\_Sold,1)

over (order by Month\_Num)) as Mom\_Increase\_Percentage

from

monthly\_quantity;

-- sales , orders and quantity on the date 2023-05-18 (yyyy-mm-dd)

select

sum(unit\_price\*transaction\_qty) as Sales,

count(transaction\_id) as Orders,

sum(transaction\_qty) as Quantity

from

coffee\_shop\_sales

where

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

-- sales , orders and quantity in thousands(K) on the date 2023-05-18 (yyyy-mm-dd)

select

concat(round(sum(unit\_price\*transaction\_qty)/1000.0,1),'K') as Sales,

CONCAT( cast(round(count(transaction\_id)/1000.0,1) as float), 'K') as Orders,

concat(cast(round(sum(transaction\_qty)/1000.0,1) as float),'K') as Quantity

from

coffee\_shop\_sales

where

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

--AVG SALES TREND OVER PERIOD

with Sales as (

select

transaction\_date as d,

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

from

coffee\_shop\_sales

group by

transaction\_date

)

select

month(d) as transaction\_month,

avg(daily\_sales) as Average\_Sales

from

Sales

group by

month(d)

order by

transaction\_month;

--DAILY SALES FOR MONTH SELECTED

with Sales as (

select

transaction\_date as d,

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

from

coffee\_shop\_sales

group by

transaction\_date

)

select

month(d) as Month\_of\_The\_Year,

DAY(d) as Day\_of\_The\_Month,

round(daily\_sales,1) as Daily\_Sales

from

Sales

order by

d;

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

with Sales as (

select

transaction\_date as d,

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

avg(sum(unit\_price\*transaction\_qty)) over() as average\_sales

from

coffee\_shop\_sales

group by

transaction\_date

)

select

month(d) as Month\_of\_The\_Year,

DAY(d) as Day\_of\_The\_Month,

round(daily\_sales,1) as Daily\_Sales,

case

when daily\_sales > average\_sales then 'ABOVE AVERAGE'

when daily\_sales < average\_sales then 'BELOW AVERAGE'

else 'Average'

end as Sales\_Status

from

Sales

order by

d;

-- count of sales Status from all dates

select

distinct(Sales\_Status),

count(Sales\_Status) as count

from (

select

month(d) as Month\_of\_The\_Year,

DAY(d) as Day\_of\_The\_Month,

round(daily\_sales,1) as Daily\_Sales,

case

when daily\_sales > average\_sales then 'ABOVE AVERAGE'

when daily\_sales < average\_sales then 'BELOW AVERAGE'

else 'Average'

end as Sales\_Status

from

(select

transaction\_date as d,

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

avg(sum(unit\_price\*transaction\_qty)) over() as average\_sales

from

coffee\_shop\_sales

group by

transaction\_date) as Sales

) as all\_sales

group by Sales\_Status;

-- SALES BY WEEKDAY / WEEKEND OVER MONTHS

select

month(transaction\_date) as month\_num,

case

when DATEPART(WEEKDAY,transaction\_date) in (1,7) then 'Weekend'

else 'Weekday'

end as day\_type,

sum(unit\_price\*transaction\_qty) as sales

from

coffee\_shop\_sales

group by

month(transaction\_date),

case

when DATEPART(WEEKDAY,transaction\_date) in (1,7) then 'Weekend'

else 'Weekday'

end

order by

month\_num;

--SALES BY STORE LOCATION

select

datename(MONTH,transaction\_date) as month,

store\_location,

sum(unit\_price\*transaction\_qty) as Sales

from

coffee\_shop\_sales

group by

store\_location,

datename(MONTH,transaction\_date)

order by

month, Sales desc;

--SALES BY PRODUCT CATEGORY

select

datename(MONTH,transaction\_date) as month,

product\_category,

round(sum(unit\_price\*transaction\_qty),1) as Sales

from

coffee\_shop\_sales

group by

product\_category,

datename(MONTH,transaction\_date)

order by

month, Sales desc;

-- SALES BY PRODUCTS (TOP 10)

with monthly\_sales as (

select

FORMAT(transaction\_date,'MM') as month,

product\_type as product,

round(sum(unit\_price\*transaction\_qty),1) as Sales,

ROW\_NUMBER() over (

partition by (FORMAT(transaction\_date,'MM'))

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

) as rn

from

coffee\_shop\_sales

group by

product\_type,

FORMAT(transaction\_date,'MM')

)

select

month,

product,

Sales

from

monthly\_sales

where

rn < 11;

-- sales, quantity, orders on tuesdays in month may between 8-9 am

select

round(sum(unit\_price\*transaction\_qty),0) as total\_sales,

count(transaction\_id) as no\_of\_orders,

sum(transaction\_qty) as quantity\_sold

from

coffee\_shop\_sales

where

DATEPART(MONTH,transaction\_date) = 5 and

DATEPART(WEEKDAY,transaction\_date) = 3 and

DATEPART(HOUR, transaction\_time) = 8;

-- TO GET SALES ANY WEEKDAY FOR MONTH OF MAY

-- type1

with monthwise\_weekly\_day\_sales as (

select

datepart(month,transaction\_date) as month\_num,

case

when DATEPART(weekday, transaction\_date) = 2 then 'Monday'

when DATEPART(weekday, transaction\_date) = 3 then 'Tuesday'

when DATEPART(weekday, transaction\_date) = 4 then 'Wednesday'

when DATEPART(weekday, transaction\_date) = 5 then 'Thursday'

when DATEPART(weekday, transaction\_date) = 6 then 'Friday'

when DATEPART(weekday, transaction\_date) = 7 then 'Saturday'

else 'Sunday'

end as Weekday\_Name,

round(sum(unit\_price\*transaction\_qty),0) as total\_sales

from

coffee\_shop\_sales

group by

DATEPART(weekday, transaction\_date),

datepart(month,transaction\_date)

)

select

month\_num,

Weekday\_Name,

total\_sales

from

monthwise\_weekly\_day\_sales

where

month\_num = 5;

-- type2

with monthwise\_weekly\_day\_sales as (

select

datepart(month,transaction\_date) as month\_num,

DATEPART(weekday, transaction\_date) as weekday\_num,

case

when DATEPART(weekday, transaction\_date) = 2 then 'Monday'

when DATEPART(weekday, transaction\_date) = 3 then 'Tuesday'

when DATEPART(weekday, transaction\_date) = 4 then 'Wednesday'

when DATEPART(weekday, transaction\_date) = 5 then 'Thursday'

when DATEPART(weekday, transaction\_date) = 6 then 'Friday'

when DATEPART(weekday, transaction\_date) = 7 then 'Saturday'

else 'Sunday'

end as Weekday\_Name,

round(sum(unit\_price\*transaction\_qty),0) as total\_sales

from

coffee\_shop\_sales

group by

DATEPART(weekday, transaction\_date),

datepart(month,transaction\_date)

)

select

month\_num,

Weekday\_Name,

total\_sales

from

monthwise\_weekly\_day\_sales

order by

month\_num ,weekday\_num;

-- TO GET SALES FOR ALL HOURS FOR ANY MONTH

select

DATEPART(MONTH,transaction\_date) as month\_num,

DATEPART(hour,transaction\_time) as sale\_hour,

round(sum(unit\_price\*transaction\_qty),0) as total\_sales

from

coffee\_shop\_sales

group by

DATEPART(MONTH,transaction\_date),

DATEPART(hour,transaction\_time)

order by

DATEPART(MONTH,transaction\_date),

DATEPART(hour,transaction\_time);