**Sql query list**

**Coffee shop analysis**

**-- sales analysis:**

1.SELECT EXTRACT(MONTH FROM transaction\_date) AS month,

ROUND(SUM(unit\_price \* transaction\_qty)::numeric, 2) AS sales

FROM cs\_detail

GROUP BY EXTRACT(MONTH FROM transaction\_date);

2. with cte as (

select extract(month from transaction\_date) as month,

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

from cs\_detail

group by extract(month from transaction\_date)

)

select month,

total\_sales

from cte

3.with cte as (

select \*,extract(month from transaction\_date) as month

from cs\_detail)

select \*,

concat(round(sum(transaction\_qty\*unit\_price::numeric)

over(order by month)/1000 ,2)::text ,'k')

as total\_sales

from cte

4. with monthly\_sales as(

select

extract(month from transaction\_date) as month,

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

from cs\_detail

where extract(month from transaction\_date) in (4,5)

group by extract(month from transaction\_date)

)

select month,round(sales),

(sales - lag(sales)over(order by month))/lag(sales) over(order by month)\*100

as percentage\_sales

from monthly\_sales

order by month

**--total order analysis:**

**1.order by month on month**

select extract(month from transaction\_date),

count(transaction\_id)

from cs\_detail

group by extract(month from transaction\_date);

**2. order by respective month**

select

count(transaction\_id)

from cs\_detail

where extract(month from transaction\_date) = '4'

**3. percentage increase**

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month,

COUNT(transaction\_id) AS total\_orders,

(COUNT(transaction\_id) - LAG(COUNT(transaction\_id), 1)

OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date))) /

LAG(COUNT(transaction\_id), 1)

OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date)) \* 100 AS mom\_increase\_percentage

FROM

coffee\_shop\_sales

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (4, 5) -- for April and May

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date);

* **Total sales analysis:**

**1.sold by month on month**

select extract(month from transaction\_date),

count(transaction\_qty)

from cs\_detail

group by extract(month from transaction\_date);

**2. sold by respective month**

select

count(transaction\_qty)

from cs\_detail

where extract(month from transaction\_date) = '5'

**3. mom increase or decrease**

with monthly\_sales as(

select

extract(month from transaction\_date) as month,

sum(transaction\_qty) as sold

from cs\_detail

where extract(month from transaction\_date) in (2,3,4)

group by extract(month from transaction\_date)

)

select month,round(sold),

coalesce( lag(sold)over(order by month),0),

COALESCE(

(sold - lag(sold,1) over(order by month))\*100/lag(sold,1) over(order by month) , 0)

as percentage\_sold

from monthly\_sales

order by month

* **Heatmap**

1. select
2. concat(round(sum(transaction\_qty\*unit\_price)/1000.0)::text,'k')
3. as total\_sales,
4. concat(sum(transaction\_qty)/1000,'k') as total\_sold,
5. concat(count(transaction\_id)/1000 , 'k') as total\_order
6. from cs\_detail
7. where transaction\_date = '2023-06-13'

* **SALES analysis weekdays and weekend**

select

case when extract(dow from transaction\_date) in (2,3,4,5,6)

then 'weekdays'

else 'weekend'

end as wow\_report,

concat(round(sum(unit\_price\*transaction\_qty)/1000::numeric)::text , 'k')

as total\_sales

from cs\_detail

where extract(month from transaction\_date) = 2 --feb month

group by

case when extract(dow from transaction\_date) in (2,3,4,5,6)

then 'weekdays'

else 'weekend'

end

* **sales analysis by store location**

select store\_location,

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

from cs\_detail

where extract(month from transaction\_date) = 5

group by store\_location

order by total\_sells desc

* **average sales per day in month**

select

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

from

(select

round(sum(unit\_price\*transaction\_qty))

as total\_sales

from cs\_detail

where extract(month from transaction\_date) = 4

group by transaction\_date

) as internal\_query

* **sale status**

select day\_of\_month,

case

when total\_sales> average\_sales then 'above average'

when total\_sales< average\_sales then 'below average'

else 'equal to average'

end as sales\_status,

total\_sales

from(

select

extract(day from transaction\_date) as day\_of\_month,

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

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

from cs\_detail

where extract(month from transaction\_date) = 5

group by transaction\_date

) as internal\_query

order by day\_of\_month

A screenshot of a computer

Description automatically generated

* **sales analysis by product\_details**

select product\_category,

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

from cs\_detail

group by product\_category

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

* **select time value date**

select

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

round(sum(transaction\_qty)) as total\_sold,

count(\*) as total\_order

from cs\_detail

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

and extract(dow from transaction\_date ) = 1 --sunday

and extract(hour from transaction\_time) = 16 --4:00 pm

* **sales analysis by hours**

select extract(hour from transaction\_time) as time,

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

round(sum(transaction\_qty)) as total\_sold,

count(\*) as total\_order

from cs\_detail

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

group by extract(hour from transaction\_time)

A screenshot of a computer

Description automatically generated

* **sales analysis by week**

SELECT

CASE

WHEN EXTRACT(DOW FROM transaction\_date) = 1 THEN 'monday'

WHEN EXTRACT(DOW FROM transaction\_date) = 2 THEN 'tuesday'

WHEN EXTRACT(DOW FROM transaction\_date) = 3 THEN 'wednesday'

WHEN EXTRACT(DOW FROM transaction\_date) = 4 THEN 'thursday'

WHEN EXTRACT(DOW FROM transaction\_date) = 5 THEN 'friday'

WHEN EXTRACT(DOW FROM transaction\_date) = 6 THEN 'saturday'

ELSE 'sunday'

END AS day\_of\_week,

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

FROM

cs\_detail

WHERE

EXTRACT(MONTH FROM transaction\_date) = 3

GROUP BY

EXTRACT(DOW FROM transaction\_date)

ORDER BY

CASE

WHEN EXTRACT(DOW FROM transaction\_date) = 1 THEN 1

WHEN EXTRACT(DOW FROM transaction\_date) = 2 THEN 2

WHEN EXTRACT(DOW FROM transaction\_date) = 3 THEN 3

WHEN EXTRACT(DOW FROM transaction\_date) = 4 THEN 4

WHEN EXTRACT(DOW FROM transaction\_date) = 5 THEN 5

WHEN EXTRACT(DOW FROM transaction\_date) = 6 THEN 6

ELSE 7

END;

A screenshot of a computer

Description automatically generated