**Part-A**

1. How many unique nodes are there on the Data Bank system?

select count(distinct(node\_id)) as nodes from customer\_nodes;

A screenshot of a computer

Description automatically generated

1. What is the number of nodes per region?

select count(customer\_nodes.region\_id) as count, region\_name from customer\_nodes

join regions on customer\_nodes.region\_id=regions.region\_id

group by region\_name

order by region\_name;

A screenshot of a computer

Description automatically generated

1. How many customers are allocated to each region?

select regions.region\_id,count(distinct(customer\_id))as count from customer\_nodes

join regions on customer\_nodes.region\_id=regions.region\_id

group by customer\_nodes.region\_id;

A screenshot of a computer

Description automatically generated

1. How many days on average are customers reallocated to a different node?

with cte1 as

(

SELECT customer\_id,customer\_nodes.region\_id,node\_id,start\_date,end\_date,region\_name,

DENSE\_RANK() OVER (PARTITION BY customer\_id ORDER BY node\_id) AS d\_rank, datediff(end\_date,start\_date) as days

FROM customer\_nodes

JOIN regions ON customer\_nodes.region\_id = regions.region\_id

where end\_date!='9999-12-31'

)

select customer\_id,d\_rank,avg(days),region\_id from cte1

group by customer\_id,d\_rank

order by customer\_id,d\_rank;

A screenshot of a data

Description automatically generated

Thoughts : This is quite subjective and more open to interpretation. I have not seen concrete solutions and the way I have interpreted this question this seems a fair response where one can look into the average avg days each customer spent at each node. We can further break this down if required.

1. What is the median, 80th and 95th percentile for this same reallocation days metric for each region?

with cte1 as

(

SELECT customer\_id,customer\_nodes.region\_id,node\_id,start\_date,end\_date,region\_name,

DENSE\_RANK() OVER (PARTITION BY customer\_id ORDER BY node\_id) AS d\_rank, datediff(end\_date,start\_date) as days

FROM customer\_nodes

JOIN regions ON customer\_nodes.region\_id = regions.region\_id

where end\_date!='9999-12-31'

),

cte2 as

(

select customer\_id,d\_rank,avg(days) as avg\_days,region\_id from cte1

group by customer\_id,d\_rank

order by customer\_id,d\_rank

),

cte3 as (

SELECT \*,

percent\_rank() over(partition by region\_id order by avg\_days)\*100 as p

FROM

cte2)

select region\_id,avg\_days from cte3

where p>50

group by region\_id

order by region\_id;

change values of p>50 to p>80 and p>90 to get the results.

Thoughts :- This is open to interpretation, should we again do the average or just select the first value ? I might have misunderstood the question all together.

**Part-B**

1. **What is the unique count and total amount for each transaction type?**

select txn\_type, count(distinct(customer\_id)) as unq\_cust, sum(txn\_amount) as amount from customer\_transactions

group by txn\_type;

A screenshot of a computer

Description automatically generated

1. **What is the average total historical deposit counts and amounts for all customers?**

select customer\_id,(count((customer\_id))) as count, avg(txn\_amount) as avg\_amount,txn\_type from customer\_transactions

where txn\_type='deposit'

group by customer\_id

order by customer\_id;

A screenshot of a computer

Description automatically generated

**3 .For each month - how many Data Bank customers make more than 1 deposit and either 1 purchase or 1 withdrawal in a single month?**

with cte1 as

(

select \*,sum(if(txn\_type='deposit',1,0)) as deposit\_count,

sum(if(txn\_type='withdrawal',1,0)) as withdrawal\_count,

sum(if(txn\_type='purchase',1,0)) as purchase\_count,

month(txn\_date) as month

from customer\_transactions

group by month,customer\_id

order by customer\_id

)

select month,count(distinct(customer\_id)) from cte1

where deposit\_count>1 and (withdrawal\_count=1 or purchase\_count=1)

group by month;

A screenshot of a computer

Description automatically generated

1. What is the closing balance for each customer at the end of the month?

with cte1 as

(

select \*, month(txn\_date) as month,sum(case

when txn\_type='deposit' then txn\_amount

else -txn\_amount

end ) as balance\_month

from customer\_transactions

group by month(txn\_date),customer\_id

order by customer\_id,month(txn\_date)

)

select customer\_id,month,sum(balance\_month) over(partition by customer\_id order by month) as closing\_balance from cte1;

A screenshot of a computer

Description automatically generated

1. What is the percentage of customers who increase their closing balance by more than 5%?

with cte1 as

(

select \*, month(txn\_date) as month,sum(case

when txn\_type='deposit' then txn\_amount

else -txn\_amount

end ) as balance\_month

from customer\_transactions

group by month(txn\_date),customer\_id

order by customer\_id,month(txn\_date)

),

cte2 as

(

select customer\_id,month,sum(balance\_month) over(partition by customer\_id order by month) as closing\_balance, dense\_rank () over(partition by customer\_id order by month) as rank\_col from cte1

),

cte3 as

(

select \*,max(rank\_col) as max\_rank from cte2

group by customer\_id

),

cte4 as

(

select cte2.customer\_id,cte3.closing\_balance as open\_balance,cte2.closing\_balance from cte2

join cte3 on cte2.customer\_id=cte3.customer\_id and cte2.rank\_col=cte3.max\_rank

),

cte5 as

(

select \*,((closing\_balance-open\_balance)/open\_balance)\*100 as p\_change from cte4

)

select \* from cte5

where p\_change>=5;

The above query will fetch the table below:

A screenshot of a data

Description automatically generated

Thoughts :

1. First the question does not ask if they want month over month or overall. I have considered overall balance i.e. opening balance of the customer at first month and the last month closing balance. I did not do month over month balance increase. I might update this in future based on month on month increase which we already got in previous question.
2. Ofcourse a lot of CTE were used to breakdown it down. This can be optimized but for learning purpose let us not focus into that however suggestions are welcome.
3. Some customers have accounts, where the negative balance is increased. This is bad for them as it mean it is still a debt. However we will include this in our analysis.

Finally to get the count we modify the last line of the query as :

select (count(distinct(customer\_id))/500)\*100 as percentage from cte5

where p\_change>=5;

A screenshot of a computer

Description automatically generated

Which means overall 44 % of the customers increased their closing balance overall by 5% ( debt & deposit included)

**Part-C**

**Note : - I did not have much clarity on the scope of question and the what is exactly needed. However I have answered it as per the question to the best of my knowledge.**

**Option 1: data is allocated based off the amount of money at the end of the previous month**

with cte1 as

(

select \*, month(txn\_date) as month,sum(case

when txn\_type='deposit' then txn\_amount

else -txn\_amount

end ) as balance\_month

from customer\_transactions

group by month(txn\_date),customer\_id

order by customer\_id,month(txn\_date)

),

cte2 as

(

select customer\_id,month,sum(balance\_month) over(partition by customer\_id order by month) as closing\_balance from cte1

)

select month, sum(abs(closing\_balance)) from cte2

group by month;

A screenshot of a computer

Description automatically generated

**Thoughts : While amount of money could be negative the data used to store cannot be negative even if we store a positive or negative interger. Hence I am doing absolute(abs) value of the closing balance since the data allocation cannot be negative in real life. Feel free to remove the abs() and check out the results also.**

**Option 2: data is allocated on the average amount of money kept in the account in the previous 30 days**

with cte1 as

(

select \*, month(txn\_date) as month,avg(case

when txn\_type='deposit' then txn\_amount

else -txn\_amount

end ) as balance\_month

from customer\_transactions

group by month(txn\_date),customer\_id

order by customer\_id,month(txn\_date)

),

cte2 as

(

select customer\_id,month,sum((balance\_month)) over(partition by customer\_id order by month) as closing\_balance from cte1

)

select month,sum(abs(closing\_balance)) from cte2

group by month;

**A screenshot of a computer

Description automatically generated**

**Option 3: data is updated real-time**

WITH transaction\_amt\_cte AS

(SELECT \*,

month(txn\_date) AS txn\_month,

SUM(CASE

WHEN txn\_type="deposit" THEN txn\_amount

ELSE -txn\_amount

END) AS net\_transaction\_amt

FROM customer\_transactions

GROUP BY customer\_id,

txn\_date

ORDER BY customer\_id,

txn\_date),

running\_customer\_balance\_cte AS

(SELECT customer\_id,

txn\_date,

txn\_month,

txn\_type,

txn\_amount,

net\_transaction\_amt,

sum(net\_transaction\_amt) over(PARTITION BY customer\_id

ORDER BY txn\_month ROWS BETWEEN UNBOUNDED preceding AND CURRENT ROW) AS running\_customer\_balance

FROM transaction\_amt\_cte)

select \* from running\_customer\_balance\_cte;

SELECT txn\_month,

SUM(abs(running\_customer\_balance)) AS data\_required\_per\_month

FROM running\_customer\_balance\_cte

GROUP BY txn\_month;

A screenshot of a computer

Description automatically generated

Again abs() is used as I feel every transaction accounts for a data value.