**Task 3:**

**INTRODUCTION:**

Neo-Banks are a recent development in the financial sector; they are new banks that solely operate online.

I believed that there should be some kind of connection between the digital world, these new age institutions, and cryptocurrencies.

So I made the decision to start a new project called Data Bank!

Customers of Data Bank receive cloud data storage allotments that are directly related to the balances in their accounts. The Data Bank team needs your assistance since this business model comes with some intriguing drawbacks.

This case study focuses on metrics calculations, business growth, and smart data analysis to assist the company more accurately estimate and plan for the future.

|  |  |
| --- | --- |
| regions | |
| region\_id | int |
| region\_name | varchar |

**Schema:**

|  |  |  |
| --- | --- | --- |
| customer\_transactions | | |
| customer\_id | int |
| txn\_date | | date |
| txn\_type | | varchar |
| txn\_amount | | int |

|  |  |
| --- | --- |
| customer\_nodes | |
| customer\_id | int |
| region\_id | int |
| node\_id | int |
| start\_date | date |
| end\_date | date |

**CASE STUDY QUESTIONS**

1. How many different nodes make up the Data Bank network?

select count(distinct node\_id) as unique\_node

from customer\_nodes;

Output:



1. How many nodes are there in each region?

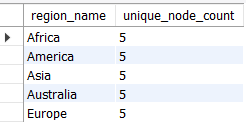
SELECT r.region\_name, COUNT(DISTINCT cn.node\_id) AS unique\_node\_count

FROM customer\_nodes cn

JOIN regions r ON cn.region\_id = r.region\_id

GROUP BY r.region\_name;

Output:



1. How many customers are divided among the regions?

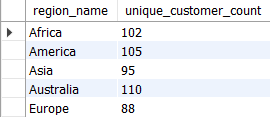
SELECT r.region\_name, COUNT(DISTINCT cn.customer\_id) AS unique\_customer\_count

FROM customer\_nodes cn

JOIN regions r ON cn.region\_id = r.region\_id

GROUP BY r.region\_name;

Output:



1. Determine the total amount of transactions for each region name.

SELECT r.region\_name, SUM(t.txn\_amount) AS total\_transaction\_amount

FROM customer\_transactions t

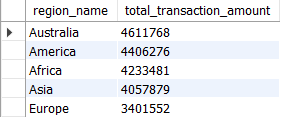
JOIN customer\_nodes cn ON t.customer\_id = cn.customer\_id

JOIN regions r ON cn.region\_id = r.region\_id

GROUP BY r.region\_name

ORDER BY total\_transaction\_amount DESC;

Output:



1. How long does it take on an average to move clients to a new node?

SELECT round(avg(datediff(end\_date, start\_date)), 2) AS avg\_days

FROM customer\_nodes

WHERE end\_date!='9999-12-31';

Output:



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

SELECT

txn\_type,

COUNT(DISTINCT customer\_id) AS unique\_transaction\_count,

SUM(txn\_amount) AS total\_amount

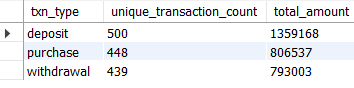
FROM

customer\_transactions

GROUP BY

txn\_type;

Output:



1. What is the average number and size of past deposits across all customers?

WITH customer\_deposits AS (

SELECT customer\_id,

COUNT(\*) AS num\_deposits,

SUM(txn\_amount) AS total\_deposit\_amount

FROM customer\_transactions

WHERE txn\_type = 'deposit'

GROUP BY customer\_id

)

SELECT

AVG(num\_deposits) AS avg\_deposits\_per\_customer,

AVG(total\_deposit\_amount) AS avg\_deposit\_amount\_per\_customer

FROM customer\_deposits;

Output:



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

WITH txn\_with\_month AS (

SELECT

customer\_id,

txn\_type,

DATE\_FORMAT(txn\_date, '%Y-%m') AS txn\_month

FROM customer\_transactions

),

deposit\_counts AS (

SELECT

customer\_id,

txn\_month,

COUNT(\*) AS deposit\_count

FROM txn\_with\_month

WHERE txn\_type = 'deposit'

GROUP BY customer\_id, txn\_month

),

pw\_flags AS (

SELECT DISTINCT

customer\_id,

txn\_month

FROM txn\_with\_month

WHERE txn\_type IN ('purchase', 'withdrawal')

),

qualified\_customers AS (

SELECT d.customer\_id, d.txn\_month

FROM deposit\_counts d

JOIN pw\_flags p

ON d.customer\_id = p.customer\_id AND d.txn\_month = p.txn\_month

WHERE d.deposit\_count > 1

)

SELECT

txn\_month,

COUNT(DISTINCT customer\_id) AS qualified\_customer\_count

FROM qualified\_customers

GROUP BY txn\_month

ORDER BY txn\_month;

Output:

