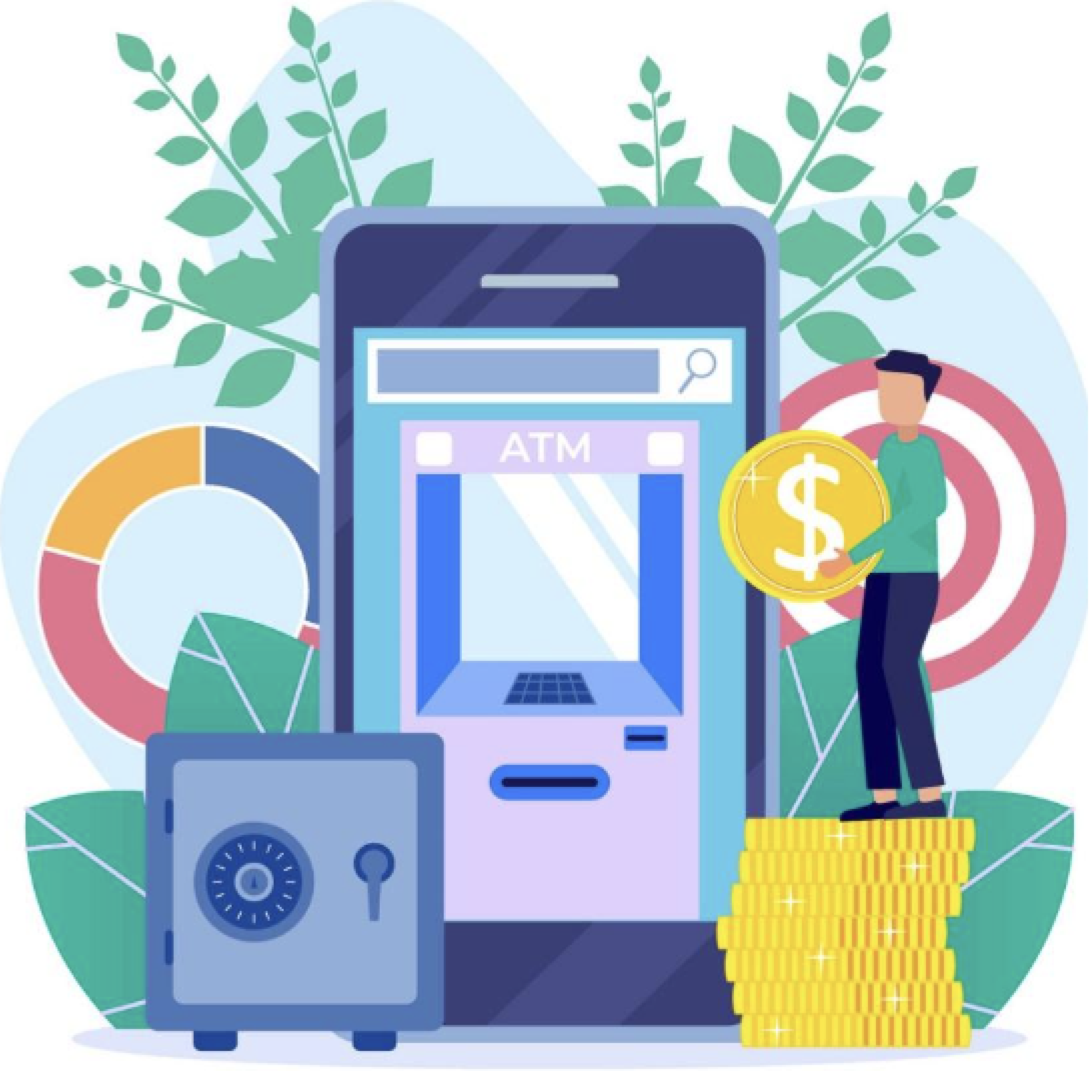
SQL Case Study : Data Bank

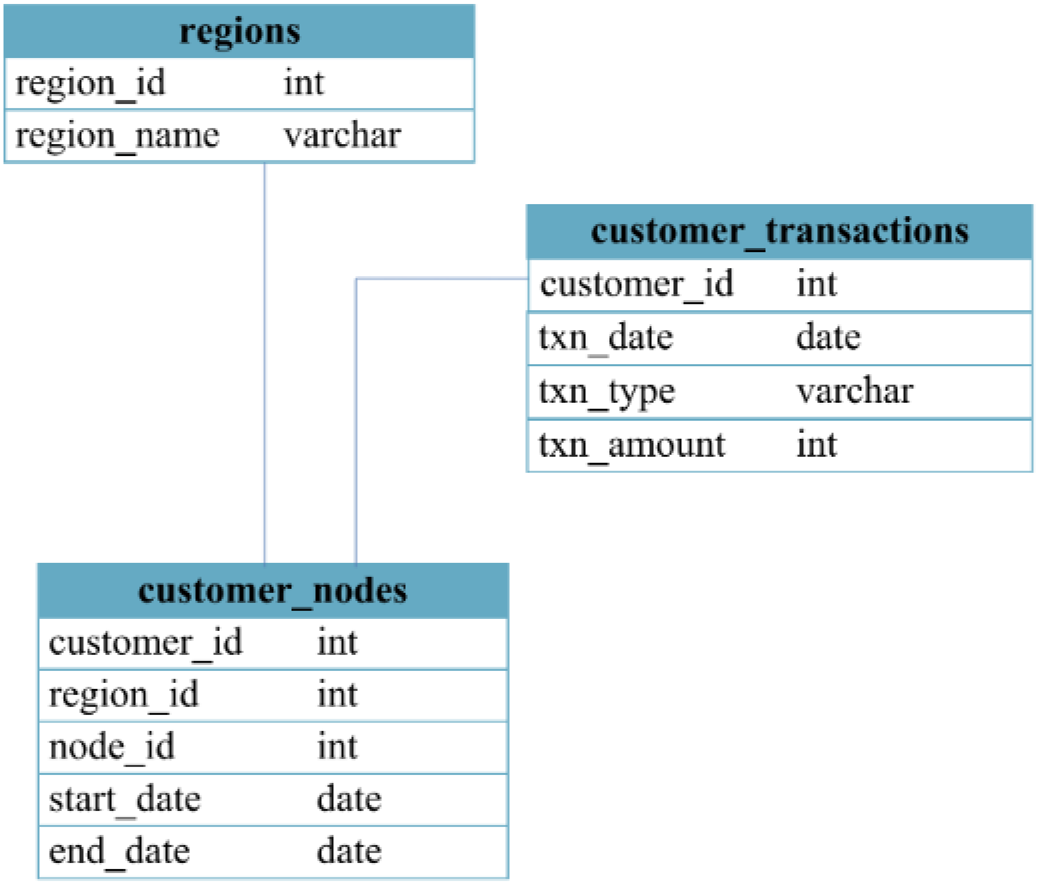


INTRODUCTION:

In the rapidly evolving financial sector, Neo-Banks have emerged as innovative institutions that operate exclusively online. Recognizing the potential synergy between the digital realm, these modern financial entities, and cryptocurrencies, I embarked on a new venture called Data Bank. At Data Bank, our customers are provided with cloud data storage allocations that are directly linked to the balances in their accounts. However, this unique business model presents us with intriguing challenges that require your expertise.

This case study revolves around calculating metrics, fostering business growth, and leveraging smart data analysis to help Data Bank accurately forecast and strategize for the future.

SCHEMA USED



regions	
region_id	int
region_name	varchar

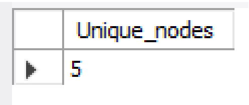
customer_transactions	
customer_id	int
txn_date	date
txn_type	varchar
txn_amount	int

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_nodes
FROM
  customer_nodes;
```



	Unique_nodes
▶	5

Explanation: The query retrieves the count of distinct node IDs from the `customer_nodes` table using `COUNT(DISTINCT node_id)`.This provides the number of unique nodes in the Data Bank network.

Insights:

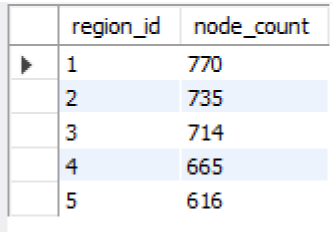
- The result of this query gives an indication of the network's scale by determining the number of different nodes present.
- It can help in understanding the complexity and size of the network infrastructure.
- Monitoring the number of nodes over time can provide insights into network growth and expansion.

Suggestions:

- Analyzing the distribution of nodes across regions can provide additional insights into the network's geographical coverage and potential areas for expansion.

2. How many nodes are there in each region?

```
SELECT
  region_id, COUNT(node_id) AS node_count
FROM
  customer_nodes
     INNER JOIN
  regions USING (region_id)
GROUP BY region_id;
```



region_id	node_count
1	770
2	735
3	714
4	665
5	616

Explanation:The query joins the `customer_nodes` table with the `regions` table based on the `region_id` to get the count of nodes in each region using `COUNT(node_id)`.The result includes the region ID and the corresponding count of nodes.

Insights:

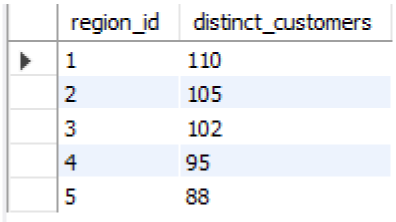
- This query helps in understanding the distribution of nodes across different regions.
- It provides insights into regions with higher or lower node concentrations, which can help identify areas for potential infrastructure improvements or resource allocation.

Suggestions:

- Visualizing the node distribution on a map can provide a clearer overview of regional imbalances and facilitate decision-making for optimizing the network's infrastructure.

3. How many customers are divided among the regions?

```
SELECT
  region_id, COUNT(DISTINCT customer_id) AS distinct_customers
FROM
  customer_nodes
     INNER JOIN
  regions USING (region_id)
GROUP BY region_id;
```



region_id	distinct_customers
1	110
2	105
3	102
4	95
5	88

Explanation: The query joins the `customer_nodes` table with the `regions` table based on the `region_id` to count the number of distinct customers in each region using `COUNT(DISTINCT customer_id)`.

Insights:

- This query provides insights into the distribution of customers among different regions.
- It helps in understanding customer concentrations and identifying regions with high customer activity.

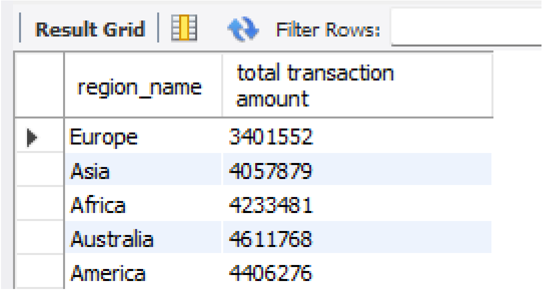
Suggestions:

- Analyzing customer demographics, preferences, and behavior in each region can provide valuable insights for targeted marketing strategies and resource allocation.

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

```
SELECT
  region_name, SUM(txn_amount) AS 'total transaction amount'
FROM
  regions,
  customer_nodes,
  customer_transactions
WHERE
```

```
    regions.region_id = customer_nodes.region_id
       AND customer_nodes.customer_id = customer_transactions.customer_id
  GROUP BY region_name;
```



region_name	total transaction amount
Europe	3401552
Asia	4057879
Africa	4233481
Australia	4611768
America	4406276

Explanation: The query combines the `regions`, `customer_nodes`, and `customer_transactions` tables using appropriate join conditions.It retrieves the region name and calculates the sum of transaction amounts (`txn_amount`) for each region using `SUM(txn_amount)`.

Insights:

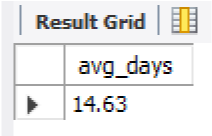
- This query helps in understanding the transaction volume in different regions.
- It provides insights into regions with higher financial activity and can be useful for evaluating regional performance or identifying regions with potential growth opportunities.

Suggestions:

- Analyzing transaction patterns, trends, and correlations with other variables (e.g., customer demographics) can provide deeper insights into regional financial dynamics and help in devising targeted marketing or investment strategies.

5.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';
```



avg_days
14.63

Explanation: The query calculates the average duration for moving clients to a new node by computing the average difference between the `end_date` and `start_date` columns.It uses

`DATEDIFF` to calculate the duration in days and `AVG` to compute the average.The result is rounded to two decimal places using `ROUND`.

Insights:

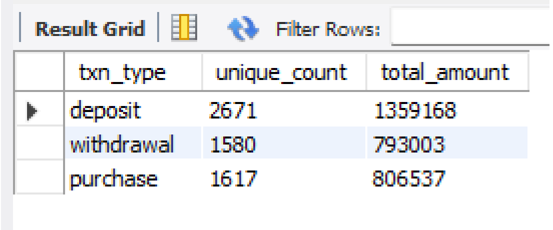
- This query provides insights into the average time it takes to migrate clients to new nodes.
- It helps in assessing the efficiency of client migration processes and identifying any potential delays or bottlenecks.

Suggestions:

- Analyzing the distribution of migration durations and identifying factors that affect the migration

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

```
SELECT
  txn_type,
  COUNT(*) AS unique_count,
  SUM(txn_amount) AS total_amount
FROM
  customer_transactions
GROUP BY txn_type;
```



txn_type	unique_count	total_amount
deposit	2671	1359168
withdrawal	1580	793003
purchase	1617	806537

Explanation: The query retrieves the transaction type, counts the number of occurrences for each type using `COUNT(*)`, and calculates the total transaction amount using `SUM(txn_amount)`.It groups the results by transaction type using `GROUP BY`.

Insights:

- This query provides insights into transaction types, their occurrence count, and the total transaction amount for each type.
- It helps in understanding the distribution and importance of different transaction types.

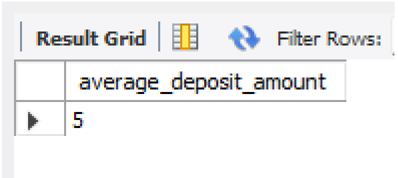
Suggestions:

- Analyzing transaction types with high occurrence counts and high total amounts can help identify areas of focus for improving customer engagement or optimizing transaction processes.

- Comparing transaction types across different customer segments or regions can reveal variations in transaction preferences and guide marketing strategies.

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

```
SELECT
  ROUND(COUNT(customer_id) / (SELECT
            COUNT(DISTINCT customer_id)
          FROM
            customer_transactions)) AS average_deposit_amount
FROM
  customer_transactions
WHERE
  txn_type = 'deposit';
```



```
average_deposit_amount
5
```

Explanation: The query calculates the average deposit amount by dividing the count of customer IDs (`COUNT(customer_id)`) by the count of distinct customer IDs (`COUNT(DISTINCT customer_id)`).The `ROUND` function is used to round the result

Insights:

- This query provides insights into the average number and size of deposits made by customers.
- It helps in understanding the deposit behavior and preferences of customers.

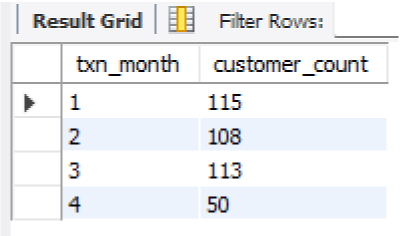
Suggestions:

- Analyzing the average deposit amount over time can reveal trends in customer savings habits or changes in deposit patterns.
- Comparing the average deposit amounts across different customer segments or demographics can provide insights for targeted marketing or product offerings.

8. 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 transaction_count_per_month_cte AS
(SELECT
  customer_id,
  MONTH(txn_date) AS txn_month,
  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
FROM
  customer_transactions
GROUP BY customer_id , MONTH(txn_date))

SELECT
  txn_month, COUNT(DISTINCT customer_id) AS customer_count
FROM
  transaction_count_per_month_cte
WHERE
  deposit_count > 1
     AND (purchase_count = 1
     OR withdrawal_count = 1)
GROUP BY txn_month;
```



txn_month	customer_count
1	115
2	108
3	113
4	50

Explanation: The query uses a Common Table Expression (CTE) named `transaction_count_per_month_cte` to calculate the count of deposits, withdrawals, and purchases for each customer in each month.It then selects the distinct count of customer IDs (`COUNT(DISTINCT customer_id)`) for months where the deposit count is greater than 1 and either the purchase count or withdrawal count is 1.

Insights:

- This query provides insights into the number of customers who exhibit specific transaction behavior (multiple deposits and at least one purchase or withdrawal) for each month.
- It helps in identifying customers with active transactional behavior and potential cross-selling opportunities.

Suggestions:

- Analyzing the customer count over time can reveal seasonal patterns or changes in transaction behavior.
- Segmenting customers based on their transactional behavior can guide personalized marketing campaigns or loyalty programs.

Valuable Insights:

- The Data Bank network comprises a diverse set of nodes, indicating scalability and potential for expansion.
- Node distribution across regions highlights areas with varying infrastructure needs and customer concentration.
- Customer distribution among regions provides insights into market penetration and growth opportunities.
- Transaction analysis by region helps allocate resources, manage risk, and tailor services based on regional financial activity.
- The average time to move clients to a new node affects operational efficiency and customer onboarding experience.
- Transaction type analysis reveals popular services and revenue potential for specific transaction types.

This comprehensive report provides valuable insights into the Data Bank network, including the number of nodes, their distribution by region, customer distribution, transaction details, average client movement time, and monthly customer behavior. These findings can aid in optimizing operations, identifying trends, and making informed business decisions for the Data Bank network.