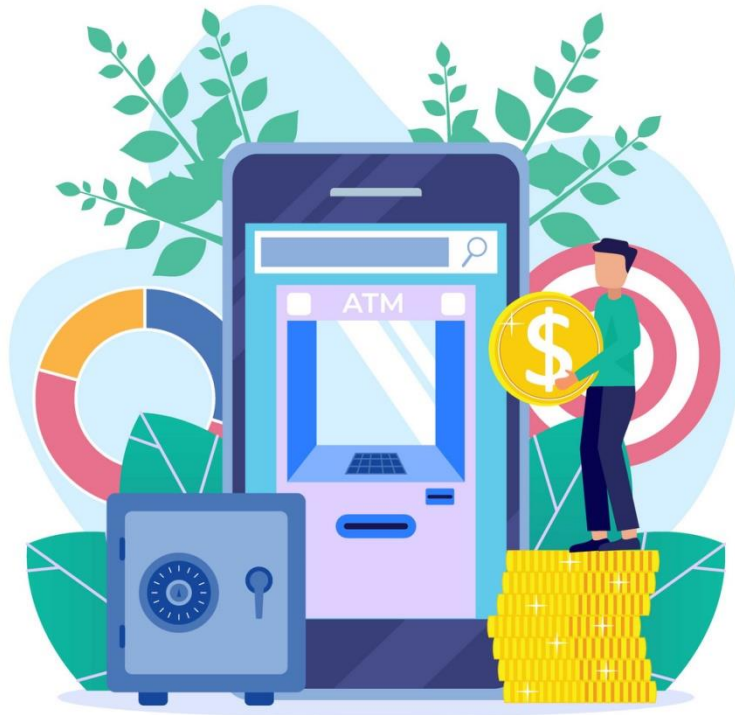


SQL CASE STUDY : DATA BANK



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.

Schema Link : <https://github.com/brijeshbabariya6/DATA-BANK-Analysis-Using-MySQL/upload/main>

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

CREATE DATABASE:

CREATE TABLES:

CASE STUDY QUESTIONS

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of databases including 'data_bank', 'datamart', 'date_time', 'db', and 'leetcode'. The 'data_bank' database is selected, showing tables like 'customer_nodes', 'customer_transaction', and 'regions'. The main editor window contains a SQL query: `## Q1. How many different nodes make up the Data Bank network?`
`SELECT`
`COUNT(DISTINCT node_id) AS unique_nodes`
`FROM`
`customer_nodes;`
The 'Result Grid' at the bottom shows the query results: a single row with the column 'unique_nodes' and the value '5'. The 'Action Output' panel at the bottom shows the execution log, including the query execution at 21:13:10, which returned 1 row(s).

#	Time	Action	Message	Duration / Fetch
9379	21:09:57	INSERT INTO regions(region_id,region_name) VALUES (5,'Europe')	1 row(s) affected	0.000 sec
9380	21:13:10	SELECT COUNT(DISTINCT node_id) AS unique_nodes FROM customer_nodes	1 row(s) returned	0.000 sec / 0.000 sec

2. How many nodes are there in each region?

MySQL Workbench

DATA BANK x Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

data_bank

- Tables
 - customer_nodes
 - customer_transactions
 - regions
- Views
- Stored Procedures
- Functions

Administration Schemas

Information

No object selected

Query 1

```
1  ##Q2 How many nodes are there in each region?
2  SELECT
3      region_id,
4      region_name,
5      COUNT(node_id) AS node_count
6  FROM
7      customer_nodes INNER JOIN regions
8      USING(region_id)
9  GROUP BY
10     region_id,
11     region_name
12 ORDER BY
13     region_id;
```

Result Grid

	region_id	region_name	node_count
1	Australia	770	
2	America	735	
3	Africa	714	
4	Asia	665	
5	Europe	616	

Result 4 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	21:19:07	SELECT region_id, region_name, COUNT(node_id) AS node_count FROM customer_nodes INNER JOIN regions...	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

3. How many customers are divided among the regions?

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'data_bank' database with tables 'customer_nodes', 'customer_transaction', and 'regions'. The main query editor contains the following SQL code:

```
1  ##Q3 How many customers are divided among the regions?
2  SELECT
3      region_id,
4      region_name,
5      COUNT(DISTINCT customer_id) AS customer_count
6  FROM
7      customer_nodes INNER JOIN regions
8      USING(region_id)
9  GROUP BY
10     region_id,
11     region_name;
```

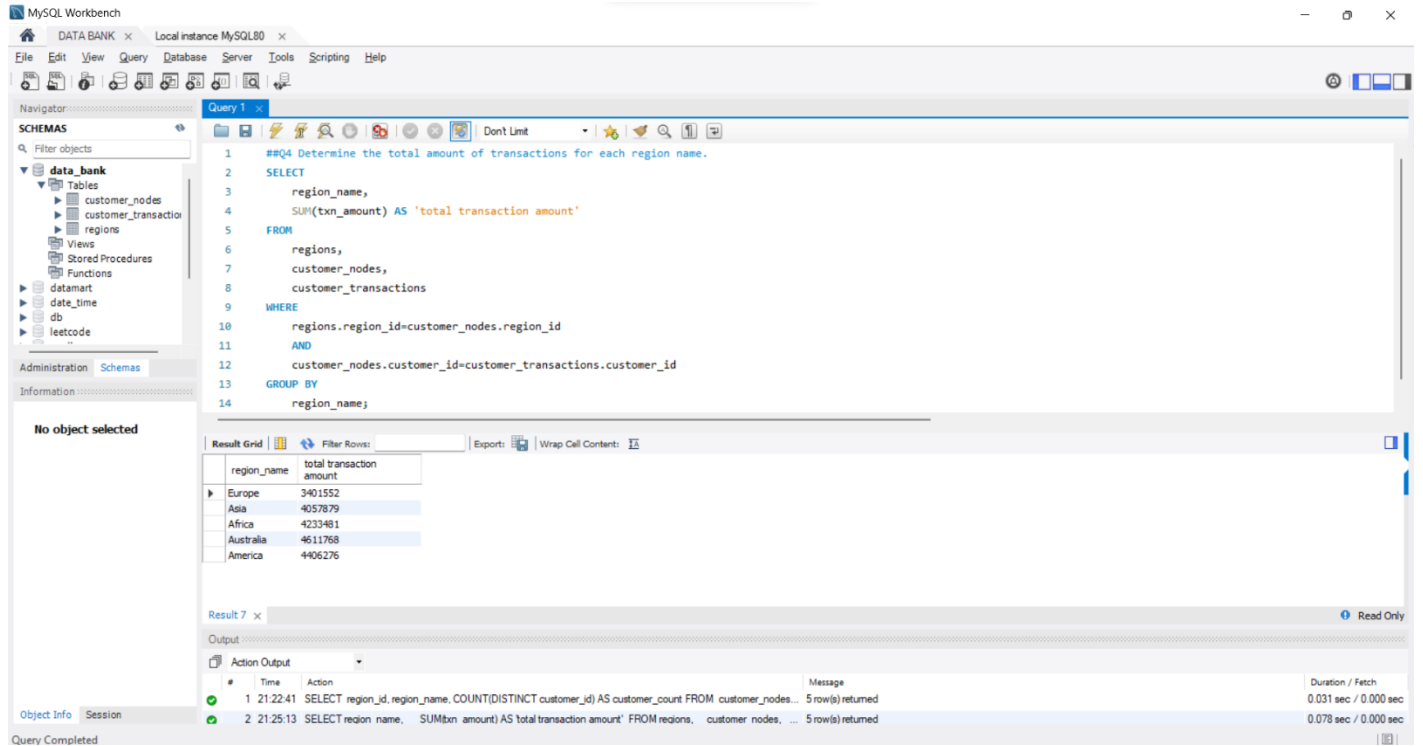
Below the query editor, the 'Result Grid' shows the output of the query:

region_id	region_name	customer_count
1	Australia	110
2	America	105
3	Africa	102
4	Asia	95
5	Europe	88

At the bottom, the 'Output' pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	21:22:41	SELECT region_id, region_name, COUNT(DISTINCT customer_id) AS customer_count FROM customer_nodes...	5 row(s) returned	0.031 sec / 0.000 sec

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



The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'Schemas' tree with 'data_bank' selected, containing tables like 'customer_nodes', 'customer_transactions', and 'regions'. The main query editor contains the following SQL code:

```
1  ##Q4 Determine the total amount of transactions for each region name.
2  SELECT
3      region_name,
4      SUM(txn_amount) AS 'total transaction amount'
5  FROM
6      regions,
7      customer_nodes,
8      customer_transactions
9  WHERE
10     regions.region_id=customer_nodes.region_id
11     AND
12     customer_nodes.customer_id=customer_transactions.customer_id
13  GROUP BY
14     region_name;
```

The 'Result Grid' shows the output of the query:

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

The bottom panel shows the 'Output' tab with a log of executed queries and their execution times.

#	Time	Action	Message	Duration / Fetch
1	21:22:41	SELECT region_id, region_name, COUNT(DISTINCT customer_id) AS customer_count FROM customer_nodes...	5 row(s) returned	0.031 sec / 0.000 sec
2	21:25:13	SELECT region name, SUM(txn amount) AS total transaction amount FROM regions, customer nodes, ...	5 row(s) returned	0.078 sec / 0.000 sec

Query Completed

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

The screenshot shows the MySQL Workbench interface with a query executed in the 'Query 1' tab. The query calculates the average time to move clients to a new node by averaging the difference between end_date and start_date for the 'customer_nodes' table, filtered by end_date != '9999-12-31'. The result grid shows a single value of 14.63 for the 'avg_days' column.

Query 1

```
1  ##Q5 How long does it take on an average to move clients to a new node?
2  SELECT
3    round(avg(datediff(end_date, start_date)), 2) AS avg_days
4  FROM
5    customer_nodes
6  WHERE
7    end_date != '9999-12-31';
8
```

Result Grid

avg_days
14.63

Result 1

Output

#	Time	Action	Message	Duration / Fetch
2	20:15:33	use data_bank	0 row(s) affected	0.000 sec
3	20:15:37	SELECT round(avg(datediff(end_date, start_date)), 2) AS avg_days FROM customer_nodes WHERE end d...	1 row(s) returned	0.016 sec / 0.000 sec

Query Completed

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'data_bank' selected. The main editor shows a SQL query for 'Query 1' with the following code:

```
1  ##Q6. What is the unique count and total amount for each transaction type?
2  *
3  SELECT
4      txn_type,
5      count(*) AS unique_count,
6      sum(txn_amount) AS total_amount
7  FROM
8      customer_transactions
9  GROUP BY
10     txn_type;
```

The 'Result Grid' at the bottom displays the query results in a table with three columns: 'txn_type', 'unique_count', and 'total_amount'.

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

Below the result grid, the 'Output' tab shows the execution log. The first entry indicates that 0 rows were affected by the query.

#	Time	Action	Message	Duration / Fetch
2	20:26:06	use data_bank	0 row(s) affected	0.000 sec
3	20:26:15	SET SQL_SELECT for txn_type, count(*) AS unique_count, sum(txn_amount) AS total_amount FROM customer_transactions	3 row(s) total	0.015 sec / 0.000 sec

The status bar at the bottom indicates 'Query Completed'.

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with 'data_bank' selected. The main query editor contains the following SQL query:

```
1  #Q7. What is the average number and size of past deposits across all customers?
2  SELECT round(count(customer_id)/
3             (SELECT count(DISTINCT customer_id)
4              FROM customer_transactions)) AS average_deposit_count,
5         concat('$', round(avg(txn_amount), 2)) AS average_deposit_amount
6  FROM customer_transactions
7  WHERE txn_type = "deposit";
8
```

The 'Result Grid' shows the following data:

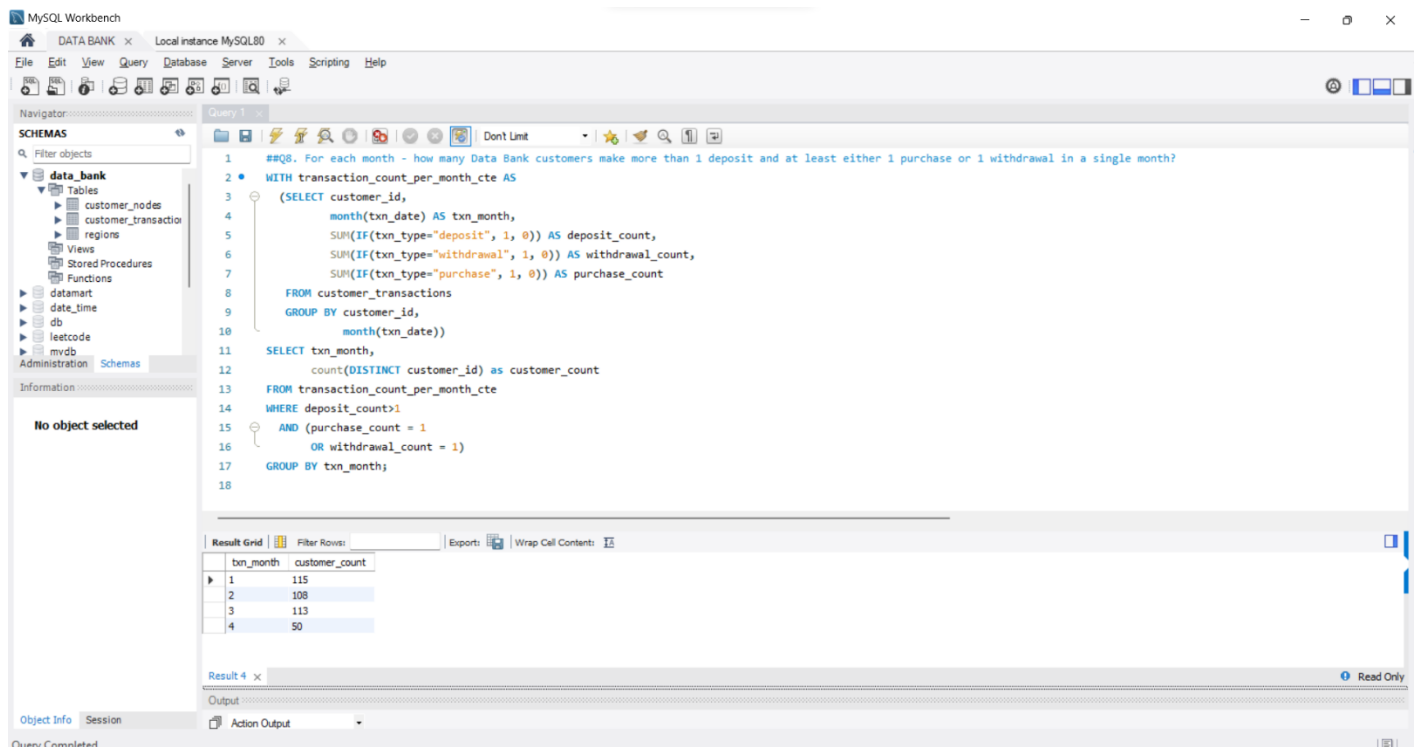
average_deposit_count	average_deposit_amount
5	\$508.86

The 'Output' panel at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
4	20:17:04	SELECT txn_type, count(*) AS unique_count, sum(txn_amount) AS total_amount FROM customer_transactions...	3 row(s) returned	0.015 sec / 0.000 sec
5	20:18:11	SELECT round(count(customer_id)/ (SELECT count(DISTINCT customer_id) FROM custo...	1 row(s) returned	0.015 sec / 0.000 sec

The status bar at the bottom indicates 'Query Completed'.

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?



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' panel with a tree view of the 'data_bank' database, including tables like 'customer_nodes', 'customer_transactions', and 'regions'. The main query editor contains a SQL query that uses a Common Table Expression (CTE) to calculate transaction counts per month and then filters for customers with more than 1 deposit and at least 1 purchase or withdrawal.

```
1  ##Q8. 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?
2  WITH transaction_count_per_month_cte AS
3  (
4      SELECT customer_id,
5             month(txn_date) AS txn_month,
6             SUM(IF(txn_type="deposit", 1, 0)) AS deposit_count,
7             SUM(IF(txn_type="withdrawal", 1, 0)) AS withdrawal_count,
8             SUM(IF(txn_type="purchase", 1, 0)) AS purchase_count
9      FROM customer_transactions
10     GROUP BY customer_id,
11              month(txn_date))
12  SELECT txn_month,
13         count(DISTINCT customer_id) as customer_count
14  FROM transaction_count_per_month_cte
15  WHERE deposit_count > 1
16         AND (purchase_count = 1
17              OR withdrawal_count = 1)
18  GROUP BY txn_month;
```

The 'Result Grid' at the bottom shows the output of the query, which is a table with two columns: 'txn_month' and 'customer_count'. The results are as follows:

txn_month	customer_count
1	115
2	108
3	113
4	50

