Challenge-4

DATA BANK

By: Harshita Aswani

INTRODUCTION

There is an innovation in the financial industry called Neo-Banks: new aged digital-only banks without physical branches.

Danny thought that there should be some sort of intersection between these new-age banks, cryptocurrency and the data world…so he decided to launch a new initiative - Data Bank!

Data Bank runs just like any other digital bank - but it isn't only for banking activities, they also have the world's most secure distributed data storage platform!

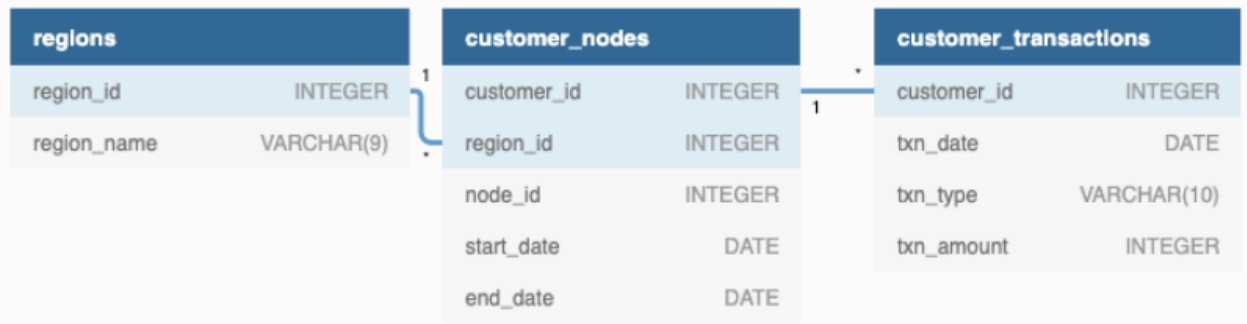
Customers are allocated cloud data storage limits which are directly linked to how much money they have in their accounts. There are a few interesting caveats that go with this business model, and this is where the Data Bank team need my help!

The management team at Data Bank want to increase their total customer base - but also needs some help tracking just how much data storage their customers will need.

This case study is all about calculating metrics, and growth and helping the business smartly analyse their data to better forecast and plan for their future developments!

Entity Relationship Diagram

Danny has shared with me 3 key datasets for this case study whose entity relationship diagram is given below:



Tables

Just like popular cryptocurrency platforms - Data Bank is also run off a network of nodes where both money and data is stored across the globe. In a traditional banking sense - you can think of these nodes as bank branches or stores that exist around the world.

This regions table contains the region_id and their respective region_name values

region_id	region_name
1	Africa
2	America
3	Asia
4	Europe
5	Oceania

Customers are randomly distributed across the nodes according to their region - this also specifies exactly which node contains both their cash and data.

This random distribution changes frequently to reduce the risk of hackers getting into Data Bank's system and stealing customers' money and data!

Below is a sample of the top 10 rows of the data_bank.customer_nodes

customer_id	region_id	node_id	start_date	end_date
1	3	4	2020-01-02	2020-01-03
2	3	5	2020-01-03	2020-01-17
3	5	4	2020-01-27	2020-02-18
4	5	4	2020-01-07	2020-01-19
5	3	3	2020-01-15	2020-01-23
6	1	1	2020-01-11	2020-02-06
7	2	5	2020-01-20	2020-02-04
8	1	2	2020-01-15	2020-01-28
9	4	5	2020-01-21	2020-01-25
10	3	4	2020-01-13	2020-01-14

Customer Transactions table stores all customer deposits, withdrawals and purchases made using their Data Bank debit card.

customer_id	txn_date	txn_type	txn_amount
429	2020-01-21	deposit	82
155	2020-01-10	deposit	712
398	2020-01-01	deposit	196
255	2020-01-14	deposit	563
185	2020-01-29	deposit	626
309	2020-01-13	deposit	995
312	2020-01-20	deposit	485
376	2020-01-03	deposit	706
188	2020-01-13	deposit	601
138	2020-01-11	deposit	520

Case Study Questions

Customer Nodes

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

a. Query SQL

```
1 SELECT COUNT(DISTINCT node_id) AS unique_nodes
2 FROM customer_nodes;
```

b.

unique_nodes
5

II. What is the number of nodes per region?

a. Query SQL

```
SELECT region_id, COUNT(DISTINCT node_id) AS nodes_per_region
FROM customer_nodes
GROUP BY region_id;
```

b.

region_id	nodes_per_region
1	5
2	5
3	5
4	5
5	5

III. How many customers are allocated to each region?

a. Query SQL

```
SELECT region_id, COUNT(DISTINCT customer_id) AS customers_per_region
FROM customer_nodes
GROUP BY region_id;
```

b.

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

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

a. Query SQL

```
SELECT AVG(DATEDIFF(end_date, start_date)) AS average_reallocation_days
FROM customer_nodes;
```

b.

average_reallocation_days
416373.4117

Customer Transactions

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

a. Query SQL

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

b.

txn_type	unique_count	total_amount
deposit	500	1359168
purchase	448	806537
withdrawal	439	793003

VI. What are the average total historical deposit counts and amounts for all customers?

a. Query SQL

```sql
SELECT
  AVG(deposit_count) AS avg_deposit_count,
  AVG(total_deposit_amount) AS avg_total_deposit_amount
FROM (
  SELECT
    customer_id,
    COUNT(*) AS deposit_count,
    SUM(txn_amount) AS total_deposit_amount
  FROM
    customer_transactions
  WHERE
    txn_type = 'deposit'
  GROUP BY
    customer_id
) AS HistoricalDeposits;
```

b.

avg_deposit_count	avg_total_deposit_amount
5.3420	2718.3360

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

a.

Query SQL

```sql
SELECT
  COUNT(DISTINCT customer_id) AS customers_count
FROM (
  SELECT
    customer_id,
    EXTRACT(MONTH FROM txn_date) AS transaction_month,
    COUNT(DISTINCT CASE WHEN txn_type = 'deposit' THEN txn_date END) AS deposit_count,
    COUNT(DISTINCT CASE WHEN txn_type IN ('purchase', 'withdrawal') THEN txn_date END) AS purchase_withdrawal_count
  FROM
    customer_transactions
  GROUP BY
    customer_id, transaction_month
) AS MonthlyTransactions
WHERE
  deposit_count > 1 AND purchase_withdrawal_count > 0;
```

b.

customers_count
346

VIII. What is the closing balance for customer_id 1 at the end of January?

a.

Query SQL

```sql
SELECT
  customer_id,
  EXTRACT(MONTH FROM txn_date) AS transaction_month,
  SUM(CASE WHEN txn_type = 'deposit' THEN txn_amount ELSE -txn_amount END) AS monthly_balance
FROM
  customer_transactions
WHERE
  customer_id = 1
  AND EXTRACT(MONTH FROM txn_date) = 1
GROUP BY
  customer_id, transaction_month;
```

b.

customer_id	transaction_month	monthly_balance
1	1	312