



Bank Data Analysis using SQL

Cithub link : <https://github.com/giovaldir/Bank-Data-Analysis-using-SQL/tree/main>

SQL Project



Acorns helps you grow your money

Take control with all-in-one investment, retirement, checking and more.

Get started





Project Description

About

Data Bank operates as a digital bank with a unique feature: aside from banking services, it offers the world's most secure distributed data storage platform. Cloud data storage limits for customers are directly tied to their account balances. However, there are intriguing complexities related to this approach. The Data Bank team seeks assistance to enhance customer acquisition and track data storage requirements. This case study focuses on calculating metrics, facilitating growth, and enabling intelligent data analysis for accurate future development forecasts and planning.

Data Understanding

customer_nodes

customer_id	region_id	node_id	start_date	end_date
integer	integer	integer	date	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

regions

region_id	region_name
integer	character varying (9)
1	Australia
2	America
3	Africa
4	Asia
5	Europe

customer_transaction

customer_id	txn_date	txn_type	txn_amount
integer	date	character varying (10)	integer
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

Query Writing & Execution

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

```
SELECT COUNT(DISTINCT node_id) AS num_of_node
FROM customer_nodes
```

num_of_node
bigint
5

3. How many customers are allocated to each region?

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

region_id	num_of_customer
integer	bigint
1	110
2	105
3	102
4	95
5	88

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

```
SELECT ROUND(COUNT(txn_type)/COUNT(DISTINCT(customer_id)),2) AS deposit_counts,
ROUND(AVG(txn_amount),2) AS avg_amount_deposit
FROM customer_transactions
WHERE txn_type = 'deposit'
```

deposit_counts	avg_amount_deposit
numeric	numeric
5.00	508.86

2. What is the number of nodes per region?

```
SELECT region_id, COUNT(node_id) AS num_of_node
FROM customer_nodes
GROUP BY region_id
ORDER BY region_id
```

region_id	num_of_node
integer	bigint
1	770
2	735
3	714
4	665
5	616

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

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

txn_type	unique_count	total_amount
character varying (10)	bigint	bigint
deposit	500	1359168
purchase	448	806537
withdrawal	439	793003



Query Writing & Execution

6. 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 count_deposit_by_month AS (
    SELECT
        customer_id,
        EXTRACT(MONTH FROM txn_date) AS month,
        COUNT(txn_type) AS count_deposit
    FROM
        customer_transactions
    WHERE
        txn_type = 'deposit'
    GROUP BY
        customer_id, month
),
count_withdrawal_by_month AS (
    SELECT
        customer_id,
        EXTRACT(MONTH FROM txn_date) AS month,
        COUNT(txn_type) AS count_withdrawal
    FROM
        customer_transactions
    WHERE
        txn_type = 'withdrawal'
    GROUP BY
        customer_id, month
),
count_purchase_by_month AS (
    SELECT
        customer_id,
        EXTRACT(MONTH FROM txn_date) AS month,
        COUNT(txn_type) AS count_purchase
    FROM
        customer_transactions
    WHERE
        txn_type = 'purchase'
    GROUP BY
        customer_id, month
)
SELECT
    count_deposit_by_month.customer_id,
    count_deposit_by_month.month,
    count_deposit_by_month.count_deposit,
    count_withdrawal_by_month.count_withdrawal,
    count_purchase_by_month.count_purchase
FROM
    count_deposit_by_month
LEFT JOIN
    count_withdrawal_by_month ON count_deposit_by_month.customer_id = count_withdrawal_by_month.customer_id
LEFT JOIN
    count_purchase_by_month ON count_deposit_by_month.customer_id = count_purchase_by_month.customer_id;
```

customer_id	month	count_deposit	count_withdrawal	count_purchase
5	3	2	2	3
5	3	2	1	3
5	3	2	1	3
6	1	2	1	4
6	1	2	1	1
6	1	2	1	1
6	1	2	2	4

7. Running customer balance column that includes the impact each transaction

```
WITH temp1 AS (
    SELECT
        customer_id,
        txn_date,
        txn_type,
        txn_amount,
        CASE
            WHEN txn_type = 'purchase' OR txn_type = 'withdrawal'
            THEN txn_amount * -1
            ELSE txn_amount
        END AS actual_transaction
    FROM
        customer_transactions
)
SELECT
    customer_id,
    txn_date,
    actual_transaction,
    SUM(actual_transaction) OVER (PARTITION BY customer_id ORDER BY txn_date) AS balance
FROM
    temp1
ORDER BY
    customer_id, txn_date;
```

customer_id	txn_date	actual_transaction	balance
1	2020-01-02	312	312
1	2020-03-05	-612	-300
1	2020-03-17	324	24
1	2020-03-19	-664	-640
2	2020-01-03	549	549
2	2020-03-24	61	610
3	2020-01-27	144	144
3	2020-02-22	-865	-821
3	2020-03-05	-213	-1034

8. Customer balance at the end of each month

```
WITH temp1 AS (
    SELECT
        customer_id,
        txn_date,
        txn_type,
        txn_amount,
        CASE
            WHEN txn_type = 'purchase' OR txn_type = 'withdrawal'
            THEN txn_amount * -1
            ELSE txn_amount
        END AS actual_transaction
    FROM
        customer_transactions
),
temp2 AS (
    SELECT
        customer_id,
        txn_date,
        actual_transaction,
        SUM(actual_transaction) OVER (PARTITION BY customer_id ORDER BY txn_date) AS balance
    FROM
        temp1
), -- Calculate the balance after each transaction
temp3 AS (
    SELECT
        *,
        MAX(txn_date) OVER (PARTITION BY customer_id, EXTRACT(MONTH FROM txn_date)) AS max_date
    FROM
        temp2
) -- Filter out the last transaction date for each user each month
SELECT
    customer_id,
    TO_CHAR(txn_date, 'Month') AS month,
    balance
FROM
    temp3
WHERE
    txn_date = max_date;
```

customer_id	month	balance
1	January	312
1	March	-640
2	January	549
2	March	610
3	January	144
3	February	-821
3	March	-1222
3	April	-729
4	January	848
4	March	655
5	January	954