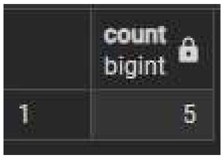
CASE STUDY #4

A. Customer Nodes Exploration

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

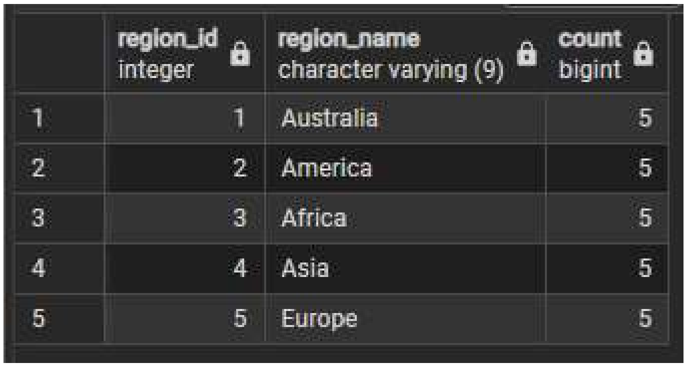
```
select count(distinct(node_id))
from customer_nodes as unique_nodes
```



	count bigint
1	5

– 2) What is the number of nodes per region?

```
select region_id, region_name, count(distinct(node_id))
from customer_nodes
join regions using (region_id)
group by region_id, region_name
```

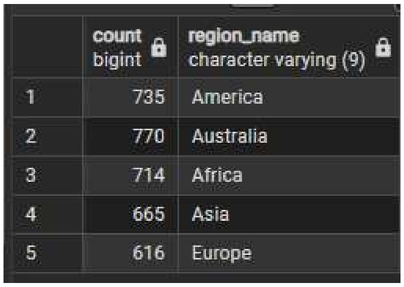


	region_id integer	region_name character varying (9)	count bigint
1	1	Australia	5
2	2	America	5
3	3	Africa	5
4	4	Asia	5
5	5	Europe	5

--3) How many customers are allocated to each region?

```
select count(customer_id), region_name
from regions
join customer_nodes using (region_id)
```

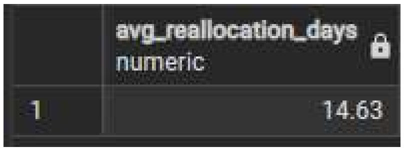
```
group by region_name
```



	count bigint	region_name character varying (9)
1	735	America
2	770	Australia
3	714	Africa
4	665	Asia
5	616	Europe

-- 4) How many days on average are customers reallocated to a different node?

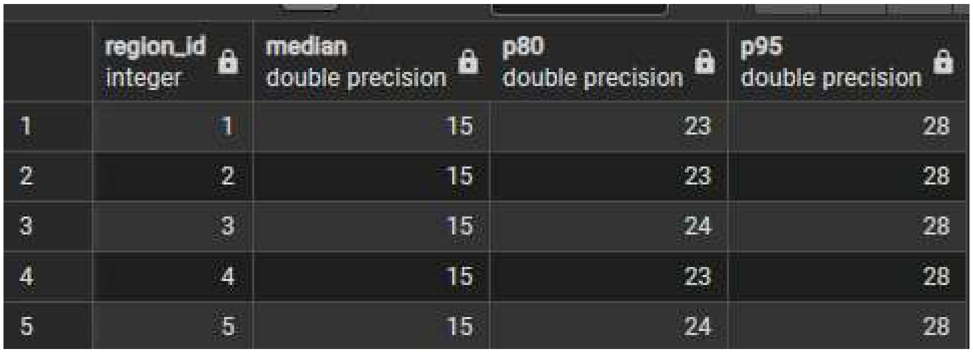
```
SELECT
 ROUND(AVG(end_date - start_date), 2) AS avg_reallocation_days
FROM customer_nodes
WHERE end_date IS NOT NULL
AND end_date <> '9999-12-31';
```



	avg_reallocation_days numeric
1	14.63

-- 5) What is the median, 80th and 95th percentile for this same reallocation days metric for each region?

```
WITH reallocation_days AS (
SELECT region_id, end_date - start_date AS days_diff
   FROM customer_nodes
   WHERE end_date <> DATE '9999-12-31'
)

SELECT
   region_id,
   PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY days_diff) AS
median,
   PERCENTILE_CONT(0.8) WITHIN GROUP (ORDER BY days_diff) AS
p80,
   PERCENTILE_CONT(0.95) WITHIN GROUP (ORDER BY days_diff) AS
p95
FROM reallocation_days
GROUP BY region_id
ORDER BY region_id;
```

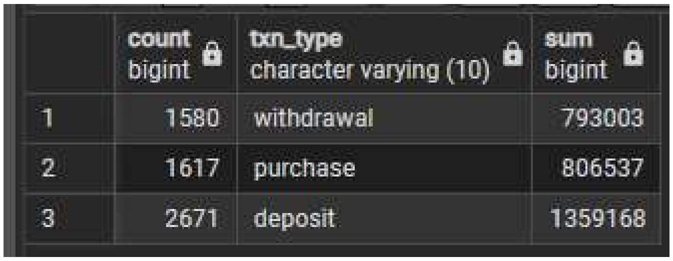


	region_id integer	median double precision	p80 double precision	p95 double precision
1	1	15	23	28
2	2	15	23	28
3	3	15	24	28
4	4	15	23	28
5	5	15	24	28

B. Customer Transactions

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

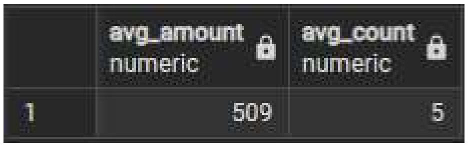
```
select distinct(count(customer_id)), txn_type,

sum(txn_amount)
from customer_transactions
group by txn_type
```



	count bigint	txn_type character varying (10)	sum bigint
1	1580	withdrawal	793003
2	1617	purchase	806537
3	2671	deposit	1359168

– 2) What is the average total historical deposit counts and amounts for all customers?

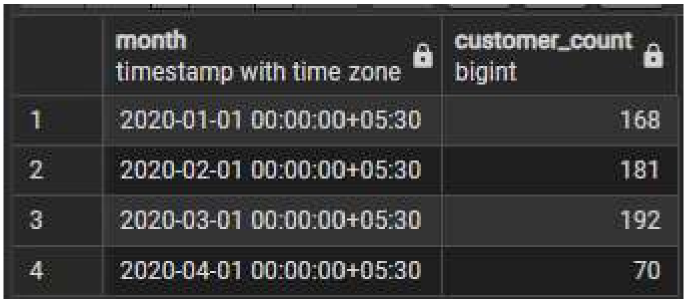
```
select round(avg(txn_amount)) as avg_amount,
ROUND(COUNT(*)::numeric / COUNT(DISTINCT customer_id), 0) as
avg_count
from customer_transactions
where txn_type = 'deposit'
```



	avg_amount numeric	avg_count numeric
1	509	5

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

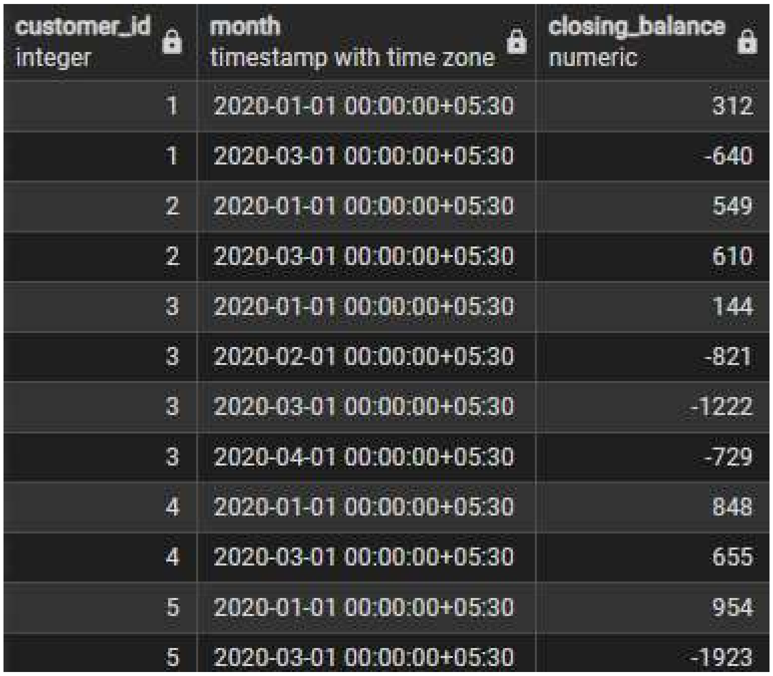
```
WITH monthly_activity AS (
   select
        customer_id,
        DATE_TRUNC('month', txn_date) AS month,
        SUM(CASE WHEN txn_type = 'deposit' THEN 1 ELSE 0 END) AS
deposit_count,
        SUM(CASE WHEN txn_type = 'purchase' THEN 1 ELSE 0 END) AS
purchase_count,
        SUM(CASE WHEN txn_type = 'withdrawal' THEN 1 ELSE 0 END) AS
withdrawal_count
    FROM customer_transactions
    GROUP BY customer_id, DATE_TRUNC('month', txn_date)
)

select
    month,
    COUNT(customer_id) AS customer_count
FROM monthly_activity
WHERE deposit_count > 1
  AND (purchase_count >= 1 OR withdrawal_count >= 1)
GROUP BY month
ORDER BY month;
```



	month timestamp with time zone	customer_count bigint
1	2020-01-01 00:00:00+05:30	168
2	2020-02-01 00:00:00+05:30	181
3	2020-03-01 00:00:00+05:30	192
4	2020-04-01 00:00:00+05:30	70

-- 4) What is the closing balance for each customer at the end of the month?

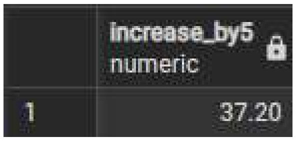
```
select customer_id, month,
    SUM(net_amount) OVER (
        PARTITION BY customer_id
        ORDER BY month
    ) AS closing_balance
FROM (
    select
        customer_id,
        DATE_TRUNC('month', txn_date) AS month,
        SUM(
            CASE
                WHEN txn_type = 'deposit' THEN txn_amount
                ELSE -txn_amount
            END
        ) AS net_amount
    FROM customer_transactions
    GROUP BY customer_id, DATE_TRUNC('month', txn_date)
) AS monthly_net
ORDER BY customer_id, month;
```



customer_id integer	month timestamp with time zone	closing_balance numeric
1	2020-01-01 00:00:00+05:30	312
1	2020-03-01 00:00:00+05:30	-640
2	2020-01-01 00:00:00+05:30	549
2	2020-03-01 00:00:00+05:30	610
3	2020-01-01 00:00:00+05:30	144
3	2020-02-01 00:00:00+05:30	-821
3	2020-03-01 00:00:00+05:30	-1222
3	2020-04-01 00:00:00+05:30	-729
4	2020-01-01 00:00:00+05:30	848
4	2020-03-01 00:00:00+05:30	655
5	2020-01-01 00:00:00+05:30	954
5	2020-03-01 00:00:00+05:30	-1923

-- 5) What is the percentage of customers who increase their closing balance by more than 5%?

```
WITH monthly_hst AS (
    SELECT
        customer_id,
        EXTRACT(YEAR FROM txn_date) AS txn_year,
        EXTRACT(MONTH FROM txn_date) AS txn_month,
        SUM(CASE WHEN txn_type = 'deposit' THEN txn_amount ELSE 0 END)
AS deposit_total,
        SUM(CASE WHEN txn_type = 'purchase' THEN txn_amount ELSE 0
END) AS purchase_total,
        SUM(CASE WHEN txn_type = 'withdrawal' THEN txn_amount ELSE 0
END) AS withdrawal_total
    FROM customer_transactions
    GROUP BY customer_id, EXTRACT(YEAR FROM txn_date),
EXTRACT(MONTH FROM txn_date)
),

monthly_balance AS (
    SELECT
        customer_id,
        txn_year,
        txn_month,
        deposit_total - (purchase_total + withdrawal_total) AS closing_amount,
        LAG(deposit_total - (purchase_total + withdrawal_total))
            OVER (PARTITION BY customer_id ORDER BY txn_year, txn_month)
AS prev_closing_amount
    FROM monthly_hst
)

SELECT

round(count( customer_id) * 100.0/
(SELECT COUNT( Distinct customer_id)
FROM customer_transactions),2) as increase_by5

from monthly_balance
where prev_closing_amount is not null and
(closing_amount - prev_closing_amount)/prev_closing_amount >0.05
```



```
     increase_by5
     numeric
1           37.20
```