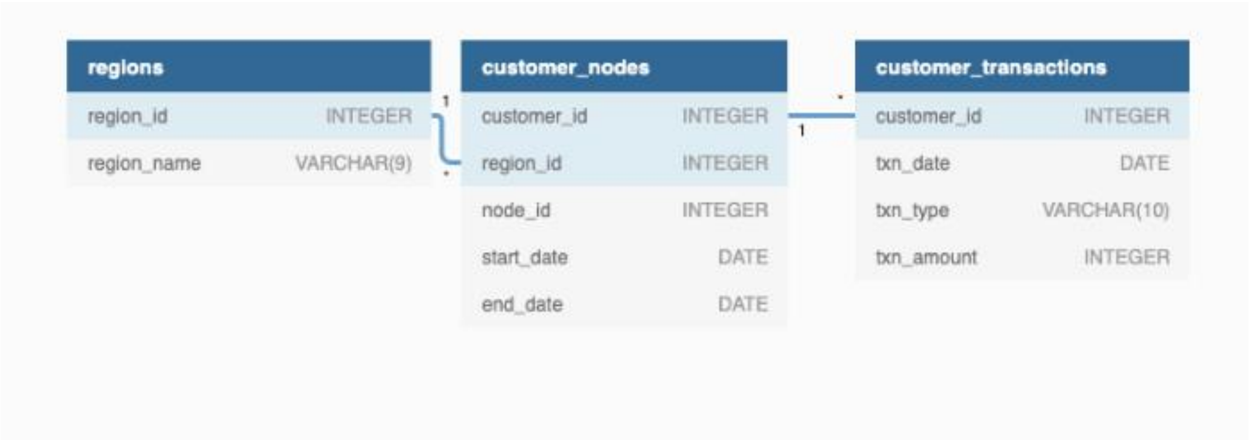
Name: Sumit Chauhan

Department: Data Engineering

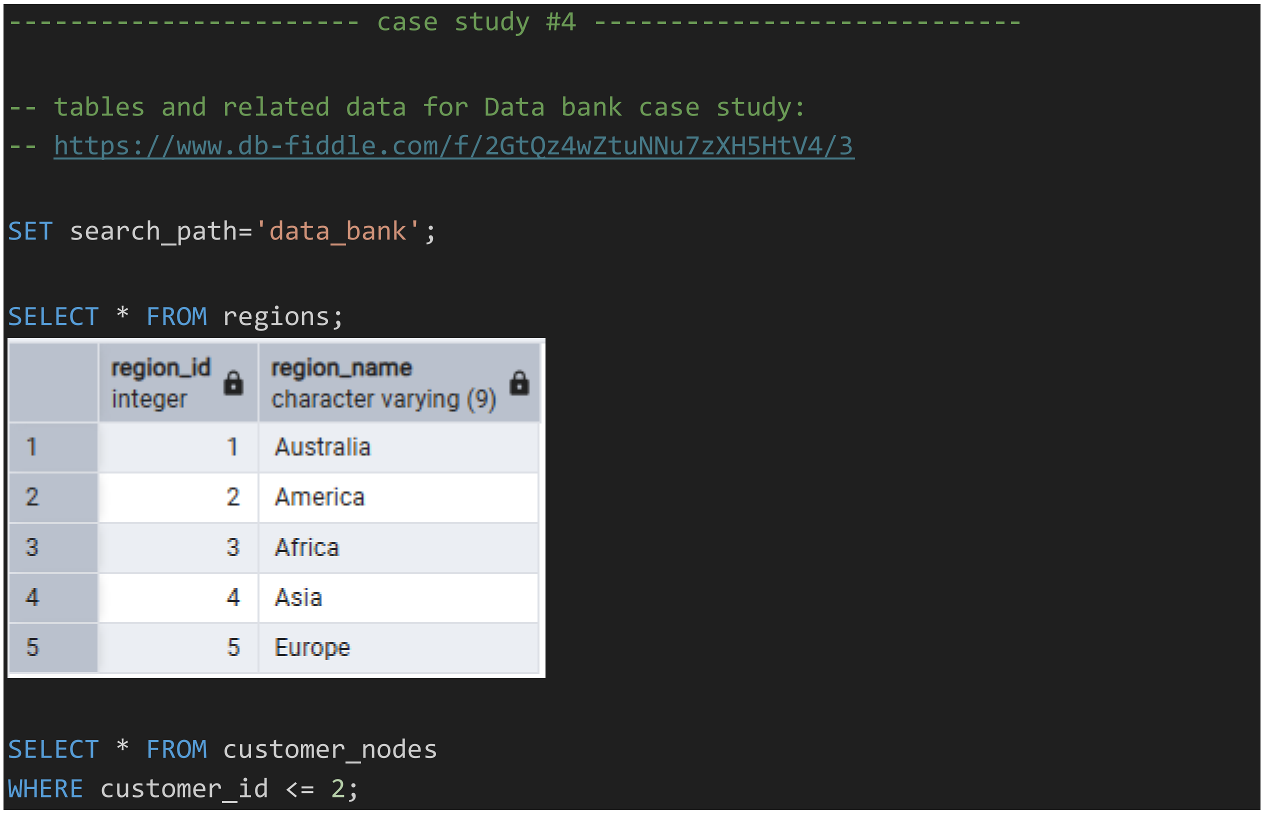
SQL 8 Week Challenge

Case Study #4

(Data bank)



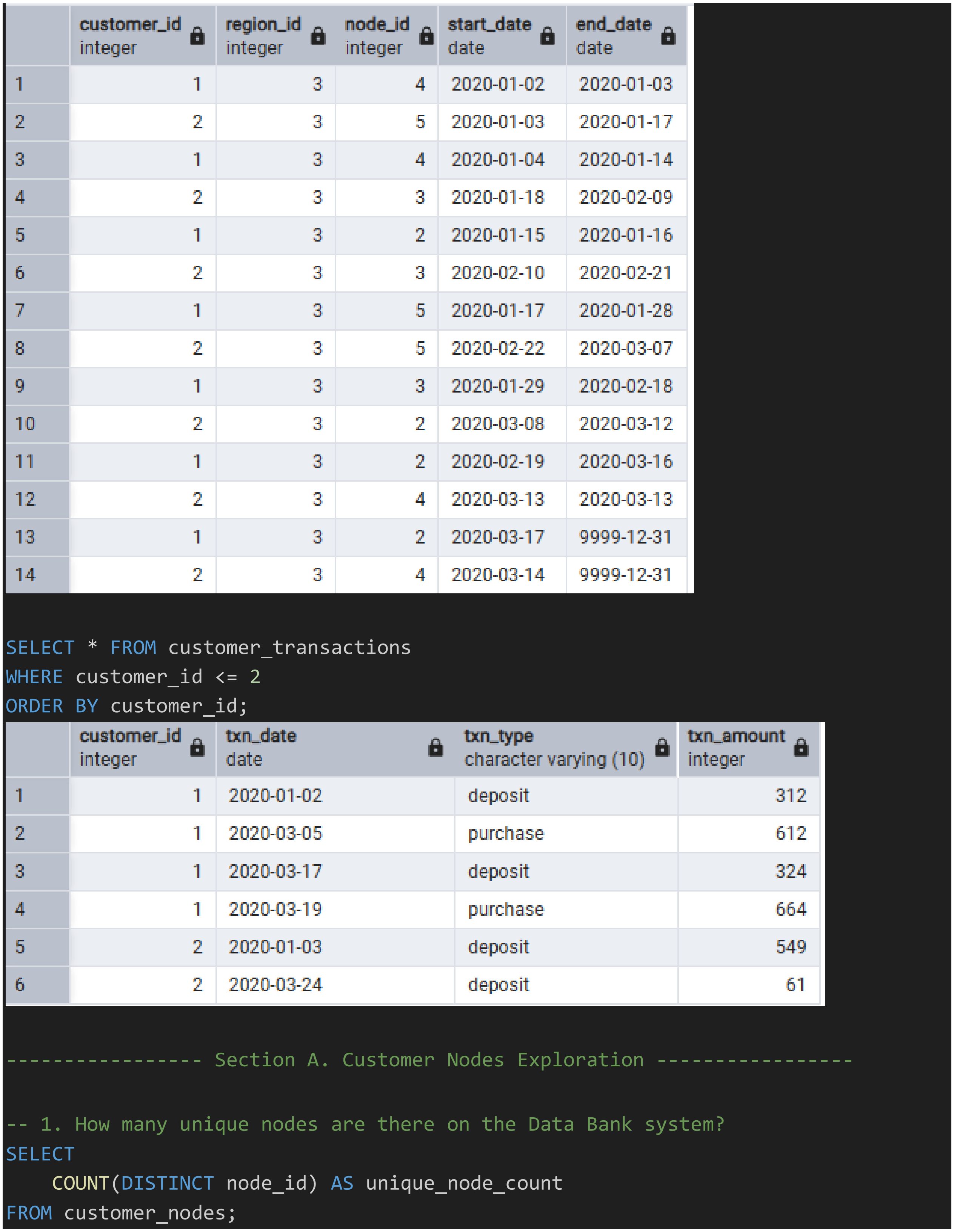
.sql file:



```
----------------------- case study #4 ----------------------------

-- tables and related data for Data bank case study:
-- https://www.db-fiddle.com/f/2GtQz4wZtuNNu7zXH5HtV4/3

SET search_path='data_bank';

SELECT * FROM regions;
```

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

```
SELECT * FROM customer_nodes
WHERE customer_id <= 2;
```

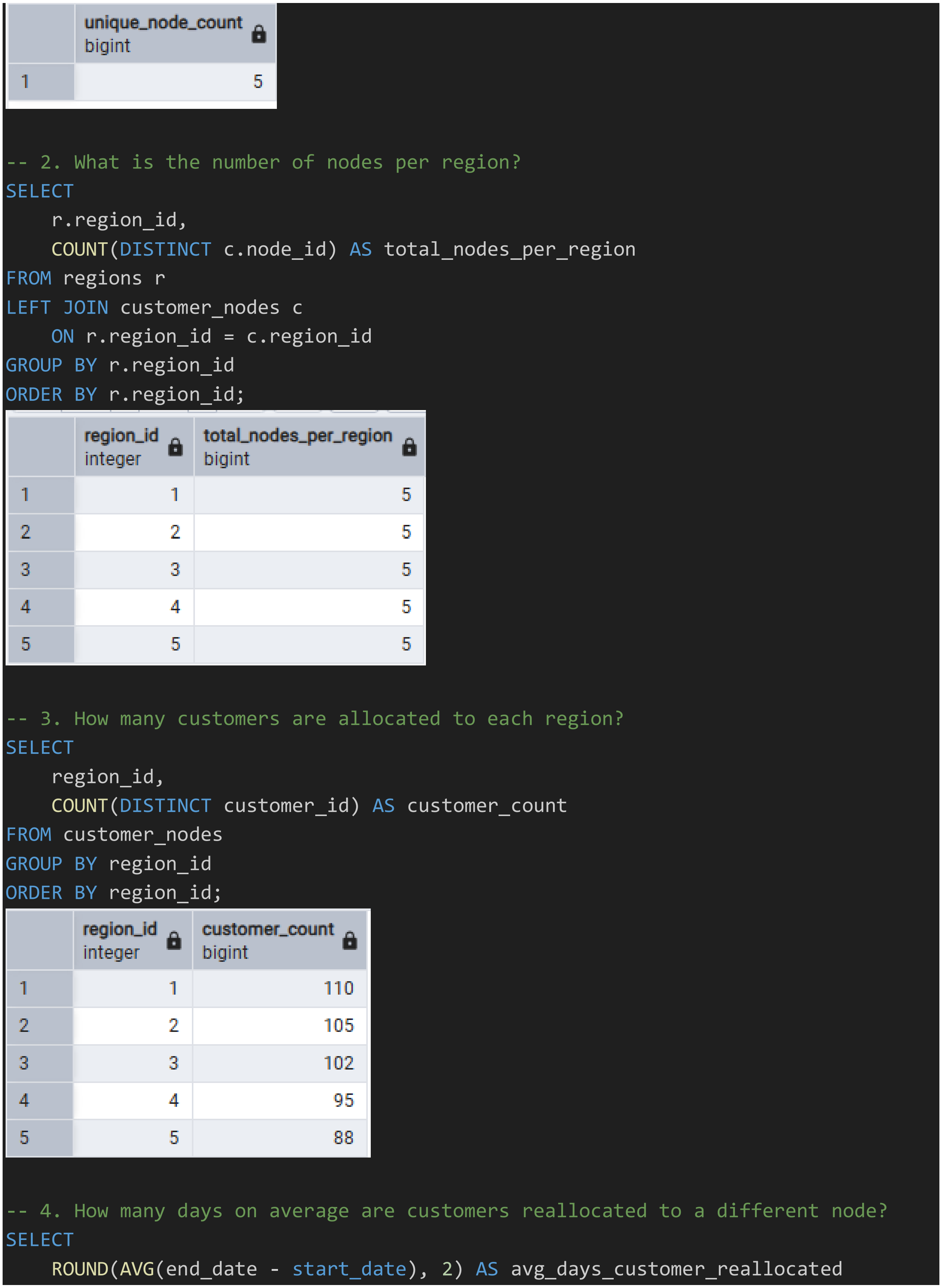


	customer_id integer	region_id integer	node_id integer	start_date date	end_date date
1	1	3	4	2020-01-02	2020-01-03
2	2	3	5	2020-01-03	2020-01-17
3	1	3	4	2020-01-04	2020-01-14
4	2	3	3	2020-01-18	2020-02-09
5	1	3	2	2020-01-15	2020-01-16
6	2	3	3	2020-02-10	2020-02-21
7	1	3	5	2020-01-17	2020-01-28
8	2	3	5	2020-02-22	2020-03-07
9	1	3	3	2020-01-29	2020-02-18
10	2	3	2	2020-03-08	2020-03-12
11	1	3	2	2020-02-19	2020-03-16
12	2	3	4	2020-03-13	2020-03-13
13	1	3	2	2020-03-17	9999-12-31
14	2	3	4	2020-03-14	9999-12-31

```sql
SELECT * FROM customer_transactions
WHERE customer_id <= 2
ORDER BY customer_id;
```

	customer_id integer	txn_date date	txn_type character varying (10)	txn_amount integer
1	1	2020-01-02	deposit	312
2	1	2020-03-05	purchase	612
3	1	2020-03-17	deposit	324
4	1	2020-03-19	purchase	664
5	2	2020-01-03	deposit	549
6	2	2020-03-24	deposit	61

```sql
----------------- Section A. Customer Nodes Exploration -----------------

-- 1. How many unique nodes are there on the Data Bank system?
SELECT
    COUNT(DISTINCT node_id) AS unique_node_count
FROM customer_nodes;
```



	unique_node_count bigint
1	5

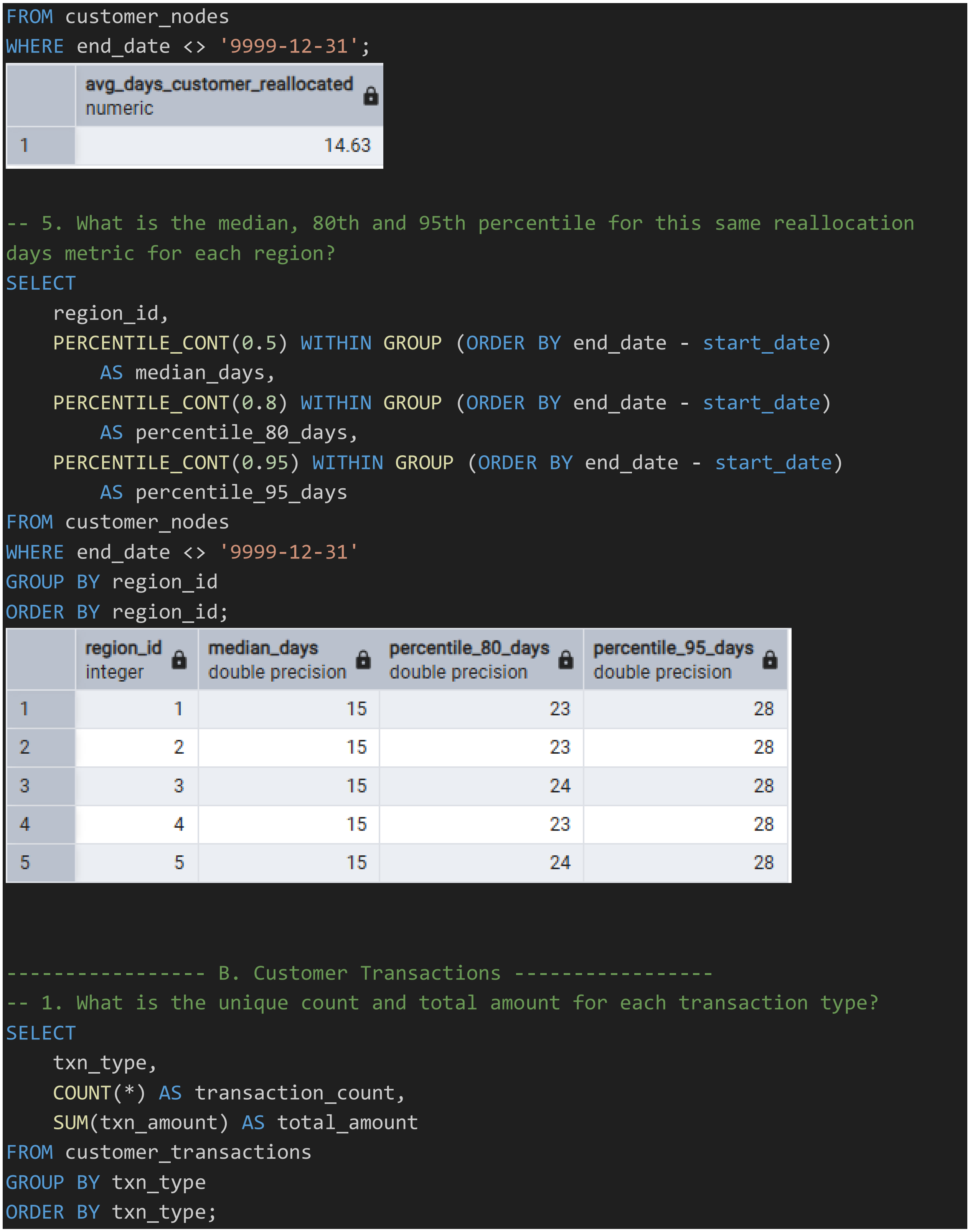
```sql
-- 2. What is the number of nodes per region?
SELECT
    r.region_id,
    COUNT(DISTINCT c.node_id) AS total_nodes_per_region
FROM regions r
LEFT JOIN customer_nodes c
    ON r.region_id = c.region_id
GROUP BY r.region_id
ORDER BY r.region_id;
```

	region_id integer	total_nodes_per_region bigint
1	1	5
2	2	5
3	3	5
4	4	5
5	5	5

```sql
-- 3. How many customers are allocated to each region?
SELECT
    region_id,
    COUNT(DISTINCT customer_id) AS customer_count
FROM customer_nodes
GROUP BY region_id
ORDER BY region_id;
```

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

```sql
-- 4. How many days on average are customers reallocated to a different node?
SELECT
    ROUND(AVG(end_date - start_date), 2) AS avg_days_customer_reallocated
```



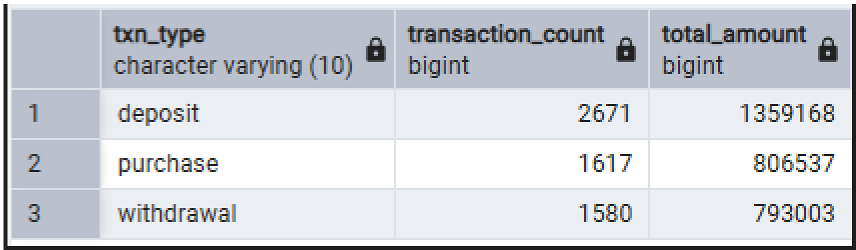
```sql
FROM customer_nodes
WHERE end_date <> '9999-12-31';
```

	avg_days_customer_reallocated numeric
1	14.63

```sql
-- 5. What is the median, 80th and 95th percentile for this same reallocation
days metric for each region?
SELECT
    region_id,
    PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY end_date - start_date)
        AS median_days,
    PERCENTILE_CONT(0.8) WITHIN GROUP (ORDER BY end_date - start_date)
        AS percentile_80_days,
    PERCENTILE_CONT(0.95) WITHIN GROUP (ORDER BY end_date - start_date)
        AS percentile_95_days
FROM customer_nodes
WHERE end_date <> '9999-12-31'
GROUP BY region_id
ORDER BY region_id;
```

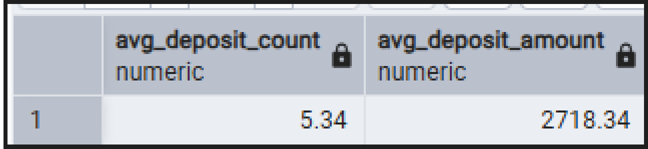
	region_id integer	median_days double precision	percentile_80_days double precision	percentile_95_days double precision
1	1	15	23	28
2	2	15	23	28
3	3	15	24	28
4	4	15	23	28
5	5	15	24	28

```sql
----------------- B. Customer Transactions -----------------
-- 1. What is the unique count and total amount for each transaction type?
SELECT
    txn_type,
    COUNT(*) AS transaction_count,
    SUM(txn_amount) AS total_amount
FROM customer_transactions
GROUP BY txn_type
ORDER BY txn_type;
```



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

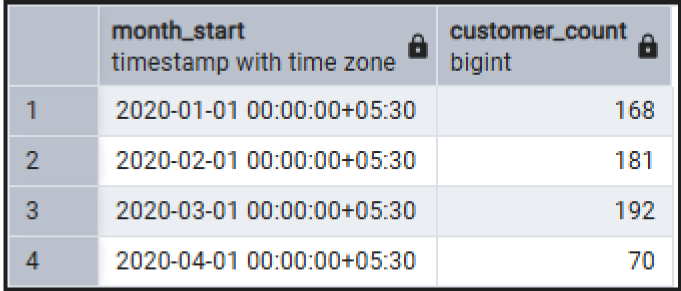
```
-- 2. What is the average total historical deposit counts and amounts for all
customers?
WITH customer_deposits AS (
    SELECT
        customer_id,
        COUNT(*) AS deposit_count,
        SUM(txn_amount) AS deposit_amount
    FROM customer_transactions
    WHERE txn_type = 'deposit'
    GROUP BY customer_id
)

SELECT
    ROUND(AVG(deposit_count), 2) AS avg_deposit_count,
    ROUND(AVG(deposit_amount), 2) AS avg_deposit_amount
FROM customer_deposits;
```



	avg_deposit_count numeric	avg_deposit_amount numeric
1	5.34	2718.34

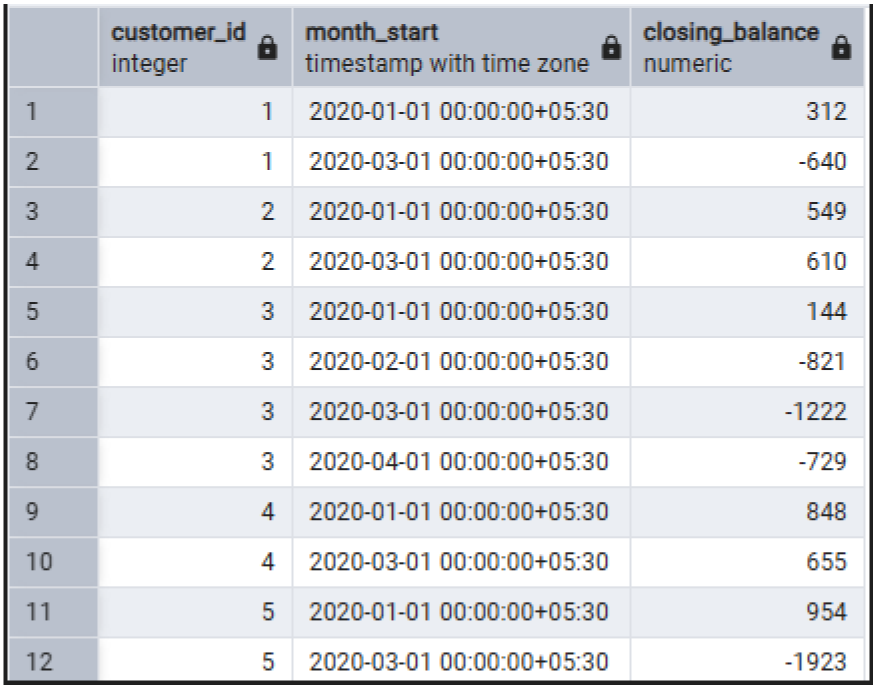
```
-- 3. For each month - how many Data Bank customers make more than 1 deposit and
either 1 purchase or 1 withdrawal in a single month?
WITH monthly_summary AS (
    SELECT
        customer_id,
        DATE_TRUNC('month', txn_date) AS month_start,
        SUM(CASE WHEN txn_type = 'deposit' THEN 1 ELSE 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
```

```sql
    month_start,
    COUNT(*) AS customer_count
FROM monthly_summary
WHERE deposit_count > 1
  AND (purchase_count >= 1 OR withdrawal_count >= 1)
GROUP BY month_start
ORDER BY month_start;
```



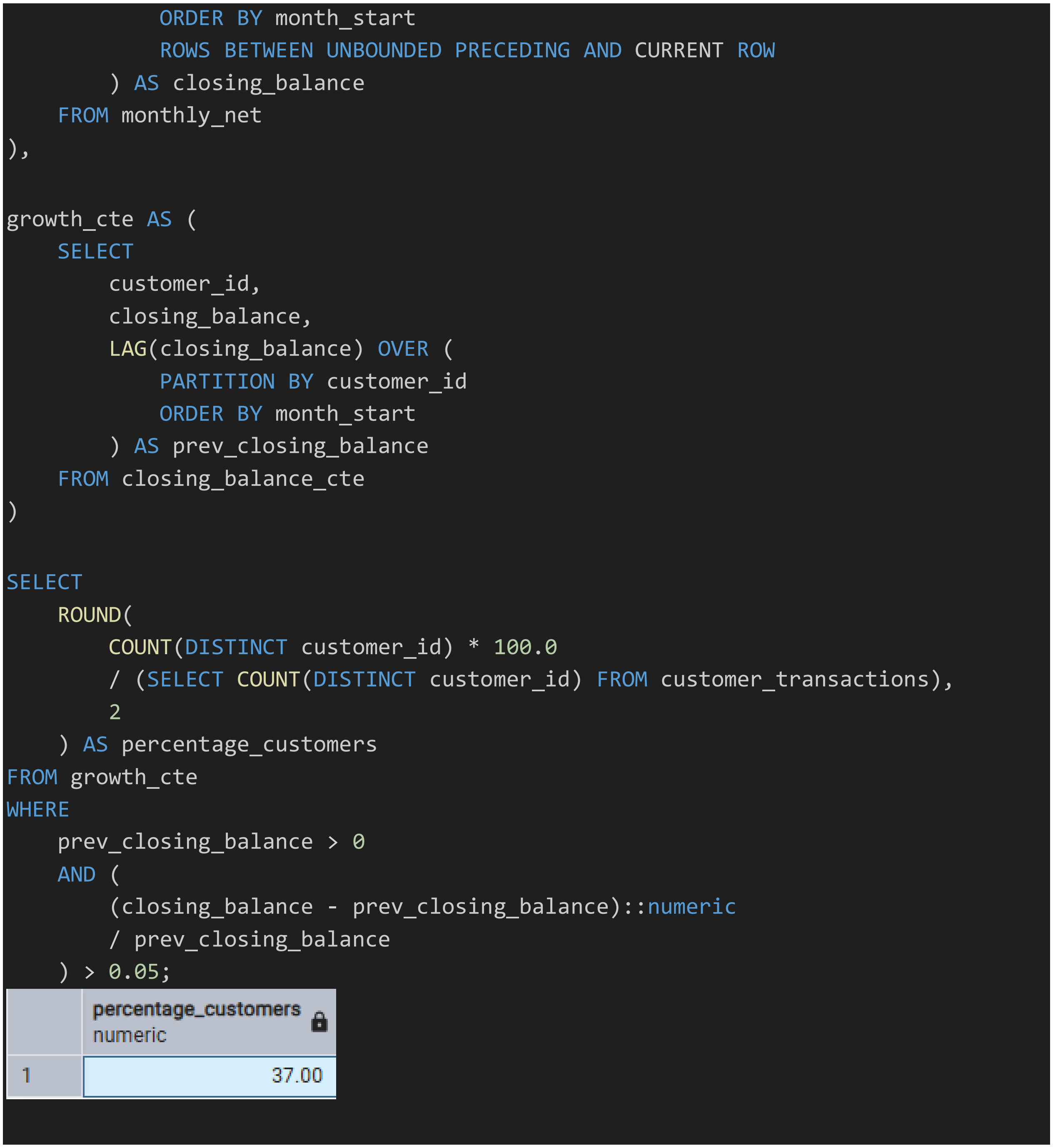
	month_start timestamp with time zone	customer_count bigint
1	2020-01-01 00:00:00+05:30	168
2	2020-02-01 00:00:00+05:30	181
3	2020-03-01 00:00:00+05:30	192
4	2020-04-01 00:00:00+05:30	70

```sql
-- 4. What is the closing balance for each customer at the end of the month?
WITH monthly_net AS (
    SELECT
        customer_id,
        DATE_TRUNC('month', txn_date) AS month_start,
        SUM(
            CASE
                WHEN txn_type = 'deposit' THEN txn_amount
                WHEN txn_type IN ('withdrawal', 'purchase') THEN -txn_amount
            END
        ) AS monthly_change
    FROM customer_transactions
    GROUP BY customer_id, DATE_TRUNC('month', txn_date)
)

SELECT
    customer_id,
    month_start,
    SUM(monthly_change) OVER (
        PARTITION BY customer_id
        ORDER BY month_start
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS closing_balance
FROM monthly_net
ORDER BY customer_id, month_start;
```



	customer_id integer	month_start timestamp with time zone	closing_balance numeric
1	1	2020-01-01 00:00:00+05:30	312
2	1	2020-03-01 00:00:00+05:30	-640
3	2	2020-01-01 00:00:00+05:30	549
4	2	2020-03-01 00:00:00+05:30	610
5	3	2020-01-01 00:00:00+05:30	144
6	3	2020-02-01 00:00:00+05:30	-821
7	3	2020-03-01 00:00:00+05:30	-1222
8	3	2020-04-01 00:00:00+05:30	-729
9	4	2020-01-01 00:00:00+05:30	848
10	4	2020-03-01 00:00:00+05:30	655
11	5	2020-01-01 00:00:00+05:30	954
12	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_net AS (
    SELECT
        customer_id,
        DATE_TRUNC('month', txn_date) AS month_start,
        SUM(
            CASE
                WHEN txn_type = 'deposit' THEN txn_amount
                WHEN txn_type IN ('withdrawal', 'purchase') THEN -txn_amount
            END
        ) AS monthly_change
    FROM customer_transactions
    GROUP BY customer_id, DATE_TRUNC('month', txn_date)
),

closing_balance_cte AS (
    SELECT
        customer_id,
        month_start,
        SUM(monthly_change) OVER (
            PARTITION BY customer_id
```



```sql
            ORDER BY month_start
            ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
        ) AS closing_balance
    FROM monthly_net
),

growth_cte AS (
    SELECT
        customer_id,
        closing_balance,
        LAG(closing_balance) OVER (
            PARTITION BY customer_id
            ORDER BY month_start
        ) AS prev_closing_balance
    FROM closing_balance_cte
)

SELECT
    ROUND(
        COUNT(DISTINCT customer_id) * 100.0
        / (SELECT COUNT(DISTINCT customer_id) FROM customer_transactions),
        2
    ) AS percentage_customers
FROM growth_cte
WHERE
    prev_closing_balance > 0
    AND (
        (closing_balance - prev_closing_balance)::numeric
        / prev_closing_balance
    ) > 0.05;
```

	percentage_customers numeric
1	37.00