Analysis of the Data of a Digital Banking Platform using SQL.

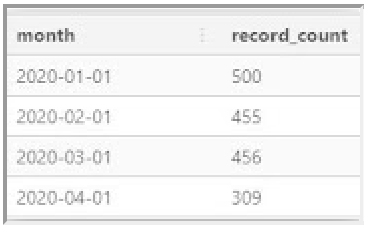
The tables in the dataset include the customer transactions table, regional table and customer table.

Data Analysis of the Dataset

- Range of months in the dataset

```
SELECT
  DATE_TRUNC('month', txn_date)::DATE AS month,
  COUNT(DISTINCT customer_id) AS record_count
FROM data_bank.customer_transactions
GROUP BY month
ORDER BY month;
```

Output:

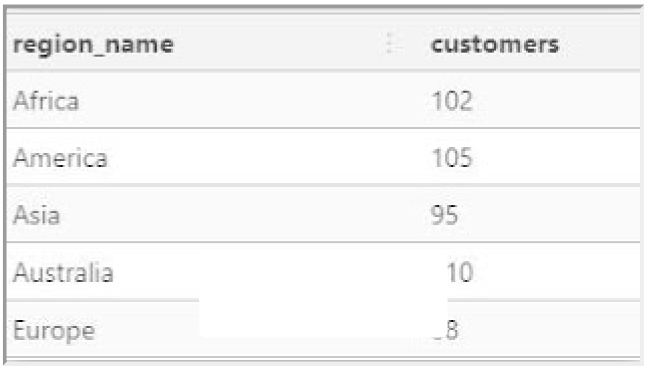


month	record_count
2020-01-01	500
2020-02-01	455
2020-03-01	456
2020-04-01	309

- Customers in each region

```
SELECT
  region_name,
  COUNT(distinct customer_nodes.customer_id) AS customers
FROM data_bank.customer_nodes
INNER JOIN data_bank.regions
  ON customer_nodes.region_id = regions.region_id
GROUP BY region_name
ORDER BY region_name;
```

Output:

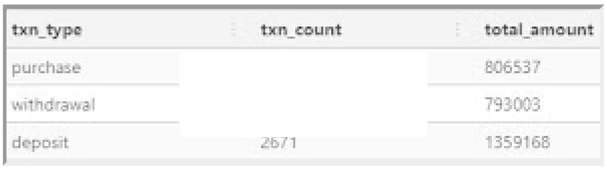


region_name	customers
Africa	102
America	105
Asia	95
Australia	10
Europe	8

- Unique count and total amount for each transaction type

```
SELECT
txn_type,
COUNT(*) AS txn_count,
SUM(txn_amount) AS total_amount
FROM data_bank.customer_transactions
GROUP BY txn_type;
```

Output:

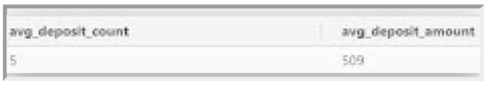


txn_type	txn_count	total_amount
purchase		806537
withdrawal		793003
deposit	2671	1359168

- Average total historical deposit counts and amounts for all customers

```
WITH cte_customer AS (
SELECT
customer_id,
COUNT(*) AS deposit_count,
SUM(txn_amount) AS total_deposit_amount
FROM data_bank.customer_transactions
WHERE txn_type = 'deposit'
GROUP BY customer_id
)
SELECT
ROUND(AVG(deposit_count)) AS avg_deposit_count,
ROUND(SUM(total_deposit_amount) / SUM(deposit_count)) AS avg_deposit_amount
FROM cte_customer;
```

Output:

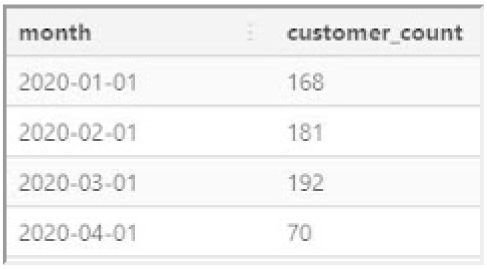


avg_deposit_count	avg_deposit_amount
5	509

- Customers who make more than 1 deposit and at least either 1 purchase or 1 withdrawal in a single month in each month

```
WITH cte_customer_months AS (
SELECT
DATE_TRUNC('month', txn_date)::DATE AS month,
customer_id,
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_coun
FROM data_bank.customer_transactions
GROUP BY month, customer_id
)
SELECT
month,
COUNT(DISTINCT customer_id) AS customer_count
FROM cte_customer_months
WHERE deposit_count > 1 AND (
purchase_count >= 1 OR withdrawal_count >= 1
)
GROUP BY month
ORDER BY month;
```

Output:

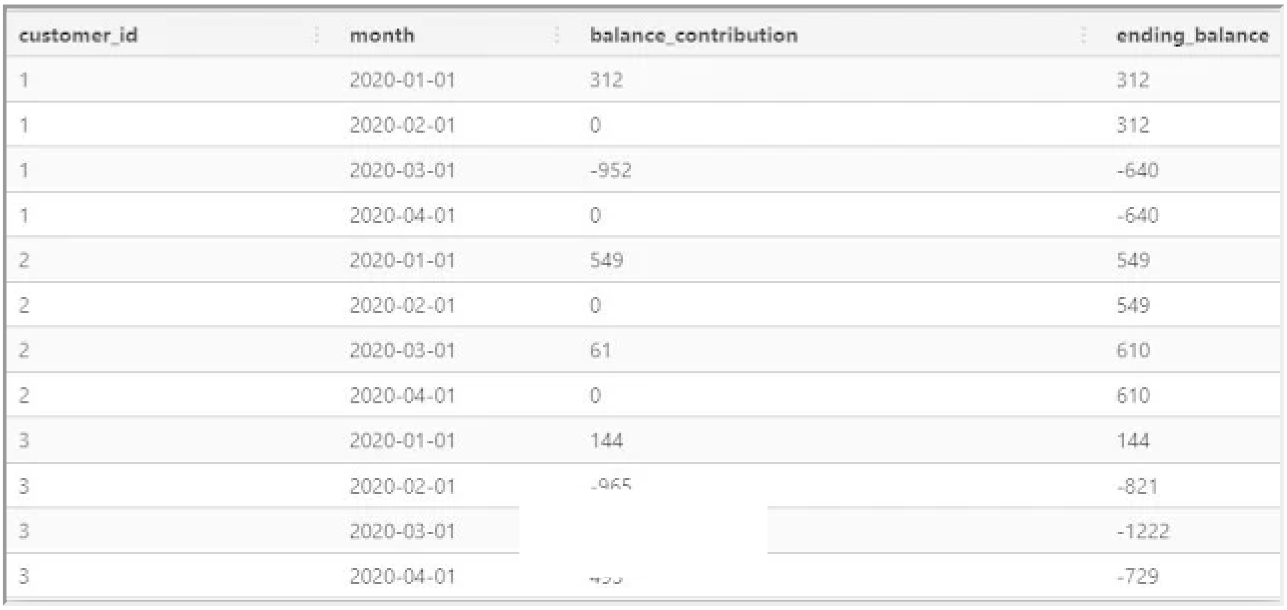


month	customer_count
2020-01-01	168
2020-02-01	181
2020-03-01	192
2020-04-01	70

- The closing balance for each customer at the end of the month

```
WITH cte_monthly_balances AS (
  SELECT
    customer_id,
    DATE_TRUNC('mon', txn_date)::DATE AS month,
    SUM(
      CASE
        WHEN txn_type = 'deposit' THEN txn_amount
        ELSE (-txn_amount)
        END
    ) AS balance
  FROM data_bank.customer_transactions
  GROUP BY customer_id, month
  ORDER BY customer_id, month
),
cte_generated_months AS (
  SELECT
    DISTINCT customer_id,
    (
      '2020-01-01'::DATE +
      GENERATE_SERIES(0, 3) * INTERVAL '1 MONTH'
    )::DATE AS month
  FROM data_bank.customer_transactions
)
SELECT
  cte_generated_months.customer_id,
  cte_generated_months.month,
  COALESCE(cte_monthly_balances.balance, 0) AS balance_contribution,
  SUM(cte_monthly_balances.balance) OVER (
    PARTITION BY cte_generated_months.customer_id
    ORDER BY cte_generated_months.month
    ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
  ) AS ending_balance
FROM cte_generated_months
LEFT JOIN cte_monthly_balances
  ON cte_generated_months.month = cte_monthly_balances.month
  AND cte_generated_months.customer_id = cte_monthly_balances.customer_id
WHERE cte_generated_months.customer_id BETWEEN 1 and 3;
```

Output:



customer_id	month	balance_contribution	ending_balance
1	2020-01-01	312	312
1	2020-02-01	0	312
1	2020-03-01	-952	-640
1	2020-04-01	0	-640
2	2020-01-01	549	549
2	2020-02-01	0	549
2	2020-03-01	61	610
2	2020-04-01	0	610
3	2020-01-01	144	144
3	2020-02-01	-965	-821
3	2020-03-01	[illegible]	-1222
3	2020-04-01	493	-729