# GCash:

## User Engagement Analysis

Presented by: Gabriel Leoj Rosales

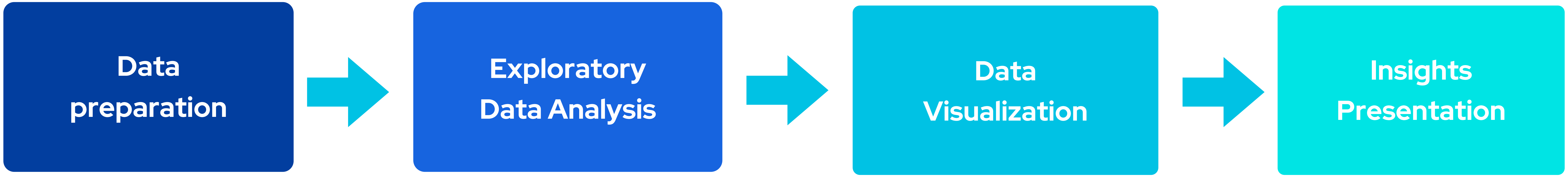
# Objective

Understand user engagement patterns to identify key opportunities for improving user experience and retention.

# Scope

- Analyze app usage and transaction behaviors
- Segment users based on engagement
- Identify trends by location, transaction type, and merchant
- Perform cohort analysis

# Steps taken for analysis



Data preparation

Exploratory Data Analysis

Data Visualization

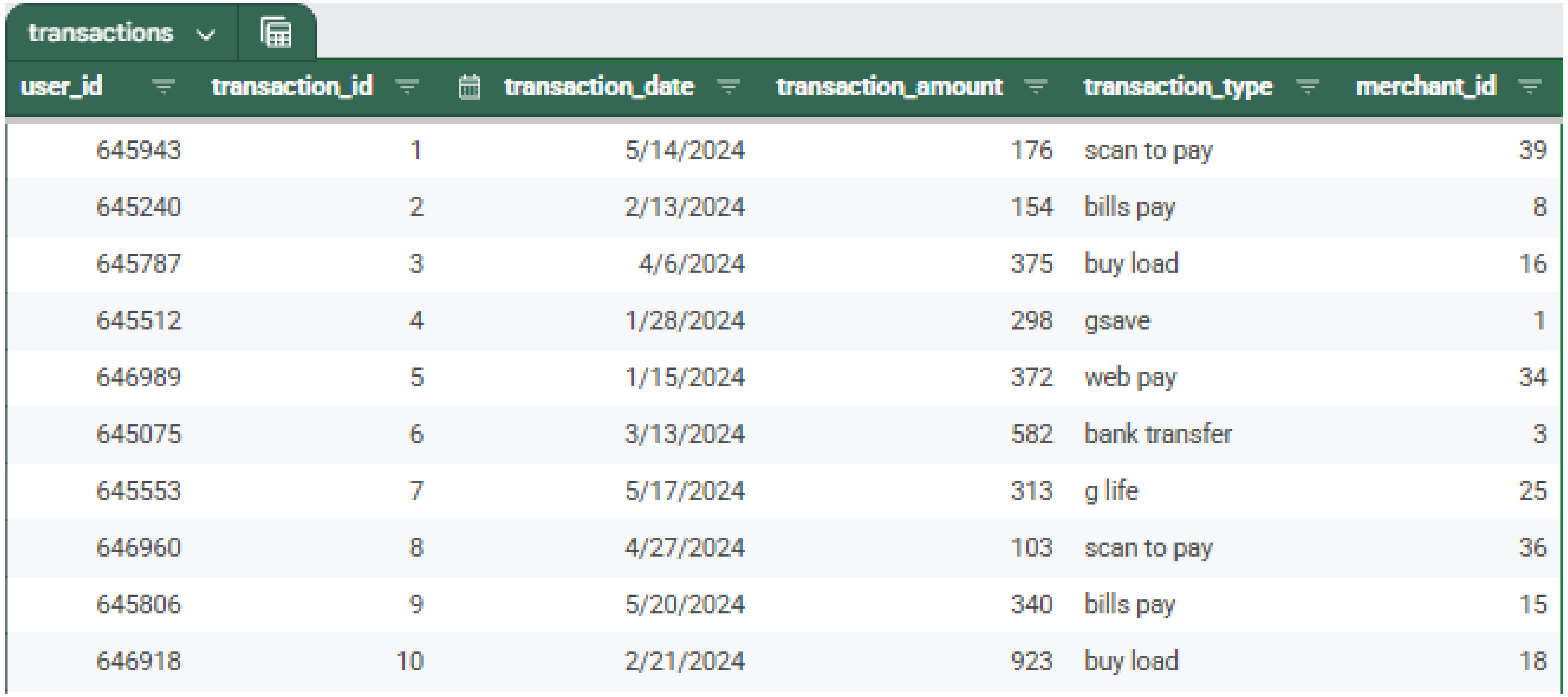
Insights Presentation

## Process Flow

# Data Preparation

# Datasets Used

Transactions
(raw data)



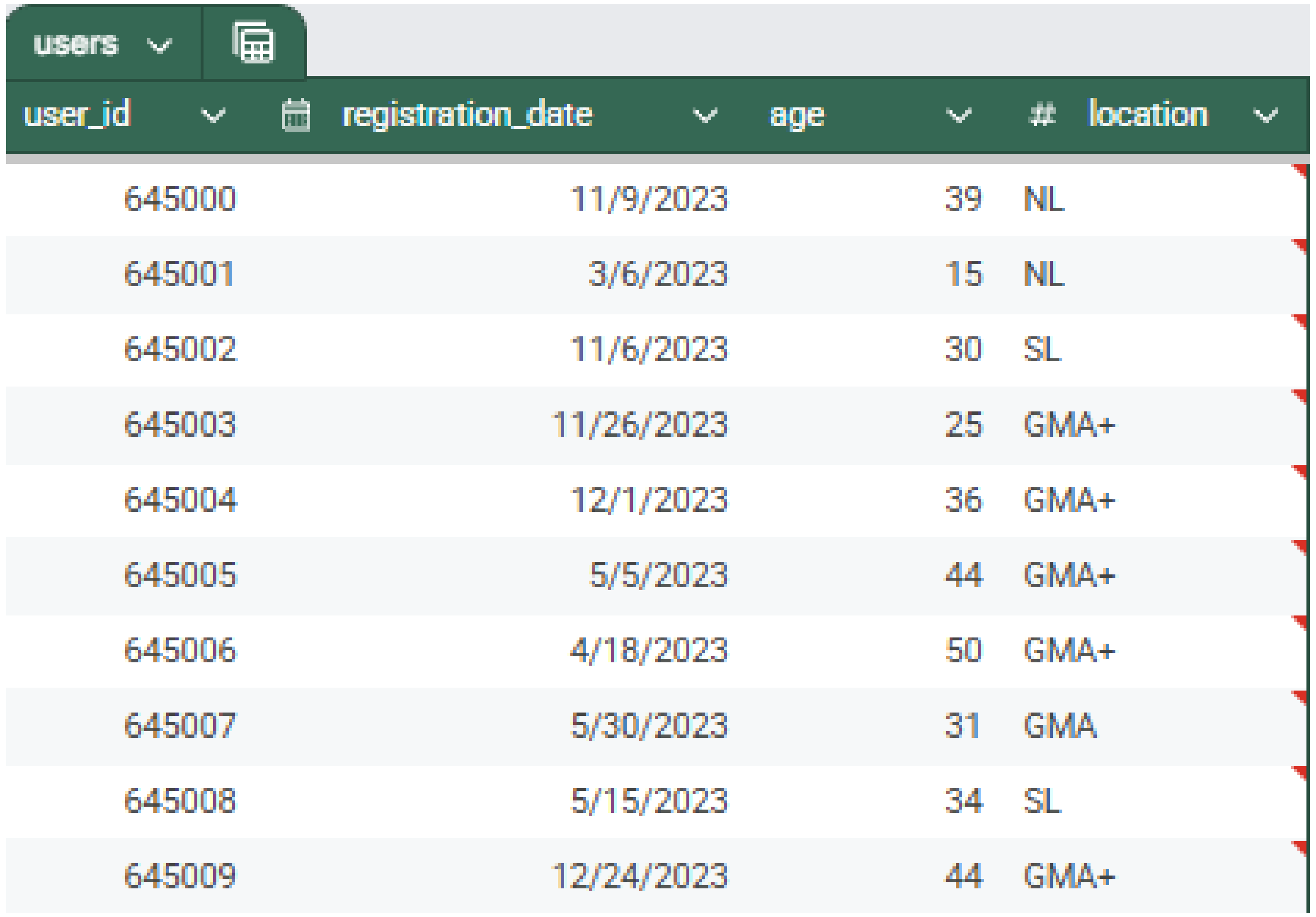
transactions

| user_id | transaction_id | transaction_date | transaction_amount | transaction_type | merchant_id |
|---|---|---|---|---|---|
| 645943 | 1 | 5/14/2024 | 176 | scan to pay | 39 |
| 645240 | 2 | 2/13/2024 | 154 | bills pay | 8 |
| 645787 | 3 | 4/6/2024 | 375 | buy load | 16 |
| 645512 | 4 | 1/28/2024 | 298 | gsave | 1 |
| 646989 | 5 | 1/15/2024 | 372 | web pay | 34 |
| 645075 | 6 | 3/13/2024 | 582 | bank transfer | 3 |
| 645553 | 7 | 5/17/2024 | 313 | g life | 25 |
| 646960 | 8 | 4/27/2024 | 103 | scan to pay | 36 |
| 645806 | 9 | 5/20/2024 | 340 | bills pay | 15 |
| 646918 | 10 | 2/21/2024 | 923 | buy load | 18 |

Data
Preparation

# Datasets Used

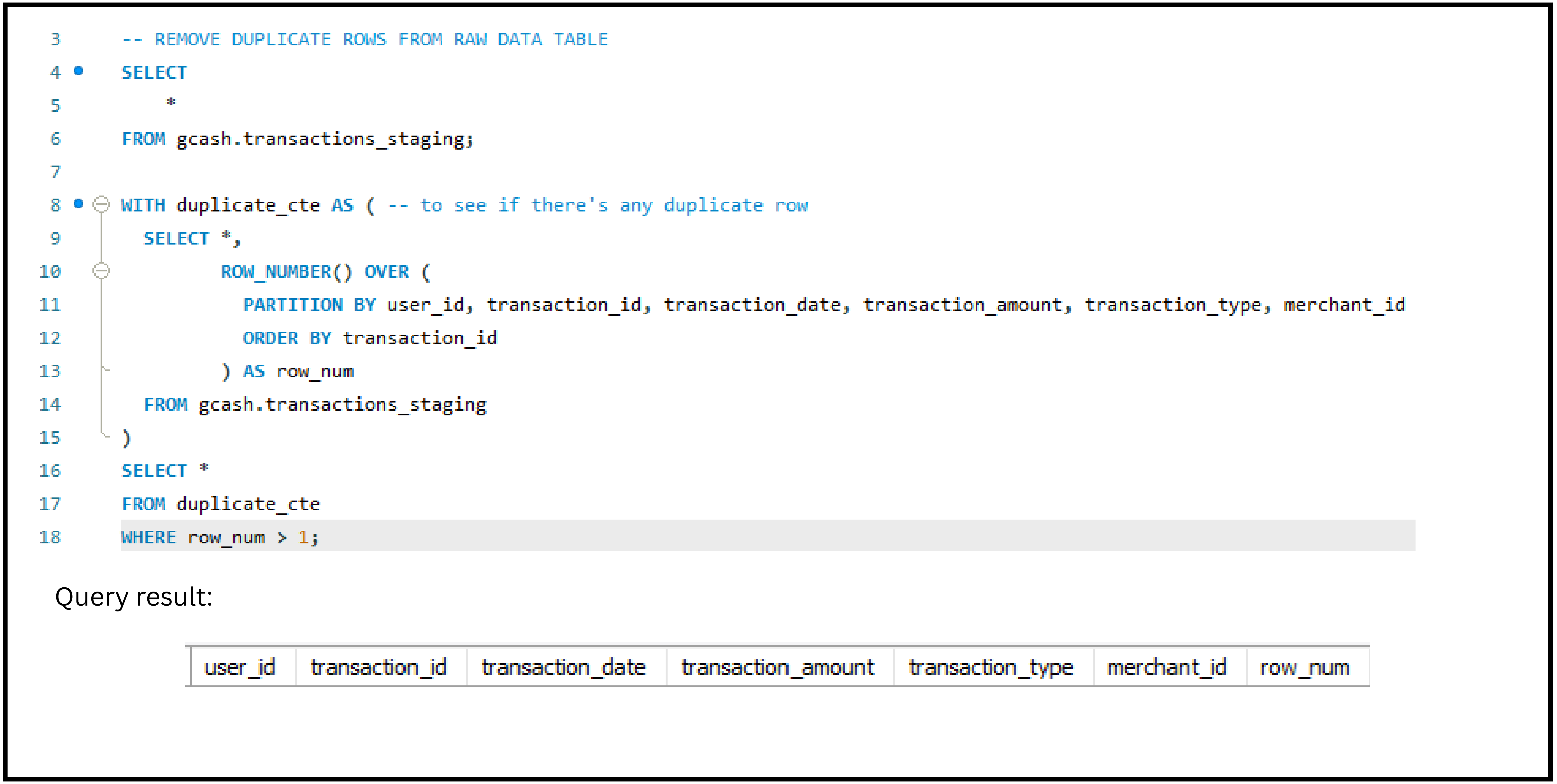
Users



| user_id | registration_date | age | location |
|---|---|---|---|
| 645000 | 11/9/2023 | 39 | NL |
| 645001 | 3/6/2023 | 15 | NL |
| 645002 | 11/6/2023 | 30 | SL |
| 645003 | 11/26/2023 | 25 | GMA+ |
| 645004 | 12/1/2023 | 36 | GMA+ |
| 645005 | 5/5/2023 | 44 | GMA+ |
| 645006 | 4/18/2023 | 50 | GMA+ |
| 645007 | 5/30/2023 | 31 | GMA |
| 645008 | 5/15/2023 | 34 | SL |
| 645009 | 12/24/2023 | 44 | GMA+ |

Data Preparation

# 1. Remove Duplicates



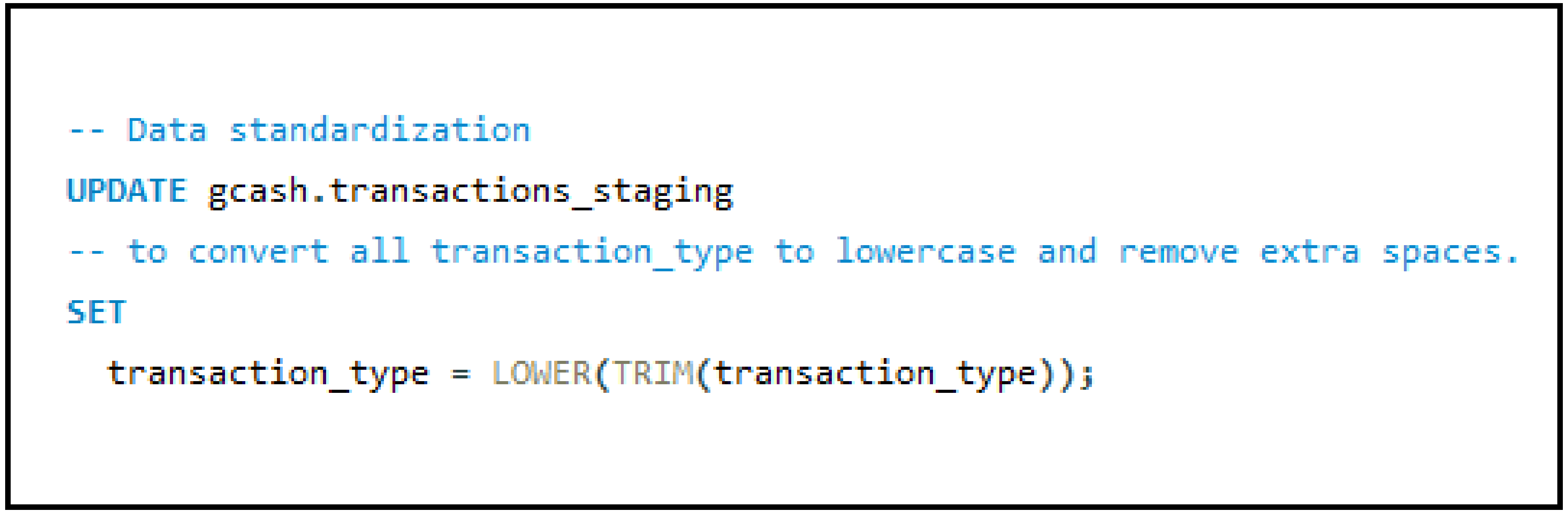
```
-- REMOVE DUPLICATE ROWS FROM RAW DATA TABLE
SELECT
    *
FROM gcash.transactions_staging;

WITH duplicate_cte AS ( -- to see if there's any duplicate row
  SELECT *,
        ROW_NUMBER() OVER (
          PARTITION BY user_id, transaction_id, transaction_date, transaction_amount, transaction_type, merchant_id
          ORDER BY transaction_id
        ) AS row_num
  FROM gcash.transactions_staging
)
SELECT *
FROM duplicate_cte
WHERE row_num > 1;
```

Query result:

| user_id | transaction_id | transaction_date | transaction_amount | transaction_type | merchant_id | row_num |
|---|---|---|---|---|---|---|

Data Preparation

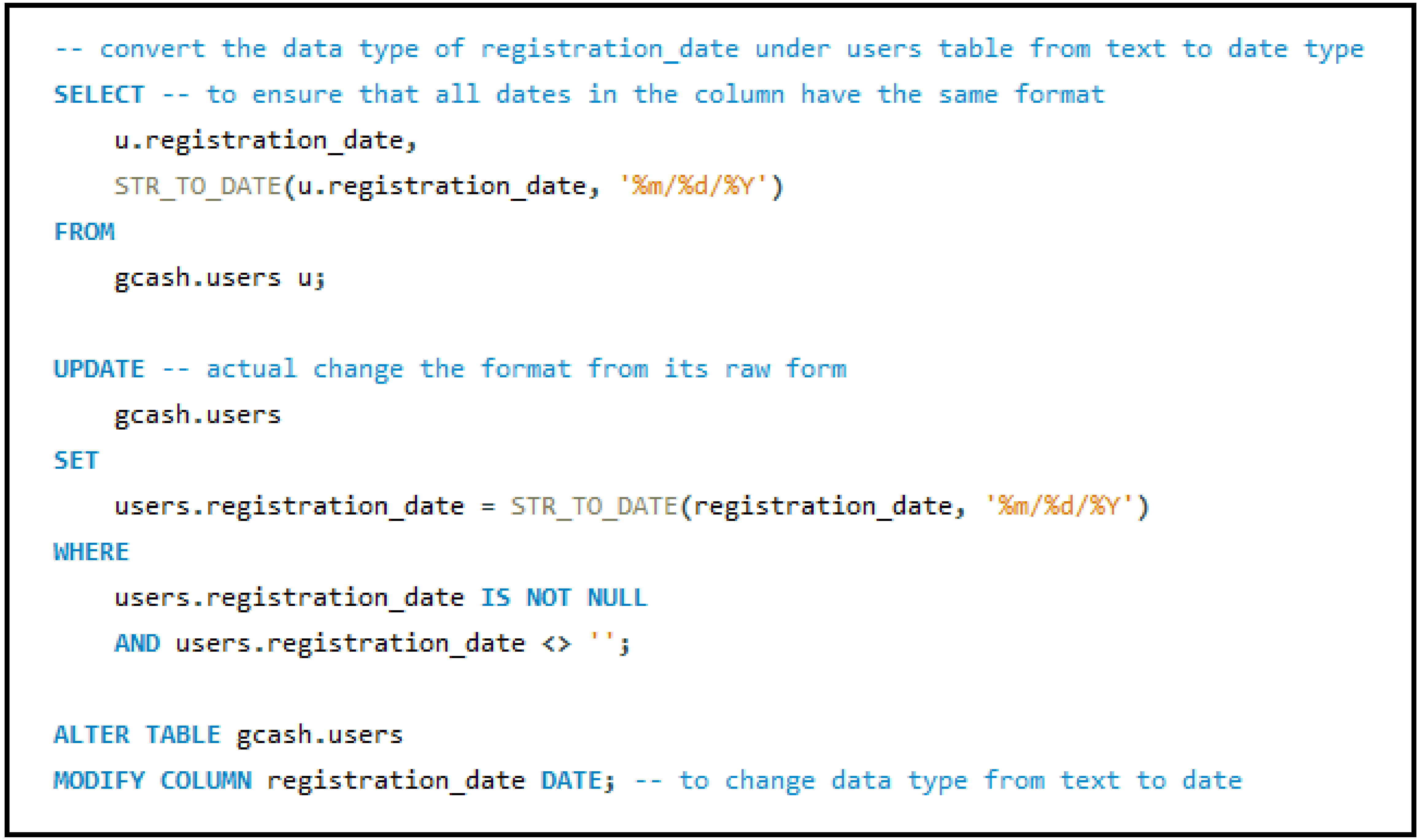
# 2. Standardized Formats



```sql
-- Data standardization
UPDATE gcash.transactions_staging
-- to convert all transaction_type to lowercase and remove extra spaces.
SET
  transaction_type = LOWER(TRIM(transaction_type));
```

Data Preparation

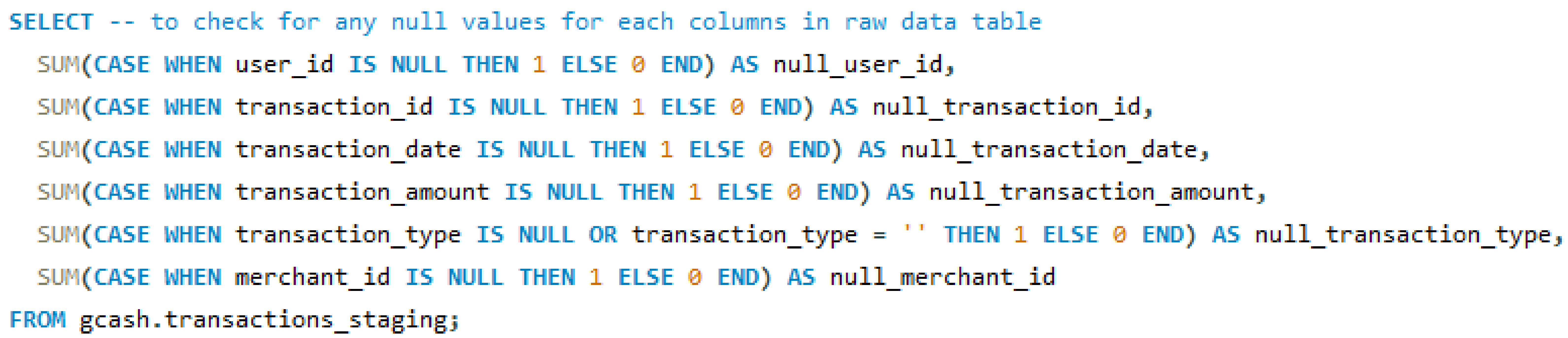
# 2. Standardized Formats



```sql
-- convert the data type of registration_date under users table from text to date type
SELECT -- to ensure that all dates in the column have the same format
    u.registration_date,
    STR_TO_DATE(u.registration_date, '%m/%d/%Y')
FROM
    gcash.users u;

UPDATE -- actual change the format from its raw form
    gcash.users
SET
    users.registration_date = STR_TO_DATE(registration_date, '%m/%d/%Y')
WHERE
    users.registration_date IS NOT NULL
    AND users.registration_date <> '';

ALTER TABLE gcash.users
MODIFY COLUMN registration_date DATE; -- to change data type from text to date
```

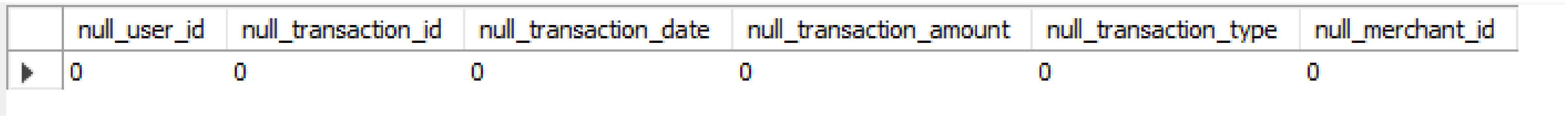
Data
Preparation

# 3. Check for null values under transactions table



```
SELECT -- to check for any null values for each columns in raw data table
  SUM(CASE WHEN user_id IS NULL THEN 1 ELSE 0 END) AS null_user_id,
  SUM(CASE WHEN transaction_id IS NULL THEN 1 ELSE 0 END) AS null_transaction_id,
  SUM(CASE WHEN transaction_date IS NULL THEN 1 ELSE 0 END) AS null_transaction_date,
  SUM(CASE WHEN transaction_amount IS NULL THEN 1 ELSE 0 END) AS null_transaction_amount,
  SUM(CASE WHEN transaction_type IS NULL OR transaction_type = '' THEN 1 ELSE 0 END) AS null_transaction_type,
  SUM(CASE WHEN merchant_id IS NULL THEN 1 ELSE 0 END) AS null_merchant_id
FROM gcash.transactions_staging;
```

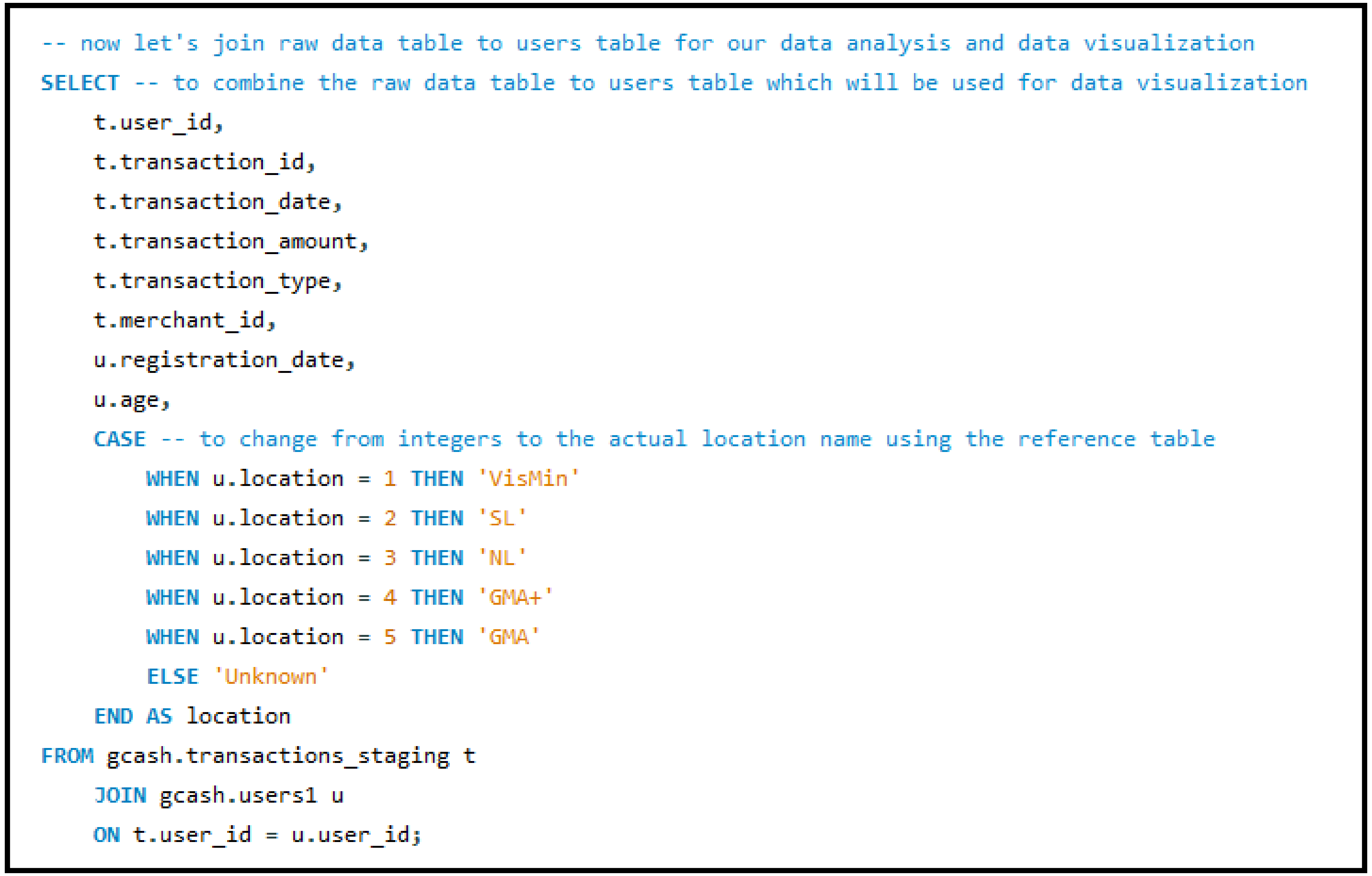
Query result:



| | null_user_id | null_transaction_id | null_transaction_date | null_transaction_amount | null_transaction_type | null_merchant_id |
|---|---|---|---|---|---|---|
| ▶ | 0 | 0 | 0 | 0 | 0 | 0 |

Data Preparation

# 4. Join transactions table to users table for data visualization

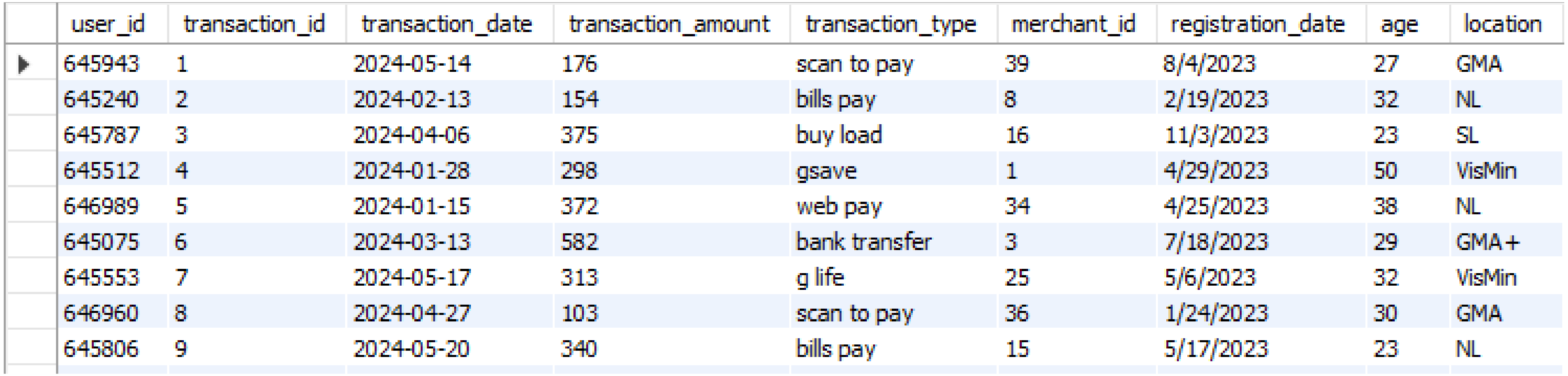


```sql
-- now let's join raw data table to users table for our data analysis and data visualization
SELECT -- to combine the raw data table to users table which will be used for data visualization
    t.user_id,
    t.transaction_id,
    t.transaction_date,
    t.transaction_amount,
    t.transaction_type,
    t.merchant_id,
    u.registration_date,
    u.age,
    CASE -- to change from integers to the actual location name using the reference table
        WHEN u.location = 1 THEN 'VisMin'
        WHEN u.location = 2 THEN 'SL'
        WHEN u.location = 3 THEN 'NL'
        WHEN u.location = 4 THEN 'GMA+'
        WHEN u.location = 5 THEN 'GMA'
        ELSE 'Unknown'
    END AS location
FROM gcash.transactions_staging t
    JOIN gcash.users1 u
    ON t.user_id = u.user_id;
```

Data
Preparation

# 4. Join transactions table to users table for data visualization

Query result:

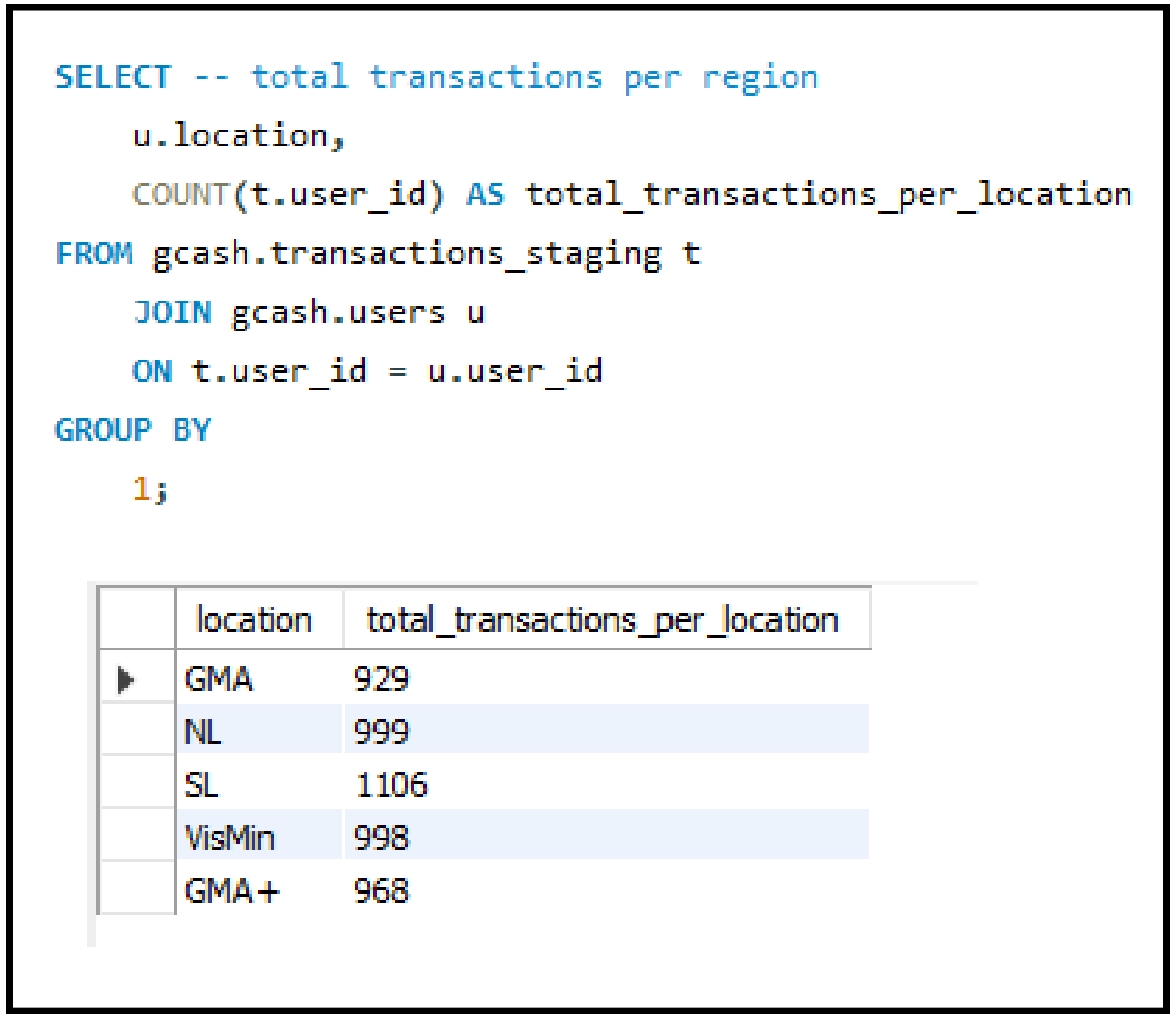


| | user_id | transaction_id | transaction_date | transaction_amount | transaction_type | merchant_id | registration_date | age | location |
|---|---|---|---|---|---|---|---|---|---|
| ▶ | 645943 | 1 | 2024-05-14 | 176 | scan to pay | 39 | 8/4/2023 | 27 | GMA |
| | 645240 | 2 | 2024-02-13 | 154 | bills pay | 8 | 2/19/2023 | 32 | NL |
| | 645787 | 3 | 2024-04-06 | 375 | buy load | 16 | 11/3/2023 | 23 | SL |
| | 645512 | 4 | 2024-01-28 | 298 | gsave | 1 | 4/29/2023 | 50 | VisMin |
| | 646989 | 5 | 2024-01-15 | 372 | web pay | 34 | 4/25/2023 | 38 | NL |
| | 645075 | 6 | 2024-03-13 | 582 | bank transfer | 3 | 7/18/2023 | 29 | GMA+ |
| | 645553 | 7 | 2024-05-17 | 313 | g life | 25 | 5/6/2023 | 32 | VisMin |
| | 646960 | 8 | 2024-04-27 | 103 | scan to pay | 36 | 1/24/2023 | 30 | GMA |
| | 645806 | 9 | 2024-05-20 | 340 | bills pay | 15 | 5/17/2023 | 23 | NL |

Data Preparation

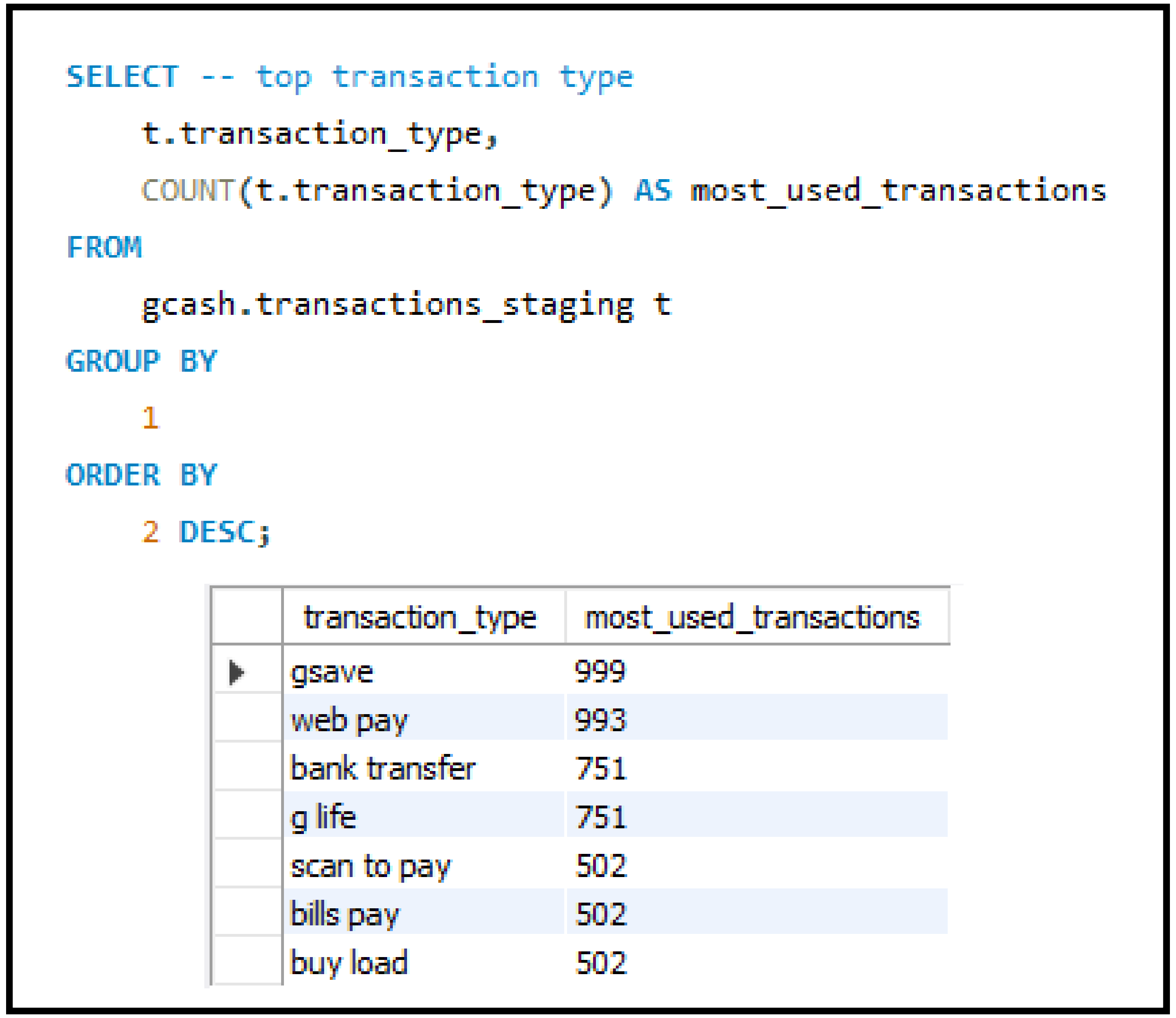
# Exploratory Data Analysis

# Initial Calculations



```
SELECT -- total transactions per region
    u.location,
    COUNT(t.user_id) AS total_transactions_per_location
FROM gcash.transactions_staging t
    JOIN gcash.users u
    ON t.user_id = u.user_id
GROUP BY
    1;
```

| location | total_transactions_per_location |
|---|---|
| GMA | 929 |
| NL | 999 |
| SL | 1106 |
| VisMin | 998 |
| GMA+ | 968 |

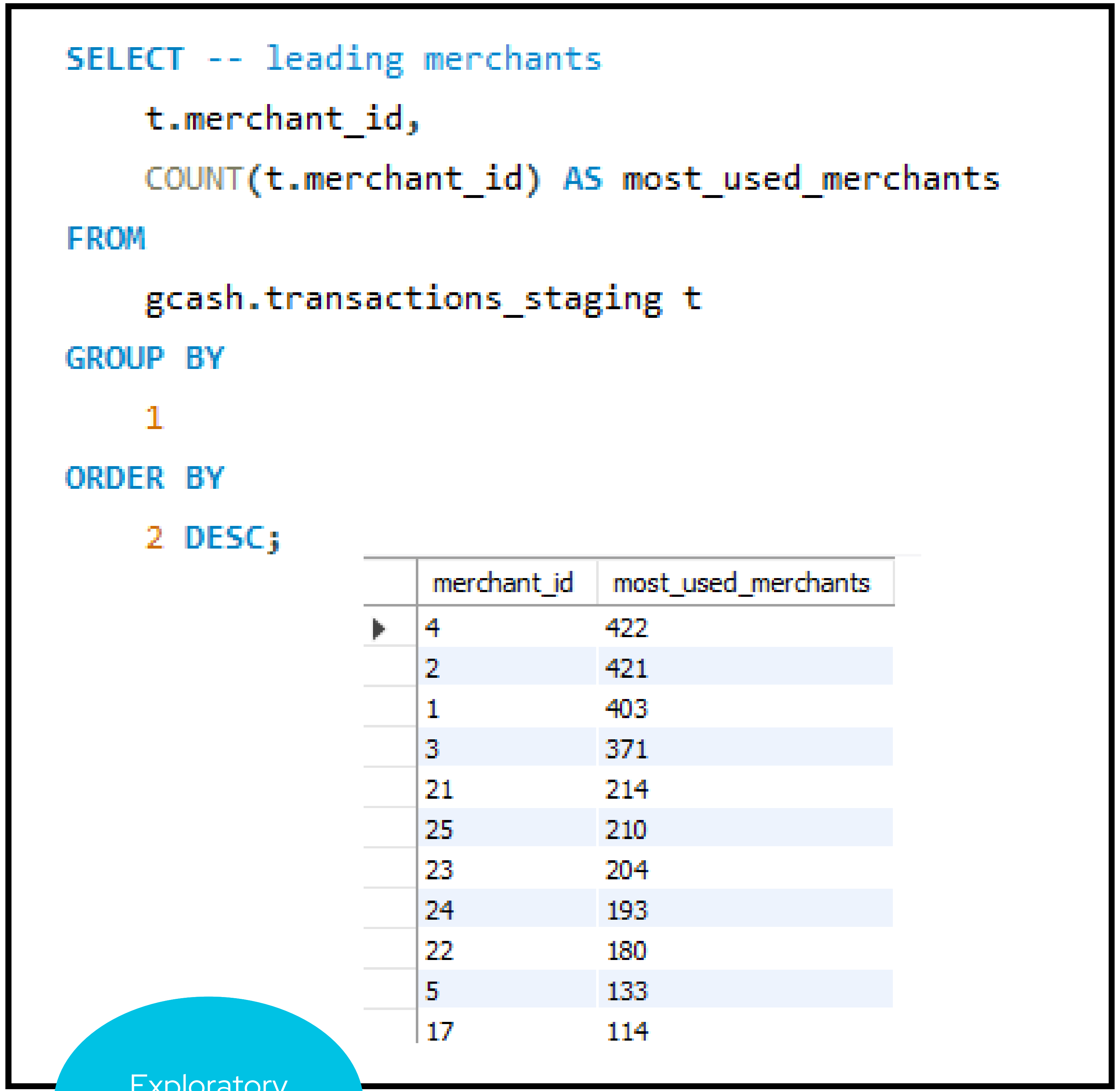


```
SELECT -- top transaction type
    t.transaction_type,
    COUNT(t.transaction_type) AS most_used_transactions
FROM
    gcash.transactions_staging t
GROUP BY
    1
ORDER BY
    2 DESC;
```

| transaction_type | most_used_transactions |
|---|---|
| gsave | 999 |
| web pay | 993 |
| bank transfer | 751 |
| g life | 751 |
| scan to pay | 502 |
| bills pay | 502 |
| buy load | 502 |

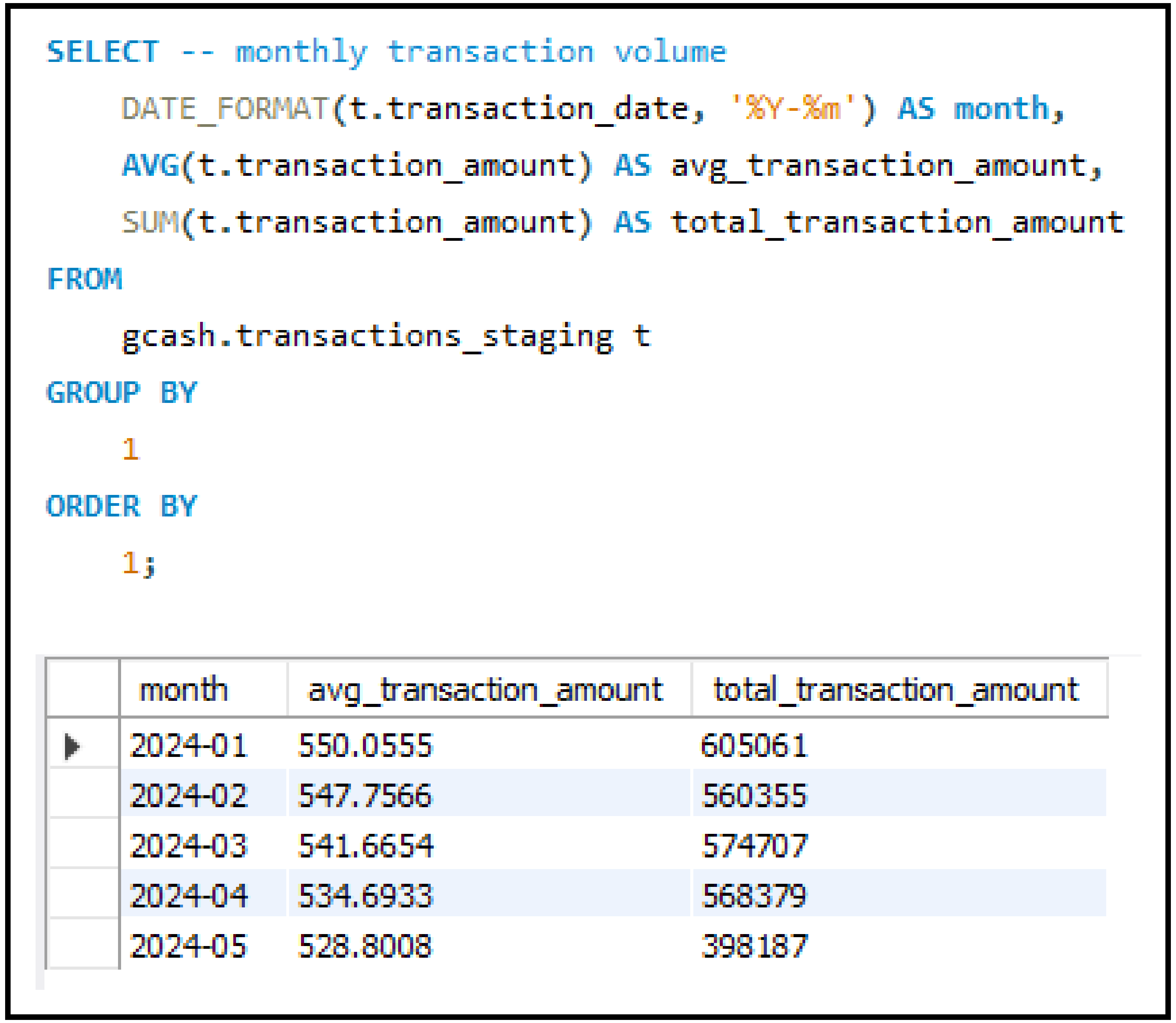
Exploratory
Data Analysis

# Initial Calculations



```
SELECT -- leading merchants
    t.merchant_id,
    COUNT(t.merchant_id) AS most_used_merchants
FROM
    gcash.transactions_staging t
GROUP BY
    1
ORDER BY
    2 DESC;
```

| merchant_id | most_used_merchants |
|---|---|
| 4 | 422 |
| 2 | 421 |
| 1 | 403 |
| 3 | 371 |
| 21 | 214 |
| 25 | 210 |
| 23 | 204 |
| 24 | 193 |
| 22 | 180 |
| 5 | 133 |
| 17 | 114 |

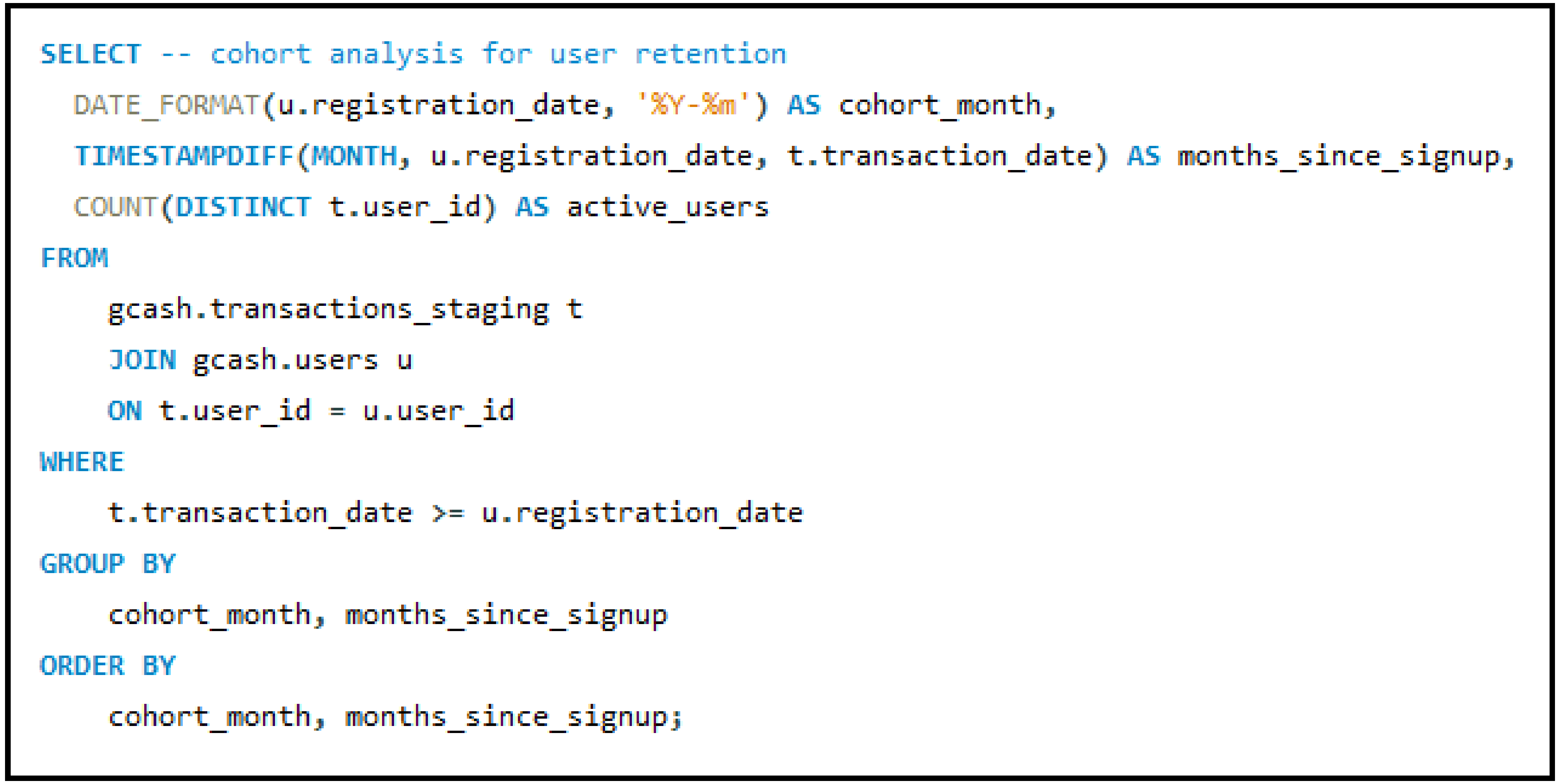


```
SELECT -- monthly transaction volume
    DATE_FORMAT(t.transaction_date, '%Y-%m') AS month,
    AVG(t.transaction_amount) AS avg_transaction_amount,
    SUM(t.transaction_amount) AS total_transaction_amount
FROM
    gcash.transactions_staging t
GROUP BY
    1
ORDER BY
    1;
```

| month | avg_transaction_amount | total_transaction_amount |
|---|---|---|
| 2024-01 | 550.0555 | 605061 |
| 2024-02 | 547.7566 | 560355 |
| 2024-03 | 541.6654 | 574707 |
| 2024-04 | 534.6933 | 568379 |
| 2024-05 | 528.8008 | 398187 |

Exploratory
Data Analysis

# Cohort Analysis

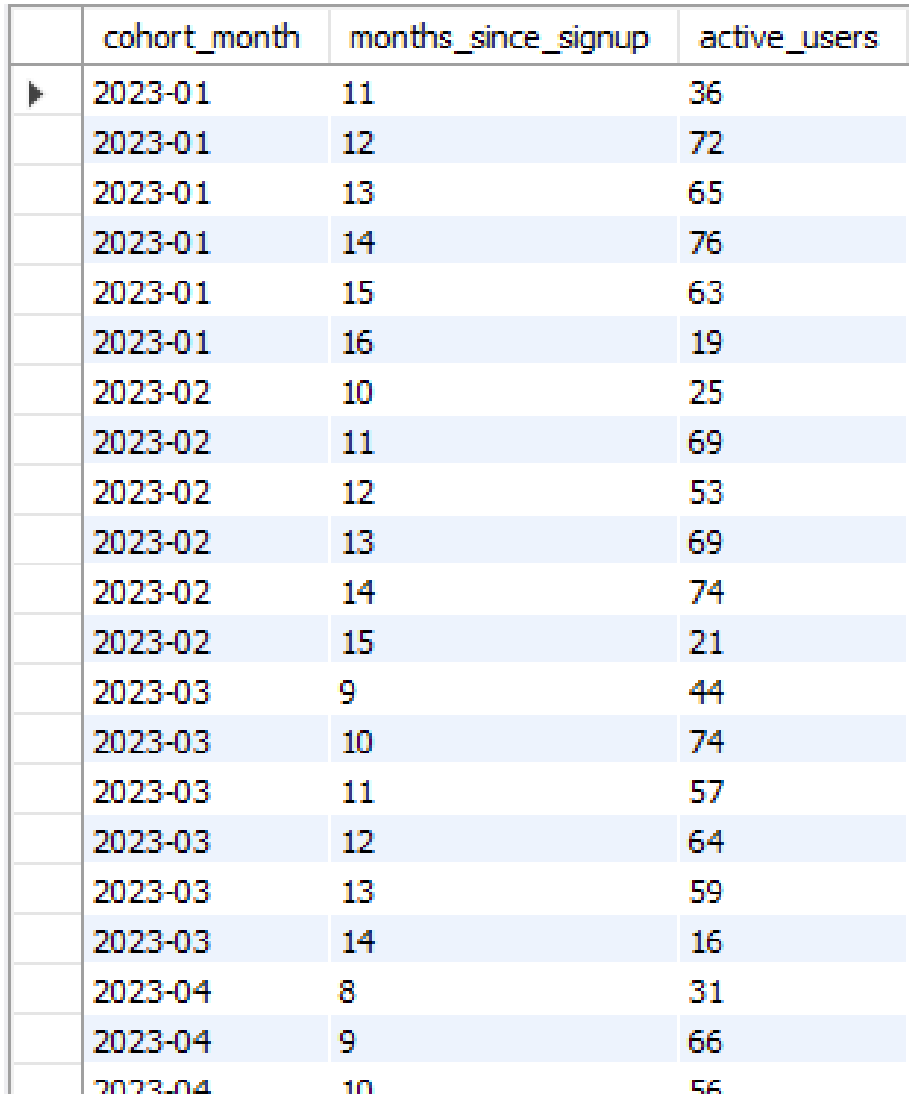


```sql
SELECT -- cohort analysis for user retention
  DATE_FORMAT(u.registration_date, '%Y-%m') AS cohort_month,
  TIMESTAMPDIFF(MONTH, u.registration_date, t.transaction_date) AS months_since_signup,
  COUNT(DISTINCT t.user_id) AS active_users
FROM
    gcash.transactions_staging t
    JOIN gcash.users u
    ON t.user_id = u.user_id
WHERE
    t.transaction_date >= u.registration_date
GROUP BY
    cohort_month, months_since_signup
ORDER BY
    cohort_month, months_since_signup;
```

Exploratory
Data Analysis

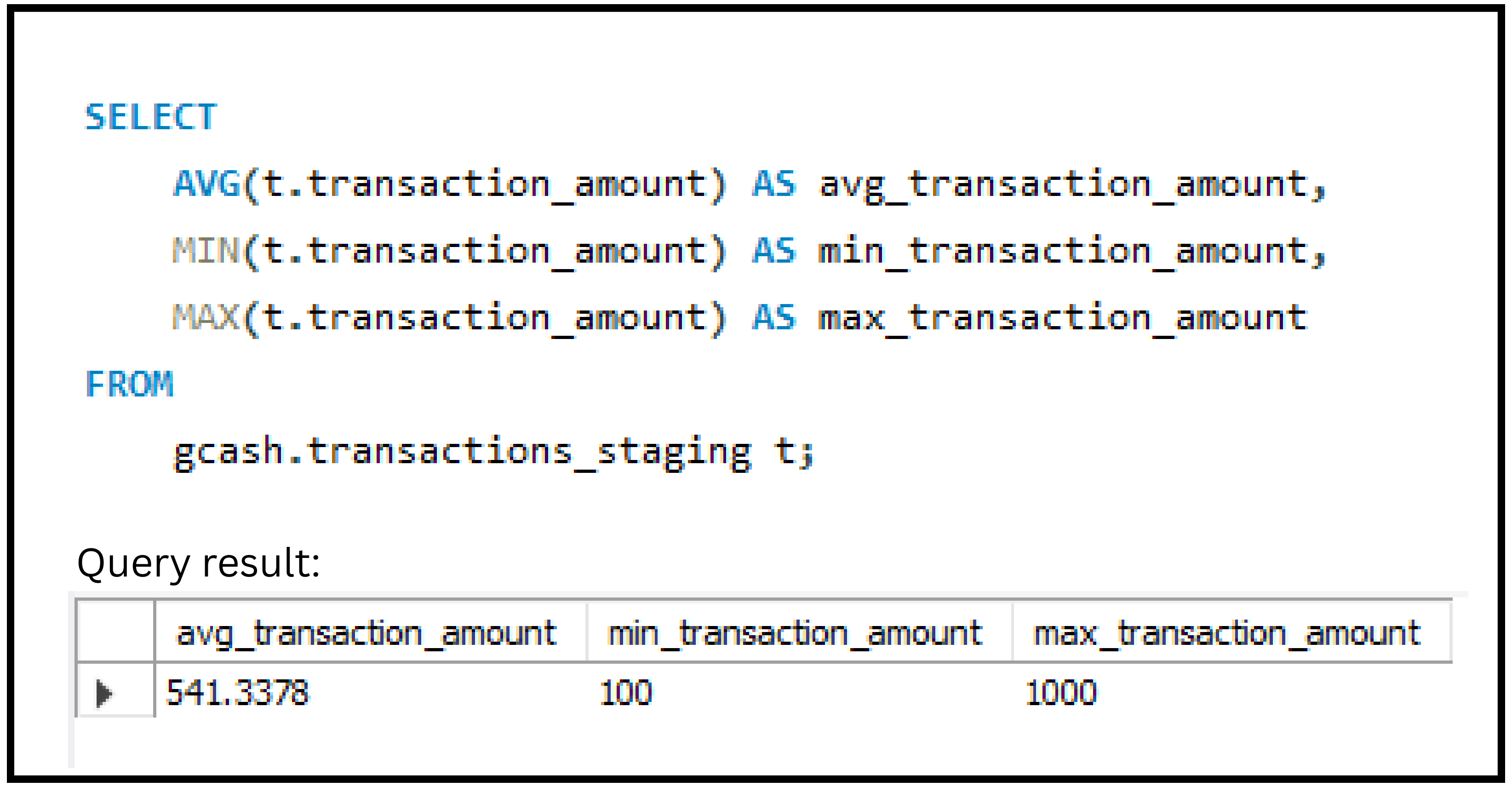
# Cohort Analysis

Query result:



| cohort_month | months_since_signup | active_users |
|---|---|---|
| 2023-01 | 11 | 36 |
| 2023-01 | 12 | 72 |
| 2023-01 | 13 | 65 |
| 2023-01 | 14 | 76 |
| 2023-01 | 15 | 63 |
| 2023-01 | 16 | 19 |
| 2023-02 | 10 | 25 |
| 2023-02 | 11 | 69 |
| 2023-02 | 12 | 53 |
| 2023-02 | 13 | 69 |
| 2023-02 | 14 | 74 |
| 2023-02 | 15 | 21 |
| 2023-03 | 9 | 44 |
| 2023-03 | 10 | 74 |
| 2023-03 | 11 | 57 |
| 2023-03 | 12 | 64 |
| 2023-03 | 13 | 59 |
| 2023-03 | 14 | 16 |
| 2023-04 | 8 | 31 |
| 2023-04 | 9 | 66 |
| 2023-04 | 10 | 56 |

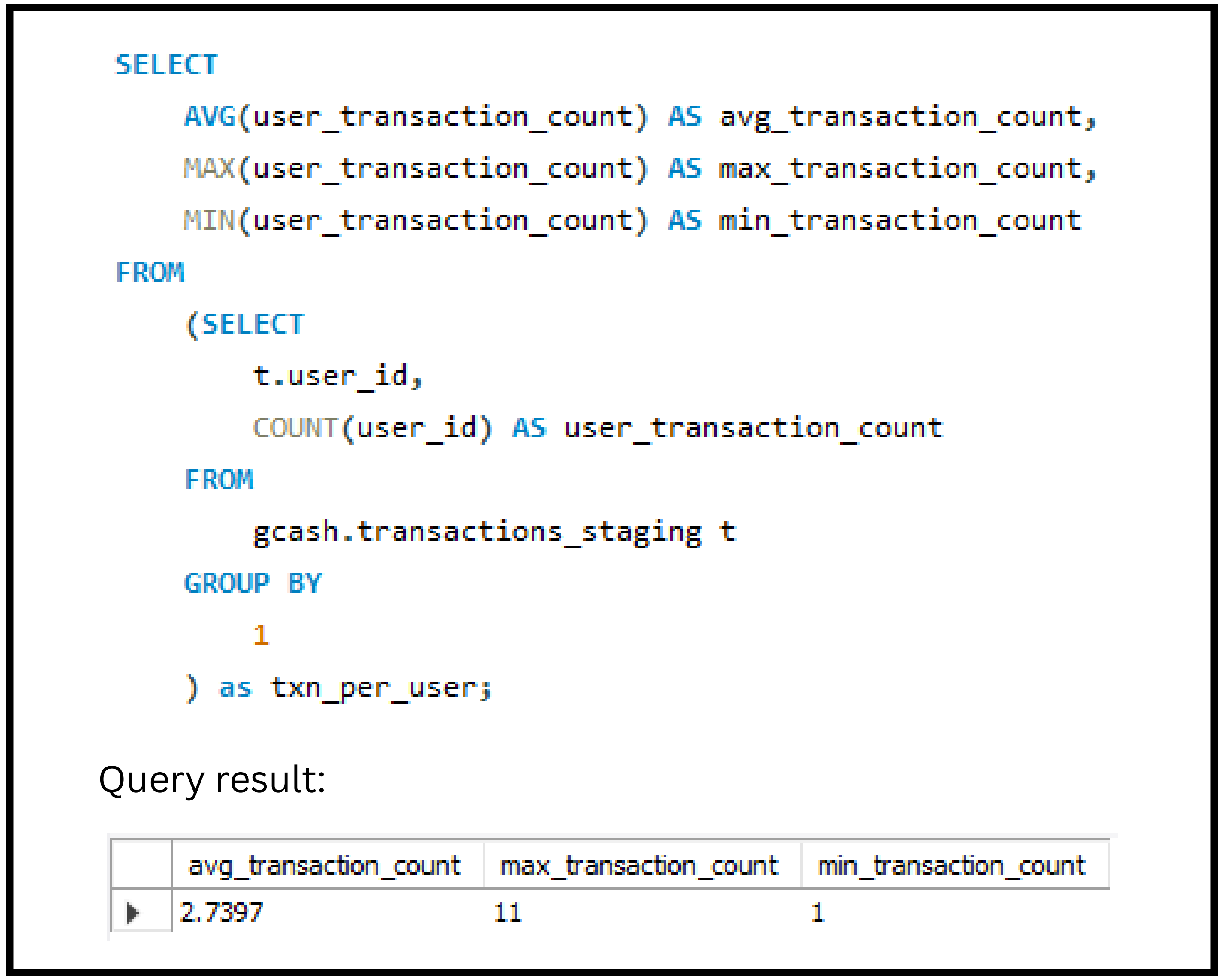
# User Segmentation



```
SELECT
    AVG(t.transaction_amount) AS avg_transaction_amount,
    MIN(t.transaction_amount) AS min_transaction_amount,
    MAX(t.transaction_amount) AS max_transaction_amount
FROM
    gcash.transactions_staging t;
```

Query result:

| avg_transaction_amount | min_transaction_amount | max_transaction_amount |
|---|---|---|
| 541.3378 | 100 | 1000 |



```
SELECT
    AVG(user_transaction_count) AS avg_transaction_count,
    MAX(user_transaction_count) AS max_transaction_count,
    MIN(user_transaction_count) AS min_transaction_count
FROM
    (SELECT
        t.user_id,
        COUNT(user_id) AS user_transaction_count
    FROM
        gcash.transactions_staging t
    GROUP BY
        1
    ) as txn_per_user;
```

Query result:

| avg_transaction_count | max_transaction_count | min_transaction_count |
|---|---|---|
| 2.7397 | 11 | 1 |

Exploratory
Data Analysis

# User Segmentation

## Frequency Score (1-3)

3 points =
Transaction Count >= 5

2 points =
Transaction Count >= 2

1 point =
Transaction Count < 2

## Value Score (1-3)

3 points =
Transaction Amount >= 700

2 points =
Transaction Amount >= 400

1 point =
Transaction Amount < 400

## Recency Score (1)

1 point = Any transaction within April 22, 2024 to May 21, 2024

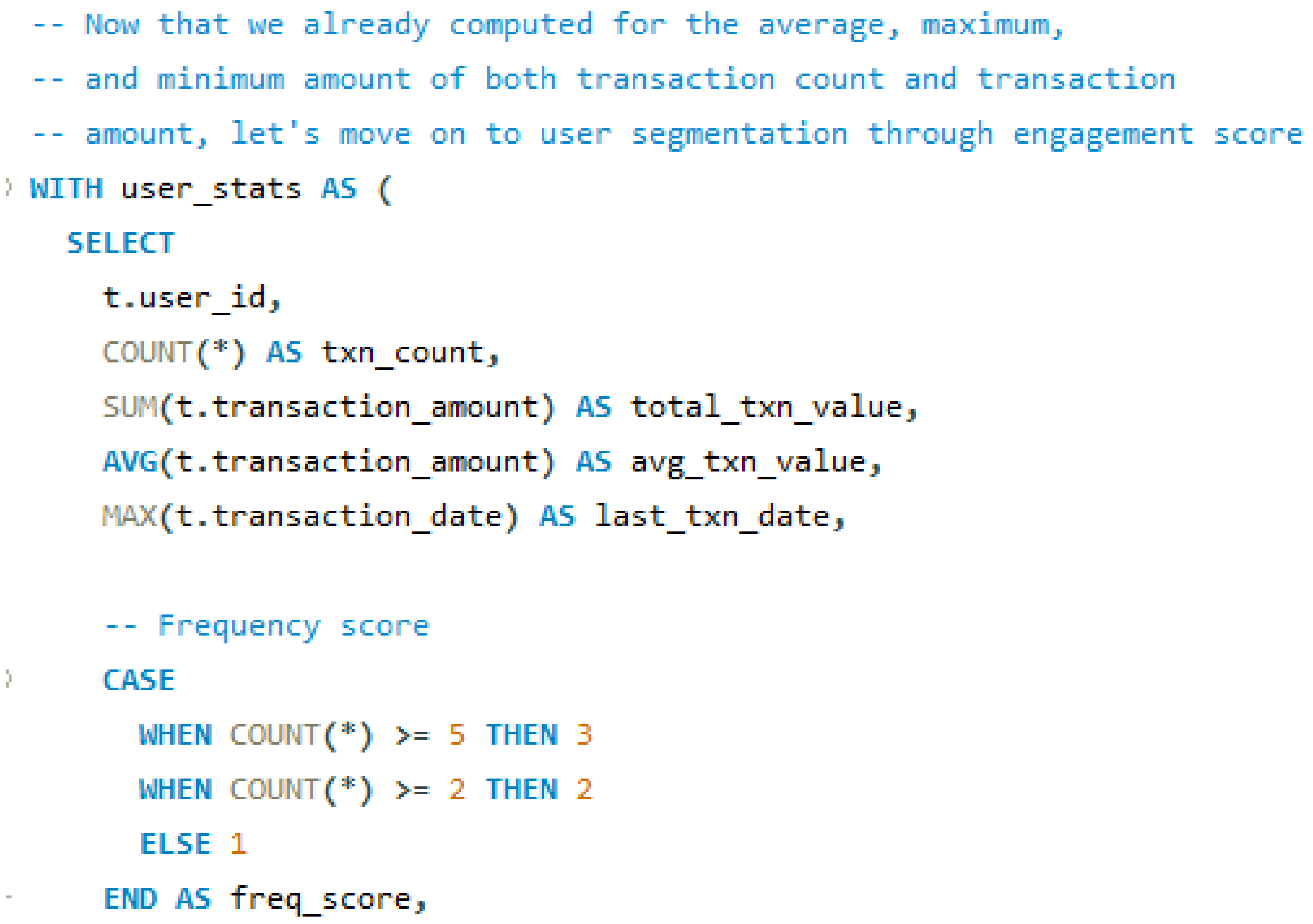
# User Segmentation

1 - 3
=
Low Engagement

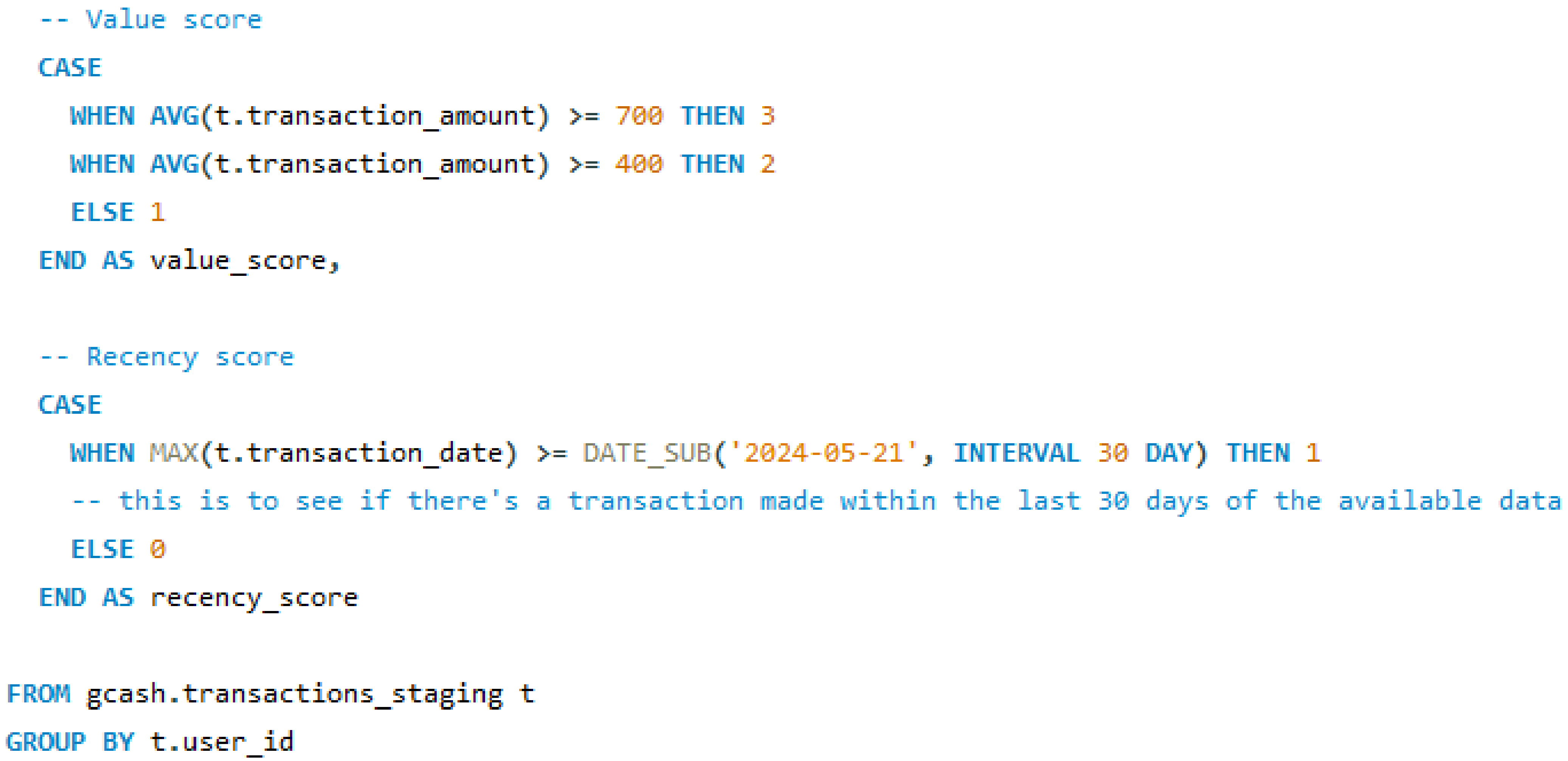
4 - 6
=
At Risk

7
=
Highly Engaged

# User Segmentation



```sql
-- Now that we already computed for the average, maximum,
-- and minimum amount of both transaction count and transaction
-- amount, let's move on to user segmentation through engagement score
WITH user_stats AS (
 SELECT
  t.user_id,
  COUNT(*) AS txn_count,
  SUM(t.transaction_amount) AS total_txn_value,
  AVG(t.transaction_amount) AS avg_txn_value,
  MAX(t.transaction_date) AS last_txn_date,

  -- Frequency score
  CASE
   WHEN COUNT(*) >= 5 THEN 3
   WHEN COUNT(*) >= 2 THEN 2
   ELSE 1
  END AS freq_score,
```

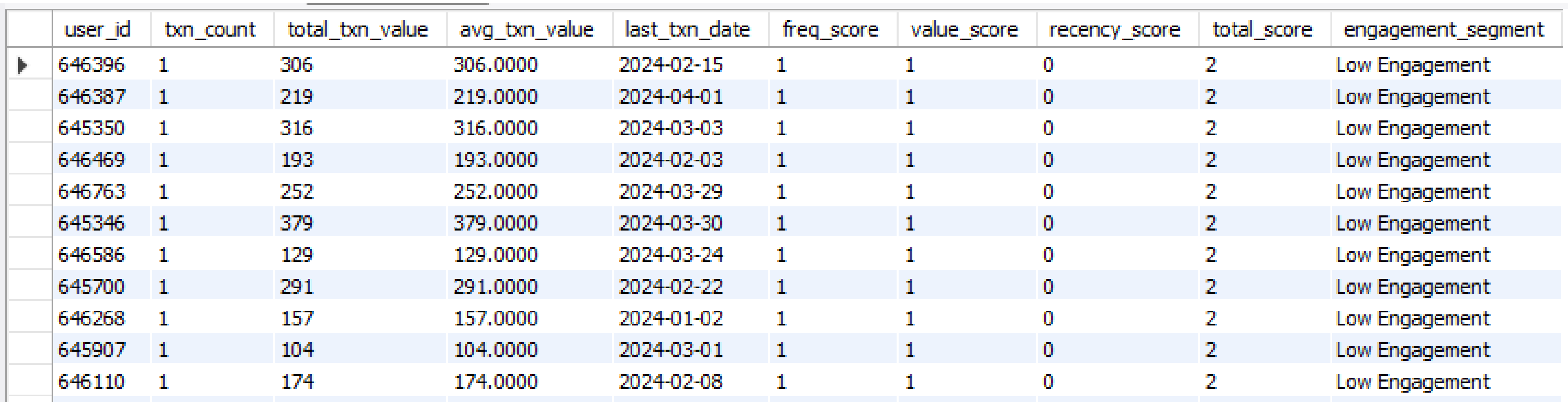


```sql
  -- Value score
  CASE
   WHEN AVG(t.transaction_amount) >= 700 THEN 3
   WHEN AVG(t.transaction_amount) >= 400 THEN 2
   ELSE 1
  END AS value_score,

  -- Recency score
  CASE
   WHEN MAX(t.transaction_date) >= DATE_SUB('2024-05-21', INTERVAL 30 DAY) THEN 1
   -- this is to see if there's a transaction made within the last 30 days of the available data
   ELSE 0
  END AS recency_score

 FROM gcash.transactions_staging t
 GROUP BY t.user_id
```

Exploratory Data Analysis

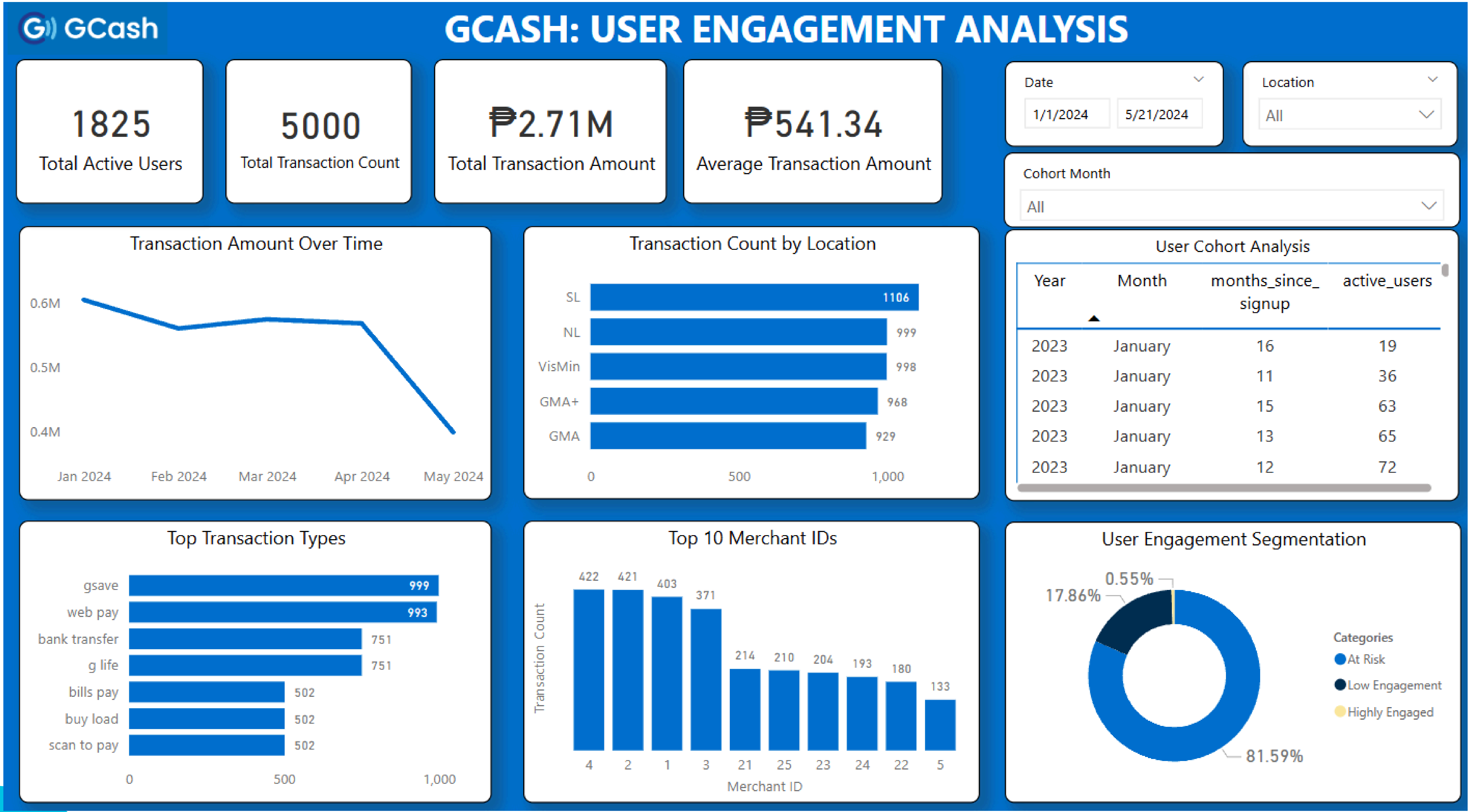
# User Segmentation

```
SELECT
  user_id,
  txn_count,
  total_txn_value,
  avg_txn_value,
  last_txn_date,
  freq_score,
  value_score,
  recency_score,
  (freq_score + value_score + recency_score) AS total_score,

  -- Segment based on score
  CASE
    WHEN (freq_score + value_score + recency_score) = 7 THEN 'Highly Engaged'
    WHEN (freq_score + value_score + recency_score) BETWEEN 4 AND 6 THEN 'At Risk'
    ELSE 'Low Engagement'
  END AS engagement_segment

FROM
    user_stats
ORDER BY
    total_score;
```



| user_id | txn_count | total_txn_value | avg_txn_value | last_txn_date | freq_score | value_score | recency_score | total_score | engagement_segment |
|---|---|---|---|---|---|---|---|---|---|
| 646396 | 1 | 306 | 306.0000 | 2024-02-15 | 1 | 1 | 0 | 2 | Low Engagement |
| 646387 | 1 | 219 | 219.0000 | 2024-04-01 | 1 | 1 | 0 | 2 | Low Engagement |
| 645350 | 1 | 316 | 316.0000 | 2024-03-03 | 1 | 1 | 0 | 2 | Low Engagement |
| 646469 | 1 | 193 | 193.0000 | 2024-02-03 | 1 | 1 | 0 | 2 | Low Engagement |
| 646763 | 1 | 252 | 252.0000 | 2024-03-29 | 1 | 1 | 0 | 2 | Low Engagement |
| 645346 | 1 | 379 | 379.0000 | 2024-03-30 | 1 | 1 | 0 | 2 | Low Engagement |
| 646586 | 1 | 129 | 129.0000 | 2024-03-24 | 1 | 1 | 0 | 2 | Low Engagement |
| 645700 | 1 | 291 | 291.0000 | 2024-02-22 | 1 | 1 | 0 | 2 | Low Engagement |
| 646268 | 1 | 157 | 157.0000 | 2024-01-02 | 1 | 1 | 0 | 2 | Low Engagement |
| 645907 | 1 | 104 | 104.0000 | 2024-03-01 | 1 | 1 | 0 | 2 | Low Engagement |
| 646110 | 1 | 174 | 174.0000 | 2024-02-08 | 1 | 1 | 0 | 2 | Low Engagement |

# Data Visualization



# GCASH: USER ENGAGEMENT ANALYSIS

GCash

1825 Total Active Users

5000 Total Transaction Count

₱2.71M Total Transaction Amount

₱541.34 Average Transaction Amount

Date: 1/1/2024 – 5/21/2024

Location: All

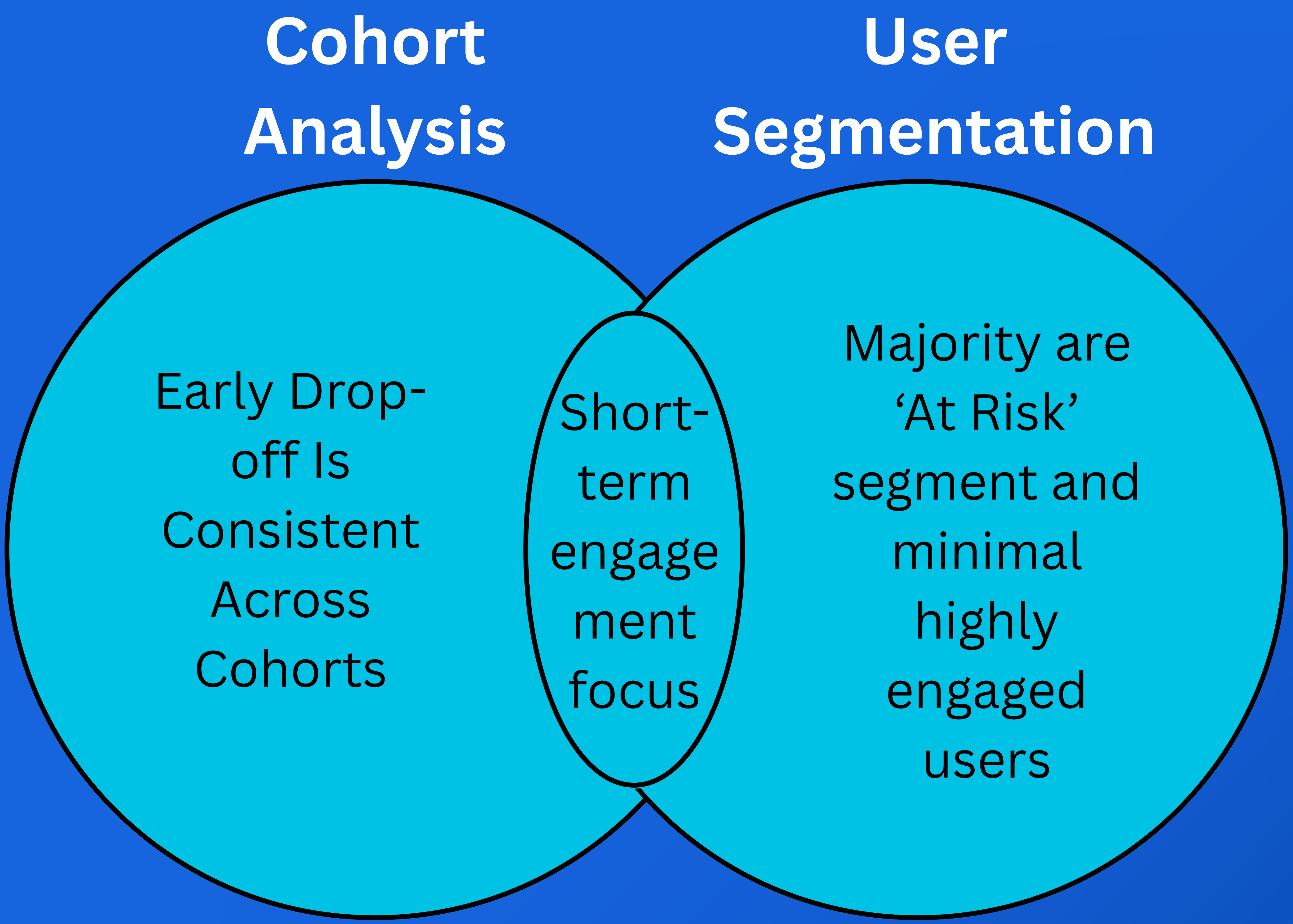
Cohort Month: All

User Cohort Analysis

| Year | Month | months_since_signup | active_users |
|---|---|---|---|
| 2023 | January | 16 | 19 |
| 2023 | January | 11 | 36 |
| 2023 | January | 15 | 63 |
| 2023 | January | 13 | 65 |
| 2023 | January | 12 | 72 |

Data Visualization

# Insights



Cohort Analysis

User Segmentation

Early Drop-off Is Consistent Across Cohorts

Short-term engagement focus

Majority are ‘At Risk’ segment and minimal highly engaged users

Insights

Declining transaction trend signals retention concerns.

Location-based gaps point to uneven app penetration.

Engagement is anchored on financial transactions.

E-commerce players show high transaction frequency.

Online gaming and betting are emerging engagement drivers.

Insights

# Thank you