Data Quality Issues:

Transactions Table:

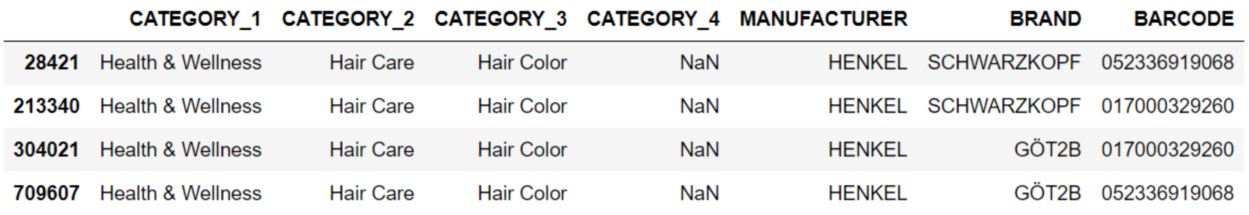
- 5,762 rows are missing BARCODE values, and 12,765 rows have a null value in FINAL_SALE.
- FINAL_QUANTITY contains zero values, which are then converted to 0.
- 335 duplicate values exist in the transaction data.
- Rows with FINAL_SALE or FINAL_QUANTITY as 0 are repeated, but corresponding rows contain the correct FINAL_SALE and FINAL_QUANTITY values.
- 47 receipts show a scan date earlier than the purchase date.
- BARCODE contains invalid values, such as -1.

USERS Table:

- Missing data includes 3.7% of BIRTH_DATE, 4.8% of STATE, 30.5% of STATE, and 5.9% of GENDER values.
- Inconsistent formatting is found in the GENDER column, with phrases like “Prefer not to say,” “Non-binary,” and “Gender isn’t listed” written in various ways.

PRODUCTS Table:

- 4,025 products lack a barcode, with 92% of CATEGORY_4, 27% of MANUFACTURER, and 27% of BRAND column values missing.
- Duplicate barcodes exist, representing different brands.



	CATEGORY_1	CATEGORY_2	CATEGORY_3	CATEGORY_4	MANUFACTURER	BRAND	BARCODE
28421	Health & Wellness	Hair Care	Hair Color	NaN	HENKEL	SCHWARZKOPF	052336919068
213340	Health & Wellness	Hair Care	Hair Color	NaN	HENKEL	SCHWARZKOPF	017000329260
304021	Health & Wellness	Hair Care	Hair Color	NaN	HENKEL	GÖT2B	017000329260
709607	Health & Wellness	Hair Care	Hair Color	NaN	HENKEL	GÖT2B	052336919068

Data Cleaning:

Transactions Table:

- Converted the BARCODE column to a string to preserve leading zeros removed during CSV import due to INT format.
- Replaced null values in BARCODE with “NA.”
- Converted FINAL_QUANTITY zeros and null values in FINAL_SALE to 0, changing both columns to FLOAT type.
- Removed the 335 duplicate values.
- Changed PURCHASE_DATE and SCAN_DATE to DateTime format.
- There are some rows with the same RECEIPT_ID and BARCODE, where either FINAL_SALE or FINAL_QUANTITY is zero. In such cases, there are duplicate rows with the same RECEIPT_ID, BARCODE, STORE_NAME, and USER_ID, where the other row

contains the correct values. So, I removed the rows containing either FINAL_QUANTITY or FINAL_SALE is 0.

USERS Table:

- Converted CREATED_DATE and BIRTH_DATE to DateTime format.
- Replaced null values in GENDER, LANGUAGE, and STATE columns with “NA.”
- Replaced null values in BIRTH_DATE with random values from existing records to maintain distribution.
- Standardized the GENDER column values by mapping terms to “Prefer not to say,” “Non-Binary,” and “My gender isn’t listed.”
- Created an AGE column by calculating the difference between BIRTH_DATE and the current date for each user.

PRODUCTS Table:

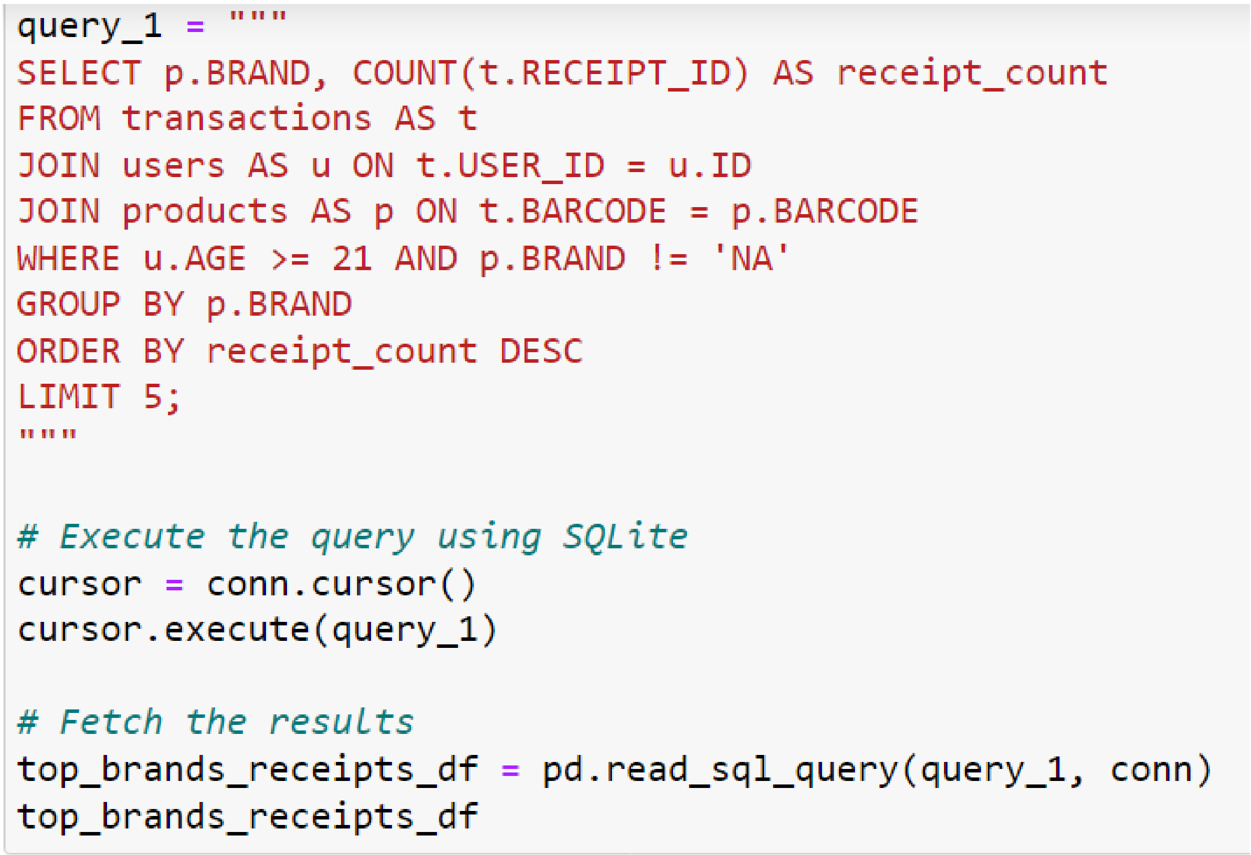
- Removed rows without a barcode, as they are not useful for analysis.
- Retained only the first instance of each duplicated barcode, removing subsequent duplicates.
- Dropped the CATEGORY_4 column due to 92% missing values.
- Replaced remaining null values with “NA.

Closed-ended Questions:

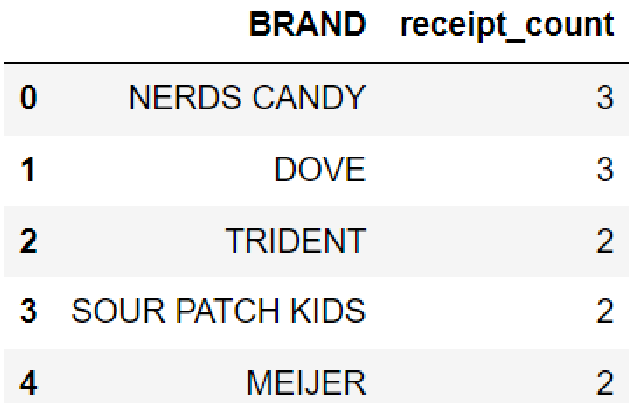
These questions are answered using SQLite3.

1. What are the top 5 brands by receipts scanned among users 21 and over?

Not considered brands with null values.

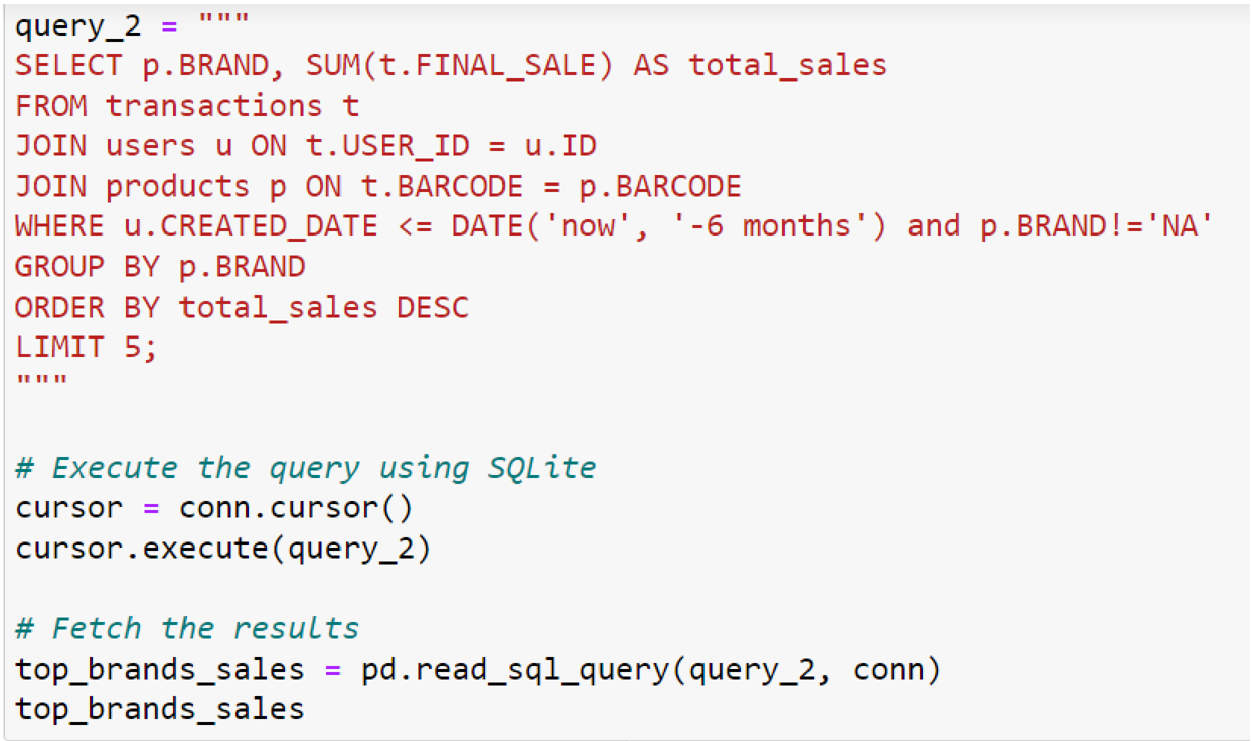


```
query_1 = """
SELECT p.BRAND, COUNT(t.RECEIPT_ID) AS receipt_count
FROM transactions AS t
JOIN users AS u ON t.USER_ID = u.ID
JOIN products AS p ON t.BARCODE = p.BARCODE
WHERE u.AGE >= 21 AND p.BRAND != 'NA'
GROUP BY p.BRAND
ORDER BY receipt_count DESC
LIMIT 5;
"""

# Execute the query using SQLite
cursor = conn.cursor()
cursor.execute(query_1)

# Fetch the results
top_brands_receipts_df = pd.read_sql_query(query_1, conn)
top_brands_receipts_df
```

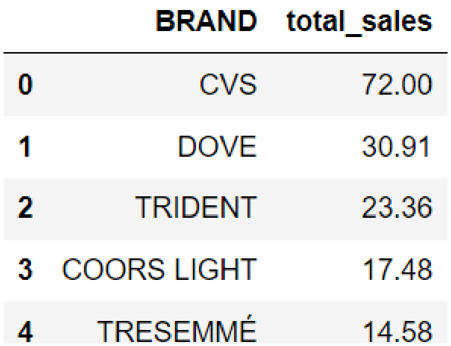


	BRAND	receipt_count
0	NERDS CANDY	3
1	DOVE	3
2	TRIDENT	2
3	SOUR PATCH KIDS	2
4	MEIJER	2

2. **What are the top 5 brands by sales among users that have had their account for at least six months?**



```
query_2 = """
SELECT p.BRAND, SUM(t.FINAL_SALE) AS total_sales
FROM transactions t
JOIN users u ON t.USER_ID = u.ID
JOIN products p ON t.BARCODE = p.BARCODE
WHERE u.CREATED_DATE <= DATE('now', '-6 months') and p.BRAND!='NA'
GROUP BY p.BRAND
ORDER BY total_sales DESC
LIMIT 5;
"""

# Execute the query using SQLite
cursor = conn.cursor()
cursor.execute(query_2)

# Fetch the results
top_brands_sales = pd.read_sql_query(query_2, conn)
top_brands_sales
```

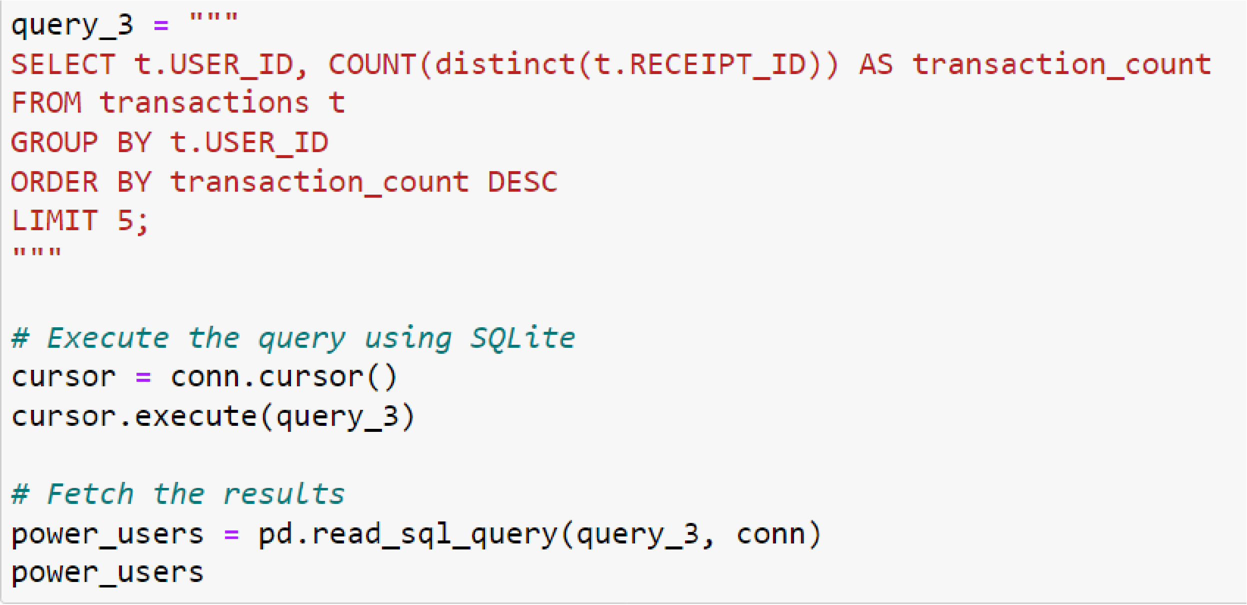


	BRAND	total_sales
0	CVS	72.00
1	DOVE	30.91
2	TRIDENT	23.36
3	COORS LIGHT	17.48
4	TRESEMMÉ	14.58

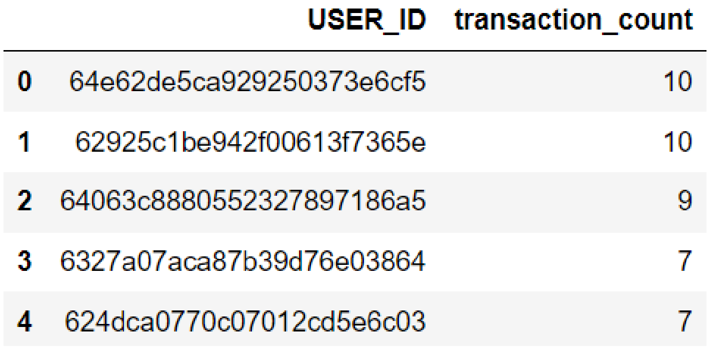
Open-ended questions:

3. Who are Fetch's power users?

Fetch's power users are customers who exhibit high engagement and frequent activity on the platform. So, I took Fetch Power users as the users with the highest distinct receipt count. So the users who have high number of unique receipts are considered power users.



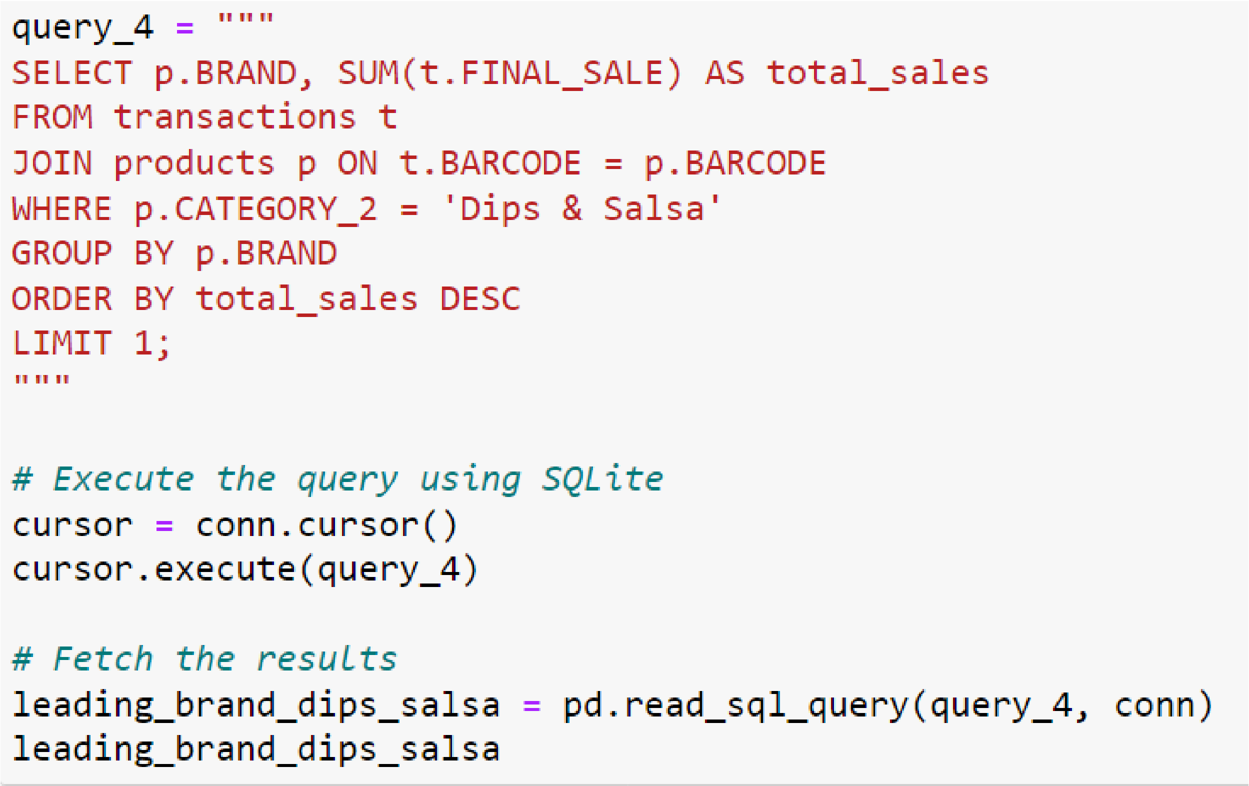
```
query_3 = """
SELECT t.USER_ID, COUNT(distinct(t.RECEIPT_ID)) AS transaction_count
FROM transactions t
GROUP BY t.USER_ID
ORDER BY transaction_count DESC
LIMIT 5;
"""

# Execute the query using SQLite
cursor = conn.cursor()
cursor.execute(query_3)

# Fetch the results
power_users = pd.read_sql_query(query_3, conn)
power_users
```



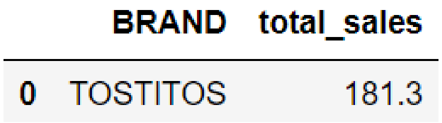
	USER_ID	transaction_count
0	64e62de5ca929250373e6cf5	10
1	62925c1be942f00613f7365e	10
2	64063c8880552327897186a5	9
3	6327a07aca87b39d76e03864	7
4	624dca0770c07012cd5e6c03	7

4. Which is the leading brand in the Dips & Salsa category?

Here, the leading brand is taken as the brand with the highest FINAL_SALE. So, for this data is taken from the transactions table and the products table. Joined both tables on BARCODE column and took the CATEGORY_2 as 'Dips & Salsa' grouped by brand and then ordered by sum of FINAL_SALE and then took the top product. I have taken the cleaned data for this.



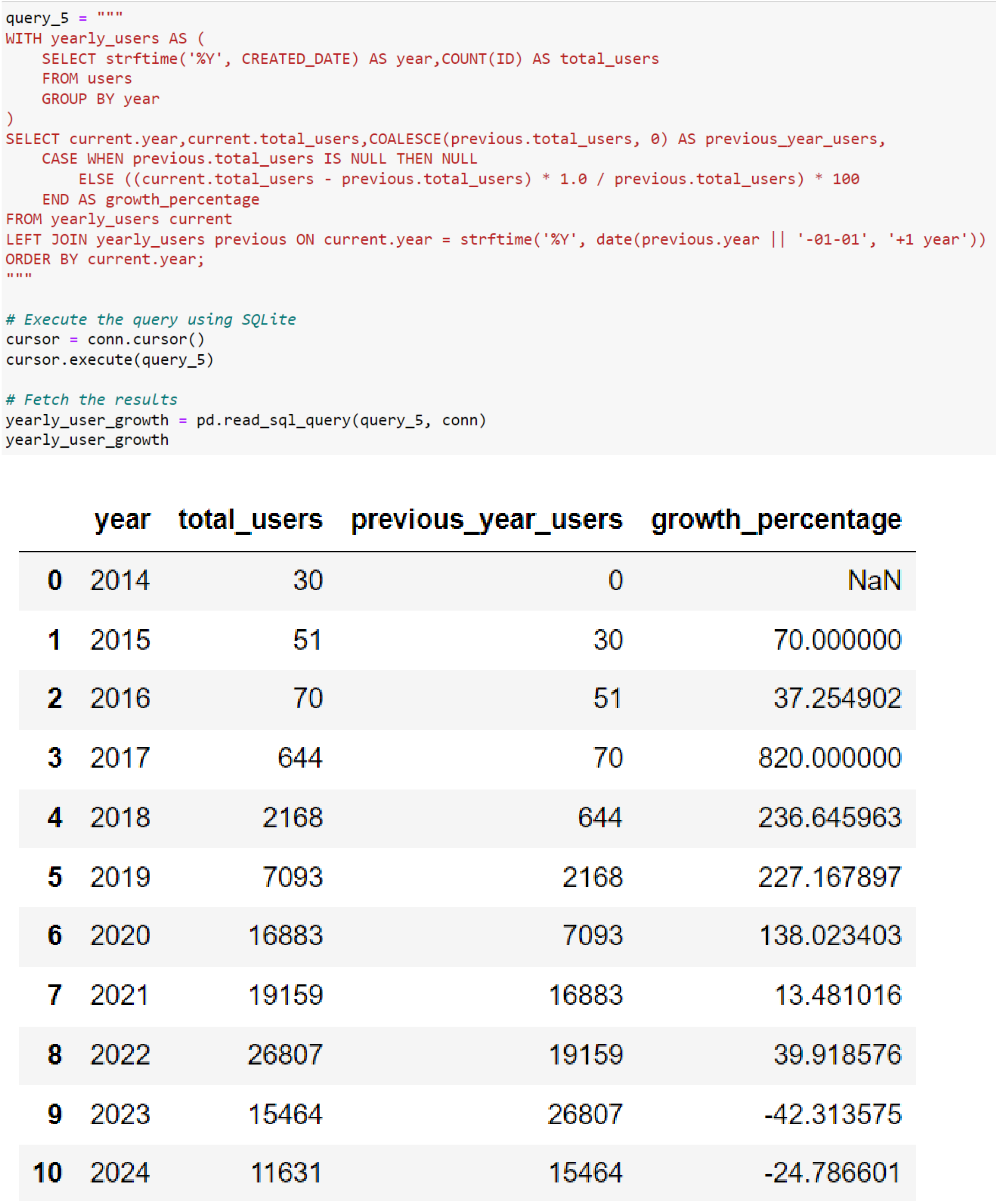
```
query_4 = """
SELECT p.BRAND, SUM(t.FINAL_SALE) AS total_sales
FROM transactions t
JOIN products p ON t.BARCODE = p.BARCODE
WHERE p.CATEGORY_2 = 'Dips & Salsa'
GROUP BY p.BRAND
ORDER BY total_sales DESC
LIMIT 1;
"""

# Execute the query using SQLite
cursor = conn.cursor()
cursor.execute(query_4)

# Fetch the results
leading_brand_dips_salsa = pd.read_sql_query(query_4, conn)
leading_brand_dips_salsa
```



	BRAND	total_sales
0	TOSTITOS	181.3

5. At what percent has Fetch grown year over year?

For this, I considered the growth in terms as the number of users created over time. We can take sales growth also, but we have only 3 months of data. I assumed created date in user_data table as user created date and the percentage change in current year users to previous year users is calculated as growth percentage.



```
query_5 = """
WITH yearly_users AS (
    SELECT strftime('%Y', CREATED_DATE) AS year,COUNT(ID) AS total_users
    FROM users
    GROUP BY year
)
SELECT current.year,current.total_users,COALESCE(previous.total_users, 0) AS previous_year_users,
    CASE WHEN previous.total_users IS NULL THEN NULL
        ELSE ((current.total_users - previous.total_users) * 1.0 / previous.total_users) * 100
    END AS growth_percentage
FROM yearly_users current
LEFT JOIN yearly_users previous ON current.year = strftime('%Y', date(previous.year || '-01-01', '+1 year'))
ORDER BY current.year;
"""

# Execute the query using SQLite
cursor = conn.cursor()
cursor.execute(query_5)

# Fetch the results
yearly_user_growth = pd.read_sql_query(query_5, conn)
yearly_user_growth
```

	year	total_users	previous_year_users	growth_percentage
0	2014	30	0	NaN
1	2015	51	30	70.000000
2	2016	70	51	37.254902
3	2017	644	70	820.000000
4	2018	2168	644	236.645963
5	2019	7093	2168	227.167897
6	2020	16883	7093	138.023403
7	2021	19159	16883	13.481016
8	2022	26807	19159	39.918576
9	2023	15464	26807	-42.313575
10	2024	11631	15464	-24.786601

Challenging fields to understand:

1. **Transactions Table:**
   - The 335 duplicate values: Are they genuinely duplicated transactions, or could they represent cases where the user bought two items with each scanned twice? Should we treat them as duplicates?
   - 47 receipts with SCAN_DATE earlier than PURCHASE_DATE: How should these be interpreted? Are they potentially fraudulent transactions, and should they be removed?
   - Rows with FINAL_SALE or FINAL_QUANTITY as 0 but with corresponding rows containing correct values: Could these indicate returned items or data entry errors?
   - Items with BARCODE as -1: What does this represent, and how can a barcode have a value of -1?

2. **Users Table:**
   - BIRTH_DATE column has missing values and many records have same birth date 1970-01-01. Is this the default date? Should we replace null values with the default date.
3. **Products table:**
   - Two brands share the same barcode: Which brand should retain the barcode, and which should be removed?
   - Over 4,000 items lack a barcode: What might be the cause? Should we remove these rows?
   - The CATEGORY_4 column has 92% missing values: Is this column necessary for analysis?