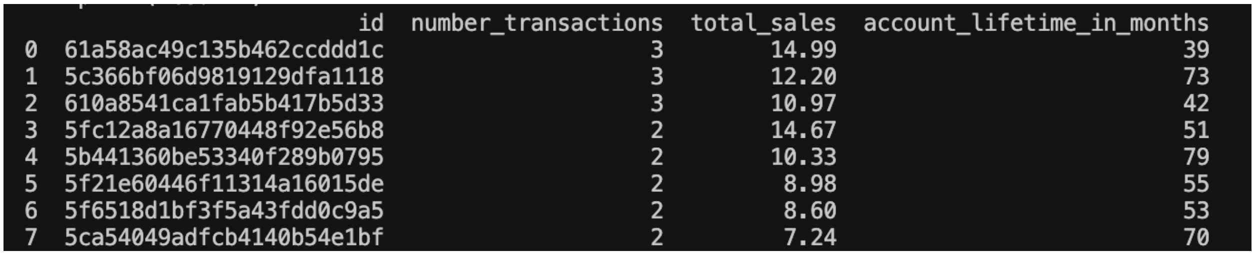
Q1. Who are Fetch’s power users?

```
# Who are Fetch’s Power Users?

# Assumptions:
# A power user in Fetch is defined as a user who:
# 1. Has a transaction count above the average -> shows high engagement.
# 2. Generates sales more than the average -> indicates high spending behavior.
# 3. Has been active on Fetch longer than the average user -> indicated loyalty &
retention.
# Not including the product table here because there are some barcodes which are not
present in the products table.
# This might be because it's just a sample data provided here instead of the actual
entire products table

# For reference avg_transactions = 1.373626,  avg_sales = 6.62967  avg_lifetime =
36.131868 months

query4 = """
WITH valid_users as (select
   id,
   COUNT(distinct receipt_id) as number_transactions,
   SUM(final_sale) as total_sales,
   (strftime('%Y', 'now') - strftime('%Y', created_date)) * 12 +
      (strftime('%m', 'now') - strftime('%m', created_date)) AS
account_lifetime_in_months
FROM transactions t
INNER JOIN users u on u.id = t.user_id
WHERE final_sale != 0 and final_quantity != 0
GROUP BY id, account_lifetime_in_months
),

cte_avg as (
   SELECT
       id,
       number_transactions,
       total_sales,
       account_lifetime_in_months,
       AVG(number_transactions) OVER () AS avg_transactions,
       AVG(total_sales) OVER () AS avg_sales,
       AVG(account_lifetime_in_months) OVER () AS avg_lifetime
   FROM valid_users
)

SELECT
   id,
   number_transactions,
   total_sales,
   account_lifetime_in_months
FROM cte_avg
WHERE
   number_transactions > avg_transactions
   AND total_sales > avg_sales
   AND account_lifetime_in_months > avg_lifetime
ORDER BY number_transactions DESC, total_sales DESC, account_lifetime_in_months DESC;
"""

result4 = pd.read_sql(query4, conn)
print(result4)
```

OUTPUT:



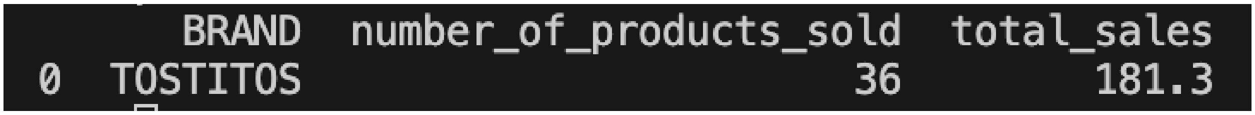
```
                         id  number_transactions  total_sales  account_lifetime_in_months
0  61a58ac49c135b462ccddd1c                    3        14.99                          39
1  5c366bf06d9819129dfa1118                    3        12.20                          73
2  610a8541ca1fab5b417b5d33                    3        10.97                          42
3  5fc12a8a16770448f92e56b8                    2        14.67                          51
4  5b441360be53340f289b0795                    2        10.33                          79
5  5f21e60446f11314a16015de                    2         8.98                          55
6  5f6518d1bf3f5a43fdd0c9a5                    2         8.60                          53
7  5ca54049adfcb4140b54e1bf                    2         7.24                          70
```

Q2. Which is the leading brand in the Dips & Salsa category?

```
# Q. Which is the leading brand in the Dips & Salsa category?
# Assumption: The brand which had the most sales can be considered a leading brand

query5 = """
SELECT
   brand,
   COUNT(t.barcode) as number_of_products_sold,
   SUM(final_sale) as total_sales
FROM products p
INNER JOIN transactions t on p.barcode = t.barcode
WHERE category_2 = 'Dips & Salsa' and final_sale != 0 and final_quantity != 0
GROUP BY brand
ORDER BY total_sales desc
```

```
LIMIT 1
"""
result5 = pd.read_sql(query5, conn)
print(result5)
```

OUTPUT:

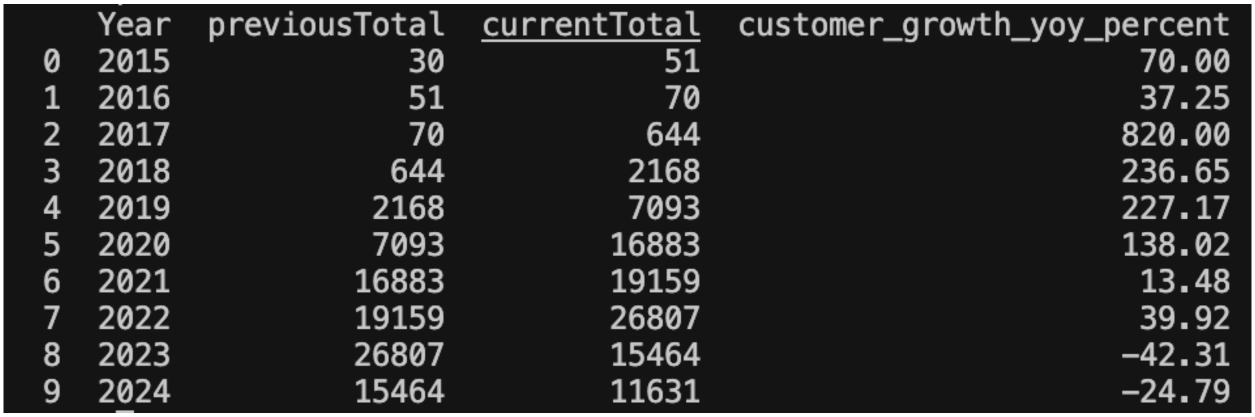


```
      BRAND  number_of_products_sold  total_sales
0  TOSTITOS                       36        181.3
```

Q3. At what percent has Fetch grown year over year?

```
# Q. At what percent has Fetch grown year over year?
# Assumptions:
# Fetch's growth can be considered as follows:
# 1. By what percentage the customer base is growing each year

query6 = """
WITH user_growth as (
   SELECT
       strftime('%Y', created_date) as Year,
       LAG (count(id), 1, 0) OVER (ORDER BY  strftime('%Y', created_date)) as
previousTotal,
       COUNT(id) as currentTotal
   FROM users
   GROUP BY Year
)
SELECT
   Year,
   previousTotal,
   currentTotal,
   ROUND(((currentTotal - previousTotal)/(previousTotal * 1.0)) * 100.00,2) as
customer_growth_yoy_percent
FROM user_growth
WHERE previousTotal != 0
"""

result6 = pd.read_sql(query6, conn)
print(result6)
```

OUTPUT:



```
   Year  previousTotal  currentTotal  customer_growth_yoy_percent
0  2015             30            51                        70.00
1  2016             51            70                        37.25
2  2017             70           644                       820.00
3  2018            644          2168                       236.65
4  2019           2168          7093                       227.17
5  2020           7093         16883                       138.02
6  2021          16883         19159                        13.48
7  2022          19159         26807                        39.92
8  2023          26807         15464                       -42.31
9  2024          15464         11631                       -24.79
```