The following SQL Queries have been executed using SQLLiteonline server and the datasets have not been cleaned before running the queries(data quality issues have been mentioned in Python file)

**Closed-ended questions:**

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

SELECT p.brand, COUNT(t.RECEIPT\_ID) as receipts\_scanned

from PRODUCTS p

JOIN TRANSACTION t on p.barcode = t.BARCODE

join USER u on t.USER\_ID = u.ID

where (CURRENT\_DATE-u.birth\_date)>= 21

group by 1

ORDER BY receipts\_scanned desc

limit 5

A screenshot of a computer

Description automatically generated

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

SELECT BRAND, sum(final\_sale) AS TOTAL\_SALES

From PRODUCTS p

JOIN TRANSACTION t on p.barcode = t.BARCODE

JOIN USER u on t.USER\_ID = u.ID

WHERE julianday('now') - julianday(CREATED\_DATE) >= 180

GROUP BY 1

ORDER BY 2 DESC

LIMIT 5

A black and grey striped background

Description automatically generated

**Open-ended questions:**

1. Who are Fetch’s power users?

Since Fetch is a rewards apps using the receipts that are being scanned by users. Based on the way the company works, I am assuming power users are those who:

* Scans more number of receipts every month (this shows they are highly interested in Fetch). Assuming only users who scans more than 10 receipts per month
* Longevity as a Fetch user
* Have highest total spending

WITH MonthlyActiveUsers AS (

-- Count receipts per user per month

SELECT USER\_ID,

COUNT(RECEIPT\_ID) AS receipts\_per\_month

FROM TRANSACTION1

WHERE PURCHASE\_DATE >= date('now', '-3 month') -- Only consider last month's data

GROUP BY USER\_ID

HAVING receipts\_per\_month > 5 -- Only include users who scan more than 10 receipts per month

),

UserLongevity AS (

-- Compute user longevity in days

SELECT ID,

(julianday('now') - julianday(CREATED\_DATE)) AS longevity\_days

FROM USER

),

UserSpending AS (

-- Compute total spending per user

SELECT USER\_ID,

SUM(FINAL\_SALE) AS total\_spent

FROM TRANSACTION1

WHERE FINAL\_SALE IS NOT NULL

GROUP BY USER\_ID

)

-- Join all three conditions

SELECT u.ID,

u.longevity\_days,

ua.receipts\_per\_month,

us.total\_spent

FROM UserLongevity u

JOIN MonthlyActiveUsers ua ON u.ID = ua.USER\_ID

JOIN UserSpending us ON u.ID = us.USER\_ID

ORDER BY u.longevity\_days DESC, us.total\_spent DESC,3