Provide SQL Queries

Answer three of the following questions with at least one question coming from the closed-ended and one from the open-ended question set. Each question should be answered using one query.

Closed-ended questions:

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

   Brands like **DOVE** and **NERDS CANDY** have the highest number of receipts scanned (3 each), followed by **COCA-COLA, GREAT VALUE, HERSHEY'S** (2 each). (Multiple brands share the same rank due to identical receipt counts, leading to a **dense ranking** system).

2. What are the **top 5 brands** by sales among **users** that have had their **account for at least six months**?
   CVS, DOVE, TRIDENT, **COORS LIGHT,** and **TRESEMME** top the list by sales.

3. What is the **percentage** of **sales** in the **Health & Wellness category by generation**?

   Baby Boomers contribute 54.09%, Gen X 23.90%, and Millennials 22.01% of the category's sales. **Older generations** account for a **larger** share of **Health & Wellness sales**.

Open-ended questions: for these, make assumptions and clearly state them when answering the question.

1. Who are **Fetch's Power Users**?

   Top 20% spenders, ranging from $10 to $76 spent. Most have 1-3 receipts and varying scan delays (0-12 days). Some have been inactive for 150+ days, suggesting churn risk.
   Assumptions: Identified by spending, not transaction frequency. High spenders with few receipts are still considered power users. Spending is the primary indicator of loyalty.

2. **Leading Brand in the Dips & Salsa Category**

   Tostitos leads with 36 receipts, 38 units sold, and $182 in sales ($5.06 avg. per transaction).

3. **Year-over-Year Growth of Fetch**

   Cannot be calculated due to limited data (June 12 to September 8, 2024).

Data Integrity Issue: Some user_id values in the Transactions table don't exist in the Users table, indicating possible data loss and leading to inaccurate transaction analysis.

(All analysis assumes the provided data is complete, despite identified discrepancies such as missing user_id values in the Users table.)

SQL Queries

```
-- DROP DATABASE takehome_db;
CREATE DATABASE takehome_db;
USE takehome_db;

CREATE TABLE Users (
    id VARCHAR(255) PRIMARY KEY,
    created_date DATETIME NULL,
    birth_date DATETIME NULL,
    state VARCHAR(255) NULL,
    language VARCHAR(255) NULL,
    gender VARCHAR(255) NULL
);

CREATE TABLE Products (
    category_1 VARCHAR(255) NULL,
    category_2 VARCHAR(255) NULL,
    category_3 VARCHAR(255) NULL,
    category_4 VARCHAR(255) NULL,
    manufacturer VARCHAR(255) NULL,
    brand VARCHAR(255) NULL,
    barcode BIGINT PRIMARY KEY
);

CREATE TABLE Transactions (
    receipt_id VARCHAR(255) NULL,
    purchase_date DATETIME NULL,
    scan_date DATETIME NULL,
    store_name VARCHAR(255) NULL,
```

```
    user_id VARCHAR(255) NULL,
    barcode BIGINT NULL,
    quantity NUMERIC NULL,
    sale NUMERIC NULL
);


-- ** Top 5 brands by receipts scanned among users 21 and over **
WITH user_age AS (
    -- Calculates user age and filters out users under 21 (kept 21 and older)
    SELECT
        ID
    FROM Users
      WHERE  TIMESTAMPDIFF(YEAR, STR_TO_DATE(BIRTH_DATE, '%Y-%m-%d'),
CURDATE()) >= 21
    AND YEAR(BIRTH_DATE) != 1900 -- Filters out 1900, as blank dates were set to
'1900-01-01' for consistency
)
SELECT
    p.BRAND,
    COUNT(t.RECEIPT_ID) as receipt_count,
    DENSE_RANK() OVER (ORDER BY COUNT(t.RECEIPT_ID) DESC) AS ranking
FROM Transactions t
JOIN Products p ON t.BARCODE = p.BARCODE
JOIN user_age u ON t.USER_ID = u.ID
WHERE p.BRAND != ''  -- Exclude empty brand names
GROUP BY p.BRAND
ORDER BY receipt_count DESC, p.BRAND;
```

```
-- ** Top 5 brands by sales among users that have had their account for at least 6 months **
WITH experienced_users AS (
    -- Filters users who have had an account for at least six months
    SELECT
        ID
    FROM Users
    WHERE TIMESTAMPDIFF(MONTH, STR_TO_DATE(CREATED_DATE,
'%Y-%m-%d'), CURDATE())  >= 6
)
SELECT
    p.BRAND,
    SUM(t.sale) as total_sales
FROM Transactions t
JOIN Products p ON t.BARCODE = p.BARCODE
JOIN experienced_users u ON t.USER_ID = u.ID
WHERE p.BRAND != ''  -- Exclude empty brand names
GROUP BY p.BRAND
ORDER BY total_sales DESC, p.BRAND
LIMIT 5;


-- ** Percentage of sales in the Health & Wellness category by generation **
WITH user_generation AS (
    -- Categorize users into generations based on birth year
    SELECT
        ID,
        BIRTH_DATE,
        CASE
            WHEN YEAR(BIRTH_DATE) >= 1997 THEN 'Gen Z'
            WHEN YEAR(BIRTH_DATE) BETWEEN 1981 AND 1996 THEN 'Millennials'
```

```
        WHEN YEAR(BIRTH_DATE) BETWEEN 1965 AND 1980 THEN 'Gen X'
        WHEN YEAR(BIRTH_DATE) BETWEEN 1946 AND 1964 THEN 'Baby Boomers'
        ELSE 'Silent Generation'
    END AS generation
  FROM Users
  WHERE YEAR(BIRTH_DATE) != 1900 -- Filters out 1900, as blank dates were set to '1900-01-01' for consistency
),
sales_by_generation AS (
  -- Calculate total sales in the Health & Wellness category by generation
  SELECT
    u.generation,
    SUM(t.SALE) AS total_sales
  FROM Transactions t
  JOIN user_generation u ON t.USER_ID = u.ID
  JOIN Products p ON t.BARCODE = p.BARCODE
  WHERE p.CATEGORY_1 = 'Health & Wellness'
  GROUP BY u.generation
),
total_sales AS (
  -- Calculate total Health & Wellness sales across all generations
  SELECT SUM(total_sales) AS grand_total FROM sales_by_generation
)
SELECT
  s.generation,
  s.total_sales,
  (s.total_sales / t.grand_total) * 100 AS sales_percentage  -- Calculate percentage contribution by generation
FROM sales_by_generation s
CROSS JOIN total_sales t
```

```
ORDER BY sales_percentage DESC;



-- ** Fetch’s Power Users **
WITH user_metrics AS (
   -- Calculate user transaction metrics: receipt count, total spend, avg scan time, last activity date
   SELECT
      u.ID,
      COUNT(DISTINCT t.RECEIPT_ID) as receipt_count, -- Total distinct receipts per user
      SUM(t.SALE) as total_spend,  -- Total spend by user
            ROUND(AVG(DATEDIFF(STR_TO_DATE(t.SCAN_DATE, '%Y-%m-%d'),
                        STR_TO_DATE(t.PURCHASE_DATE, '%Y-%m-%d'))), 2) as avg_scan_time, -- Average time between scan and purchase
      MAX(STR_TO_DATE(t.SCAN_DATE, '%Y-%m-%d')) as last_activity  -- Most recent scan date
   FROM Users u
   JOIN Transactions t ON u.ID = t.USER_ID
   GROUP BY u.ID
),
spend_percentile AS (
   -- Determine the spend decile for users, dividing into 10 groups based on spend
   SELECT
      total_spend,
      NTILE(10) OVER (ORDER BY total_spend) as spend_quartile
   FROM user_metrics
)
SELECT
   um.ID,
   u.STATE,
   u.LANGUAGE,
```

```
    U.GENDER,
    um.total_spend,
      um.receipt_count,
    um.avg_scan_time,
    DATEDIFF(CURDATE(), um.last_activity) as days_since_last_active
FROM user_metrics um
JOIN Users u ON um.ID = u.ID
-- Filter to only include users in the 9th sp end decile (top 20% by total spend)
AND um.total_spend >= (
 SELECT MIN(total_spend)
   FROM spend_percentile
   WHERE spend_quartile = 9)
ORDER BY um.total_spend DESC, um.receipt_count, um.avg_scan_time;



-- ** Leading brand in the Dips & Salsa category **
SELECT
    p.BRAND,
      COUNT(DISTINCT t.RECEIPT_ID) as number_of_receipts,
    ROUND(SUM(t.QUANTITY),2) as total_units_sold,
    SUM(t.SALE) as total_sales,
    AVG(t.SALE) as average_sale
FROM Transactions t
JOIN Products p ON t.BARCODE = p.BARCODE
WHERE p.CATEGORY_2 = 'Dips & Salsa'
AND p.BRAND != ''
GROUP BY p.BRAND
ORDER BY total_sales DESC
LIMIT 1;
```