Second: Write queries that directly answer predetermined questions from a business stakeholder

1. Introduction

This report presents an analysis of the Fetch dataset using **Snowflake SQL**, focusing on key business questions related to **brand popularity, user spending behavior, and transaction trends**. The objective is to extract meaningful insights that can drive **marketing strategies, user engagement, and business decisions**.

The dataset consists of three primary entities:

- **Users**: Information about Fetch app users.
- **Brands**: Details of various brands included in scanned receipts.
- **Receipts**: Purchase transactions submitted by users.

To ensure accurate and efficient analysis, the **raw JSON data was structured, cleaned, and transformed** into a relational format suitable for SQL querying.

2. Data Preparation & Processing

Data Loading & Warehousing

- A **Snowflake warehouse (loading_wh)** was created to manage data ingestion.
- A **database (FETCH)** was set up to store structured tables for analysis.
- The **context** is set to ensure queries run in the correct database and warehouse.

Table Structuring & Cleaning

To optimize performance and maintain data integrity, the following tables were created:

Table Name	Description	Key Enhancements
FETCH.PUBLIC.USERS	Stores unstructured users JSON data	Original format for data ingestion
FETCH.PUBLIC.USERS_CLEANED	Cleaned Users data for analysis	Structured, indexed, and partitioned
FETCH. PUBLIC.BRANDS	Stores unstructured brands JSON data	Original format for data ingestion
FETCH.PUBLIC.BRANDS_CLEANED	Cleaned brand data	Structured, indexed, and partitioned
FETCH.PUBLIC.RECEIPTS	Stores unstructured receipt JSON data	Original format for data ingestion
FETCH.PUBLIC.RECEIPTS_CLEANED	Structured receipts data	Structured, indexed, and partitioned for analysis
FETCH.PUBLIC.RECEIPT_ITEMS	Item-level purchase data	Flattened structure for easier querying, linked to RECEIPTS_CLEANED via receipt_id and to BRANDS via barcode

Note: The table names and structures were aligned with the sample dataset provided. In a production environment, naming conventions, data loading strategies, and table configurations would adhere to the organization's **data architecture and governance standards**.

Queries-:

```
--Creating a loading warehouse

CREATE OR REPLACE WAREHOUSE loading_wh WITH

WAREHOUSE_SIZE='X-SMALL'

AUTO_RESUME = TRUE -- default

AUTO_SUSPEND = 600 -- default

INITIALLY_SUSPENDED = TRUE; -- default


--Create a new database named Fetch

CREATE DATABASE IF NOT EXSITS FETCH;


--Set the Context

USE fetch.public;

USE WAREHOUSE loading_wh;
```

The structured and flattened tables are named based on the sample dataset used for analysis. In a production environment, data loading, table naming conventions, and database structuring would align with the organization's data architecture and best practices.

Additionally, data cleaning, structuring, and flattening are conducted before analysis to ensure data integrity and optimize queries for addressing the given business questions effectively.

3. Analytical Approach & Methodology

Purpose of RECEIPTS_SUMMARY View

The RECEIPTS_SUMMARY view is created to **aggregate key transactional data from multiple tables** (RECEIPTS_CLEANED, RECEIPT_ITEMS, and BRANDS_CLEANED). This view **simplifies analysis by providing a structured summary of each receipt**, including total spending, item counts, and purchase details.

Queries-:

```
CREATE OR REPLACE VIEW FETCH.PUBLIC.receipts_summary AS
SELECT
    rc.receipt_id,
    rc.user_id,
    rc.purchase_date,
    rc.total_spent,
    rc.receipt_status,
    SUM(ri.quantity_purchased) AS total_quantity_purchased,  -- Ensuring accurate item count
    SUM(ri.final_price) AS total_item_value
FROM FETCH.PUBLIC.receipts_cleaned rc
JOIN FETCH.PUBLIC.receipt_items ri ON rc.receipt_id = ri.receipt_id
LEFT JOIN FETCH.PUBLIC.brands_cleaned b ON ri.item_barcode = b.barcode
GROUP BY rc.receipt_id, rc.user_id, rc.purchase_date, rc.total_spent, rc.receipt_status;
```

Analytical approach followed for Questions 1&2 -:

Data Filtering (valid_receipts CTE)

- I have extracted **only relevant receipts** that have **valid purchase dates and brand associations**.
- This ensured we excluded incomplete or erroneous data.

Identifying the Most Recent and Previous Month (latest_month & previous_month CTEs)

- I have determined the **most recent** and **previous month** where transactions were recorded.
- This allows us to dynamically compare **trends over time** without hardcoding dates.

Ranking Brands Based on Receipt Counts (brand_rankings CTE)

- I have **aggregated** the number of **unique receipts per brand per month**.
- The **RANK() function** was used to rank brands based on the total number of receipts scanned in each month.
- This ranking helps in **identifying leading brands and tracking changes over time**.

Comparing Brand Rankings (Final Selection Query)

- The **latest month's rankings** were joined with the **previous month's rankings** to enable direct comparisons.
- This allows us to see which brands **maintained their position, improved, or dropped in rankings**.
- The **LEFT JOIN** ensures that brands that appear in the latest month but not in the previous month are still included.

Query-:

```
-- 1&2

WITH valid_receipts AS (
    -- Filtering out receipts with valid brands and purchase dates
    SELECT
        rc.receipt_id,
        b.brand_id,
        DATE_TRUNC('month', rc.purchase_date) AS receipt_month,
        b.brand_code
    FROM fetch.public.receipts_cleaned rc
    JOIN fetch.public.receipt_items ri ON rc.receipt_id = ri.receipt_id
    JOIN fetch.public.brands_cleaned b ON ri.item_barcode = b.barcode
```

```
    WHERE rc.purchase_date IS NOT NULL
),
latest_month AS (
    -- Getting the most recent month with valid receipt data
    SELECT MAX(receipt_month) AS latest_month FROM valid_receipts
),
previous_month AS (
    -- Getting the previous month relative to the latest available month
    SELECT DATEADD('month', -1, latest_month) AS previous_month FROM latest_month
),
brand_rankings AS (
    -- Ranking brands based on receipt counts for each month
    SELECT
        receipt_month,
        brand_id,
        brand_code,
        COUNT(DISTINCT receipt_id) AS receipt_count,  -- Renamed for clarity
        RANK() OVER (PARTITION BY receipt_month ORDER BY COUNT(DISTINCT receipt_id) DESC) AS
rank_position
    FROM valid_receipts
    GROUP BY receipt_month, brand_id, brand_code
)
-- Final Selection for Top 5 Brands & Rank Comparison
SELECT
    latest_ranked.brand_code,
    latest_ranked.receipt_count AS latest_receipt_count,
    latest_ranked.rank_position AS latest_rank,
    previous_ranked.rank_position AS previous_rank
FROM brand_rankings latest_ranked
```

```
LEFT JOIN brand_rankings previous_ranked
    ON latest_ranked.brand_code = previous_ranked.brand_code
    AND previous_ranked.receipt_month = (SELECT previous_month FROM previous_month)
WHERE latest_ranked.receipt_month = (SELECT latest_month FROM latest_month)
ORDER BY latest_rank
LIMIT 5;
```

Query Result-:

	BRAND_CODE	LATEST_RECEIPT_COUNT	LATEST_RANK	PREVIOUS_RANK
1	TOSTITOS	11	1	null
2	SWANSON	11	1	null
3	CRACKER BARREL	10	3	null
4	PREGO	4	4	null
5	DIETCHRIS2	4	4	null

Note: The above result corresponds to questions 1 and 2, identifying the top five brands by receipts scanned in the most recent month and comparing their rankings to the previous month. The **previous_rank** column contains null values because there is no data available for the previous month in the **receipts_cleaned** table. As a result, no historical comparison can be made.

Analysis of Results:

- **Tostitos** and **Swanson** had the highest number of scanned receipts (**11 each**), ranking **first**.
- **Cracker Barrel** followed closely with **10 receipts**, securing the **third position**.
- **Prego** and **Dietchris2** had **4 receipts each**, ranking **fourth**.
- Since there is no data for the previous month, this ranking reflects only the latest available trends.

Ranking Approach & Optimization

I have used **RANK()** in this analysis to assign rankings based on the number of scanned receipts per brand. However, depending on the business requirements, we can also use **DENSE_RANK()**, which ensures no gaps in ranking if multiple brands share the same receipt count. The agreed-upon approach can be adapted based on the analytical needs to ensure accurate insights

Alignment with Business Modeling Best Practices

Clean, Efficient, and Performant Data Model-:

- Used well-structured CTEs for logical, efficient query breakdown.
- Performed aggregations at the **brand-month level** to reduce complexity.
- Enabled dynamic time selection (MAX(receipt_month)) for reusability.

Capturing Broader Insights-:

- **Brand Popularity Trends** – Tracked ranking shifts over time.
- **Seasonality Analysis** – Identified fluctuations in brand performance.
- **Market Shifts** – Detected emerging brands entering the top 5.

This approach ensured **scalability, automation, and actionable insights** beyond just the questions asked.

Analytical Approach Followed for Questions 3 & 4

Data Filtering & Structuring (receipts_summary View)

- Extracted **only relevant receipt data** containing receipt_status, total_spent, and total_quantity_purchased.
- Ensured **accurate aggregation** by using SUM(quantity_purchased), avoiding overcounting.
- Grouped data at the **receipt level**, making it easier to compute overall trends.

Filtering & Aggregation for Analysis (accepted_rejected_summary View)

- The analysis was conducted using receipts where **receipt_status is 'FINISHED' and 'REJECTED'**, as these are the available categories in the dataset. (While the original question requested an analysis on **'ACCEPTED'** receipts, the sample data provided does not contain this category. **'FINISHED'** was used as a logical alternative to align with real-world scenarios where completed transactions represent accepted purchases).
- Used **AVG (total_spent)** to compute the **average spending per receipt** for both statuses.
- Applied **SUM (total_quantity_purchased)** to determine **which category had more items purchased**.
- Grouped by receipt_status to enable a direct comparison of ACCEPTED vs. REJECTED purchases.

Query-:

```
--3&4

CREATE OR REPLACE VIEW fetch.public.accepted_rejected_summary AS

SELECT

   receipt_status,
```

```
    AVG(total_spent) AS avg_total_spent,  -- Getting the average spend
    SUM(total_quantity_purchased) AS total_quantity_purchased
FROM fetch.public.receipts_summary
WHERE receipt_status IN ('FINISHED', 'REJECTED')
GROUP BY receipt_status;


-- Get avg spend & total items for Finished/Rejected Receipts & Get only receipt status & total items purchased
SELECT * FROM fetch.public.finished_rejected_summary;
```

Query Result-:

	RECEIPT_STATUS	AVG_TOTAL_SPENT	TOTAL_QUANTITY_PURCHASED
1	FINISHED	81.167693798	8183
2	REJECTED	24.355147059	141

Analysis of Results

- **Average Spend:** FINISHED receipts show a much higher average spend ($81.17) compared to REJECTED receipts ($24.36), indicating that completed transactions tend to have higher value.
- **Total Items Purchased:** FINISHED receipts account for **8,183 items**, whereas REJECTED receipts have only **141**, showing a clear drop in purchases for rejected transactions.
- **Business Impact:** The lower spend and item count for rejected receipts suggest issues like failed transactions, incorrect scans, or abandoned purchases. Understanding these causes could help improve conversion rates and user satisfaction.

Alignment with Business Modeling Best Practices

Building a Smart, Scalable Data Model-:

- Focused on meaningful receipt statuses (FINISHED and REJECTED) to get the most relevant insights.
- Ensured accurate spending and item count calculations by **aggregating at the receipt level**, avoiding duplication.
- Designed the approach to be **flexible and scalable**, so we can easily include ACCEPTED transactions if the data becomes available.

Capturing Broader Insights-:

- **Customer Spending Behavior** – Highlights how much users typically spend in completed vs. rejected transactions.
- **Purchase Volume Trends** – Reveals the disparity in total items purchased between successful and failed transactions.
- **Operational Efficiency** – Identifies potential issues leading to rejections, helping optimize the user experience and reduce lost revenue.

This approach ensures **actionable insights** beyond the initial questions, providing a foundation for improving transaction success rates and customer retention.

Analytical Approach Followed for Questions 5 & 6

Data Filtering & Structuring (new_users & user_brand_summary CTEs)

- **Identified new users** by selecting those who created their accounts in the past six months. Since the dataset does not contain recent transactions, I've dynamically referenced MAX(purchase_date) instead of CURRENT_DATE to determine the timeframe.
- **Aggregated spending and transaction data at the brand level** by linking purchases to brands and computing SUM(total_spent) and COUNT(DISTINCT receipt_id).
- **Grouped data by brand_code**, ensuring accurate calculations of spending and transaction counts.

Filtering & Aggregation for Analysis

- The analysis focuses on identifying:
    - The brand with the highest **total spend** among new users.
    - The brand with the **most transactions** among new users.
- Used **ORDER BY total_spend DESC** to find the top-spending brand.
- Applied **ORDER BY transaction_count DESC** to find the brand with the most transactions.

Query-:

```
--5&6

WITH new_users AS (

  SELECT user_id

  FROM fetch.public.users_cleaned

  -- The database does not contain receipts from the past 6 months, so instead of using CURRENT_DATE,
```

```
    -- we consider MAX(purchase_date) from the receipts dataset as a reference point.
    WHERE created_date >= DATEADD(MONTH, -6, (SELECT MAX(purchase_date) FROM
fetch.public.receipts_cleaned))
),
user_brand_summary AS (
    SELECT
        b.brand_code,
        SUM(rc.total_spent) AS total_spend,
        COUNT(DISTINCT rc.receipt_id) AS transaction_count
    FROM fetch.public.receipt_items ri
    JOIN fetch.public.receipts_cleaned rc ON ri.receipt_id = rc.receipt_id
    JOIN fetch.public.brands_cleaned b ON ri.item_barcode = b.barcode
    WHERE rc.user_id IN (SELECT user_id FROM new_users)
    GROUP BY b.brand_code
)
SELECT
    -- Identifying the brand with the highest total spend among new users
    (SELECT brand_code FROM user_brand_summary ORDER BY total_spend DESC LIMIT 1) AS
top_brand_by_spend,
    -- Identifying the brand with the most transactions among new users
    (SELECT brand_code FROM user_brand_summary ORDER BY transaction_count DESC LIMIT 1) AS
top_brand_by_transactions;
```

Query Result-:

	TOP_BRAND_BY_SPEND	TOP_BRAND_BY_TRANSACTIONS
1	TOSTITOS	SWANSON

Analysis of Results

- **Top Brand by Spend:** Tostitos had the highest total spending among new users, meaning they spent the most on this brand.

- **Top Brand by Transactions:** Swanson had the most transactions, indicating frequent purchases by new users.
- **Business Impact:**
    - **Tostitos' high spend** suggests strong brand loyalty or premium product positioning.
    - **Swanson's frequent transactions** indicate a high purchase frequency, making it a staple brand for new users.

Analysis of Results

- **Top Brand by Spend:** Tostitos had the highest total spending among new users, meaning they spent the most on this brand.
- **Top Brand by Transactions:** Swanson had the most transactions, indicating frequent purchases by new users.
- **Business Impact:**
    - **Tostitos' high spend** suggests strong brand loyalty or premium product positioning.
    - **Swanson's frequent transactions** indicate a high purchase frequency, making it a staple brand for new users.
    - Understanding user behavior can help **optimize promotions and targeted marketing** for high-value and frequently purchased brands.

This approach not only answers business questions but also uncovers deeper user-brand interactions, enabling smarter marketing and engagement strategies.

4. Business Impact & Recommendations

This structured analytical approach **not only answers the business questions** but also uncovers valuable insights into **user spending behavior, brand engagement, and transaction success rates**.

Key Takeaways:

→ **Understanding Brand Performance**

- The ranking insights help identify **top brands**, enabling **targeted promotions** for high-performing products.

→ **Improving Transaction Success Rates**

- The disparity between FINISHED and REJECTED receipts suggests areas for improvement in **receipt scanning accuracy** and **user experience**.

→ **Enhancing New User Engagement**

- Identifying the top brands purchased by new users helps optimize **onboarding promotions** and **personalized marketing strategies**.

5. Conclusion

This analysis demonstrates how structured **data modeling, efficient SQL querying, and strategic insights** can support **business decision-making**. By transforming raw JSON data into a **clean, structured format**, we have built a foundation for **scalable, automated, and insightful analytics**.

In addition to the analysis of Question 2, I have included a **document** in the same folder comprising of Snowflake SQL queries that can be run in **Snowflake environment**. This notebook contains the complete set of queries used for data cleaning, transformation, and analysis, ensuring full transparency and reproducibility of the results.