**Question 1: What are the top 5 brands by receipts scanned for the most recent month?**

**Task**: To identify the top 5 brands based on the number of receipts scanned during the most recent month recorded in the receipts\_main table.

WITH recent AS (

SELECT TOP 1

YEAR(r.scanned\_date) AS year, MONTH(r.scanned\_date) AS month

FROM receipts\_main r

ORDER BY r.scanned\_date DESC

),

res AS (

SELECT TOP 5

i.barcode

FROM receipts\_main r

JOIN receipt\_items i ON r.receipt\_id = i.receipt\_id

WHERE YEAR(r.scanned\_date) = (SELECT year FROM recent)

AND MONTH(r.scanned\_date) = (SELECT month FROM recent)

GROUP BY i.barcode

ORDER BY COUNT(i.barcode) DESC

)

SELECT b.brand\_id, b.brand\_name FROM brands\_data b

JOIN res ON b.barcode = res.barcode;

**Explanation:**

1. **CTE - recent**:

* This CTE retrieves the most recent year and month based on the scanned\_date from the receipts\_main table.
* It sorts the records by scanned\_date in descending order and selects the top entry, ensuring that we capture the latest month for analysis.

1. **CTE - res**:

* This CTE finds the top 5 barcodes from the receipt\_items table, which are associated with the receipts scanned in the most recent month.
* It joins receipts\_main with receipt\_items on receipt\_id, filters the records by the year and month obtained from the first CTE, groups the results by barcode, and orders them by the count of receipts scanned in descending order.

1. **Final Selection**:

* The main query retrieves brand\_id and brand\_name from the brands\_data table by joining it with the results from the res CTE based on the barcode.

#### **Results:**

* Upon executing the query, the most recent data indicates that the most recent year is **2021** and the month is **March**
* The query identifies a list of the top 5 barcodes scanned during March 2021. However, it is important to note that the barcodes obtained from this analysis are not present in the brands\_data table.
* As a result, the final output of the query is **empty**, indicating that there are no matching brands for the barcodes scanned in the most recent month.

**Question 3: When considering *average spend* from receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?**

**Task:** To evaluate the average spending of customers for receipts categorized by their rewardsReceiptStatus.

Notably, the provided dataset does not include any receipts with a status of 'Accepted.'

As a result, the query is written n order to extracts the total spent and average spent for the various statuses present in the dataset, which includes 'FLAGGED,' 'FINISHED,' 'PENDING,' 'REJECTED,' and 'SUBMITTED.'

SELECT

rewards\_receipt\_status AS STATUS,

SUM(total\_spent) AS TotalSpent,

(SUM(total\_spent) / COUNT(rewards\_receipt\_status)) AS AvgSpent

FROM receipts\_main

GROUP BY rewards\_receipt\_status

ORDER BY AvgSpent DESC;

**Explanation:**

1. **SELECT Statement**:

* rewards\_receipt\_status AS STATUS: Retrieves the status of the rewards receipt, aliased as STATUS for clarity.
* SUM(total\_spent) AS TotalSpent: Calculates the total amount spent for each rewards receipt status, labeled as TotalSpent.
* (SUM(total\_spent) / COUNT(rewards\_receipt\_status)) AS AvgSpent: Computes the average spending by dividing the total spent by the count of receipts for each status, labeled as AvgSpent.

1. **FROM Clause**:

* FROM receipts\_main: Indicates that the data is pulled from the receipts\_main table.

1. **GROUP BY Clause**:

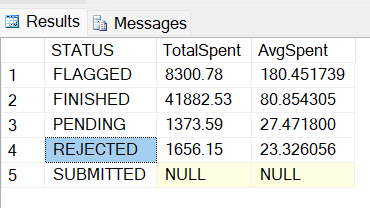
* GROUP BY rewards\_receipt\_status: Groups the results by the rewards\_receipt\_status, enabling separate calculations for each status present in the data.

1. **ORDER BY Clause**:

* ORDER BY AvgSpent DESC: Orders the results in descending order based on average spending, allowing for easy identification of which status has a higher average spend.

**Results:**

The output of the query:



* The analysis of the average spending across different rewards receipt statuses highlights that, among the available data, the 'FLAGGED' status shows the highest average spend per receipt.
* However, it is important to note that there is no data for the 'ACCEPTED' status, which limits the completeness of this analysis.

**Question 4: When considering *total number of items purchased* from receipts with 'rewardsReceiptStatus’ of ‘Accepted’ or ‘Rejected’, which is greater?**

Task: To evaluate the total number of items purchased from receipts categorized by their rewardsReceiptStatus.

Notably, the provided dataset does not include any receipts with a status of 'Accepted.'

As a result, the query extracts the total number of purchased items for the various statuses present in the dataset.

SELECT

rewards\_receipt\_status AS STATUS,

SUM(purchased\_item\_count) AS ItemsCount

FROM receipts\_main

GROUP BY rewards\_receipt\_status

ORDER BY ItemsCount DESC;

Explanation:

1. **SELECT Statement**:

* rewards\_receipt\_status AS STATUS: This retrieves the status of the rewards receipt, aliased as STATUS for clarity in the output.
* SUM(purchased\_item\_count) AS ItemsCount: This calculates the total number of items purchased for each rewards receipt status, labelled as ItemsCount.

1. **FROM Clause**:

* FROM receipts\_main: This indicates that the data is being pulled from the receipts\_main table.

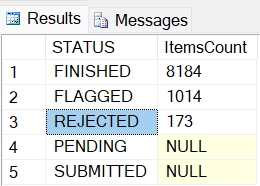
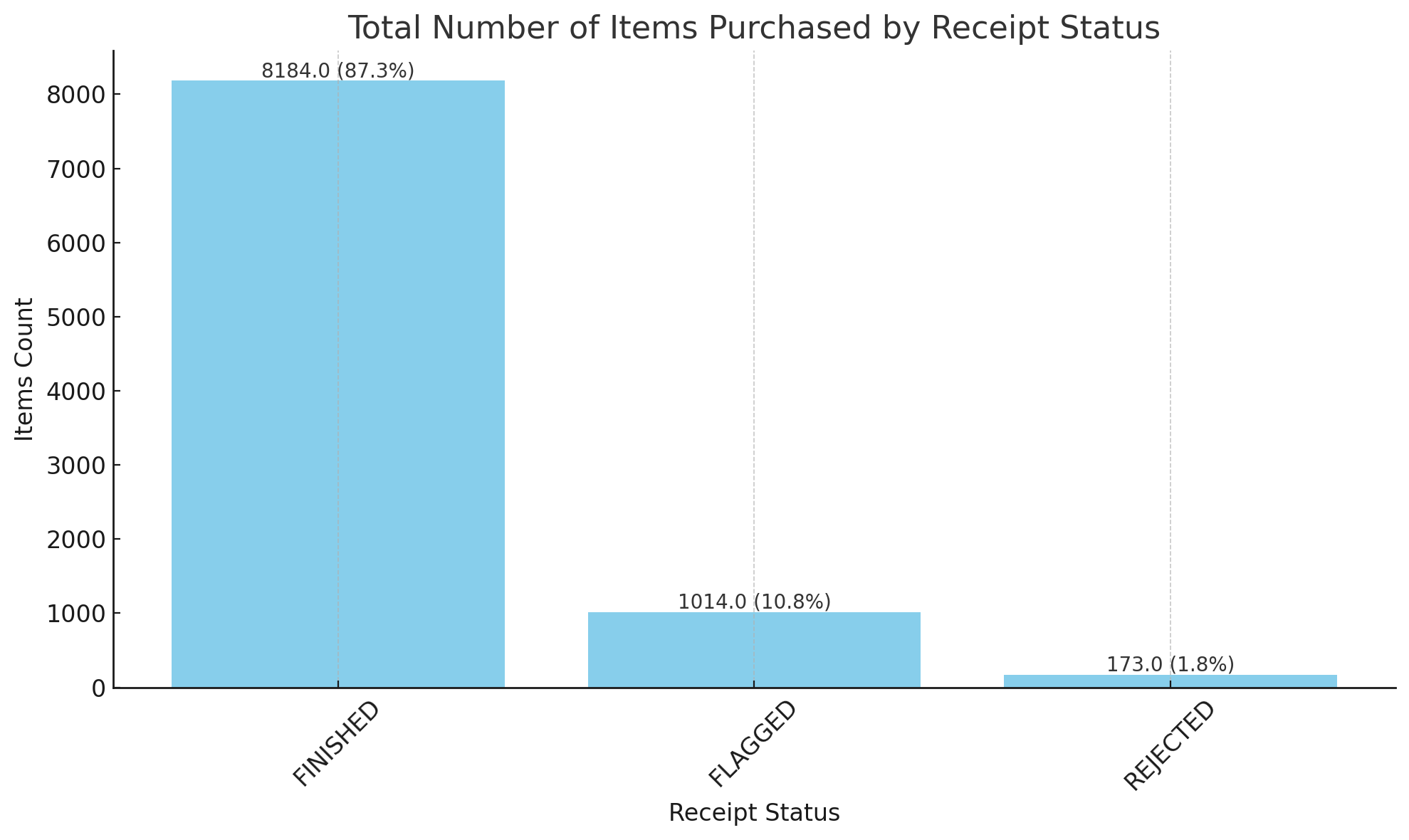
1. **GROUP BY Clause**:

* GROUP BY rewards\_receipt\_status: This groups the results by the rewards\_receipt\_status, allowing for separate calculations for each status present in the data.

1. **ORDER BY Clause**:

* ORDER BY ItemsCount DESC: This orders the final results in descending order based on the total number of items purchased, allowing for easy identification of which status has a higher count.

**Result:**

The output of the query:

* From the analysis, it is evident that the 'FINISHED' status has the highest total number of items purchased, significantly surpassing other statuses.
* The 'FLAGGED' status also shows a notable count, while 'REJECTED' has a much lower total.

**Question 5: Which brand has the most *spend* among users who were created within the past 6 months?**

**Task:** to identify the brand that has the highest total spending from users who were created within the last six months. By analyzing the purchasing behavior of these new users, businesses can tailor their marketing strategies and product offerings to better engage this demographic.

It is important to note that instead of using the current date to fetch data regarding the past six months, we are basing our analysis on the most recent date available in the dataset. This means we are calculating the last six months from the latest recorded user creation date

WITH recent AS (

SELECT user\_id AS users FROM users\_data

WHERE created\_date > DATEADD(MONTH, -6, (SELECT TOP 1 created\_date FROM users\_data

ORDER BY created\_date DESC))

),

barcode\_spending AS (

SELECT i.barcode, SUM(r.total\_spent) AS TotalSpent FROM receipts\_main r

JOIN receipt\_items i ON r.receipt\_id = i.receipt\_id

WHERE r.user\_id IN (SELECT users FROM recent)

GROUP BY i.barcode

)

SELECT bs.barcode, bs.TotalSpent, b.brand\_name FROM barcode\_spending bs

LEFT JOIN brands\_data b ON bs.barcode = b.barcode

ORDER BY bs.TotalSpent DESC;

#### **Explanation**

#### Common Table Expression (CTE) - recent:

#### This CTE identifies users who have been created in the last six months by comparing their created\_date to the most recent date recorded in the users\_data table.

1. Common Table Expression (CTE) - barcode\_spending:

* This CTE calculates the total amount spent by these recent users for each barcode associated with their purchases. It aggregates the spending from the receipts\_main and receipt\_items tables.

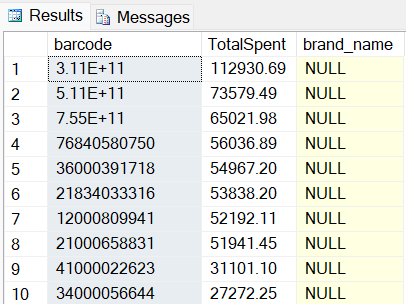
1. Final Selection:

* The final query selects the barcode, total spending, and the corresponding brand\_name by performing a LEFT JOIN with the brands\_data table. This ensures that all barcodes with total spending are displayed, even if there is no matching brand name.

1. Ordering:

* The results are ordered by TotalSpent in descending order, allowing for the quick identification of which brand has the highest spending.

Result:

* Upon executing the query, the output will provide a list of barcodes along with their total spending and associated brand names.
* This query effectively identifies the brand with the highest total spending among new users.



**Question 6: Which brand has the most *transactions* among users who were created within the past 6 months?**

Task: To identify the brand that has the highest number of transactions from users who were created within the last six months. By analyzing the transaction behavior of these new users, businesses can gain insights into brand engagement and customer preferences, allowing them to optimize marketing strategies and product offerings.

It is important to note that instead of using the current date to fetch data regarding the past six months, we are basing our analysis on the most recent date available in the dataset.

WITH recent AS (

SELECT user\_id AS users FROM users\_data

WHERE created\_date > DATEADD(MONTH, -6, (SELECT TOP 1 created\_date FROM users\_data

ORDER BY created\_date DESC))

)

SELECT i.barcode, COUNT(i.barcode) AS Transactions, b.brand\_name FROM receipts\_main r

JOIN receipt\_items i ON r.receipt\_id = i.receipt\_id

LEFT JOIN brands\_data b ON i.barcode = b.barcode

WHERE r.user\_id IN (SELECT users FROM recent)

GROUP BY i.barcode, b.brand\_name

ORDER BY Transactions DESC;

Explanation:

1. **Common Table Expression (CTE) - recent**:

* This CTE identifies users who have been created in the last six months by comparing their created\_date to the most recent date recorded in the users\_data table.

1. **Main Query**:

* The main query selects the barcode, counts the number of transactions for each barcode, and retrieves the corresponding brand\_name.
* The query joins the receipts\_main and receipt\_items tables to aggregate the transaction data and uses a LEFT JOIN with the brands\_data table to include brand names.

1. **WHERE Clause**:

* This filters the records to include only those transactions associated with users created in the last six months.

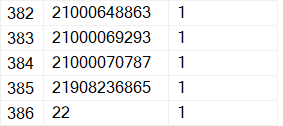
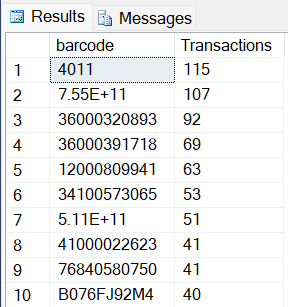
1. **GROUP BY Clause**:

* The results are grouped by both barcode and brand\_name to ensure accurate counting of transactions for each unique barcode-brand combination.

1. **Ordering**:

* The results are ordered by the Transactions count in descending order, allowing for quick identification of which brand has the most transactions.

### **Results**



* This query effectively identifies the brand with the highest number of transactions among new users.
* By focusing on users created within the last six months, businesses can gain insights into which brands are engaging new customers most effectively.