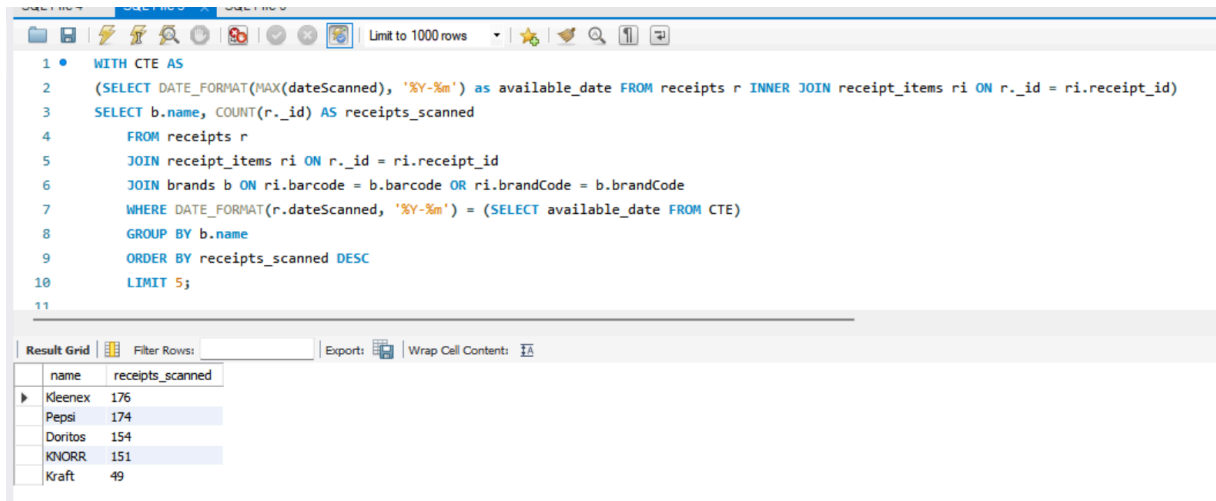## 1) top 5 brands for recent month
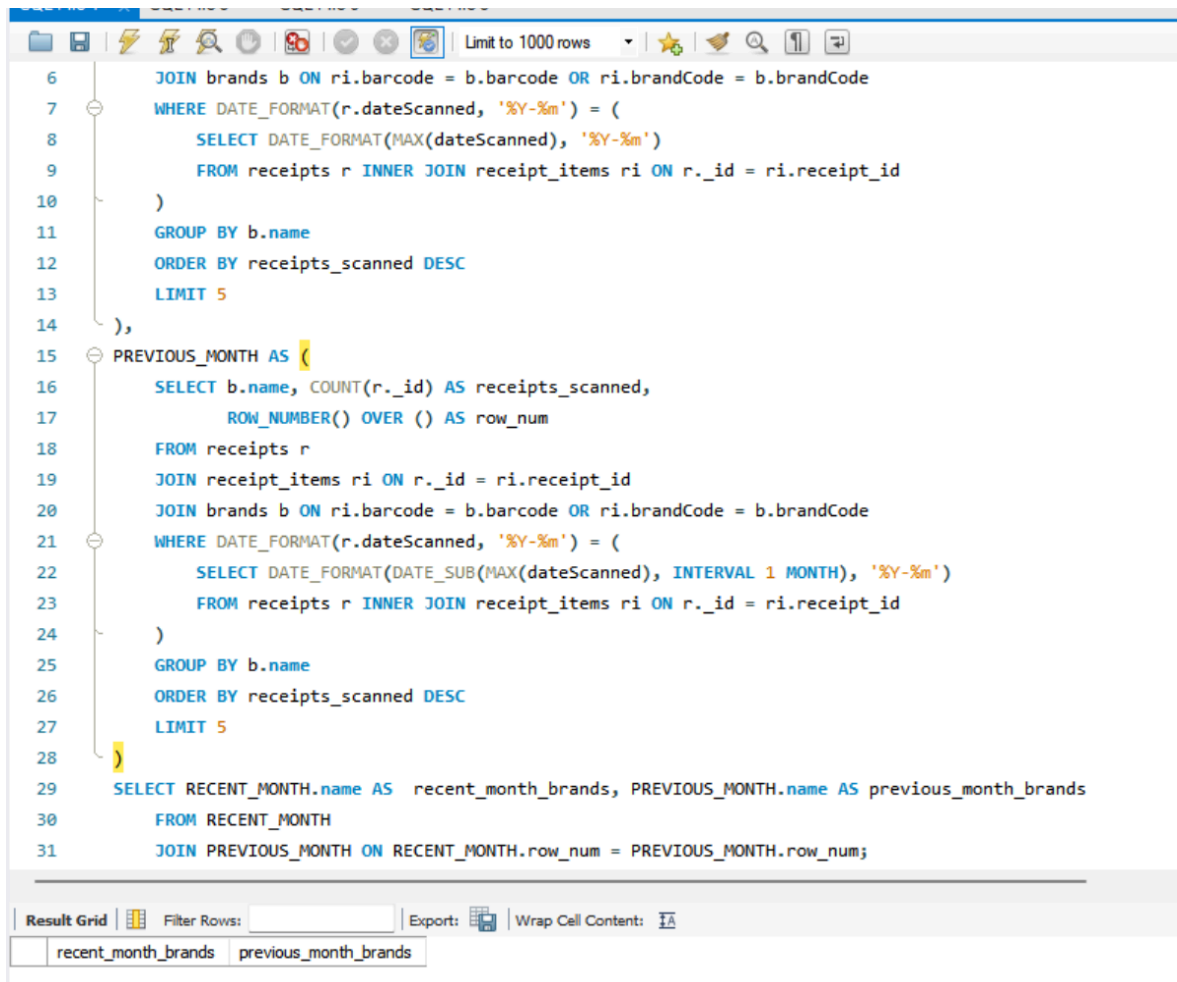
```sql
WITH CTE AS
  (SELECT DATE_FORMAT(MAX(dateScanned), '%Y-%m') as available_date FROM receipts r INNER JOIN receipt_items ri ON r._id = ri.receipt_id)
  SELECT b.name, COUNT(r._id) AS receipts_scanned
    FROM receipts r
    JOIN receipt_items ri ON r._id = ri.receipt_id
    JOIN brands b ON ri.barcode = b.barcode OR ri.brandCode = b.brandCode
    WHERE DATE_FORMAT(r.dateScanned, '%Y-%m') = (SELECT available_date FROM CTE)
    GROUP BY b.name
    ORDER BY receipts_scanned DESC
    LIMIT 5;
```

name	receipts_scanned
Kleenex	176
Pepsi	174
Doritos	154
KNORR	151
Kraft	49

2) top 5 brands while comparing recent and previous months
getting empty result here as previous month has no data after joining with brands

```sql
      JOIN brands b ON ri.barcode = b.barcode OR ri.brandCode = b.brandCode
      WHERE DATE_FORMAT(r.dateScanned, '%Y-%m') = (
          SELECT DATE_FORMAT(MAX(dateScanned), '%Y-%m')
          FROM receipts r INNER JOIN receipt_items ri ON r._id = ri.receipt_id
      )
      GROUP BY b.name
      ORDER BY receipts_scanned DESC
      LIMIT 5
  ),
  PREVIOUS_MONTH AS (
      SELECT b.name, COUNT(r._id) AS receipts_scanned,
             ROW_NUMBER() OVER () AS row_num
      FROM receipts r
      JOIN receipt_items ri ON r._id = ri.receipt_id
      JOIN brands b ON ri.barcode = b.barcode OR ri.brandCode = b.brandCode
      WHERE DATE_FORMAT(r.dateScanned, '%Y-%m') = (
          SELECT DATE_FORMAT(DATE_SUB(MAX(dateScanned), INTERVAL 1 MONTH), '%Y-%m')
          FROM receipts r INNER JOIN receipt_items ri ON r._id = ri.receipt_id
      )
      GROUP BY b.name
      ORDER BY receipts_scanned DESC
      LIMIT 5
  )
  SELECT RECENT_MONTH.name AS  recent_month_brands, PREVIOUS_MONTH.name AS previous_month_brands
      FROM RECENT_MONTH
      JOIN PREVIOUS_MONTH ON RECENT_MONTH.row_num = PREVIOUS_MONTH.row_num;
```

recent_month_brands	previous_month_brands

3) accepted vs rejected for average of total spent

```sql
1 • SELECT r.rewardsReceiptStatus AS receipt_status, AVG(r.totalSpent) as avg_total_spent
2   FROM receipts r
3   WHERE r.rewardsReceiptStatus IN ('FINISHED', 'REJECTED')
4   GROUP BY r.rewardsReceiptStatus
5   ORDER BY AVG(r.totalSpent) DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

receipt_status	avg_total_spent
FINISHED	84.08475732320716
REJECTED	7.7389999071757

4) accepted vs rejected for total quantities of items purchased
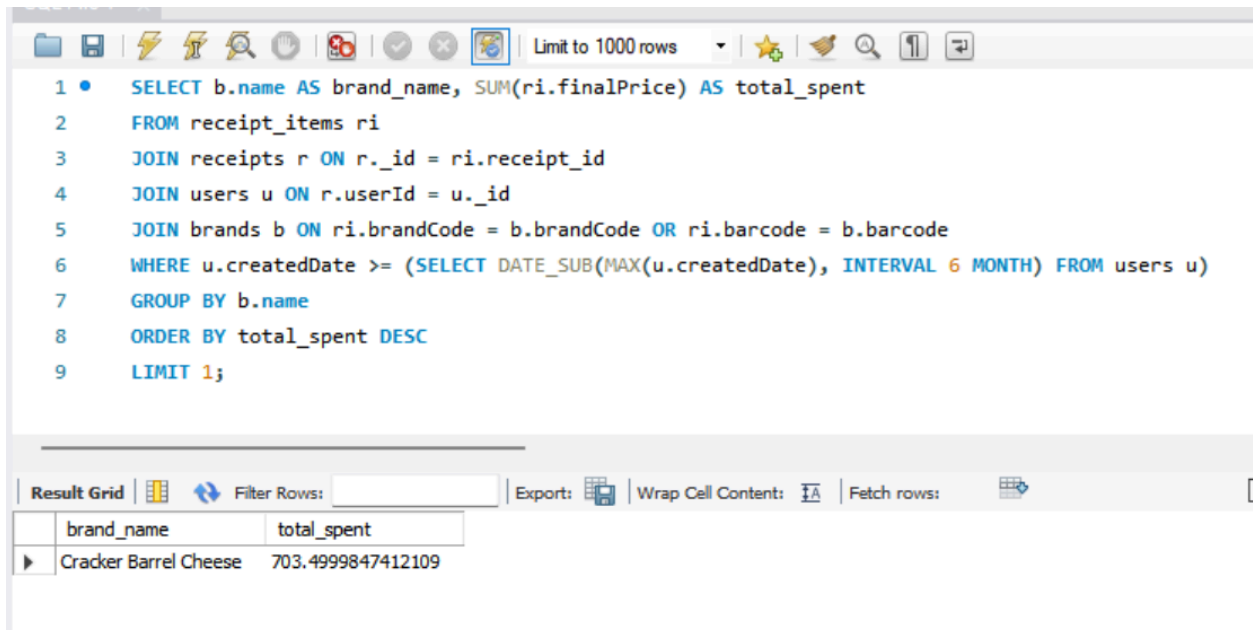
```sql
1 • SELECT r.rewardsReceiptStatus AS receipt_status, SUM(r.purchasedItemCount) as total_items_purchased
2   FROM receipts r
3   WHERE r.rewardsReceiptStatus IN ('FINISHED', 'REJECTED')
4   GROUP BY r.rewardsReceiptStatus
5   ORDER BY SUM(r.purchasedItemCount) DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

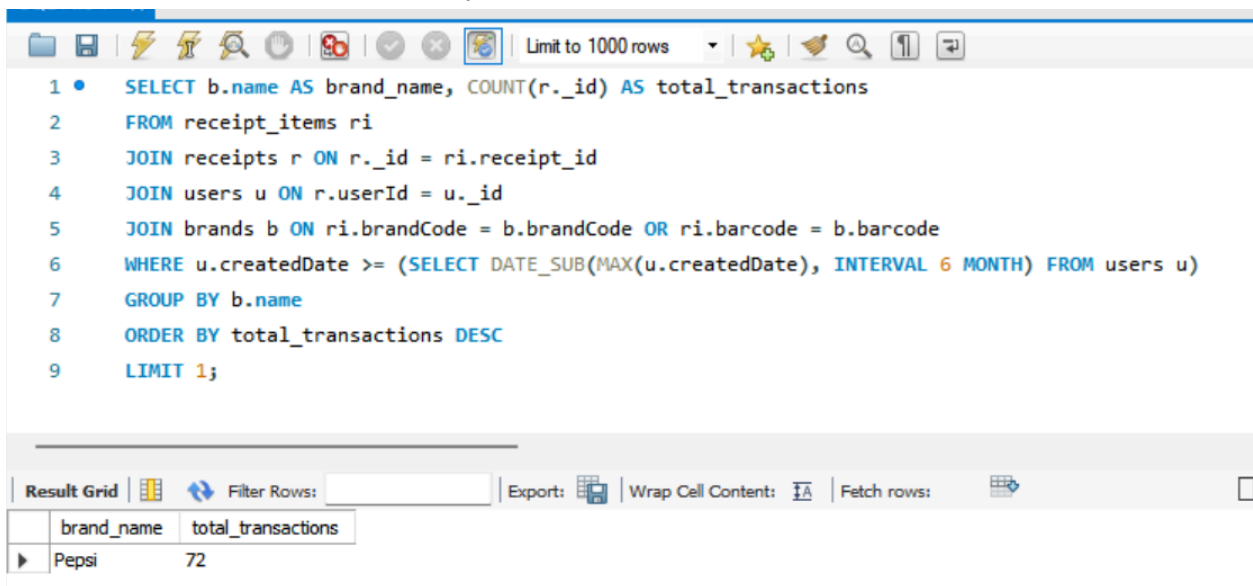| receipt_status	total_items_purchased
FINISHED	8124
REJECTED	91

5) top brand for most spent by new users (past 6 months)

```sql
SELECT b.name AS brand_name, SUM(ri.finalPrice) AS total_spent
FROM receipt_items ri
JOIN receipts r ON r._id = ri.receipt_id
JOIN users u ON r.userId = u._id
JOIN brands b ON ri.brandCode = b.brandCode OR ri.barcode = b.barcode
WHERE u.createdDate >= (SELECT DATE_SUB(MAX(u.createdDate), INTERVAL 6 MONTH) FROM users u)
GROUP BY b.name
ORDER BY total_spent DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

brand_name	total_spent
Cracker Barrel Cheese	703.4999847412109

6) top brand for most transactions by new users (past 6 months)

```sql
SELECT b.name AS brand_name, COUNT(r._id) AS total_transactions
FROM receipt_items ri
JOIN receipts r ON r._id = ri.receipt_id
JOIN users u ON r.userId = u._id
JOIN brands b ON ri.brandCode = b.brandCode OR ri.barcode = b.barcode
WHERE u.createdDate >= (SELECT DATE_SUB(MAX(u.createdDate), INTERVAL 6 MONTH) FROM users u)
GROUP BY b.name
ORDER BY total_transactions DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

brand_name	total_transactions
Pepsi	72