- **What are the top 5 brands by receipts scanned for the most recent month?**

```
WITH RankedBrands AS
(
   SELECT b.name AS brand_name, COUNT(r._id) AS
receipts_count,     DENSE_RANK() OVER (ORDER BY COUNT(r._id) DESC) AS
brand_rank
   FROM Brand b
   JOIN RewardsReceiptItemList r ON b.barcode = r.barcode
   WHERE DATE_TRUNC('month', r.dateScanned) = DATE_TRUNC('month',
CURRENT_DATE)

GROUP BY b.name
)
SELECT brand_name,  receipts_count
 FROM RankedBrands
WHERE brand_rank <= 5;
```

- **How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?**

```
WITH RankedBrands AS
(
   SELECT b.name AS brand_name, COUNT(r._id) AS receipts_count,
      DATE_TRUNC('month', r.dateScanned) AS scan_month,
      DENSE_RANK() OVER (PARTITION BY DATE_TRUNC('month', r.dateScanned)
ORDER BY COUNT(r._id) DESC) AS brand_rank
   FROM Brand b
   JOIN RewardsReceiptItemList r ON b.barcode = r.barcode
   WHERE
      DATE_TRUNC('month', r.dateScanned) IN (DATE_TRUNC('month',
CURRENT_DATE), DATE_TRUNC('month', CURRENT_DATE - INTERVAL '1'
MONTH))
   GROUP BY b.name, scan_month
)
SELECT brand_name,
   MAX(CASE WHEN scan_month = DATE_TRUNC('month', CURRENT_DATE) THEN
brand_rank END) AS current_month_rank,
   MAX(CASE WHEN scan_month = DATE_TRUNC('month', CURRENT_DATE -
INTERVAL '1' MONTH) THEN brand_rank END) AS previous_month_rank
FROM RankedBrands
WHERE brand_rank <= 5
GROUP BY Brand_name;
```

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

```
SELECT
    CASE
        WHEN rewardsReceiptStatus = 'Accepted' THEN 'Accepted'
        WHEN rewardsReceiptStatus = 'Rejected' THEN 'Rejected'
    END AS status,
    AVG(totalSpent) AS average_spend
FROM Receipts
WHERE rewardsReceiptStatus IN ('Accepted', 'Rejected')
GROUP BY rewardsReceiptStatus;
```

If you want a single result indicating which status has a greater average spend, you can further modify the query:

```
SELECT
    MAX(status) AS greater_status,
    MAX(average_spend) AS greater_average_spend
FROM
(
    SELECT
        CASE
            WHEN rewardsReceiptStatus = 'Accepted' THEN 'Accepted'
            WHEN rewardsReceiptStatus = 'Rejected' THEN 'Rejected'
        END AS status,
        AVG(totalSpent) AS average_spend
    FROM Receipts
    WHERE rewardsReceiptStatus IN ('Accepted', 'Rejected')
    GROUP BY rewardsReceiptStatus
) AS subquery;
```

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

```
SELECT
    CASE
        WHEN rewardsReceiptStatus = 'Accepted' THEN 'Accepted'
        WHEN rewardsReceiptStatus = 'Rejected' THEN 'Rejected'
    END AS status,
    SUM(purchasedItemCount) AS total_items_purchased
```

```
FROM Receipts
WHERE rewardsReceiptStatus IN ('Accepted', 'Rejected')
GROUP BY rewardsReceiptStatus;
```

If you want a single result indicating which status has a greater total number of items purchased, you can further modify the query:

```
SELECT
   MAX(status) AS greater_status,
   MAX(total_items_purchased) AS greater_total_items_purchased
FROM
(
   SELECT
      CASE
         WHEN rewardsReceiptStatus = 'Accepted' THEN 'Accepted'
         WHEN rewardsReceiptStatus = 'Rejected' THEN 'Rejected'
      END AS status,
      SUM(purchasedItemCount) AS total_items_purchased
   FROM Receipts
   WHERE rewardsReceiptStatus IN ('Accepted', 'Rejected')
   GROUP BY rewardsReceiptStatus
) AS subquery;
```

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

```
WITH RecentUsers AS
(
   SELECT  _id AS user_id, createdDate
   FROM Users
   WHERE createdDate >= CURRENT_DATE - INTERVAL '6' MONTH
),
UserBrandSpend AS
(
   SELECT u.user_id, r.totalSpent, b.name AS brand_name
   FROM RecentUsers u
   JOIN Receipts r ON u.user_id = r.userId
   JOIN RewardsReceiptItemList ri ON r._id = ri.rewardsReceiptId
   JOIN Brand b ON ri.barcode = b.barcode
)
SELECT brand_name, SUM(totalSpent) AS total_spend
FROM UserBrandSpend
GROUP BY brand_name
ORDER BY total_spend DESC
LIMIT 1;
```

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

```
WITH RecentUsers AS
(
   SELECT  _id AS user_id, createdDate
   FROM Users
   WHERE createdDate >= CURRENT_DATE - INTERVAL '6' MONTH
),
UserBrandTransactions AS
(
   SELECT u.user_id, r._id AS receipt_id, b.name AS brand_name
   FROM RecentUsers u
   JOIN Receipts r ON u.user_id = r.userId
   JOIN RewardsReceiptItemList ri ON r._id = ri.rewardsReceiptId
   JOIN Brand b ON ri.barcode = b.barcode
)
SELECT brand_name, COUNT(DISTINCT receipt_id) AS transaction_count
FROM UserBrandTransactions
GROUP BY brand_name
ORDER BY  transaction_count DESC
LIMIT 1;
```