Second: SQL Queries on the Structural Relational Model

- **Top 5 Brands by Receipts Scanned for the Most Recent Month:**

```
SELECT
   b.name AS brand_name,
   COUNT(DISTINCT r.receipt_id) AS receipts_scanned
FROM fact_receipts r
JOIN dim_items i ON r.receipt_id = i.receipt_id
JOIN dim_brands b ON i.barcode = b.barcode
WHERE r.date_scanned >= date_trunc('month', current_date)
  AND r.date_scanned < date_trunc('month', current_date) + interval '1 month'
GROUP BY b.name
ORDER BY receipts_scanned DESC
LIMIT 5;
```

- **Compare Top 5 Brands by Receipts Scanned: Recent Month vs. Previous Month**

```
WITH current_month AS (
   SELECT
      b.name AS brand_name,
      COUNT(DISTINCT r.receipt_id) AS receipts_scanned,
      RANK() OVER (ORDER BY COUNT(DISTINCT r.receipt_id) DESC) AS current_rank
   FROM fact_receipts r
   JOIN dim_items i ON r.receipt_id = i.receipt_id
   JOIN dim_brands b ON i.barcode = b.barcode
   WHERE r.date_scanned >= date_trunc('month', current_date)
     AND r.date_scanned < date_trunc('month', current_date) + interval '1 month'
   GROUP BY b.name
),
previous_month AS (
   SELECT
      b.name AS brand_name,
      COUNT(DISTINCT r.receipt_id) AS receipts_scanned,
```

```
        RANK() OVER (ORDER BY COUNT(DISTINCT r.receipt_id) DESC) AS previous_rank
    FROM fact_receipts r
    JOIN dim_items i ON r.receipt_id = i.receipt_id
    JOIN dim_brands b ON i.barcode = b.barcode
    WHERE r.date_scanned >= date_trunc('month', current_date - interval '1 month')
      AND r.date_scanned < date_trunc('month', current_date)
    GROUP BY b.name
)
SELECT
    COALESCE(c.brand_name, p.brand_name) AS brand_name,
    c.receipts_scanned AS current_month_scanned,
    c.current_rank,
    p.receipts_scanned AS previous_month_scanned,
    p.previous_rank
FROM current_month c
FULL OUTER JOIN previous_month p ON c.brand_name = p.brand_name
ORDER BY current_rank;
```

- **Compare Average Spend for Receipts with Status 'Accepted' vs. 'Rejected'**

```
WITH avg_spend AS (
    SELECT
        rewards_receipt_status,
        AVG(total_spent::numeric) AS avg_spend
    FROM fact_receipts
    WHERE rewards_receipt_status IN ('Accepted', 'Rejected')
    GROUP BY rewards_receipt_status
)
SELECT
    CASE
        WHEN (SELECT avg_spend FROM avg_spend WHERE rewards_receipt_status =
'Accepted') >
            (SELECT avg_spend FROM avg_spend WHERE rewards_receipt_status =
'Rejected')
        THEN 'Accepted has greater average spend'
```

```
    ELSE 'Rejected has greater average spend'
  END AS result;
```

- **Compare Total Number of Items Purchased for Receipts with Status 'Accepted' vs. 'Rejected'**

```
SELECT
  rewards_receipt_status,
  SUM(purchased_item_count) AS total_items
FROM fact_receipts
WHERE rewards_receipt_status IN ('Accepted', 'Rejected')
GROUP BY rewards_receipt_status;
```

- **Brand with the Most Spend Among Users Created Within the Past 6 Months**

```
SELECT
  b.name AS brand_name,
  SUM(r.total_spent::numeric) AS total_spend
FROM fact_receipts r
JOIN dim_users u ON r.user_id = u.user_id
JOIN dim_items i ON r.receipt_id = i.receipt_id
JOIN dim_brands b ON i.barcode = b.barcode
WHERE u.created_date >= current_date - interval '6 months'
GROUP BY b.name
ORDER BY total_spend DESC
LIMIT 1;
```

- **Brand with the Most Transactions Among Users Created Within the Past 6 Months**

```
SELECT
   b.name AS brand_name,
   COUNT(DISTINCT r.receipt_id) AS transactions
FROM fact_receipts r
JOIN dim_users u ON r.user_id = u.user_id
JOIN dim_items i ON r.receipt_id = i.receipt_id
JOIN dim_brands b ON i.barcode = b.barcode
WHERE u.created_date >= current_date - interval '6 months'
GROUP BY b.name
ORDER BY transactions DESC
LIMIT 1;
```