**Fetch Rewards Coding Exercise**

**Brianna Lytle**

# Pt.2 SQL Queries

|  |  |  |
| --- | --- | --- |
| Index | Question | Query |
| 1 | What are the top 5 brands by receipts scanned for the most recent month? | WITH RecentMonth AS (      SELECT MAX(EXTRACT(YEAR from r.purchaseDate)) AS MaxMonth      FROM df\_receipts r  ),  ReceiptsbyBrand AS (      SELECT          b.name AS brand\_name,          COUNT(DISTINCT r.\_id) as transaction\_count,          DATE\_TRUNC('month', r.purchaseDate) AS Month      FROM df\_receipts r      INNER JOIN df\_items it on r.\_id =it.\_id      INNER JOIN df\_brands b ON it.barcode = b.barcode or it.brandCode = b.brandCode      GROUP BY b.name, Month  )  SELECT      rb.brand\_name,      rb.transaction\_count  FROM ReceiptsbyBrand rb  INNER JOIN RecentMonth rm ON rb.Month = rm.MaxMonth  ORDER BY rb.transaction\_count DESC |
| 2 | 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 MonthInfo AS (      SELECT          DATE\_TRUNC('month', MAX(r.purchaseDate)) AS MostRecentMonth,          DATE\_TRUNC('month', MAX(r.purchaseDate)) - INTERVAL '1 month' AS PreviousMonth      FROM df\_receipts r  ),  ReceiptsByBrand AS (      SELECT          b.name AS BrandName,          COUNT(DISTINCT CASE WHEN DATE\_TRUNC('month', r.purchaseDate) = m.MostRecentMonth THEN r.\_id END) AS RecentMonthReceipts,          COUNT(DISTINCT CASE WHEN DATE\_TRUNC('month', r.purchaseDate) = m.PreviousMonth THEN r.\_id END) AS PreviousMonthReceipts      FROM df\_receipts r      INNER JOIN df\_items it ON r.\_id = it.\_id      LEFT JOIN df\_brands b ON it.barcode = b.barcode OR it.brandCode = b.brandCode      CROSS JOIN MonthInfo m      GROUP BY b.name  )  SELECT      rb.BrandName,      rb.RecentMonthReceipts,      rb.PreviousMonthReceipts  FROM ReceiptsByBrand rb  WHERE rb.RecentMonthReceipts > 0 OR rb.PreviousMonthReceipts > 0  ORDER BY rb.RecentMonthReceipts DESC, rb.PreviousMonthReceipts DESC  LIMIT 5 |
| 3 | When considering *average spend* from receipts with ‘rewardsReceiptStatus of ‘Accepted’ or ‘Rejected’, which is greater? | SELECT          rewardsReceiptStatus,          AVG(totalSpent)      FROM df\_receipts      GROUP BY 1      ORDER BY 2 DESC |
| 4 | When considering *total number of items purchased* from receipts with ‘rewardsReceiptStatus of ‘Accepted’ or ‘Rejected’, which is greater? | SELECT          r.rewardsReceiptStatus,          SUM(it.quantityPurchased) items\_purchased      FROM df\_items it      INNER JOIN df\_receipts r ON it.\_id == r.\_id      GROUP BY 1      ORDER BY 2 DESC |
| 5 | Which brand has the most *spend* among users who were created within the past 6 months? | SELECT          b.name as BrandName,          COUNT(DISTINCT r.totalSpent) as totalspend      FROM df\_items it      INNER JOIN df\_receipts r on it.\_id = r.\_id      INNER JOIN df\_users u on r.userId = u.\_id      LEFT JOIN df\_brands b on b.barcode = it.barcode OR b.brandCode = it.brandCode      WHERE u.createdDate >= DATE\_ADD(CURRENT\_DATE, INTERVAL '-6 months')      GROUP BY b.name      ORDER BY totalspend DESC      LIMIT 1 |
| 6 | Which brand has the most *transactions* among users who were created within the past 6 months? | SELECT          b.name as BrandName,          COUNT(DISTINCT r.\_id) as transaction\_count    FROM df\_items it      INNER JOIN df\_receipts r on it.\_id = r.\_id      INNER JOIN df\_users u on r.userId = u.\_id      LEFT JOIN df\_brands b on b.barcode = it.barcode OR b.brandCode = it.brandCode      WHERE u.createdDate >= DATE\_ADD(CURRENT\_DATE, INTERVAL '-6 months')      GROUP BY b.name      ORDER BY transaction\_count DESC      LIMIT 1 |