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

WITH RecentMonthReceipts AS (

  SELECT

    DATE\_TRUNC(CURRENT\_DATE(), MONTH) AS start\_date,

    DATE\_TRUNC(CURRENT\_DATE(), MONTH) + INTERVAL 1 MONTH AS end\_date

)

SELECT

  b.name AS brandName,

  COUNT(r.receiptId) AS receiptsScanned

FROM

`fetch\_prep.receipts` r

JOIN

  `fetch\_prep.items` i ON r.receiptId = i.receiptId

JOIN

  `fetch\_prep.brands` b ON i.barcode = b.barcode

WHERE

  r.ScannedDateTime >= (SELECT start\_date FROM RecentMonthReceipts)

  AND r.ScannedDateTime < (SELECT end\_date FROM RecentMonthReceipts)

GROUP BY

  brandName

ORDER BY

  receiptsScanned DESC

LIMIT 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 RecentMonth AS (

  SELECT

    b.name AS brandName,

    COUNT(r.receiptId) AS receiptsScanned

  FROM

    `fetch\_prep.receipts` r

  JOIN

    `fetch\_prep.items` i ON r.receiptId = i.receiptId

  JOIN

    `fetch\_prep.brands` b ON i.barcode = b.barcode

  WHERE

    r.ScannedDateTime >= DATE\_TRUNC(CURRENT\_DATE(), MONTH)

    AND r.ScannedDateTime < DATE\_TRUNC(CURRENT\_DATE(), MONTH) + INTERVAL 1 MONTH

  GROUP BY

    brandName

),

PreviousMonth AS (

  SELECT

    b.name AS brandName,

    COUNT(r.receiptId) AS receiptsScanned

  FROM

    `fetch\_prep.receipts` r

  JOIN

    `fetch\_prep.items` i ON r.receiptId = i.receiptId

  JOIN

    `fetch\_prep.brands` b ON i.barcode = b.barcode

  WHERE

    r.ScannedDateTime >= DATE\_TRUNC(CURRENT\_DATE(), MONTH) - INTERVAL 1 MONTH

    AND r.ScannedDateTime < DATE\_TRUNC(CURRENT\_DATE(), MONTH)

  GROUP BY

    brandName

),

RankedCurrentMonth AS (

  SELECT

    brandName,

    receiptsScanned,

    RANK() OVER (ORDER BY receiptsScanned DESC) AS currentRank

  FROM

    RecentMonth

),

RankedPreviousMonth AS (

  SELECT

    brandName,

    receiptsScanned,

    RANK() OVER (ORDER BY receiptsScanned DESC) AS previousRank

  FROM

    PreviousMonth

)

SELECT

  COALESCE(c.brandName, p.brandName) AS brandName,

  COALESCE(c.receiptsScanned, 0) AS currentMonthReceipts,

  COALESCE(p.receiptsScanned, 0) AS previousMonthReceipts,

  c.currentRank,

  p.previousRank

FROM

  RankedCurrentMonth c

FULL OUTER JOIN

  RankedPreviousMonth p ON c.brandName = p.brandName

ORDER BY

  COALESCE(c.currentRank, p.previousRank) ASC;

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

WITH prep\_table AS (

  SELECT

    \_id\_\_oid AS receiptId,

    userId,

    bonusPointsEarned,

    bonusPointsEarnedReason,

    DATETIME(TIMESTAMP\_MILLIS(createDate\_\_date)) AS createdDateTime,

    DATETIME(TIMESTAMP\_MILLIS(dateScanned\_\_date)) AS ScannedDateTime,

    DATETIME(TIMESTAMP\_MILLIS(finishedDate\_\_date)) AS finishedDateTime,

    DATETIME(TIMESTAMP\_MILLIS(modifyDate\_\_date)) AS modifyDateTime,

    DATETIME(TIMESTAMP\_MILLIS(pointsAwardedDate\_\_date)) AS pointsAwardedDateTime,

    pointsEarned,

    DATETIME(TIMESTAMP\_MILLIS(purchaseDate\_\_date)) AS purchaseDateTime,

    purchasedItemCount,

    rewardsReceiptStatus,

    totalSpent

  FROM

    `ae-devraw.fetch.receipts`

)

-- Calculate average number of items purchased for "Accepted" and "Rejected" receipts

SELECT

  rewardsReceiptStatus,

  AVG(purchasedItemCount) AS avgItemsPurchased

FROM

  prep\_table

WHERE

  rewardsReceiptStatus IN ('ACCEPTED', 'REJECTED')

GROUP BY

  rewardsReceiptStatus;

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

WITH prep\_table AS (

  SELECT

    \_id\_\_oid AS receiptId,

    userId,

    bonusPointsEarned,

    bonusPointsEarnedReason,

    DATETIME(TIMESTAMP\_MILLIS(createDate\_\_date)) AS createdDateTime,

    DATETIME(TIMESTAMP\_MILLIS(dateScanned\_\_date)) AS ScannedDateTime,

    DATETIME(TIMESTAMP\_MILLIS(finishedDate\_\_date)) AS finishedDateTime,

    DATETIME(TIMESTAMP\_MILLIS(modifyDate\_\_date)) AS modifyDateTime,

    DATETIME(TIMESTAMP\_MILLIS(pointsAwardedDate\_\_date)) AS pointsAwardedDateTime,

    pointsEarned,

    DATETIME(TIMESTAMP\_MILLIS(purchaseDate\_\_date)) AS purchaseDateTime,

    purchasedItemCount,

    rewardsReceiptStatus,

    totalSpent

  FROM

    `ae-devraw.fetch.receipts`

)

--  average totalSpent for "Accepted" and "Rejected" receipts

SELECT

  rewardsReceiptStatus,

  AVG(totalSpent) AS avgTotalSpent

FROM

  prep\_table

WHERE

  rewardsReceiptStatus IN ('Accepted', 'Rejected')

GROUP BY

  rewardsReceiptStatus;

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

WITH RecentUsers AS (

  SELECT

    userId

  FROM

    `fetch\_prep.users`

  WHERE

    createdDateTime >= TIMESTAMP\_SUB(CURRENT\_TIMESTAMP(), INTERVAL 6 MONTH)

)

SELECT

  b.name AS brandName,

  SUM(r.totalSpent) AS totalSpend

FROM

  `fetch\_prep.receipts` r

JOIN

  `fetch\_prep.items` i ON r.receiptId = i.receiptId

JOIN

  `fetch\_prep.brands` b ON i.barcode = b.barcode

JOIN

  RecentUsers u ON r.userId = u.userId

WHERE

  r.purchaseDateTime >= DATE\_SUB(CURRENT\_DATE(), INTERVAL 6 MONTH)

GROUP BY

  brandName

ORDER BY

  totalSpend DESC

LIMIT 1;

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

WITH RecentUsers AS (

  SELECT

    userId

  FROM

    `fetch\_prep.users`

  WHERE

    createdDateTime >= TIMESTAMP\_SUB(CURRENT\_TIMESTAMP(), INTERVAL 183 DAY)  -- Approximately 6 months

)

SELECT

  b.name AS brandName,

  COUNT(r.receiptId) AS transactionCount

FROM

  `fetch\_prep.receipts` r

JOIN

  `fetch\_prep.items` i ON r.receiptId = i.receiptId

JOIN

  `fetch\_prep.brands` b ON i.barcode = b.barcode

JOIN

  RecentUsers u ON r.userId = u.userId

GROUP BY

  brandName

ORDER BY

  transactionCount DESC

LIMIT 1;