

Design Notes:

1. We have 3 original data tables i.e. **brand_table**, **receipt_table**, **user_table** as specified in Relational Diagram.
2. We also have an additional table generated from Receipts data called **ReceiptItem_Table** table which consists of **receipt_id**, **brand_id**, and all the normalized fields from **rewardsReceiptitemList** field. This can be used to join between receipt and brand table enabling us to maintain a relational structure in our data model.

3. Available Tables :

receipt_table
brand_table
user_table
ReceiptItem_Table

4. SQL Dialect: **MYSQL**

SQL 1: What are the top 5 brands by receipts scanned for the most recent month?

```
SELECT brandcode,
       Count(*) AS receipt_count
FROM   receiptitem_table src
       JOIN receipt_table tgt
         ON src.receipt_id = tgt.receipt_id
WHERE  tgt.createdate > Date_sub(Curdate(), INTERVAL 1 month)
GROUP BY brandcode
ORDER BY receipt_count DESC
LIMIT 5;
```

SQL 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 recentmonth
AS
(
    SELECT      brandcode,
                count(*)                               AS receipt_count,
                rank() over (ORDER BY count(*) DESC) AS ranking
    FROM        receiptitem_table src
    JOIN        receipt_table tgt
    ON          src.receipt_id = tgt.receipt_id
    WHERE       tgt.createdate > date_sub(curdate(), INTERVAL 1 month)
    GROUP BY    brandcode
    ORDER BY    receipt_count DESC
    LIMIT       5 ),

previousmonth
AS
(
    SELECT      brandcode,
                count(*)                               AS receipt_count,
                rank() over (ORDER BY count(*) DESC) AS ranking
    FROM        receiptitem_table src
    JOIN        receipt_table tgt
    ON          src.receipt_id = tgt.receipt_id
    WHERE       tgt.createdate >= date_sub(curdate(), INTERVAL 2 month)
    AND         tgt.createdate < date_sub(curdate(), INTERVAL 1 month)
    GROUP BY    brandcode
    ORDER BY    receipt_count DESC) --No limit here to get correct join

with previous_month
SELECT      recentmonth.brandcode
AS brand,
           recentmonth.ranking
AS recent_ranking,
           previousmonth.ranking
AS previous_ranking,

coalesce(recentmonth.ranking,0)-coalesce(previousmonth.ranking,0) AS
ranking_change
FROM        recentmonth
LEFT JOIN    previousmonth
ON          recentmonth.brandcode = previousmonth.brandcode
ORDER BY    recent_ranking;
```

SQL 3: When considering average spend from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?

```
WITH cte
  AS (SELECT rewardsreceiptstatus,
            Avg(totalspent) AS avg_spend
      FROM receipt_table
      WHERE rewardsreceiptstatus IN ( 'Accepted', 'Rejected'
    )
      GROUP BY rewardsreceiptstatus)
SELECT CASE
      WHEN avg_spend > (SELECT avg_spend
                        FROM cte
                        WHERE rewardsreceiptstatus =
'Rejected') THEN
        'Accepted'
      ELSE 'Rejected'
      END AS higher_spend
FROM cte
WHERE rewardsreceiptstatus = 'Accepted';
```

SQL 4: When considering total number of items purchased from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?

```
WITH cte AS (
  SELECT
    rewardsReceiptStatus,
    SUM(receipt_table.purchasedItemCount) AS total_items
  FROM receipt_table
  JOIN receiptitem_table ON receipt_table.receipt_id =
receiptitem_table.receipt_id
  WHERE rewardsReceiptStatus IN ('Accepted', 'Rejected')
  GROUP BY rewardsReceiptStatus
)
SELECT
  CASE WHEN total_items > (SELECT total_items FROM cte WHERE
rewardsReceiptStatus = 'Rejected')
    THEN 'Accepted'
    ELSE 'Rejected' END AS higher_count
FROM cte
WHERE rewardsReceiptStatus = 'Accepted';
```

SQL 5: Which brand has the most spend among users who were created within the past 6 months?

```
SELECT brandcode,
       Sum(finalprice * quantitypurchased) AS totalSpent
FROM   receiptitem_table src
       JOIN receipt_table tgt
       ON src.receipt_id = tgt.receipt_id
-- joining to get user_id field from receipts table
WHERE  tgt.userid IN (SELECT _id
                      FROM   user_table
                      WHERE    From_unixtime(createdate / 1000) >=
                             Date_sub(Curdate(), INTERVAL 6 month
                             ))

GROUP BY brandcode
ORDER BY totalspent DESC
LIMIT 1
```

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

```
SELECT brandcode,
       Count(*) AS transaction_count
FROM   receiptitem_table src
       JOIN receipt_table tgt
       ON src.receipt_id = tgt.receipt_id
WHERE  tgt.userid IN (SELECT _id
                      FROM   user_table
                      WHERE    From_unixtime(createdate / 1000) >
                             Date_sub(Curdate(), INTERVAL 6
month)
                      )

GROUP BY brandcode
ORDER BY totalspent DESC
LIMIT 1;
```