Second: Write a query that directly answers a predetermined question from a business stakeholder

Write a SQL query against your new structured relational data model that answers one of the following bullet points below of your choosing.

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

```
WITH brand_mostrecent AS (
   SELECT b.name, COUNT(*) AS brand_count,
      ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS rnk
   FROM Receipts r
   JOIN ReceiptsItem i ON r.Receipts_id = i.Receipts_id
   JOIN Brand b ON i.brand_id = b.brand_id AND i.barcode = b.barcode
   WHERE DATEDIFF(CURRENT_DATE(), r.dateScanned) < 180
   GROUP BY b.name
)
SELECT name
FROM brand_mostrecent
WHERE rnk <= 5
ORDER BY rnk;
```

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 brand_mostrecent AS (
   SELECT b.name, COUNT(*) AS brand_count,
      ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS rnk
   FROM Receipts r
   JOIN ReceiptsItem i ON r.Receipts_id = i.Receipts_id
   JOIN Brand b ON i.brand_id = b.brand_id AND i.barcode = b.barcode
   WHERE DATEDIFF(CURRENT_DATE(), r.dateScanned) < 180
   GROUP BY b.name
)
SELECT name
FROM brand_previousmonth
WHERE rnk <= 5
ORDER BY rnk;
```

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

```
#Calculating the average spend from receipts with 'rewardsReceiptStatus' of 'Accepted'
SELECT AVG(totalspent) AS avg_spent_accepted
FROM Receipts
WHERE rewardsReceiptStatus = 'Accepted';

#Calculating the average spend from receipts with 'rewardsReceiptStatus' of 'Rejected'
SELECT AVG(totalspent) AS avg_spent_rejected
FROM Receipts
WHERE rewardsReceiptStatus = 'Rejected';
```

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

```
#Calculate the total number of items purchased from receipts with 'rewardsReceiptStatus' of 'Accepted'
SELECT SUM(purchaseItemCount) AS total_item_count_accepted
FROM Receipts
WHERE rewardsReceiptStatus = 'Accepted';

#Calculate the total number of items purchased from receipts with 'rewardsReceiptStatus' of 'Rejected'
SELECT SUM(purchaseItemCount) AS total_item_count_rejected
FROM Receipts
WHERE rewardsReceiptStatus = 'Rejected';

#Compare the total_item_count_accepted and total_item_count_rejected to see which one is greater
SELECT
   CASE
      WHEN total_item_count_accepted > total_item_count_rejected THEN 'Accepted'
      WHEN total_item_count_accepted < total_item_count_rejected THEN 'Rejected'
      ELSE 'Equal'
   END AS greater_count
FROM (
   SELECT
      (SELECT SUM(purchaseItemCount) FROM Receipts WHERE rewardsReceiptStatus = 'Accepted') AS total_item_count_accepted,
      (SELECT SUM(purchaseItemCount) FROM Receipts WHERE rewardsReceiptStatus = 'Rejected') AS total_item_count_rejected
) AS counts;
```

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

```
WITH full_table AS (
   SELECT b.name, r.total_spent, u.createDate
   FROM Brand b
   JOIN ReceiptItem i ON i.brand_id = b.brand_id AND i.barcode = b.barcode
   JOIN Receipts r USING(receipts_id)
   JOIN Users u USING(user_id)
)
SELECT name, SUM(total_spent) AS total_spent_by_brand
FROM full_table
WHERE createDate >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)
GROUP BY name
ORDER BY total_spent_by_brand DESC
LIMIT 1;
```

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

```
SELECT b.name AS brand_name, count(*) AS num_trans
```

```
FROM receipts r
JOIN receipts_item ri ON r._id = ri._id
LEFT JOIN brands b ON ri.barcode = b.barcode
JOIN users u on r.userId = u.id
WHERE
FROM_UNIXTIME(u.createdDate/1000) > DATE_SUB((SELECT
MAX(FROM_UNIXTIME(u.createdDate/1000)) FROM users u), INTERVAL 6 MONTH)
GROUP BY
b.name
ORDER BY
SUM(totalSpent) DESC
LIMIT 5;
```