FETCH CODING CHALLENGE

MYSQL

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

```
Select b.name as brand_name, count(r._id) as receipt_id
From Brands b join Receipt_Items r on b.barcode= r.barcode join Payments p on
p.bardcode= r.barcode join Receipts r on p.receipt_id= r._id
where datadiff( ‘month’, now(), r.createDate) <=1
Group by  b.name
Order by receipt_id desc
Limit 5
```

Query to select top 5 brands from brands table based on the count receipt id’s from receipt table . Group by brand name and order by receipt ID’s and limit 5 to get the TOP 5 brands.

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

```
With cte as (Select AVG(totalSpent) as avg_spend_acc
From Receipts
Where rewardsReceiptStatus = “Accepted”) ,

Cte 2 as (Select AVG(totalSpent) as avg_spend_rej
From Receipts
Where rewardsReceiptStatus = “Rejected”)

Select

Case when avg_spend_acc > avg_spend_rej then ‘accepted is greater’
```

```
When avg_spend_acc < avg_spend_rej then ‘rejected is greater’

End as greater_receipt_status

From cte,cte2
```

The CTE1 provides avg of total spent amount when status is accepted and CTE2 provides provides avg of total spent amount when status is rejected. The query compares both the avg spending amount and provides a result using the case statement.

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

```
With cte as (Select SUM(purchasedItemCount)  as no_of_items_acc
From Receipts
Where rewardsReceiptStatus = “Accepted”) ,

Cte 2 as (Select SUM(purchasedItemCount)  as no_of_items_rej
From Receipts
Where rewardsReceiptStatus = “Rejected”)

Select

Case when no_of_items_acc > no_of_items_rej then ‘accepted is greater’

When no_of_items_acc < no_of_items_rej then ‘rejected is greater’

End as greater_receipt_status

From cte,cte2
```

The CTE1 provides total no of items when status is accepted and CTE2 provides provides provides total no of items when status is rejected. The query compares both the provides total no of items and provides a result using the case statement.

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

```
Select b.name as brand_name

From  Brands b join Receipt_Items r on b.barcode= r.barcode join Payments p on
p.bardcode= r.barcode join users u on u._id_id= p.user_id join Receipts r on
r._userId= u._id
where r.totalSpent in (select max(total_spent) from receipts r join users u on

r._userId= u._id where datediff(‘month’, now(), u.createDate) <=6)
```

The subquery provides the max spend among users in last six months. The outer query provides the brand name associated with the max total spent.

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

```
Select b.name as b.brand_name, count(p.receipt_id)  as transactions

From  Brands b join Receipt_Items r on b.barcode= r.barcode join Payments p on
p.bardcode= r.barcode join users u on u._id_id= p.user_id

where datediff(‘month’, now(), u.createDate) <=6
order by transactions desc
limit 1
```

The query provides brand name and count of transactions for each brand name. The where clause filters for recnt 6 months . The transactions are arranged in descending order for each brand. The brand with most transactions is selected using limit 1.