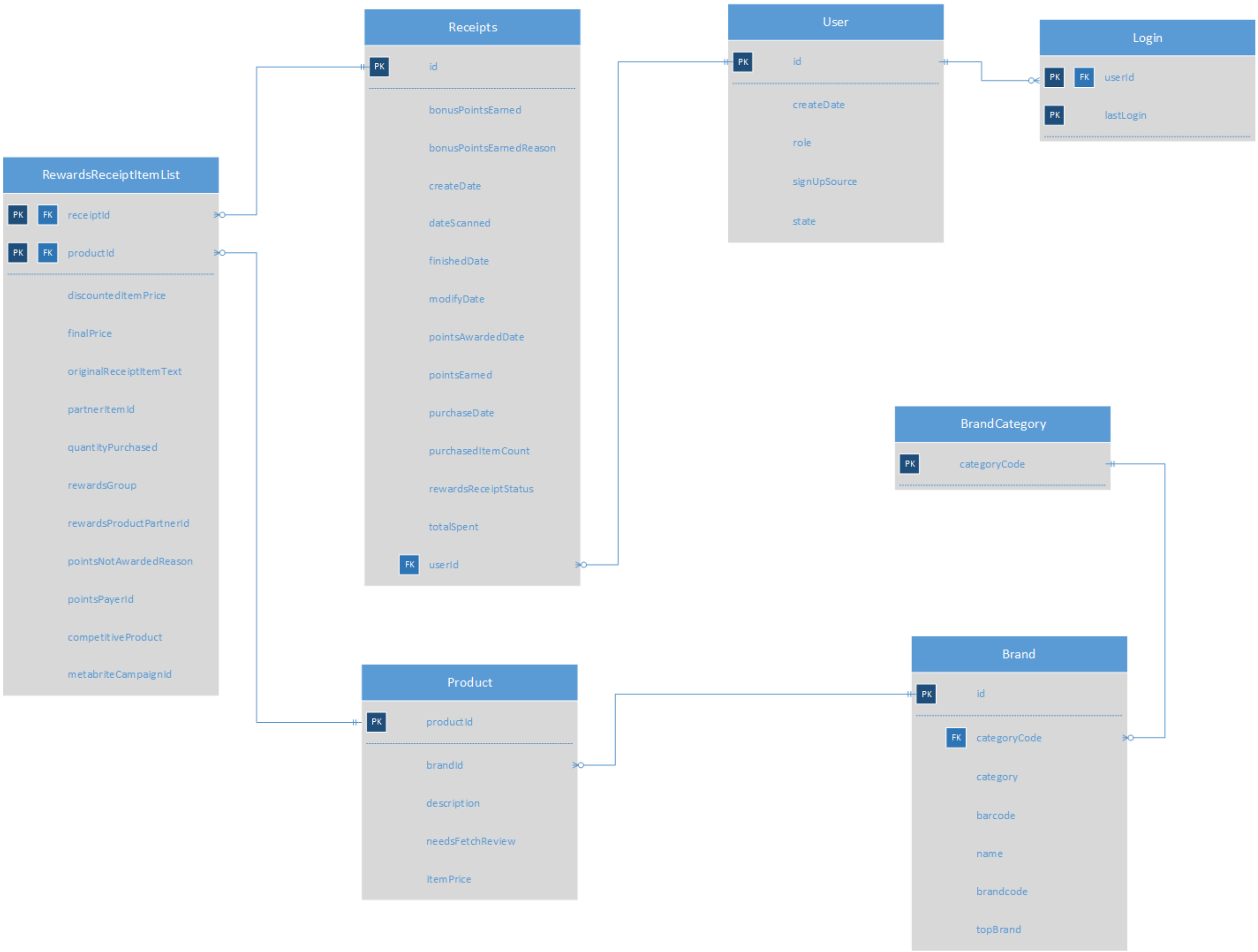
Assignment

1: Review Existing Unstructured Data and Diagram a New Structured Relational Data Model



2. Write a query that directly answers a predetermined question from a business stakeholder.

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

```
SELECT t1.accaveragespend, t2.rejaveragespend
FROM
(
SELECT AVG(totalSpent) AS accaveragespend
FROM Receipts
WHERE rewardsReceiptStatus = 'Accepted'
) t1,
(
SELECT AVG(totalSpent) AS rejaveragespend
FROM Receipts
WHERE rewardsReceiptStatus = 'Rejected'
) t2;
```

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

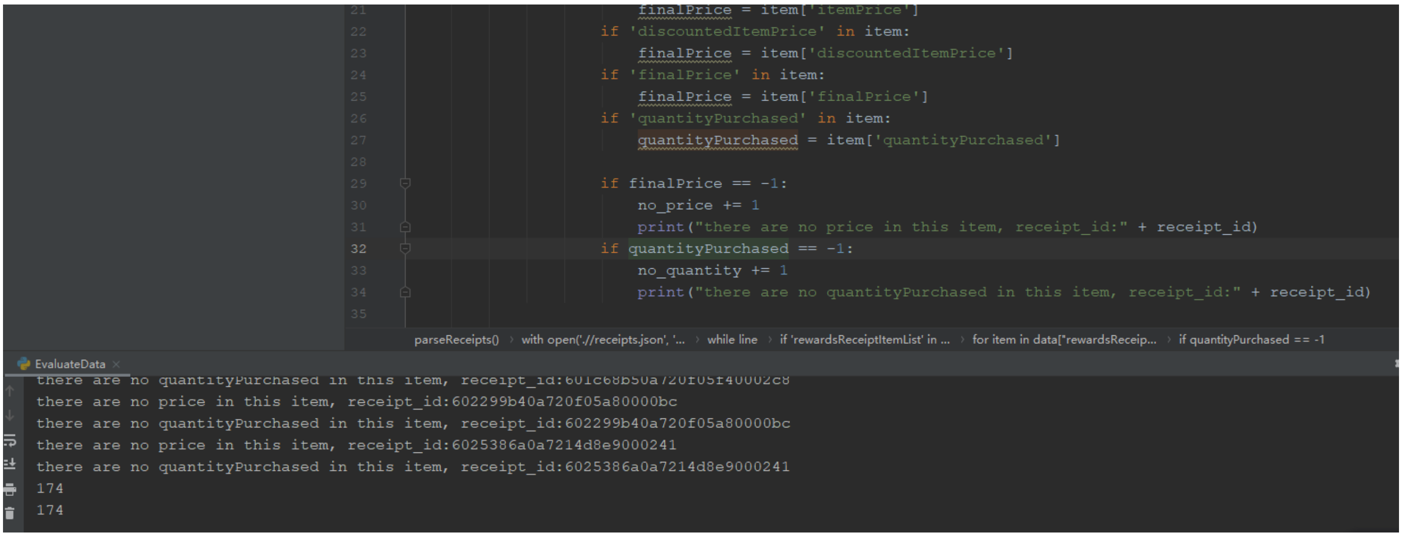
```
SELECT t1.acctotalnum, t2.rejtotalnum
FROM
(
SELECT SUM(l.quantityPurchased) AS acctotalnum
FROM Receipts r, RewardsReceiptItemList l
WHERE r.id=l.receiptId AND rewardsReceiptStatus = 'Accepted'
) t1,
(
SELECT SUM(l.quantityPurchased) AS rejtotalnum
FROM Receipts r, RewardsReceiptItemList l
WHERE r.id=l.receiptId AND rewardsReceiptStatus = 'Rejected'
) t2;
```

3. Evaluate Data Quality Issues in the Data Provided.

After I analyzed the receipts.json file by using Python like the screenshot below, I found the following data quality issues:

1, There are no “itemPrice”, “discountedItemPrice” or “finalPrice” in the item list “rewardsReceiptItemList” purchased on the receipt, which means that the purchase price of the item did not be recorded.

2. There is no “quantityPurchased” item in the item list “rewardsReceiptItemList” purchased on the receipt, which means that the purchase quantity of the item did not be recorded also.



```
        finalPrice = item['itemPrice']
    if 'discountedItemPrice' in item:
        finalPrice = item['discountedItemPrice']
    if 'finalPrice' in item:
        finalPrice = item['finalPrice']
    if 'quantityPurchased' in item:
        quantityPurchased = item['quantityPurchased']

    if finalPrice == -1:
        no_price += 1
        print("there are no price in this item, receipt_id:" + receipt_id)
    if quantityPurchased == -1:
        no_quantity += 1
        print("there are no quantityPurchased in this item, receipt_id:" + receipt_id)
```

parseReceipts() › with open('.//receipts.json', '... › while line › if 'rewardsReceiptItemList' in ... › for item in data["rewardsReceip... › if quantityPurchased == -1
EvaluateData

```
there are no quantityPurchased in this item, receipt_id:601c68b50a720f05f40002c8
there are no price in this item, receipt_id:602299b40a720f05a80000bc
there are no quantityPurchased in this item, receipt_id:602299b40a720f05a80000bc
there are no price in this item, receipt_id:6025386a0a7214d8e9000241
there are no quantityPurchased in this item, receipt_id:6025386a0a7214d8e9000241
174
174
```

4. Communicate with Stakeholders

Dear business leader:

Hope everything is going well with you. Thank you for providing me with these data for analysis. However, during my analysis, I found some following issues:

1. It seems that the “rewardsReceiptItemList” in the Receipts Data Schema does not have all the products in the receipt.
2. There are some products without price and quantityPurchased in the Receipts Data schema.

I discovered the issues above after I analyzed the data in receipts.json file using Python, In order to resolve the data quality issue, I would love to know more about how you store and export data on a daily basis in operating system, I believe that it will help me analyze the abnormal data.

In order to better optimize data assets, it would be great if you can provide all the product information and brand types in the system instead of only providing receipt-related products. With the growth of data in the system, we may encounter the problem like the data operation is running slower due to excessive data storage in production. Moreover, when the number of users is getting larger, it would take longer time when we try to access it. As for solutions for these possible problems, our system can optimize the system into a microservice and distributed database architecture to facilitate horizontal expansion of the storage and access of the system.

Best Wishes!

Allen