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

Given 3 Json files Users, Brands, and Receipts.

First, I read these 3 files using python and to be specific using pandas dataframe.

Then I found the following, for users data frame.

	_id	active	createdDate	lastLogin	role	signUpSource	state
476	{'$oid': '54943462e4b07e684...	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	NaN	NaN
177	{'$oid': '600056a3f7e5b011f...	True	{'$date': 1610634915207}	{'$date': 1610635030767}	consumer	Email	WI
86	{'$oid': '5ff7930fb3348b11c...	True	{'$date': 1610060559759}	{'$date': 1610060994656}	consumer	Email	WI
95	{'$oid': '5ff73b90eb7c7d31c...	True	{'$date': 1610038160959}	{'$date': 1610038267267}	consumer	Email	WI
214	{'$oid': '600741d06e6469120...	True	{'$date': 1611088337000}	{'$date': 1611088743299}	consumer	Email	WI
180	{'$oid': '6002475cfb296c121...	True	{'$date': 1610762076571}	NaN	consumer	Email	WI
4	{'$oid': '5ff1e194b6a9d73a3...	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Email	WI
327	{'$oid': '600f00d05edb787dc...	True	{'$date': 1611595984549}	NaN	consumer	Email	WI
236	{'$oid': '60088d84633aab121...	True	{'$date': 1611173252034}	{'$date': 1611173252079}	consumer	Email	WI
377	{'$oid': '601ac1da591789121...	True	{'$date': 1612366298705}	{'$date': 1612366298909}	consumer	Email	WI

Figure 1 users dataframe sample before cleaning

For "_id" column I found that every row contains an object called '$id' that holds the data of the id same as last Login column.

For brands dataframe

	_id	barcode	category	categoryCode	cpg	name	topBrand	bra
112	{'$oid': '5fa98944be37ce239...	511111217251	Baking	BAKING	{'$ref': 'Cogs', '$id': {'$...	test brand @1604946244133	0.0	
969	{'$oid': '5f403232be37ce5f7...	511111615705	Baking	BAKING	{'$ref': 'Cogs', '$id': {'$...	test brand @1598042674677	NaN	
1104	{'$oid': '57c08257e4b0718ff...	511111102496	NaN	NaN	{'$ref': 'Cpgs', '$id': {'$...	Hoegaarden	NaN	
852	{'$oid': '5332f5ebe4b03c9a2...	511111304050	NaN	NaN	{'$ref': 'Cpgs', '$id': {'$...	Monster	NaN	
810	{'$oid': '5e710da7ee7f2d0b3...	511111614074	Dairy & Refrigerated	DAIRY_AND_REFRIGERATED	{'$ref': 'Cogs', '$id': {'$...	COUNTRY CROCK ORIGINAL	NaN	
576	{'$oid': '5332f734e4b03c9a2...	511111003809	NaN	NaN	{'$ref': 'Cpgs', '$id': {'$...	Johnsonville	NaN	
1078	{'$oid': '5fb807bebe37ce522...	511111617570	Beer Wine Spirits	BEER_WINE_SPIRITS	{'$id': {'$oid': '5fb6d381b...	The Glenlivet® 12 Year	0.0	
524	{'$oid': '58861c7e4e8d0d20b...	511111001324	Snacks	NaN	{'$ref': 'Cogs', '$id': {'$...	Maui Style Chips	NaN	
1008	{'$oid': '5fa98944be37ce239...	511111117285	Baking	BAKING	{'$ref': 'Cogs', '$id': {'$...	test brand @1604946244833	0.0	TE
516	{'$oid': '5332f756e4b03c9a2...	511111503743	NaN	NaN	{'$ref': 'Cpgs', '$id': {'$...	Murray	NaN	

Figure 2 Brands dataframe sample before cleaning

The same here for "_id" column, but the "cpg" column had 2 objects ref object and id object so I dealt with it as a multivalued attribute and separated them into 2 columns.

For Receipts dataframe

modifyDate	pointsAwardedDate	pointsEarned	purchaseDate	purchasedItemCount	rewardsReceiptItemList	rewardsReceiptStatus
{'$date': 1612122159098}	NaN	NaN	NaN	NaN	NaN	SUBMITTED
{'$date': 1614368255062}	NaN	NaN	NaN	NaN	NaN	SUBMITTED
{'$date': 1610138583000}	{'$date': 1610138583000}	600.0	{'$date': 1609459200000}	1.0	[{'barcode': '089203700016'...	FINISHED
{'$date': 1614607657664}	NaN	NaN	NaN	NaN	NaN	SUBMITTED
{'$date': 1609687508000}	NaN	NaN	{'$date': 1509321600000}	3.0	[{'deleted': True, 'descrip...	REJECTED
{'$date': 1610566786000}	{'$date': 1610566786000}	250.0	{'$date': 1610480385000}	1.0	[{'barcode': '025800026302'...	FINISHED
{'$date': 1611762666000}	{'$date': 1611762660000}	760.0	{'$date': 1611676258000}	1.0	[{'barcode': '079400066619'...	FINISHED
{'$date': 1614382050000}	NaN	25.0	{'$date': 1597622400000}	2.0	[{'barcode': 'B076FJ92M4', ...	REJECTED
{'$date': 1612089160176}	NaN	NaN	NaN	NaN	NaN	SUBMITTED
{'$date': 1611869065000}	NaN	NaN	NaN	0.0	[{'needsFetchReview': True,...	FLAGGED

Figure 3 Receipts dataframe sample before cleaning

I found here RewardsReceiptItemList as list of objects and it's not good to deal with, so I created a new table for this list and link it with the receipts and brands tables.

Next, I created ER diagram using MYSQL workbench, but first I created 4 tables with a new design and linked them using primary and foreign keys.

NOTE: I have changed _id for all three tables (user_id, brands_id, receipts_id) to avoid confusion.

Users table

```
create table users(
user_id varchar(50) primary key not null,
active boolean,
createdDate date,
lastLogin date,
role varchar(50),
signUpSource varchar(50),
state varchar(50)
);
```

Brands table

```
create table brands(
brands_id varchar(50) primary key not null,
barcode bigint,
category varchar(50),
categoryCode varchar(50),
cpg_Id varchar(50),
cpg_Ref varchar(50),
name text,
topBrand tinyint,
brandCode text
);
```

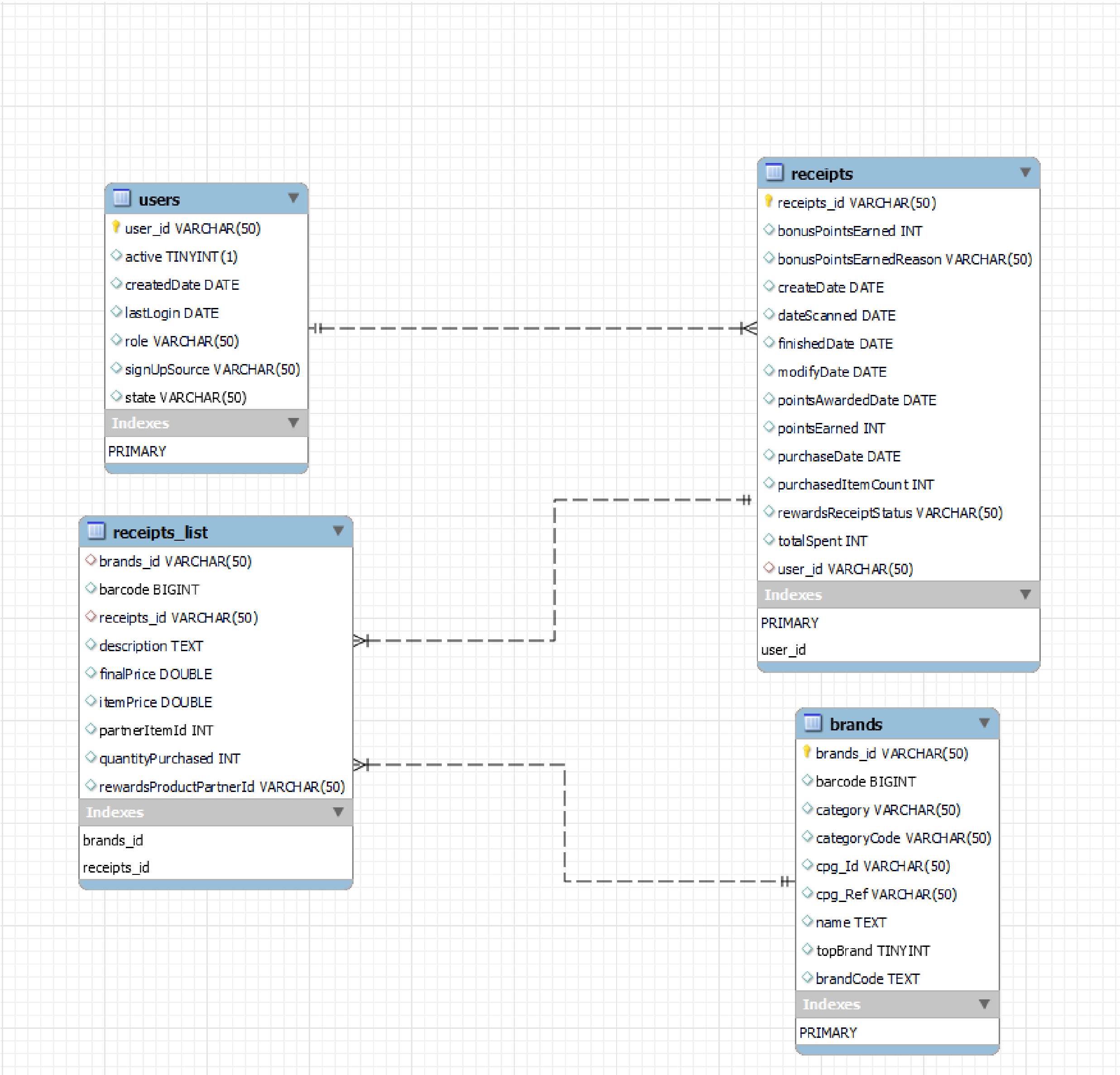
Receipt Table

```
create table receipts(
receipts_id varchar(50) primary key not null,
bonusPointsEarned int,
bonusPointsEarnedReason varchar(50),
createDate date,
dateScanned date,
finishedDate date,
modifyDate date,
pointsAwardedDate date,
pointsEarned double,
purchaseDate date,
purchasedItemCount int,
rewardsReceiptStatus varchar(50),
totalSpent double,
user_id varchar(50),
FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

Receipts List Table

```
create table receipts_list(
brands_id varchar(50),
barcode bigint,
receipts_id varchar(50),
description text,
finalPrice double,
itemPrice double,
partnerItemId int,
quantityPurchased int,
rewardsProductPartnerId varchar(50),
FOREIGN KEY (brands_id) REFERENCES brands(brands_id),
FOREIGN KEY (receipts_id) REFERENCES receipts(receipts_id)
);
```

ERD design



In this schema, tables are linked using foreign key references, forming relationships between entities. For example, the receipts_list table references both the brands table and the receipts table through foreign keys.

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

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

-- How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?

-- Query for the Recent Month:

```
SELECT b.name AS BrandName, COUNT(r.receipts_id) AS ReceiptsScanned
FROM brands b
JOIN receipts_list rl ON b.brands_id = rl.brands_id
JOIN receipts r ON rl.receipts_id = r.receipts_id
WHERE YEAR(r.dateScanned) = YEAR(CURRENT_DATE)
AND MONTH(r.dateScanned) = MONTH(CURRENT_DATE)
GROUP BY b.name
ORDER BY ReceiptsScanned DESC
LIMIT 5;
```

-- Query for the Previous Month:

```
SELECT b.name AS BrandName, COUNT(r.receipts_id) AS ReceiptsScanned
FROM brands b
JOIN receipts_list rl ON b.brands_id = rl.brands_id
JOIN receipts r ON rl.receipts_id = r.receipts_id
WHERE YEAR(r.dateScanned) = YEAR(CURRENT_DATE - INTERVAL 1 MONTH)
AND MONTH(r.dateScanned) = MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
GROUP BY b.name
ORDER BY ReceiptsScanned DESC
LIMIT 5;
```

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

```
SELECT rewardsReceiptStatus, AVG(totalSpent) AS AvgSpend
FROM receipts
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?

```
SELECT rewardsReceiptStatus, SUM(purchasedItemCount) AS TotalItemsPurchased
FROM receipts
WHERE rewardsReceiptStatus IN ('Accepted', 'Rejected')
```

```
GROUP BY rewardsReceiptStatus;
```

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

```
SELECT b.name AS BrandName, SUM(rl.finalPrice) AS TotalSpend
FROM brands b
JOIN receipts_list rl ON b.brands_id = rl.brands_id
JOIN receipts r ON rl.receipts_id = r.receipts_id
JOIN users u ON r.user_id = u.user_id
WHERE u.createdDate >= DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH)
GROUP BY b.name
ORDER BY TotalSpend DESC
LIMIT 1;
```

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

```
SELECT b.name AS BrandName, COUNT(DISTINCT r.receipts_id) AS TransactionCount
FROM brands b
JOIN receipts_list rl ON b.brands_id = rl.brands_id
JOIN receipts r ON rl.receipts_id = r.receipts_id
JOIN users u ON r.user_id = u.user_id
WHERE u.createdDate >= DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH)
GROUP BY b.name
ORDER BY TransactionCount DESC
LIMIT 1;
```

3. Evaluate Data Quality Issues in the Data Provided

Users Data (users_df):

- Duplicates: 283 duplicated rows were found and removed.
- pull out the data from the objects $id and $Date

```
# pull out the data from the objects $id and $Date
users_df['_id'] = users_df['_id'].apply(lambda entry: entry['$oid'])
users_df['createdDate'] = users_df['createdDate'].apply(lambda entry: entry['$date'])
users_df['lastLogin'] = users_df['lastLogin'].apply(lambda entry: entry['$date'] if pd.notna(entry) else np.nan)
users_df=users_df.rename(columns={'_id':'user_id'})
```

-
- Null Values: Null values were present in the 'lastLogin', 'signUpSource', and 'state' columns.

```
# check for null values
users_df.isna().sum()
```

	data
user_id	0
active	0
createdDate	0
lastLogin	40
role	0
signUpSource	5
state	6

Length: 7, dtype: int64 Open in new tab

- Data Types: The 'createdDate' and 'lastLogin' columns were adjusted to datetime data types.
- Data Cleaning: Null values in 'lastLogin' were replaced with the mean, and null values in 'signUpSource' and 'state' were replaced with the mode.

```
# Replace null values with mean for numeric columns
users_df['lastLogin']=users_df['lastLogin'].fillna(users_df['lastLogin'].mean())
# Replace null values with mode for categorical columns
users_df['signUpSource']=users_df['signUpSource'].fillna(users_df['signUpSource'].mode().iloc[0])
users_df['state']=users_df['state'].fillna(users_df['state'].mode().iloc[0])
```

```
# Adjust Date datatype
# Convert milliseconds timestamp to datetime format
users_df['createdDate'] = pd.to_datetime(users_df['createdDate'], unit='ms')
users_df['lastLogin'] = pd.to_datetime(users_df['lastLogin'], unit='ms')
```

Sample of the cleaned users data

	user_id	active	createdDate	lastLogin	role	signUpSource	state
328	60132b85a4b74c3cbc516295	True	2021-01-28 21:24:21.761	2021-01-28 21:24:21.912999936	consumer	Email	WI
129	5ffcb4bc04929111f6e92608	True	2021-01-11 20:27:40.225	2021-01-11 20:27:40.264999936	consumer	Email	WI
122	5ffc8f9704929111f6e922bf	True	2021-01-11 17:49:11.890	2021-01-11 17:50:56.750000128	consumer	Email	WI
435	5fc961c3b8cfca11a077dd33	True	2020-12-03 22:08:03.936	2021-02-26 22:39:16.799000064	fetch-staff	Email	NH
91	5ff726a0eb7c7d12096da2db	True	2021-01-07 15:20:00.299	2021-01-07 15:20:00.352999936	consumer	Email	WI
115	5ff8d634b3348b11c9337aa4	True	2021-01-08 22:01:24.238	2021-01-08 22:01:24.518000128	consumer	Email	WI
272	600f47f06fd0dc1768a34a12	True	2021-01-25 22:36:32.551	2021-01-25 22:40:13.980999936	consumer	Email	WI
337	60145a5584231211ce796cac	True	2021-01-29 18:56:21.484	2021-01-29 18:56:21.542000128	consumer	Email	WI

Brands Data (brands_df):

- Duplicates: No duplicated rows were found.

```python
# check for duplicated values
brands_df.duplicated().sum()
```

```
0
```

- Null Values: Null values were present in the 'category', 'categoryCode', 'topBrand', and 'brandCode' columns.

```python
# check for null values
brands_df.isna().sum()
```

	data
brand_id	0
barcode	0
category	155
categoryCode	650
name	0
topBrand	612
brandCode	234
cpg_id	0

- Data Types: The 'topBrand' column was converted to a boolean data type.
- Data Cleaning: Null values in 'topBrand', 'category', 'categoryCode', and 'brandCode' were replaced with the mode.

```
brands_df['topBrand']=brands_df['topBrand'].fillna(brands_df['topBrand'].mode().iloc[0])
brands_df['category']=brands_df['category'].fillna(brands_df['category'].mode().iloc[0])
brands_df['categoryCode']=brands_df['categoryCode'].fillna(brands_df['categoryCode'].mode().iloc[0])
brands_df['brandCode']=brands_df['brandCode'].fillna(brands_df['brandCode'].mode().iloc[0])
```

```
# convert top brand as type bool
brands_df['topBrand']=brands_df['topBrand'].astype(bool)
```

```
#checking for data types
brands_df.dtypes
```

	data
brand_id	object
barcode	int64
category	object
categoryCode	object
name	object
topBrand	bool
brandCode	object
cpg_id	object

Sample of the cleaned Brands data

barcode	category	categoryCode	name	topBrand	brandCode	cpg_id	cpg_ref
511111216421	Baking	BAKING	test brand @1598881723241	False		5f4cffbaa475f1050a66b573	Cogs
511111102052	Condiments & Sauces	BAKING	Kraft Mayo	False		559c2234e4b06aca36af13c6	Cogs
511111317203	Baking	BAKING	test brand @1604437351617	False	TEST BRANDCODE @1604437351617	5fa1c567be37ce402c4618ef	Cogs
511111815457	Baking	BAKING	test brand @1597527951461	False	TEST BRANDCODE @1597527951461	5f38578fbe37ce5178517ad3	Cogs
511111818694	Baking	BAKING	test brand @1610039590443	False	TEST BRANDCODE @1610039590443	5ff74126be37ce1e961f326e	Cogs
511111600916	Dairy	BAKING	BRUMMEL AND BROWN	False	BRUMMEL AND BROWN	53e10d6368abd3c7065097cc	Cogs
511111401087	Condiments & Sauces	BAKING	HP Sauce	False	HP	559c2234e4b06aca36af13c6	Cogs
511111502210	Frozen	BAKING	Boca	False	BOCA	559c2234e4b06aca36af13c6	Cogs
511111219224	Baking	BAKING	test brand @1610493497005	False	TEST BRANDCODE @1610493497005	5ffe2e38be37ce5e01754c21	Cogs
511111500179	Beer Wine Spirits	BAKING	Terrapin	False	TERRAPIN	5332f709e4b03c9a25efd0f1	Cogs

Receipts Data (receipts_df):

- Duplicates: No duplicated rows were found.

```python
# check for duplicated values
receipts_df.duplicated().sum()
```

```
0
```

- Null Values: Null values were present in several columns, including 'bonusPointsEarned', 'bonusPointsEarnedReason', 'finishedDate', 'pointsAwardedDate', 'pointsEarned', 'purchaseDate', 'purchasedItemCount', and 'totalSpent'.

```python
# check for null values
receipts_df.isna().sum()
```

	data
_id	0
bonusPointsEarned	575
bonusPointsEarnedReason	575
createDate	0
dateScanned	0
finishedDate	551
modifyDate	0
pointsAwardedDate	582
pointsEarned	510
purchaseDate	448
purchasedItemCount	484
rewardsReceiptStatus	0
totalSpent	435
userId	0

- Data Types: Several date-related columns were adjusted to datetime data types.
- Data Cleaning: Null values were replaced with the mean for numeric columns and mode for categorical columns.

```python
# Fill null values
receipts_df['bonusPointsEarnedReason']=receipts_df['bonusPointsEarnedReason'].fillna(receipts_df['bonusPointsEarnedReason'].mode().iloc[0])
receipts_df['bonusPointsEarned']=receipts_df['bonusPointsEarned'].fillna(receipts_df['bonusPointsEarned'].mean())
receipts_df['finishedDate']=receipts_df['finishedDate'].fillna(receipts_df['finishedDate'].mean())
receipts_df['pointsAwardedDate']=receipts_df['pointsAwardedDate'].fillna(receipts_df['pointsAwardedDate'].mean())
receipts_df['pointsEarned']=receipts_df['pointsEarned'].fillna(receipts_df['pointsEarned'].mean())
receipts_df['totalSpent']=receipts_df['totalSpent'].fillna(receipts_df['totalSpent'].mean())
receipts_df['purchasedItemCount']=receipts_df['purchasedItemCount'].fillna(receipts_df['purchasedItemCount'].mean())
receipts_df['purchaseDate']=receipts_df['purchaseDate'].fillna(receipts_df['purchaseDate'].mean())
```

```python
# Adjust Date datatype
# Convert milliseconds timestamp to datetime format
receipts_df['createDate'] = pd.to_datetime(receipts_df['createDate'], unit='ms')
receipts_df['dateScanned'] = pd.to_datetime(receipts_df['dateScanned'], unit='ms')
receipts_df['finishedDate'] = pd.to_datetime(receipts_df['finishedDate'], unit='ms')
receipts_df['modifyDate'] = pd.to_datetime(receipts_df['modifyDate'], unit='ms')
receipts_df['pointsAwardedDate'] = pd.to_datetime(receipts_df['pointsAwardedDate'], unit='ms')
receipts_df['purchaseDate'] = pd.to_datetime(receipts_df['purchaseDate'], unit='ms')
```

4. COMMUNICATE WITH STAKE HOLDERS

Subject: Update on Data Quality and Optimization

Hi [STAKEHOLDER],

I hope you're doing well. I want to give you a quick rundown of the progress we've achieved in advancing our data quality and optimization activities. This message is intended to keep you informed, create a thorough awareness of our work, and identify any potential issues we may be facing.

Questions About Data Quality:

While reviewing the data, I encountered some critical questions:

- Is all the data complete, or are there any gaps?
- Can we confidently rely on the accuracy of the data?
- Are we aware of the data sources and any associated concerns?
- Does the data still align with our objectives, or have there been changes?

Data Quality Issues Detected:

During the analysis, I observed certain issues that are as follows:

- There were Inconsistencies in numerical values.
- Many redundant records were present that could compromise our insights if not resolved.
- Presence of missing values was high. Though we have replaced it with the mean, it could impact the accuracy of the findings.

Proposed Solutions for Data Quality Issues:

To address these concerns, I recommend the following actions:

- Conduct a thorough review of data sources to identify and rectify inconsistencies or inaccuracies.
- Evaluate unusual numerical values and determine if the adjustments are necessary.
- Perform data cleaning to remove the duplicate entries and ensure data integrity.
- Document the data sources and any modifications made.

Enhancing Data for Optimal Use:

To further enhance the usability of our data, I'll need the following information:

- Clear understanding of our specific objectives to work on the data accordingly.

- Insights into how the data will be used to optimize its structure and content.

Anticipating Performance and Scaling Challenges:

We anticipate the following challenges:

- Increasing data volume could impact on our current infrastructure.
- quick data retrieval may be necessary for timely decision-making.
- Consideration of scalable systems might be required to accommodate growth.

Mitigating Performance Concerns:

To address these potential challenges, I'll be focusing on:

- Finding a solution to handle the increase in data volumes.
- Exploring methods to enhance data retrieval for quicker insights.
- Focusing on the need for scalable solutions to support and accommodate future demands.

I'm dedicated to resolving these issues and collaborating with the team to ensure high-quality data. Your insights and input are invaluable in guiding our approach.

Looking forward to discussing this further.

Best regards,

Analytics Team