Part3: Evaluating the data quality issues in the data provided.

Users dataset:

The initial dataset for the users table was loaded into a pandas DataFrame, and basic information and statistics were displayed.

```
[16] users = pd.read_json('https://raw.githubusercontent.com/Gops319/Fetch_AE_Takehome/main/users.json',lines=True)

[17] users
```

	_id	active	createdDate	lastLogin	role	signUpSource	state
0	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Email	WI
1	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Email	WI
2	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Email	WI
3	{'$oid': '5ff1e1eacfcf6c399c274ae6'}	True	{'$date': 1609687530554}	{'$date': 1609687530597}	consumer	Email	WI
4	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Email	WI
...	...	...	...	...	...	...	...
490	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	NaN	NaN
491	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	NaN	NaN
492	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	NaN	NaN
493	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	NaN	NaN
494	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	NaN	NaN

495 rows × 7 columns

The columns id, createddate, and lastlogin were found to be dictionary objects containing single key-value pairs, which were subsequently unpacked. The date values were in Unix epoch timestamps and were converted to standard timestamps. The id column was renamed to user_id.

```
[35] users.info()

<class 'pandas.core.frame.DataFrame'>
Index: 212 entries, 0 to 475
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   userId        212 non-null    object
 1   active        212 non-null    bool
 2   createdDate   212 non-null    datetime64[ns]
 3   lastLogin     172 non-null    datetime64[ns]
 4   role          212 non-null    object
 5   signUpSource  207 non-null    object
 6   state         206 non-null    object
dtypes: bool(1), datetime64[ns](2), object(4)
memory usage: 11.8+ KB
```

Upon checking for null values and unique values in the primary key column, it was observed that out of 495 records, only 212 user_ids were not null, indicating a significant data quality issue.

```
Number of records before removing duplicates: 495
Number of records after removing duplicates: 212
Number of null values in userId: 0
Number of duplicate values in userId: 0
Records where createdDate is later than lastLogin date:
Empty DataFrame
Columns: [userId, active, createdDate, lastLogin, role, signUpSource, state]
Index: []
Unique values in 'role' column: ['consumer' 'fetch-staff']
Unique values in 'signUpSource' column: ['Email' 'Google' nan]
Unique values in 'active' column: [ True False]
Records with null values in createdDate column:
Empty DataFrame
Columns: [userId, createdDate]
Index: []
```

After removing duplicates, the number of records reduced to 212, confirming the absence of nulls or duplicates in the user_id column. Additionally, it was verified that no createdDate was later than the lastLogin date, as users cannot log in without first creating an account.

The role column, which was expected to have a constant value of 'CONSUMER', contained some records with 'fetch-staff'. The signUpSource column had unique values of Email, Google, and Null. There are 40 records with null values in the lastLogin column and 5 records with null values in the state column.

Null values in the signUpSource column, lastLogin column, and state column pose significant data quality issues. Missing values in signUpSource can hinder analysis of user acquisition channels, affecting marketing strategies. Null values in lastLogin can obscure insights into user engagement and retention, leading to potential missteps in user activity assessment. Missing state information can impact regional analysis and targeted efforts, limiting the ability to customize user experiences or marketing efforts based on location.

```
[105] null_values_count = users.isnull().sum()
      null_values_count
```

```
userId          0
active          0
createdDate     0
lastLogin      40
role            0
signUpSource    5
state           6
dtype: int64
```

After flattening the dictionary objects and removing the duplicate records, the users dataset:

	userId	active	createdDate	lastLogin	role	signUpSource	state
0	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	consumer	Email	WI
3	5ff1e1eacfcf6c399c274ae6	True	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	consumer	Email	WI
6	5ff1e1e8cfcf6c399c274ad9	True	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	consumer	Email	WI
7	5ff1e1b7cfcf6c399c274a5a	True	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	consumer	Email	WI
9	5ff1e1f1cfcf6c399c274b0b	True	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	consumer	Email	WI
...	...	...	...	...	...	...	...
435	5fc961c3b8cfca11a077dd33	True	2020-12-03 22:08:10.752	2021-02-26 22:40:20.992	fetch-staff	Email	NH
455	5fa41775898c7a11a6bcef3e	True	2020-11-05 15:17:33.312	2021-03-04 16:02:35.648	fetch-staff	Email	NaN
456	5fa32b4d898c7a11a6bcebce	True	2020-11-04 22:30:29.120	2021-03-04 07:22:40.512	fetch-staff	Google	AL
462	5964eb07e4b03efd0c0f267b	True	2017-07-11 15:13:06.048	2021-03-04 19:08:16.768	fetch-staff	NaN	IL
475	54943462e4b07e684157a532	True	2014-12-19 14:20:41.344	2021-03-05 16:52:26.752	fetch-staff	NaN	NaN

212 rows × 7 columns

Brands dataset:

Similar to users dataset, The initial dataset for the brands table was loaded into a pandas DataFrame, and basic information and statistics were displayed.

	_id	barcode	category	categoryCode	cpg	name	topBrand	brandCode
0	{'$oid': '601ac115be37ce2ead437551'}	511111019862	Baking	BAKING	{'$id': {'$oid': '601ac114be37ce2ead437550'}, ...	test brand @1612366101024	0.0	NaN
1	{'$oid': '601c5460be37ce2ead43755f'}	511111519928	Beverages	BEVERAGES	{'$id': {'$oid': '5332f5fbe4b03c9a25efd0ba'}, ...	Starbucks	0.0	STARBUCKS
2	{'$oid': '601ac142be37ce2ead43755d'}	511111819905	Baking	BAKING	{'$id': {'$oid': '601ac142be37ce2ead437559'}, ...	test brand @1612366146176	0.0	TEST BRANDCODE @1612366146176
3	{'$oid': '601ac142be37ce2ead43755a'}	511111519874	Baking	BAKING	{'$id': {'$oid': '601ac142be37ce2ead437559'}, ...	test brand @1612366146051	0.0	TEST BRANDCODE @1612366146051
4	{'$oid': '601ac142be37ce2ead43755e'}	511111319917	Candy & Sweets	CANDY_AND_SWEETS	{'$id': {'$oid': '5332fa12e4b03c9a25efd1e7'}, ...	test brand @1612366146827	0.0	TEST BRANDCODE @1612366146827
...	...	...	...	...	...	...	...	...
1162	{'$oid': '5f77274dbe37ce6b592e90c0'}	511111116752	Baking	BAKING	{'$ref': 'Cogs', '$id': {'$oid': '5f77274dbe37...	test brand @1601644365844	NaN	NaN
1163	{'$oid': '5dc1fca91dda2c0ad7da64ae'}	511111706328	Breakfast & Cereal	NaN	{'$ref': 'Cogs', '$id': {'$oid': '53e10d6368ab...	Dippin Dots® Cereal	NaN	DIPPIN DOTS CEREAL
1164	{'$oid': '5f494c6e04db711dd8fe87e7'}	511111416173	Candy & Sweets	CANDY_AND_SWEETS	{'$ref': 'Cogs', '$id': {'$oid': '5332fa12e4b0...	test brand @1598639215217	NaN	TEST BRANDCODE @1598639215217
1165	{'$oid': '5a021611e4b00efe02b02a57'}	511111400608	Grocery	NaN	{'$ref': 'Cogs', '$id': {'$oid': '5332f5f6e4b0...	LIPTON TEA Leaves	0.0	LIPTON TEA Leaves
1166	{'$oid': '6026d757be37ce6369301468'}	511111019930	Baking	BAKING	{'$id': {'$oid': '6026d757be37ce6369301467'}, ...	test brand @1613158231643	0.0	TEST BRANDCODE @1613158231644

The initial observations are as follows. Both the 'id' and 'cpg' columns are in dictionary format, with 'cpg' containing two key-value pairs ('id' and 'ref'). While the 'topBrand' column was meant to have Boolean values, it initially contained 1s and 0s, which were corrected to True and False.

```
brands.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1167 entries, 0 to 1166
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   barcode       1167 non-null   int64
 1   category      1012 non-null   object
 2   categoryCode  517 non-null    object
 3   name          1167 non-null   object
 4   topBrand      555 non-null    float64
 5   brandCode     933 non-null    object
 6   brandId       1167 non-null   object
 7   cpgId         1167 non-null   object
 8   cpgRef        1167 non-null   object
dtypes: float64(1), int64(1), object(7)
memory usage: 82.2+ KB
```

No duplicate records were found in this dataset. The primary key column 'brandId' does not have any null values or duplicate values. The topBrand column which was supposed to have only Boolean values had 1’s and 0’s in place of true and False respectively. This has been modified, but it still has some Null values.

```
Number of records before removing duplicates: 1167
Number of records after removing duplicates: 1167
Number of null values in brandId: 0
Number of duplicate values in brandId: 0
Unique values in 'topBrand' column: [False nan True]
barcode           0
category        155
categoryCode    650
name              0
topBrand        612
brandCode       234
brandId           0
cpgId             0
cpgRef            0
dtype: int64
```

There are 14 duplicate records for same barcode. It is a significant data quality issue as same barcode cannot be assigned for multiple items. There are 234 duplicates for same brandcode and nulls and zero values were present in the brandcode column. This is another data quality issue. Duplicate barcodes and brand codes can lead to inventory mismanagement, causing confusion in product identification and potentially impacting sales and customer satisfaction. Null values in critical columns like 'topBrand' and 'category' hinder accurate data analysis and segmentation.

```
Number of unique values in 'categoryCode' column: 14
Number of unique values in 'category' column: 23
Number of unique values in 'barcode' column: 1160
Number of unique values in 'brandCode' column: 897
Number of duplicate barcode values: 14
Number of duplicate brandcode values: 234
```

	barcode	category	categoryCode	name	topBrand	brandCode	brandId	cpgId	cpgRef
0	511111019862	Baking	BAKING	test brand @1612366101024	False	NaN	601ac115be37ce2ead437551	601ac114be37ce2ead437550	Cogs
11	511111102540	NaN	NaN	MorningStar	NaN	NaN	57c08106e4b0718ff5fcb02c	5332f5f2e4b03c9a25efd0aa	Cpgs
18	511111317364	Baking	BAKING	test brand @1605535049181	False	NaN	5fb28549be37ce522e165cb5	5fb28549be37ce522e165cb4	Cogs
23	511111303947	NaN	NaN	Bottled Starbucks	NaN	NaN	5332f5fee4b03c9a25efd0bd	53e10d6368abd3c7065097cc	Cpgs
24	511111802914	NaN	NaN	Full Throttle	NaN	NaN	5332fa7ce4b03c9a25efd22e	5332f5ebe4b03c9a25efd0a8	Cpgs
...	...	...	...	...	...	...	...	...	...
1135	511111405184	NaN	NaN	Do It Yourself	NaN	NaN	5d658fca6d5f3b23d1bc7912	53e10d6368abd3c7065097cc	Cogs
1144	511111202516	NaN	NaN	Corona	NaN	NaN	57c08242e4b0718ff5fcb032	5332f7a7e4b03c9a25efd134	Cpgs
1146	511111703105	NaN	NaN	Bellatoria	NaN	NaN	5332fa12e4b03c9a25efd1e6	5332fa12e4b03c9a25efd1e7	Cpgs
1157	511111303015	NaN	NaN	DASANI	NaN	NaN	5332fa75e4b03c9a25efd221	5332f5ebe4b03c9a25efd0a8	Cpgs
1162	511111116752	Baking	BAKING	test brand @1601644365844	NaN	NaN	5f77274dbe37ce6b592e90c0	5f77274dbe37ce6b592e90bf	Cogs

234 rows × 9 columns

Since one cpgId can have multiple cpgRef values, it's not suitable for a separate table as it would require a composite primary key consisting of both columns. This approach may not align well with the intended use of a relational database. Similarly, the presence of null values in both the category and categoryCode columns within the same record makes it challenging to create a separate table for categories. Without a reliable unique identifier for each category, it's difficult to establish a primary key for the table.

After flattening the dictionary objects and removing the duplicate records, the brands dataset:

	barcode	category	categoryCode	name	topBrand	brandCode	brandId	cpgId	cpgRef
0	511111019862	Baking	BAKING	test brand @1612366101024	False	NaN	601ac115be37ce2ead437551	601ac114be37ce2ead437550	Cogs
1	511111519928	Beverages	BEVERAGES	Starbucks	False	STARBUCKS	601c5460be37ce2ead43755f	5332f5fbe4b03c9a25efd0ba	Cogs
2	511111819905	Baking	BAKING	test brand @1612366146176	False	TEST BRANDCODE @1612366146176	601ac142be37ce2ead43755d	601ac142be37ce2ead437559	Cogs
3	511111519874	Baking	BAKING	test brand @1612366146051	False	TEST BRANDCODE @1612366146051	601ac142be37ce2ead43755a	601ac142be37ce2ead437559	Cogs
4	511111319917	Candy & Sweets	CANDY_AND_SWEETS	test brand @1612366146827	False	TEST BRANDCODE @1612366146827	601ac142be37ce2ead43755e	5332fa12e4b03c9a25efd1e7	Cogs
...	...	...	...	...	...	...	...	...	...
1162	511111116752	Baking	BAKING	test brand @1601644365844	NaN	NaN	5f77274dbe37ce6b592e90c0	5f77274dbe37ce6b592e90bf	Cogs
1163	511111706328	Breakfast & Cereal	NaN	Dippin Dots® Cereal	NaN	DIPPIN DOTS CEREAL	5dc1fca91dda2c0ad7da64ae	53e10d6368abd3c7065097cc	Cogs
1164	511111416173	Candy & Sweets	CANDY_AND_SWEETS	test brand @1598639215217	NaN	TEST BRANDCODE @1598639215217	5f494c6e04db711dd8fe87e7	5332fa12e4b03c9a25efd1e7	Cogs
1165	511111400608	Grocery	NaN	LIPTON TEA Leaves	False	LIPTON TEA Leaves	5a021611e4b00efe02b02a57	5332f5f6e4b03c9a25efd0b4	Cogs
1166	511111019930	Baking	BAKING	test brand @1613158231643	False	TEST BRANDCODE @1613158231644	6026d757be37ce6369301468	6026d757be37ce6369301467	Cogs

1167 rows × 9 columns

Receipts dataset:

	_id	bonusPointsEarned	bonusPointsEarnedReason	createDate	dateScanned	finishedDate	modifyDate	pointsAwardedDate	pointsEarned	purchaseDate	purchase
0	{'$oid': '5ff1e1eb0a720f0523000575'}	500.0	Receipt number 2 completed, bonus point schedu...	{'$date': 1609687531000}	{'$date': 1609687531000}	{'$date': 1609687531000}	{'$date': 1609687536000}	{'$date': 1609687531000}	500.0	{'$date': 1609632000000}	
1	{'$oid': '5ff1e1bb0a720f052300056b'}	150.0	Receipt number 5 completed, bonus point schedu...	{'$date': 1609687483000}	{'$date': 1609687483000}	{'$date': 1609687483000}	{'$date': 1609687488000}	{'$date': 1609687483000}	150.0	{'$date': 1609601083000}	
2	{'$oid': '5ff1e1f10a720f052300057a'}	5.0	All-receipts receipt bonus	{'$date': 1609687537000}	{'$date': 1609687537000}	NaN	{'$date': 1609687542000}	NaN	5.0	{'$date': 1609632000000}	
3	{'$oid': '5ff1e1ee0a7214ada100056f'}	5.0	All-receipts receipt bonus	{'$date': 1609687534000}	{'$date': 1609687534000}	{'$date': 1609687534000}	{'$date': 1609687539000}	{'$date': 1609687534000}	5.0	{'$date': 1609632000000}	
4	{'$oid': '5ff1e1d20a7214ada1000561'}	5.0	All-receipts receipt bonus	{'$date': 1609687506000}	{'$date': 1609687506000}	{'$date': 1609687511000}	{'$date': 1609687511000}	{'$date': 1609687506000}	5.0	{'$date': 1609601106000}	
...	...	...	...	...	...	...	...	...	...	...	
1114	{'$oid': '603cc0630a720fde100003e6'}	25.0	COMPLETE_NONPARTNER_RECEIPT	{'$date': 1614594147000}	{'$date': 1614594147000}	NaN	{'$date': 1614594148000}	NaN	25.0	{'$date': 1597622400000}	
1115	{'$oid': '603d0b710a720fde1000042a'}	NaN	NaN	{'$date': 1614613361873}	{'$date': 1614613361873}	NaN	{'$date': 1614613361873}	NaN	NaN	NaN	
1116	{'$oid': '603cf5290a720fde10000413'}	NaN	NaN	{'$date': 1614607657664}	{'$date': 1614607657664}	NaN	{'$date': 1614607657664}	NaN	NaN	NaN	
1117	{'$oid': '603ce7100a7217c72c000405'}	25.0	COMPLETE_NONPARTNER_RECEIPT	{'$date': 1614604048000}	{'$date': 1614604048000}	NaN	{'$date': 1614604049000}	NaN	25.0	{'$date': 1597622400000}	
1118	{'$oid': '603c4fea0a7217c72c000389'}	NaN	NaN	{'$date': 1614565354962}	{'$date': 1614565354962}	NaN	{'$date': 1614565354962}	NaN	NaN	NaN	

1119 rows × 15 columns

The rewardsReceiptItemList column, which contains multiple key-value pairs, has been unpacked into a separate table to hold item information. Similarly, other columns with single key-value pairs have been unpacked. Timestamp values originally in Unix epoch format, such as createDate, dateScanned, finishedDate, modifyDate, pointsAwardedDate, and purchaseDate, have been converted to standard timestamps. Additionally, the column id has been renamed to receipt_id.

```
[180] receipts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1119 entries, 0 to 1118
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   receiptId               1119 non-null   object
 1   bonusPointsEarned       544 non-null    float64
 2   bonusPointsEarnedReason 544 non-null    object
 3   createDate              1119 non-null   datetime64[ns]
 4   dateScanned             1119 non-null   datetime64[ns]
 5   finishedDate            568 non-null    datetime64[ns]
 6   modifyDate              1119 non-null   datetime64[ns]
 7   pointsAwardedDate       537 non-null    datetime64[ns]
 8   pointsEarned            609 non-null    float64
 9   purchaseDate            671 non-null    datetime64[ns]
 10  purchasedItemCount      635 non-null    float64
 11  rewardsReceiptStatus    1119 non-null   object
 12  totalSpent              684 non-null    float64
 13  userId                  1119 non-null   object
dtypes: datetime64[ns](6), float64(4), object(4)
memory usage: 122.5+ KB
```

```python
# Find user IDs from receipts that are not present in the user table
missing_user_ids = receipts[~receipts['userId'].isin(users['userId'])]
missing_user_count = len(missing_user_ids)

# Display the missing user IDs and count
print("Missing User IDs:")
print(missing_user_ids[['userId']])
print("\nNumber of Missing User IDs:", missing_user_count)
```

```
Number of records before removing duplicates: 1119
Number of records after removing duplicates: 1119
Number of null values in receiptId: 0
Number of duplicate values in receiptId: 0
Unique values in 'rewards receipt status' column: ['FINISHED' 'REJECTED' 'FLAGGED' 'SUBMITTED' 'PENDING']
Missing User IDs:
                       userId
13   5f9c74f7c88c1415cbddb839
15   5ff1e1e9b6a9d73a3a9f10f6
16   5ff1e1dfcfcf6c399c274ab3
20   5f9c74e3f1937815bd2c1d73
21   5ff1e196cfcf6c399c274a38
..                        ...
955  60253861efa6017a44dc6b50
956  60253891b54593795bf69242
966  60253891b54593795bf69242
985  60268c7bb545931ac63683af
990  60268c78efa6011bb151077d

[148 rows x 1 columns]

Number of Missing User IDs: 148
```

Throughout the dataset, no duplicate records were identified, and the primary key receiptId has no nulls or duplicate values. However, a significant data quality issue arises from 148 records containing user IDs not present in the user table, which compromises referential integrity and requires resolution.

```
receipts
```

```
Records where purchaseDate is later than pointsAwardedDate date:
                    receiptId  bonusPointsEarned  \
14   5ff1e1b20a7214ada100055a              300.0
85   5ff4ce640a7214ada10005e0               25.0
362  600887560a720f05fa000098              250.0
553  60145a3d0a7214ad50000082              750.0

                                bonusPointsEarnedReason  \
14   Receipt number 4 completed, bonus point schedu...
85                          COMPLETE_NONPARTNER_RECEIPT
362  Receipt number 3 completed, bonus point schedu...
553  Receipt number 1 completed, bonus point schedu...

                  createDate             dateScanned            finishedDate  \
14  2021-01-03 15:24:58.752 2021-01-03 15:24:58.752 2021-01-03 15:24:58.752
85  2021-01-05 20:38:46.016 2021-01-05 20:38:46.016 2021-01-05 20:38:46.016
362 2021-01-20 19:40:27.008 2021-01-20 19:40:27.008 2021-01-20 19:40:27.008
553 2021-01-29 18:55:24.672 2021-01-29 18:55:24.672 2021-01-29 18:55:24.672

                  modifyDate       pointsAwardedDate  pointsEarned  \
14  2021-01-03 15:24:58.752 2021-01-03 15:24:58.752         300.0
85  2021-01-05 20:38:46.016 2021-01-05 20:38:46.016          25.0
362 2021-01-20 19:40:27.008 2021-01-20 19:40:27.008         250.0
553 2021-01-29 18:55:24.672 2021-01-29 18:55:24.672         750.0

                purchaseDate  purchasedItemCount rewardsReceiptStatus  \
14  2021-02-03 15:23:44.000                 1.0             FINISHED
85  2021-02-05 20:39:42.336                 1.0             FINISHED
362 2021-02-20 19:41:23.328                 1.0             FINISHED
553 2021-02-28 18:56:44.544                 1.0             FINISHED

     totalSpent                    userId
14          1.0  5ff1e194b6a9d73a3a9f1052
85          1.0  5ff4ce33c3d63511e2a484b6
362         1.0  6008873eb6310511daa4e8eb
553         1.0  60145a3c84231211ce796c5d
```

Additionally, some records display a discrepancy where the points awarded date precedes the purchase date. Such inconsistencies could undermine the accuracy of rewards allocation and have adverse effects on business operations, including customer loyalty metrics and financial reporting.

After flattening the dictionary objects and removing the duplicate records, the receipts dataset:

```
receipts
```

	receiptId	bonusPointsEarned	bonusPointsEarnedReason	createDate	dateScanned	finishedDate	modifyDate	pointsAwardedDate	pointsEarned	purchaseDate	purchasedItem
0	5ff1e1eb0a720f0523000575	500.0	Receipt number 2 completed, bonus point schedu...	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	500.0	2021-01-03 00:00:55.296	
1	5ff1e1bb0a720f052300056b	150.0	Receipt number 5 completed, bonus point schedu...	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	150.0	2021-01-02 15:25:22.304	
2	5ff1e1f10a720f052300057a	5.0	All-receipts receipt bonus	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	NaT	2021-01-03 15:24:58.752	NaT	5.0	2021-01-03 00:00:55.296	
3	5ff1e1ee0a7214ada100056f	5.0	All-receipts receipt bonus	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	5.0	2021-01-03 00:00:55.296	
4	5ff1e1d20a7214ada1000561	5.0	All-receipts receipt bonus	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	2021-01-03 15:24:58.752	5.0	2021-01-02 15:25:22.304	
...	...	...	...	...	...	...	...	...	...	...	
1114	603cc0630a720fde100003e6	25.0	COMPLETE_NONPARTNER_RECEIPT	2021-03-01 10:22:59.072	2021-03-01 10:22:59.072	NaT	2021-03-01 10:22:59.072	NaT	25.0	2020-08-17 00:00:52.224	
1115	603d0b710a720fde1000042a	NaN	NaN	2021-03-01 15:41:55.584	2021-03-01 15:41:55.584	NaT	2021-03-01 15:41:55.584	NaT	NaN	NaT	
1116	603cf5290a720fde10000413	NaN	NaN	2021-03-01 14:07:59.488	2021-03-01 14:07:59.488	NaT	2021-03-01 14:07:59.488	NaT	NaN	NaT	
1117	603ce7100a7217c72c000405	25.0	COMPLETE_NONPARTNER_RECEIPT	2021-03-01 13:06:49.472	2021-03-01 13:06:49.472	NaT	2021-03-01 13:06:49.472	NaT	25.0	2020-08-17 00:00:52.224	
1118	603c4fea0a7217c72c000389	NaN	NaN	2021-03-01 02:22:23.232	2021-03-01 02:22:23.232	NaT	2021-03-01 02:22:23.232	NaT	NaN	NaT	

971 rows × 14 columns

Items dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6941 entries, 0 to 6940
Data columns (total 35 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   barcode                           3090 non-null   object
 1   description                       6560 non-null   object
 2   finalPrice                        6767 non-null   object
 3   itemPrice                         6767 non-null   object
 4   needsFetchReview                  813 non-null    object
 5   partnerItemId                     6941 non-null   object
 6   preventTargetGapPoints            358 non-null    object
 7   quantityPurchased                 6767 non-null   float64
 8   userFlaggedBarcode                337 non-null    object
 9   userFlaggedNewItem                323 non-null    object
 10  userFlaggedPrice                  299 non-null    object
 11  userFlaggedQuantity               299 non-null    float64
 12  receiptId                         6941 non-null   object
 13  needsFetchReviewReason            219 non-null    object
 14  pointsNotAwardedReason            340 non-null    object
 15  pointsPayerId                     1267 non-null   object
 16  rewardsGroup                      1731 non-null   object
 17  rewardsProductPartnerId           2269 non-null   object
 18  userFlaggedDescription            205 non-null    object
 19  originalMetaBriteBarcode          71 non-null     object
 20  originalMetaBriteDescription      10 non-null     object
 21  brandCode                         2600 non-null   object
 22  competitorRewardsGroup            275 non-null    object
 23  discountedItemPrice               5769 non-null   object
 24  originalReceiptItemText           5760 non-null   object
 25  itemNumber                        153 non-null    object
 26  originalMetaBriteQuantityPurchased 15 non-null    float64
 27  pointsEarned                      927 non-null    object
 28  targetPrice                       378 non-null    object
 29  competitiveProduct                645 non-null    object
 30  originalFinalPrice                9 non-null      object
 31  originalMetaBriteItemPrice        9 non-null      object
 32  deleted                           9 non-null      object
 33  priceAfterCoupon                  956 non-null    object
 34  metabriteCampaignId               863 non-null    object
dtypes: float64(3), object(32)
memory usage: 1.9+ MB
```

The dataset, derived from the rewardsReceiptItemList column of the receipts dataset, comprises 35 columns.

```
barcode                           3851
description                        381
finalPrice                         174
itemPrice                          174
needsFetchReview                  6128
partnerItemId                        0
preventTargetGapPoints            6583
quantityPurchased                  174
userFlaggedBarcode                6604
userFlaggedNewItem                6618
userFlaggedPrice                  6642
userFlaggedQuantity               6642
receiptId                            0
needsFetchReviewReason            6722
pointsNotAwardedReason            6601
pointsPayerId                     5674
rewardsGroup                      5210
rewardsProductPartnerId           4672
userFlaggedDescription            6736
originalMetaBriteBarcode          6870
originalMetaBriteDescription      6931
brandCode                         4341
competitorRewardsGroup            6666
discountedItemPrice               1172
originalReceiptItemText           1181
itemNumber                        6788
originalMetaBriteQuantityPurchased    6926
pointsEarned                      6014
targetPrice                       6563
competitiveProduct                6296
originalFinalPrice                6932
originalMetaBriteItemPrice        6932
deleted                           6932
priceAfterCoupon                  5985
metabriteCampaignId               6078
dtype: int64
```

While the ideal primary key should be the combination of receiptId and barcode, due to a substantial number of null values in the barcode column, an alternative composite key using receiptId and partnerItemId is adopted to uniquely identify records. All receiptIds are not present in the receipts table which compromises referential integrity.

Notably, the presence of 3851 null values in the barcode column, intended for item identification, poses a significant data quality concern. This deficiency could lower precise item tracking and affect inventory management processes. Additionally, several other columns exhibit notable null values, worsening data reliability challenges. The inability to utilize the barcode column as a foreign key referencing the brands table is also noteworthy. The barcode column in the items table could have served as a reference to the brands table, but because some item barcodes are missing in the brands table, this connection couldn't be established. In such cases, using surrogate key can be a good option.