DATA QUALITY REPORT

Methodology

1. Checking the quality of the data in individual tables.

Brands Table

a. Checking the uniqueness of the brand_uuid as it's supposed to be a unique identifier according to the limited metadata provided:

Query Query History

```
1  select count(distinct brand_uuid) from public.brands;
```

Data Output Messages Notifications

	count bigint
1	1167

Result: Brand_uuid is a unique column.

b. Checking for nulls in each column based on the importance of the column:

Query Query History

```sql
SELECT
    brand_code,
    COUNT(*) AS total_rows,
    SUM(CASE WHEN brand_code IS NULL THEN 1 ELSE 0 END) AS null_count,
    (SUM(CASE WHEN brand_code IS NULL THEN 1 ELSE 0 END) / COUNT(*)) * 100 AS null_percentage
FROM
    brands
GROUP BY
    brand_code
ORDER BY brand_code;
```

Data Output Messages Notifications

	brand_code character varying (100)	total_rows bigint	null_count bigint	null_percentage bigint
1		35	0	0
2	MILANO	1	0	0
3	SOBE	1	0	0
4	TAZO BOTTLED TEAS	1	0	0
5	09090909090	1	0	0
6	0987654321	1	0	0
7	1915 BOLTHOUSE FARMS	1	0	0
8	511111005148	1	0	0
9	511111005216	1	0	0
10	511111005377	1	0	0
11	511111005421	1	0	0
12	511111005704	1	0	0

Total rows: 898 of 898 Query complete 00:00:00.180

Result: Brand_code which is an extremely important foreign key for the brands table connecting it to the item receipts table has nulls, blank values, inconsistencies (text codes as well as numeric codes).
Fix: We should come up with a nomenclature for brand codes that should follow a consistent format with no missing values or nulls.

c. Data redundancy: I found quite a few columns with repetitive information, hence we could eliminate those columns to avoid redundancy.

Users Table

a. Checking for uniqueness of the unique identifier, user_id:

A similar query to the one above found that there are duplicate values in this column and in turn duplicate rows in the users table with only 212 unique values out of the 500+ records in the users table.

Receipts Table

a. The bonus_points_earned_reason column can be streamlined to be broken up into further columns or made to follow a discrete value selection column to support further analysis.
b. There are redundant columns with date_scanned and finished_date having the

same values.

c. The distribution of receipts scanned amongst different months is abornmal based on investigation:

Query Query History

```
select date_trunc('month', date_scanned) as month, count(*)
from public.receipts
group by date_trunc('month', date_scanned)
order by date_trunc('month', date_scanned) desc;
```

Data Output Messages Notifications

	month timestamp without time zone	count bigint
1	2021-03-01 00:00:00	22
2	2021-02-01 00:00:00	448
3	2021-01-01 00:00:00	641
4	2020-11-01 00:00:00	6
5	2020-10-01 00:00:00	2

Result: Either the data is not being completely captured or there could be business reasons for this.

d. Interesting find:

There are 258 unique users in the receipts table but only 212 unique users in our users table. Hence we are not capturing all the information in our users table or

there is something wrong in the receipts table:

Query Query History

```
SELECT user_id, count(*) FROM public.receipts
group by user_id
order by count DESC
```

Data Output Messages Notifications

	user_id character varying (24)	count bigint
1	5fc961c3b8cfca11a077dd33	436
2	59c124bae4b0299e55b0f330	58
3	54943462e4b07e684157a5...	50
4	5fa41775898c7a11a6bcef3e	21
5	5ff5d15aeb7c7d12096d91a2	20
6	600fb1ac73c60b12049027bb	16
7	5ff1e194b6a9d73a3a9f1052	14
8	5ff47392c3d63511e2a47881	10
9	600987d77d983a11f63cfa92	10
10	5a43c08fe4b014fd6b6a0612	9
11	6007464b6e64691717e8c1f0	8
12	5ff1e1eacfcf6c399c274ae6	8
13	5fff4beedf9ace121f0c17ea	8
14	5ff370c562fde912123a5e0e	7
15	6010bddaa4b74c120bd19dfb	7
16	5fff2698b3348b03eb45bb10	7

Item Receipts Table

a. The item receipts table has 31 columns which are highly dimensional in nature. To solve the business questions, I reduced the dimensionality by reducing the number of columns to only necessary 12 columns in the bridge_item_receipts table.

It did not have any primary key, so I created the item_receipt_id key in the table.

To obtain the foreign key relation between the bridge_item_receipts table and the dim_brands table, I checked the following which did not give sufficient outcome.

Query Query History

```sql
SELECT barcode
FROM dim_brands
WHERE barcode IN (
    SELECT barcode
    FROM bridge_item_receipts

);
```

Data Output Messages Notifications

	barcode character varying (20)
1	511111204206
2	511111104537
3	511111001768
4	511111001485
5	511111502142
6	511111104186
7	511111902690
8	511111904175
9	511111101451
10	511111518044
11	511111602118

Total rows: 17 of 17 Query complete 00:00:00.081

Barcode could not be created a foreign key.

Further, I checked the relation using brand_code as the key:

Query Query History

```sql
SELECT brand_code
FROM dim_brands
WHERE brand_code IN (
    SELECT brand_code
    FROM bridge_item_receipts

);
```

Data Output Messages Notifications

	brand_code character varying (255)
1	TACO BELL
2	COTTONELLE
3	SWANSON
4	KETTLE BRAND
5	DORITOS
6	KLONDIKE
7	PLANTERS
8	CHEETOS
9	COOL WHIP
10	TOSTITOS
11	NATURE VALLEY

Total rows: 42 of 42 Query complete 00:00:00.071

This was a better way to connect the two tables.

Query Query History

```sql
SELECT count(*)
FROM dim_brands
WHERE brand_code is null;
```

Data Output Messages Notifications

	count bigint
1	234

However, it had nulls and therefore to make it as Primary and Foreign Key, the nulls have to be removed or taken care of.

b. Insufficient information about the partner_item_id and rewards_product_partner_id and its relation with brand_code.

Query Query History

```
SELECT rewards_product_partner_id, brand_code
    FROM public.bridge_item_receipts
group by rewards_product_partner_id,brand_code
order by brand_code ;
```

Data Output Messages Notifications

	rewards_product_partner_id character varying (255)	brand_code character varying (255)
1	[null]	7UP
2	5332f5fbe4b03c9a25efd0ba	7UP
3	5d9b4f591dda2c6225a284aa	ADVIL
4	[null]	ADVIL
5	559c2234e4b06aca36af13c6	AMERICAN BEAUTY
6	[null]	AMERICAN BEAUTY
7	5fb6b608be37ce522e165cb9	ARNOLD
8	559c2234e4b06aca36af13c6	ARROWHEAD
9	[null]	ARROWHEAD
10	559c2234e4b06aca36af13c6	AZTECA
11	5ffdb69ebe37ce746df1192a	BANZA
12	[null]	BEAR CREEK COUNTRY KITCHENS
13	[null]	BEN AND JERRYS

Total rows: 325 of 325 Query complete 00:00:00.076

Query Query History

```
SELECT rewards_product_partner_id, brand_code,partner_item_id
    FROM public.bridge_item_receipts
group by rewards_product_partner_id,brand_code,partner_item_id
order by brand_code ;
```

Data Output Messages Notifications

	rewards_product_partner_id character varying (255)	brand_code character varying (255)	partner_item_id character varying (255)
1	5332f5fbe4b03c9a25efd0ba	7UP	1432
2	[null]	7UP	1422
3	5332f5fbe4b03c9a25efd0ba	7UP	1456
4	5d9b4f591dda2c6225a284aa	ADVIL	1188
5	5d9b4f591dda2c6225a284aa	ADVIL	1190
6	5d9b4f591dda2c6225a284aa	ADVIL	1187
7	5d9b4f591dda2c6225a284aa	ADVIL	1180
8	5d9b4f591dda2c6225a284aa	ADVIL	1199
9	5d9b4f591dda2c6225a284aa	ADVIL	1254
10	5d9b4f591dda2c6225a284aa	ADVIL	1189
11	5d9b4f591dda2c6225a284aa	ADVIL	1181
12	[null]	ADVIL	1198
13	559c2234e4b06aca36af13c6	AMERICAN BEAUTY	1960

Total rows: 1000 of 2993 Query complete 00:00:00.067

If this information had been given, there would be better way to create fact and dimension tables from the bridge_item_receipts table further down, which would be useful to answer several other business questions.