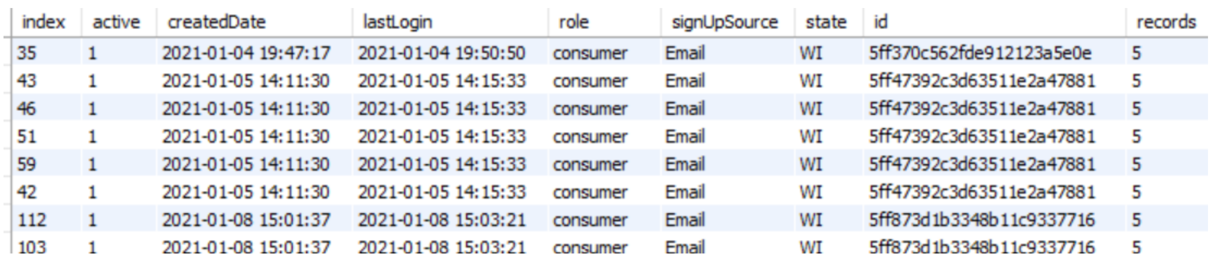
Data Quality Issues

This is a technical document for consumption by the data engineering team and database admins.

1. Duplicate Data

User records are duplicated in the *users* table. This is not a big issue since the duplicated records do not have different signUpSource, state, createdDate or Id. However, it is imperative to correct it to prevent spurious joins.



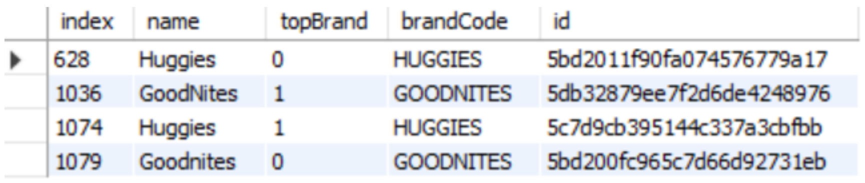
index	active	createdDate	lastLogin	role	signUpSource	state	id	records
35	1	2021-01-04 19:47:17	2021-01-04 19:50:50	consumer	Email	WI	5ff370c562fde912123a5e0e	5
43	1	2021-01-05 14:11:30	2021-01-05 14:15:33	consumer	Email	WI	5ff47392c3d63511e2a47881	5
46	1	2021-01-05 14:11:30	2021-01-05 14:15:33	consumer	Email	WI	5ff47392c3d63511e2a47881	5
51	1	2021-01-05 14:11:30	2021-01-05 14:15:33	consumer	Email	WI	5ff47392c3d63511e2a47881	5
59	1	2021-01-05 14:11:30	2021-01-05 14:15:33	consumer	Email	WI	5ff47392c3d63511e2a47881	5
42	1	2021-01-05 14:11:30	2021-01-05 14:15:33	consumer	Email	WI	5ff47392c3d63511e2a47881	5
112	1	2021-01-08 15:01:37	2021-01-08 15:03:21	consumer	Email	WI	5ff873d1b3348b11c9337716	5
103	1	2021-01-08 15:01:37	2021-01-08 15:03:21	consumer	Email	WI	5ff873d1b3348b11c9337716	5

Figure 1: Results from ***duplicate_users.sql***

The fix is simple, remove duplicates so that the *Id* column can be set as a primary key.

2. Ambiguous Data

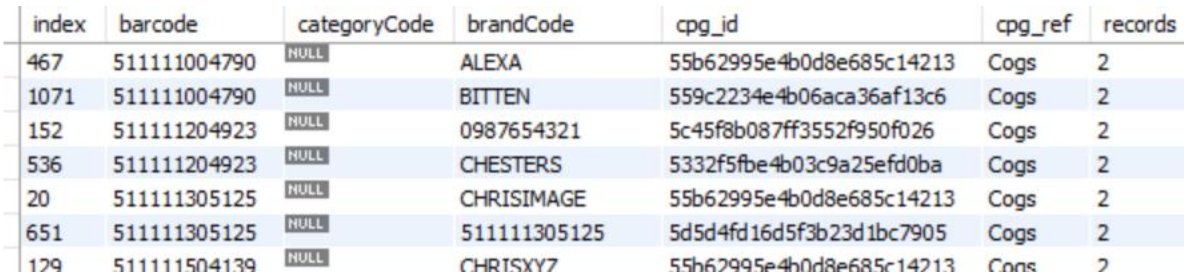
There is ambiguity in the *brands* and *items* tables. Many brands do not have a *brandCode*, this is an issue. More importantly, some brands (Huggies and Goodnite) have more than 1 records with different names but same *brandCode*. It is also unclear whether *Uuid* or *brandCode* is the unique identifier ideally.



index	name	topBrand	brandCode	id
628	Huggies	0	HUGGIES	5bd2011f90fa074576779a17
1036	GoodNites	1	GOODNITES	5db32879ee7f2d6de4248976
1074	Huggies	1	HUGGIES	5c7d9cb395144c337a3cbfbb
1079	Goodnites	0	GOODNITES	5bd200fc965c7d66d92731eb

Figure 2: Results from ***duplicate_brands.sql***

Items ideally must have unique *barcodes*, as they encode price and SKU information. I identified several items with the same *barcode* but different *brandCode*. This is spurious and would lead to inconsistencies in prices in receipt.



index	barcode	categoryCode	brandCode	cpg_id	cpg_ref	records
467	511111004790	NULL	ALEXA	55b62995e4b0d8e685c14213	Cogs	2
1071	511111004790	NULL	BITTEN	559c2234e4b06aca36af13c6	Cogs	2
152	511111204923	NULL	0987654321	5c45f8b087ff3552f950f026	Cogs	2
536	511111204923	NULL	CHESTERS	5332f5fbe4b03c9a25efd0ba	Cogs	2
20	511111305125	NULL	CHRISIMAGE	55b62995e4b0d8e685c14213	Cogs	2
651	511111305125	NULL	511111305125	5d5d4fd16d5f3b23d1bc7905	Cogs	2
129	511111504139	NULL	CHRISXYZ	55b62995e4b0d8e685c14213	Cogs	2

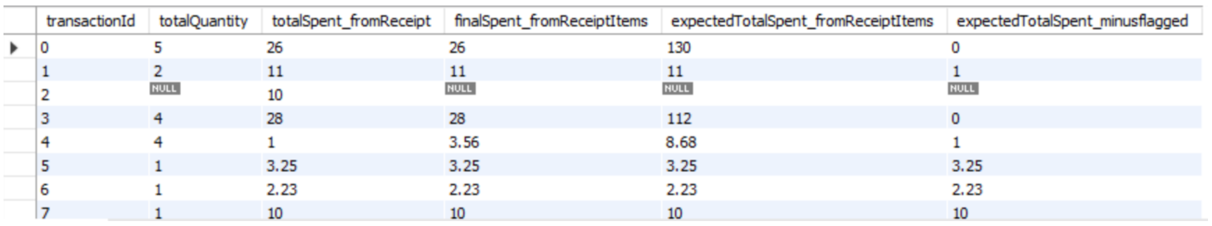
Figure 3: Results from ***duplicate_items.sql***

3. Inconsistent Data

Shoppers become savers when they earn points for the amount they spent. The total amount spent and total points earned in a receipt are given as sum totals in the data.

I ran a query to verify if the totals matched the actuals. They did not!

First, let us check *totalSpent*.



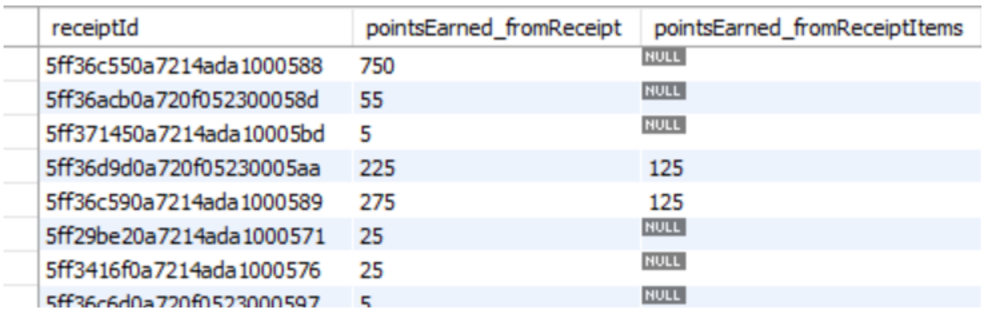
transactionId	totalQuantity	totalSpent_fromReceipt	finalSpent_fromReceiptItems	expectedTotalSpent_fromReceiptItems	expectedTotalSpent_minusflagged
0	5	26	26	130	0
1	2	11	11	11	1
2	NULL	10	NULL	NULL	NULL
3	4	28	28	112	0
4	4	1	3.56	8.68	1
5	1	3.25	3.25	3.25	3.25
6	1	2.23	2.23	2.23	2.23
7	1	10	10	10	10

Figure 4: Results from ***inconsistency_totalspent.sql***

The expenditure calculation from individual receipt items (*finalSpent_fromReceiptItems*) does not sum up to the total given in the data (*totalSpent_fromReceipt*).

Also, *itemPrice* in the receipt items data does not consider the quantity of item purchased. I presume, the amounts should end up as column *expectedTotalSpent_fromReceiptItems* after considering the quantity.

Second, I noticed the same inconsistency with *pointsEarned*.



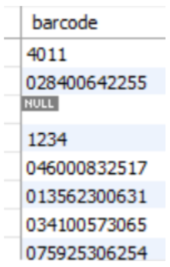
receiptId	pointsEarned_fromReceipt	pointsEarned_fromReceiptItems
5ff36c550a7214ada1000588	750	NULL
5ff36acb0a720f052300058d	55	NULL
5ff371450a7214ada10005bd	5	NULL
5ff36d9d0a720f05230005aa	225	125
5ff36c590a7214ada1000589	275	125
5ff29be20a7214ada1000571	25	NULL
5ff3416f0a7214ada1000576	25	NULL
5ff36c6d0a720f0523000597	5	NULL

Figure 5: Results from ***inconsistency_pointsearned.sql***

I propose that such totals must not be maintained as columns; they need constant reconciliation when there are adjustments in the receipt. Totals get be calculated easily used a bunch of joins.

4. Missing Data

There is not a unified schema for items. Item information is included in the original brands data and original receipts data. I separated it out into an *items* schema using both sources leaving out those which did not have *brandCode*. Then I noticed that 287 items from *receipts* were missing in the newly created *items* schema. This leads me to conclude that item data is scattered in the original data.



```
barcode
4011
028400642255
NULL
1234
046000832517
013562300631
034100573065
075925306254
```

Figure 6: Results from ***missing_items.sql***