

Fetch:

Relational Diagram:

The ER (Entity-Relationship) has the relationships between the entities: `USERS`, `RECEIPTS`, and `BRANDS`, `CPG_BRANDS` along with a linking table `ITEMS`.

Here's an explanation of the mappings:

1. `USERS` to `RECEIPTS`:

- This is a one-to-many relationship. This means that one user can have many receipts, but each receipt is associated with only one user. The `user_id` field in `RECEIPTS` is a foreign key (FK) that references the primary key (PK) `user_id` in the `USERS` table.

2. `RECEIPTS` to `ITEMS`:

- This is also a one-to-many relationship, where one receipt can contain many items, but each item is associated with only one receipt. The `receipt_id` field in `ITEMS` is a foreign key that references the primary key `receipt_id` in the `RECEIPTS` table.

3. `BRANDS` to `ITEMS`:

- This is a many-to-many relationship. It indicates that brands can have many items on different receipts, and each receipt item can be associated with different brands. The relationship is established through the `barcode` field, which is used as a foreign key in `ITEMS` to reference the `barcode` in `BRANDS`.

4. `CPG_BRANDS` to `BRANDS`:

- One-to-many relationship, where a CPG (Consumer Packaged Goods) brand can have many individual brands associated with it. The cpg_id in Brands serves as an 'FK' back to CPG_BRANDS

5. `BRANDS` to `RECEIPTS`:

- This is a many-to-many relationship. The relationship is established through 'name' field in `RECEIPTS` to reference 'name' in `BRANDS`

6. `USERS` to `BRANDS`:

- This is a many-to-many relationship. The relationship is established through 'user_id' field, which is used as a foreign key in `BRANDS`.