RELATIONAL MODEL

This final relational model includes four core tables: **Users**, **Receipts**, **Receipt_Items**, **CPG** and **Brands**. Each table is designed according to normalization principles, primarily following **Third Normal Form (3NF)**, ensuring data integrity, reducing redundancy, and allowing for efficient querying.

1. Users Table

The **Users Table** stores information about individuals who use the application to scan receipts.

Explanation:

- **Purpose**: Centralizes user-related data, such as account creation date, login activity, and role.
- **Relationships**: Linked to the **Receipts Table** in a **One-to-Many** relationship.

---

2. Receipts Table

The **Receipts Table** stores information about receipts scanned by users.

Explanation:

- **Purpose**: Tracks transaction-level data, such as points earned, total spent, and the user associated with the receipt.
- **Relationships**: Linked to the **Users Table** (One-to-Many) and **Receipt_Items Table** (One-to-Many).

---

3. Receipt_Items Table

The **Receipt_Items Table** stores individual items listed on a receipt.

Explanation:

- **Purpose**: Stores item-level details, such as barcode, price, and user-flagged corrections.
- **Relationships**: Linked to the **Receipts Table** (One-to-Many) and to the **Brands Table** (Many-to-One).

---

4. Brands Table

The **Brands Table** stores information about the product brands.

Explanation:

- **Purpose**: Captures brand-specific data for analysis and reference in receipts.
- **Relationships**: Linked to the **Receipt_Items Table** in a **Many-to-One** relationship and CPG table in **Many-to-One**

---

5. CPG (Consumer Packaged Goods) Table

In case brands need to be linked to CPGs, a **CPG Table** can store the CPG company details.

Explanation:

- **Purpose**: Stores details about CPGs, with each CPG possibly managing multiple brands.
- **Relationships**: A **One-to-Many** relationship between CPG and Brands

---

Key Relationships:

1. **Users ➔ Receipts**:
   - A **user** can have multiple **receipts**.
   - **Foreign Key**: Receipts.user_id ➔ Users.user_id
2. **Receipts ➔ Receipt_Items**:
   - A **receipt** can contain multiple **items**.
   - **Foreign Key**: Receipt_Items.receipt_id ➔ Receipts.receipt_id
3. **Brands ➔ Receipt_Items**:
   - A **brand** may be linked to multiple **items**, but each **item** belongs to only one **brand**.
   - **Foreign Key**: Receipt_Items.brand_id ➔ Brands.brand_id
4. **Brands ➔CPG**:
   - One **CPG** may be linked to multiple **brands**, but each **brand** belongs to only one **CPG**.

  - **Foreign Key**: brands.cpg_id ➔ cpg.cpg_id

Normalization:

This model follows **3rd Normal Form (3NF)**:

- Data is organized into distinct tables with minimal redundancy.
- Each table has a clear purpose, ensuring **data integrity** and **efficient querying**.

Conclusion:

This end-to-end relational model captures all relevant data from users, receipts, items, and brands in a structured format that minimizes redundancy, ensures data integrity, and supports efficient querying for analytical purposes. The relationships between these entities are well-defined, ensuring scalability and flexibility.