Fetch Assessment

1. **Entity Relationship Diagram -** Developed a simplified ER diagram after analyzing the user, receipt and brand data schema provided.

fetch-assessment/Entity Relationship Diagram.pdf at main · Sabapathy-deepeka/fetch-assessment

Created a database in PostgreSQL, developed DDL scripts to define the schema.

Brand:

```
-- Table: public.brand

-- DROP TABLE IF EXISTS public.brand;

CREATE TABLE IF NOT EXISTS public.brand
(
    brandid text COLLATE pg_catalog."default" NOT NULL,
    brandname text COLLATE pg_catalog."default",
    cpgid text COLLATE pg_catalog."default",
    cpgref text COLLATE pg_catalog."default",
    category text COLLATE pg_catalog."default",
    categorycode text COLLATE pg_catalog."default",
    barcode text COLLATE pg_catalog."default",
    brandcode text COLLATE pg_catalog."default",
    topbrand boolean,
    CONSTRAINT brand_pkey PRIMARY KEY (brandid)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.brand
    OWNER to postgres;
```

Receipts:

```
-- Table: public.receipts

-- DROP TABLE IF EXISTS public.receipts;

CREATE TABLE IF NOT EXISTS public.receipts
(
    receiptid text COLLATE pg_catalog."default" NOT NULL,
    userid text COLLATE pg_catalog."default" NOT NULL,
    bonuspointsearned integer,
    bonuspointsearnedreason text COLLATE pg_catalog."default",
    createdate timestamp with time zone,
    datescanned timestamp with time zone,
    finisheddate timestamp with time zone,
    modifydate timestamp with time zone,
    pointsawardeddate timestamp with time zone,
    pointsearned text COLLATE pg_catalog."default",
    purchasedate timestamp with time zone,
    purchaseditemcount integer,
    rewardsreceiptstatus text COLLATE pg_catalog."default",
    totalspent text COLLATE pg_catalog."default",
    brandid text COLLATE pg_catalog."default",
    CONSTRAINT receipts_pkey PRIMARY KEY (receiptid),
    CONSTRAINT fk_brand FOREIGN KEY (brandid)
        REFERENCES public.brand (brandid) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.receipts
    OWNER to postgres;
```

ReceiptItem:

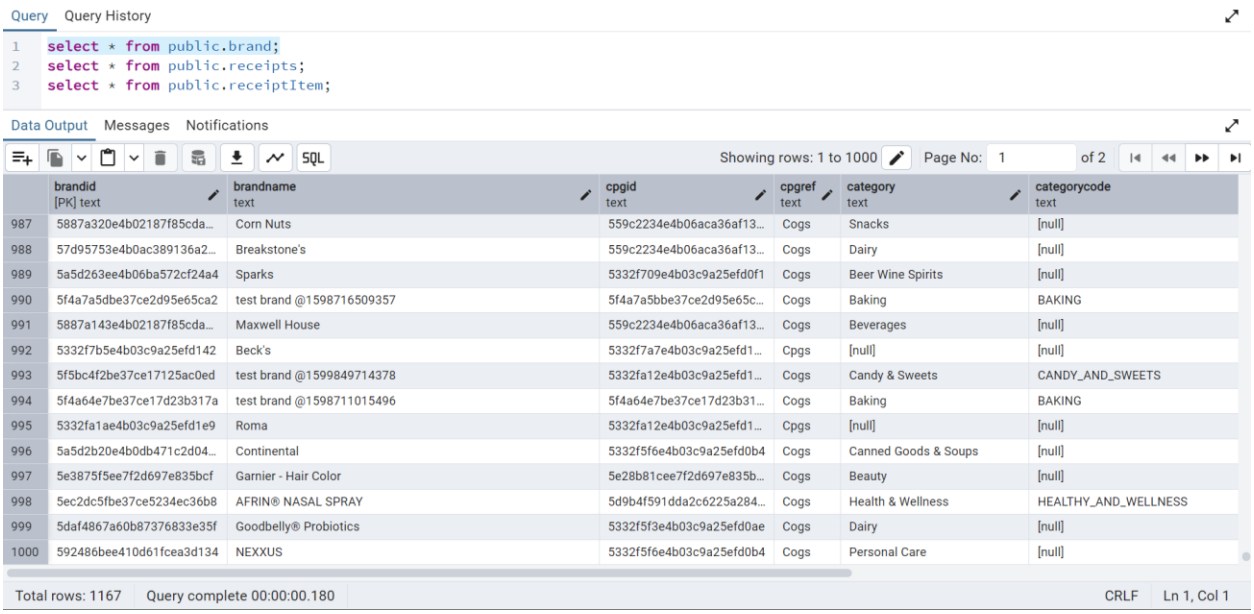
```
-- Table: public.receiptitem

-- DROP TABLE IF EXISTS public.receiptitem;

CREATE TABLE IF NOT EXISTS public.receiptitem
(
    receiptitemid text COLLATE pg_catalog."default" NOT NULL,
    receiptid text COLLATE pg_catalog."default" NOT NULL,
    brandcode text COLLATE pg_catalog."default",
    barcode text COLLATE pg_catalog."default",
    description text COLLATE pg_catalog."default",
    itemprice numeric,
    finalprice numeric,
    needsfetchreview boolean,
    needsfetchreviewreason text COLLATE pg_catalog."default",
    partneritemid integer,
    pointsnotawardedreason text COLLATE pg_catalog."default",
    pointspayerid text COLLATE pg_catalog."default",
    preventtargetgappoints boolean,
    quantitypurchased integer,
    rewardsgroup text COLLATE pg_catalog."default",
    rewardsproductpartnerid text COLLATE pg_catalog."default",
    userflaggedbarcode text COLLATE pg_catalog."default",
    userflaggeddescription text COLLATE pg_catalog."default",
    userflaggednewitem boolean,
    userflaggedprice numeric,
    userflaggedquantity integer,
    CONSTRAINT receiptitem_pkey PRIMARY KEY (receiptitemid)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.receiptitem
    OWNER to postgres;
```

Developed python scripts to parse the JSON files, extract the data fields and values, to insert the data into respective database tables

[fetch-assessment/fetch_assessment at main · Sabapathy-deepeka/fetch-assessment]

Inserted data into brand, receipts, receiptItem and Users tables

Brand:



Query | Query History

```
select * from public.brand;
select * from public.receipts;
select * from public.receiptItem;
```

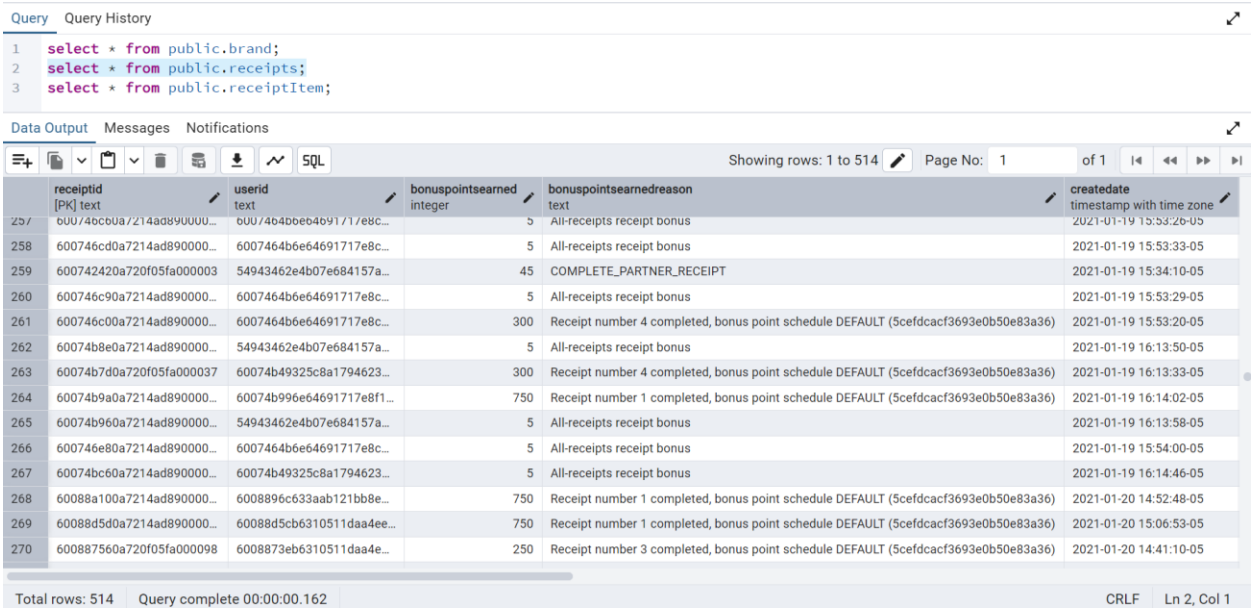
Data Output | Messages | Notifications

Showing rows: 1 to 1000 | Page No: 1 of 2

	brandid [PK] text	brandname text	cpgid text	cpgref text	category text	categorycode text
987	5887a320e4b02187f85cda...	Corn Nuts	559c2234e4b06aca36af13...	Cogs	Snacks	[null]
988	57d95753e4b0ac389136a2...	Breakstone's	559c2234e4b06aca36af13...	Cogs	Dairy	[null]
989	5a5d263ee4b06ba572cf24a4	Sparks	5332f709e4b03c9a25efd0f1	Cogs	Beer Wine Spirits	[null]
990	5f4a7a5dbe37ce2d95e65ca2	test brand @1598716509357	5f4a7a5bbe37ce2d95e65c...	Cogs	Baking	BAKING
991	5887a143e4b02187f85cda...	Maxwell House	559c2234e4b06aca36af13...	Cogs	Beverages	[null]
992	5332f7b5e4b03c9a25efd142	Beck's	5332f7a7e4b03c9a25efd1...	Cpgs	[null]	[null]
993	5f5bc4f2be37ce17125ac0ed	test brand @1599849714378	5332fa12e4b03c9a25efd1...	Cogs	Candy & Sweets	CANDY_AND_SWEETS
994	5f4a64e7be37ce17d23b317a	test brand @1598711015496	5f4a64e7be37ce17d23b31...	Cogs	Baking	BAKING
995	5332fa1ae4b03c9a25efd1e9	Roma	5332fa12e4b03c9a25efd1...	Cpgs	[null]	[null]
996	5a5d2b20e4b0db471c2d04...	Continental	5332f5f6e4b03c9a25efd0b4	Cogs	Canned Goods & Soups	[null]
997	5e3875f5ee7f2d697e835bcf	Garnier - Hair Color	5e28b81cee7f2d697e835b...	Cogs	Beauty	[null]
998	5ec2dc5fbe37ce5234ec36b8	AFRIN® NASAL SPRAY	5d9b4f591dda2c6225a284...	Cogs	Health & Wellness	HEALTHY_AND_WELLNESS
999	5daf4867a60b87376833e35f	Goodbelly® Probiotics	5332f5f3e4b03c9a25efd0ae	Cogs	Dairy	[null]
1000	592486bee410d61fcea3d134	NEXXUS	5332f5f6e4b03c9a25efd0b4	Cogs	Personal Care	[null]

Total rows: 1167 | Query complete 00:00:00.180 | CRLF | Ln 1, Col 1

Receipts:



Query | Query History

```
select * from public.brand;
select * from public.receipts;
select * from public.receiptItem;
```

Data Output | Messages | Notifications

Showing rows: 1 to 514 | Page No: 1 of 1

	receiptid [PK] text	userid text	bonuspointsearned integer	bonuspointsearnedreason text	createdate timestamp with time zone
257	60074c6c0a7214ad890000...	6007464b6e64691717e8c...	5	All-receipts receipt bonus	2021-01-19 15:53:26-05
258	600746cd0a7214ad890000...	6007464b6e64691717e8c...	5	All-receipts receipt bonus	2021-01-19 15:53:33-05
259	600742420a720f05fa000003	54943462e4b07e684157a...	45	COMPLETE_PARTNER_RECEIPT	2021-01-19 15:34:10-05
260	600746c90a7214ad890000...	6007464b6e64691717e8c...	5	All-receipts receipt bonus	2021-01-19 15:53:29-05
261	600746c00a7214ad890000...	6007464b6e64691717e8c...	300	Receipt number 4 completed, bonus point schedule DEFAULT (5cefdcacf3693e0b50e83a36)	2021-01-19 15:53:20-05
262	60074b8e0a7214ad890000...	54943462e4b07e684157a...	5	All-receipts receipt bonus	2021-01-19 16:13:50-05
263	60074b7d0a720f05fa000037	60074b49325c8a1794623...	300	Receipt number 4 completed, bonus point schedule DEFAULT (5cefdcacf3693e0b50e83a36)	2021-01-19 16:13:33-05
264	60074b9a0a7214ad890000...	60074b996e64691717e8f1...	750	Receipt number 1 completed, bonus point schedule DEFAULT (5cefdcacf3693e0b50e83a36)	2021-01-19 16:14:02-05
265	60074b960a7214ad890000...	54943462e4b07e684157a...	5	All-receipts receipt bonus	2021-01-19 16:13:58-05
266	600746e80a7214ad890000...	6007464b6e64691717e8c...	5	All-receipts receipt bonus	2021-01-19 15:54:00-05
267	60074bc60a7214ad890000...	60074b49325c8a1794623...	5	All-receipts receipt bonus	2021-01-19 16:14:46-05
268	60088a100a7214ad890000...	6008896c633aab121bb8e...	750	Receipt number 1 completed, bonus point schedule DEFAULT (5cefdcacf3693e0b50e83a36)	2021-01-20 14:52:48-05
269	60088d5d0a7214ad890000...	60088d5cb6310511daa4ee...	750	Receipt number 1 completed, bonus point schedule DEFAULT (5cefdcacf3693e0b50e83a36)	2021-01-20 15:06:53-05
270	600887560a720f05fa000098	6008873eb6310511daa4e...	250	Receipt number 3 completed, bonus point schedule DEFAULT (5cefdcacf3693e0b50e83a36)	2021-01-20 14:41:10-05

Total rows: 514 | Query complete 00:00:00.162 | CRLF | Ln 2, Col 1

ReceiptItem:

Query | Query History

```
1  select * from public.brand;
2  select * from public.receipts;
3  select * from public.receiptItem;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1000 | Page No: 1 of 3

	receiptitemid [PK] text	receiptid text	brandcode text	barcode text	description text
1	d525739b-7ed2-4aa9-9385-b48720b79…	5ff1e1eb0a720f0523000575	[null]	4011	ITEM NOT FOUND
2	761ea044-1754-436a-86f8-e6956fc0e915	5ff1e1bb0a720f052300056b	[null]	4011	ITEM NOT FOUND
3	fdff82dd-90c6-41c7-9d2f-bc9cc95460ff	5ff1e1bb0a720f052300056b	[null]	028400642255	DORITOS TORTILLA CHIP SPICY SWEET CHILI REDUCED FAT BAG 1 OZ
4	e04243b2-9a5a-48f6-9745-02a8de4e99…	5ff1e1f10a720f052300057a	[null]	[null]	[null]
5	d958388d-b975-4af4-8f16-4061a71f4cbc	5ff1e1ee0a7214ada100056f	[null]	4011	ITEM NOT FOUND
6	3db67555-0c59-4e63-a47c-5edbf964601f	5ff1e1d20a7214ada1000561	[null]	4011	ITEM NOT FOUND
7	f922b690-2dce-4d13-b188-9d3c0d9610…	5ff1e1d20a7214ada1000561	[null]	1234	[null]
8	80607141-2a08-4031-9ce4-46c15d9f89…	5ff1e1e40a7214ada1000566	[null]	4011	ITEM NOT FOUND
9	7aed8ddd-db7d-45f3-849e-6c78786de0…	5ff1e1cd0a720f052300056f	MISSION	[null]	MSSN TORTLLA
10	6df3e3c4-04c2-4c14-8666-2f354bfd09be	5ff1e1a40a720f0523000569	BRAND	046000832517	Old El Paso Mild Chopped Green Chiles, 4.5 Oz
11	50c25552-8a10-450e-9717-a58dbff023…	5ff1e1ed0a7214ada100056e	[null]	4011	ITEM NOT FOUND
12	f2cb91d4-d8e1-484a-a648-40bb13c7c1f6	5ff1e1eb0a7214ada100056b	[null]	4011	ITEM NOT FOUND
13	47c19adf-4039-4cc3-bd65-b9d8718122…	5ff1e1c50a720f052300056c	[null]	4011	ITEM NOT FOUND
14	be6fcfe8-293c-416a-a01d-43d0e6e15a3e	5ff1e1a10a720f0523000568	[null]	013562300631	Annie's Homegrown Organic White Cheddar Macaroni & Cheese Shells, 6 Oz

Total rows: 2771 | Query complete 00:00:00.269 | CRLF | Ln 3, Col 1

2. SQL Queries

1. What are the top 5 brands by receipts scanned for most recent month?

```
WITH january_2021 AS (
    SELECT
        ri.brandCode,
        r.purchaseDate,
        COUNT(ri.receiptId) AS receipt_count
    FROM receiptItem ri
-- Join with receipts and receiptItem table using receiptId
    JOIN receipts r ON ri.receiptId = r.receiptId
    WHERE r.purchaseDate >= '2021-01-01'
      AND r.purchaseDate < '2021-02-01'
    GROUP BY ri.brandCode, r.purchaseDate
)
SELECT
    b.brandId,j.brandCode, b.brandName,j.purchaseDate, j.receipt_count
   FROM january_2021 j
  JOIN brand b ON j.brandCode = b.brandCode
ORDER BY j.receipt_count DESC
LIMIT 5;
```

	brandid text	brandcode text	brandname text	purchasedate timestamp with time zone	receipt_count bigint
1	5bd2013f965c7d66d92731ec	KLEENEX	Kleenex	2021-01-14 19:00:00-05	26
2	5332f5fbe4b03c9a25efd0b9	PEPSI	Pepsi	2021-01-14 19:00:00-05	21
3	5887a372e4b02187f85cdad9	DORITOS	Doritos	2021-01-14 19:00:00-05	16
4	5bd2013f965c7d66d92731ec	KLEENEX	Kleenex	2021-01-20 19:00:00-05	12
5	585a972de4b03e62d1ce0e96	TOSTITOS	Tostitos	2021-01-08 19:00:00-05	10

2. How does the ranking of the top 5 brands by receipts scanned for the recent month compare to the ranking for the previous month?

```
WITH brand_ranking AS (
    SELECT
        ri.brandCode,
        -- Extracting the month
        DATE_TRUNC('month', r.purchaseDate) AS month,
        COUNT(DISTINCT(ri.receiptId)) AS receipt_count
    FROM receiptItem ri
    JOIN receipts r ON ri.receiptId = r.receiptId
    WHERE r.purchaseDate >= '2021-01-01'
      AND r.purchaseDate < '2021-02-01'
    GROUP BY ri.brandCode, month
),
ranked_brands AS (
    SELECT br.brandCode, br.month, br.receipt_count,
        RANK() OVER (PARTITION BY month ORDER BY br.receipt_count DESC) AS rank
    FROM brand_ranking br
)
SELECT  DISTINCT(b.brandId), rb.brandCode, rb.month, rb.receipt_count, rb.rank
FROM ranked_brands rb
JOIN brand b ON rb.brandCode = b.brandCode
ORDER BY rb.month DESC, rb.rank ASC
LIMIT 5;
```

	brandid text	brandcode text	month timestamp with time zone	receipt_count bigint	rank bigint
1	5bd2013f965c7d66d92731ec	KLEENEX	2021-01-01 00:00:00-05	15	2
2	592486bee410d61fcea3d130	KNORR	2021-01-01 00:00:00-05	12	4
3	55a41b88e4b0d0a65b3692f0	KRAFT	2021-01-01 00:00:00-05	11	9
4	5332f5fbe4b03c9a25efd0b9	PEPSI	2021-01-01 00:00:00-05	11	9
5	585a96e9e4b03e62d1ce0e8b	RICE-A-RONI	2021-01-01 00:00:00-05	11	9

3. When considering *average spend* from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?

```
SELECT
    rewardsReceiptStatus,
    AVG(cast(totalSpent AS NUMERIC)) AS avg_spend
FROM receipts
WHERE rewardsReceiptStatus IN ('FINISHED', 'REJECTED')
GROUP BY rewardsReceiptStatus;
```

	rewardsreceiptstatus text	avg_spend numeric
1	FINISHED	81.4474513618677043

4. When considering *total number of items purchased* from receipts with 'rewardsReceiptStatus' of 'Accepted' or 'Rejected', which is greater?

```
SELECT
    rewardsReceiptStatus,
    SUM(purchasedItemCount) AS total_items_purchased
FROM receipts
WHERE rewardsReceiptStatus IN ('FINISHED', 'REJECTED')
GROUP BY rewardsReceiptStatus;
```

	rewardsreceiptstatus text	total_items_purchased bigint
1	FINISHED	8178

3. Data Quality Check

Created a python script to perform data quality checks
fetch-assessment/fetch_assessment/data_quality_check.py at main · Sabapathy-deepeka/fetch-assessment

4. Communicate with stakeholders

Subject: Data Quality Findings
Hi Product/Business Leader,

I have been running data quality checks by developing a python script that runs multiple SQL queries to check for data in-consistencies. As part of that exercise, I identified key data quality issues that may impact analysis and reporting. Your input is needed to determine the next steps.

Key Issues:

1. Missing brandCode values – Critical for mapping receipt items with brands. Should we enforce brandCode as a not-null field to ensure accuracy?
2. Duplicate barcodes – Can the same barcode be linked to multiple products, or should it be unique? Identified duplicate barcode values in our dataset.
3. Receipts with totalSpent = 0.0 – Are these valid cases (e.g., refunds/cancelled transactions), or do they indicate any missing data?
4. Missing itemPrice and finalPrice – Could this impact the pointsearned/bonuspointsearned calculations. Should these fields be not null?

Recommendations to address the issues:

1. Implement data validation rules (NOT NULL, UNIQUE) to the required data fields to prevent future inconsistencies.
2. Confirm if the brandCode is critical for mapping or alternately we can use brandId.
3. Clarification on barcode uniqueness to determine if duplicates are expected.
4. Validate if the fields like totalSpend, ItemPrice, FinalPrice can be null or 0.

Production scaling concerns:
As the production data gets scaled, having too many null or missing values may slow down the queries. We need to make sure to select proper secondary indexes to optimize query performance.
Check for possibilities to implement data validation rules at the source to filter out the missing, invalid and duplicate values.

Please let me know for any questions and thoughts.
I can schedule a quick sync up call if required.
Thank you,
Deepeka