Process followed for the analysis of fetch AE assignment dataset

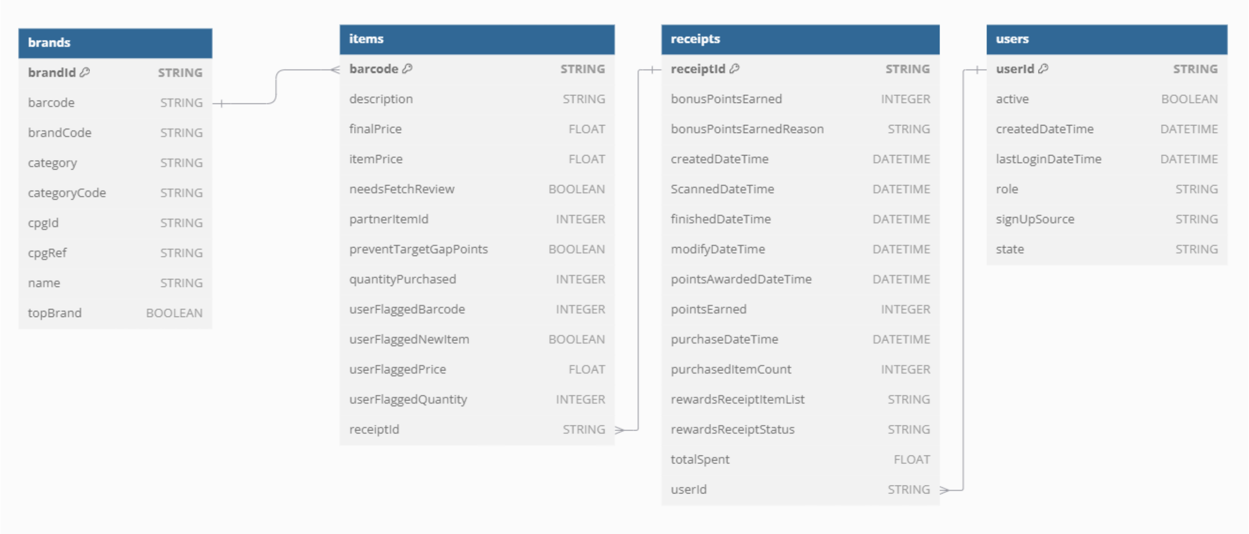
At the beginning , I have looked at the data set and realized these are mongodb json files with three datasets , after close examination , I found that receipts has nested data inside , so I thought this could be divided into 2 making the over all count to 4 datasets, which are

- Users
- Brands
- Receipts
- Items

I decided to use python to convert the json files to csv files since GCP BIG QUERY can import these files with ease, additionally I have use python to do a basic data profiling task using the library ProfileReport from ydata-profiling library, This profilereport creates html files about the profile of the dataset , this report includes dataset statistics, Distribution of values , missing values etc., it is used to determines the structure of the date , detect patterns.

once I have the overall idea of how data looked like , I have exported them converted csv files from json to BIGQUERY

Here is the ERD that can be used to represent the data



brands	
brandId	STRING
barcode	STRING
brandCode	STRING
category	STRING
categoryCode	STRING
cpgId	STRING
cpgRef	STRING
name	STRING
topBrand	BOOLEAN

items	
barcode	STRING
description	STRING
finalPrice	FLOAT
itemPrice	FLOAT
needsFetchReview	BOOLEAN
partnerItemId	INTEGER
preventTargetGapPoints	BOOLEAN
quantityPurchased	INTEGER
userFlaggedBarcode	INTEGER
userFlaggedNewItem	BOOLEAN
userFlaggedPrice	FLOAT
userFlaggedQuantity	INTEGER
receiptId	STRING

receipts	
receiptId	STRING
bonusPointsEarned	INTEGER
bonusPointsEarnedReason	STRING
createdDateTime	DATETIME
ScannedDateTime	DATETIME
finishedDateTime	DATETIME
modifyDateTime	DATETIME
pointsAwardedDateTime	DATETIME
pointsEarned	INTEGER
purchaseDateTime	DATETIME
purchasedItemCount	INTEGER
rewardsReceiptItemList	STRING
rewardsReceiptStatus	STRING
totalSpent	FLOAT
userId	STRING

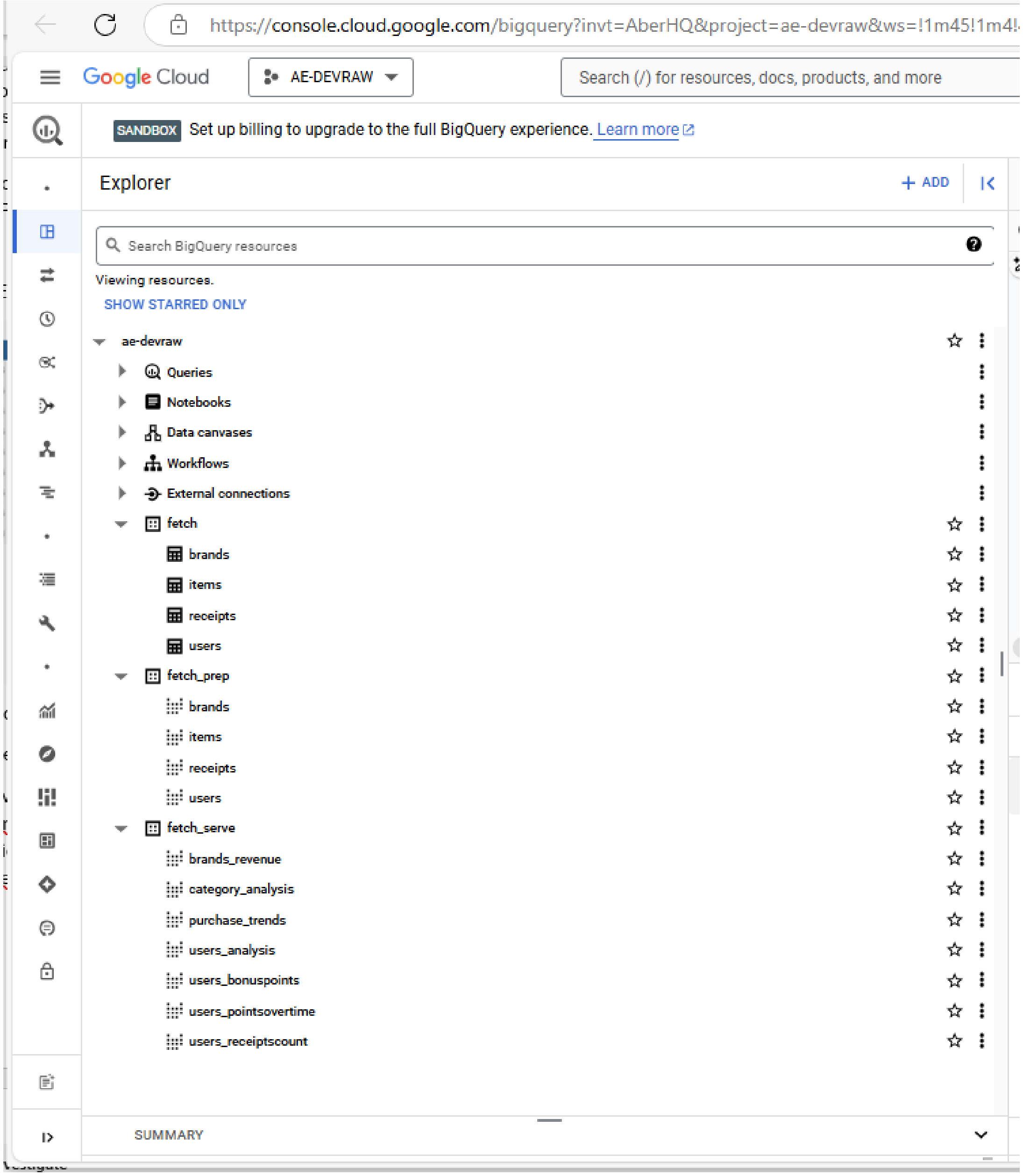
users	
userId	STRING
active	BOOLEAN
createdDateTime	DATETIME
lastLoginDateTime	DATETIME
role	STRING
signUpSource	STRING
state	STRING

Here , I thought of following medallion design in the Big query Datalake

So , I have created three datasets

- Fetch (where data is brought in as is)
- Fetch_prep (here, the transformation takes place including de duplication , datatypes conversion and column names changes etc)
- Fetch_serve (once the cleaned data is available in prep , we do the aggregation and analysis in this layer)

The layers can be seen below



https://console.cloud.google.com/bigquery?invt=AberHQ&project=ae-devraw&ws=!1m45!1m4!

Google Cloud   AE-DEVRAW

Search (/) for resources, docs, products, and more

SANDBOX Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer   + ADD

Search BigQuery resources

Viewing resources.

SHOW STARRED ONLY

- ae-devraw
  - Queries
  - Notebooks
  - Data canvases
  - Workflows
  - External connections
  - fetch
    - brands
    - items
    - receipts
    - users
  - fetch_prep
    - brands
    - items
    - receipts
    - users
  - fetch_serve
    - brands_revenue
    - category_analysis
    - purchase_trends
    - users_analysis
    - users_bonuspoints
    - users_pointsovertime
    - users_receiptscount

SUMMARY

Following link should take you to the dataset in Big query

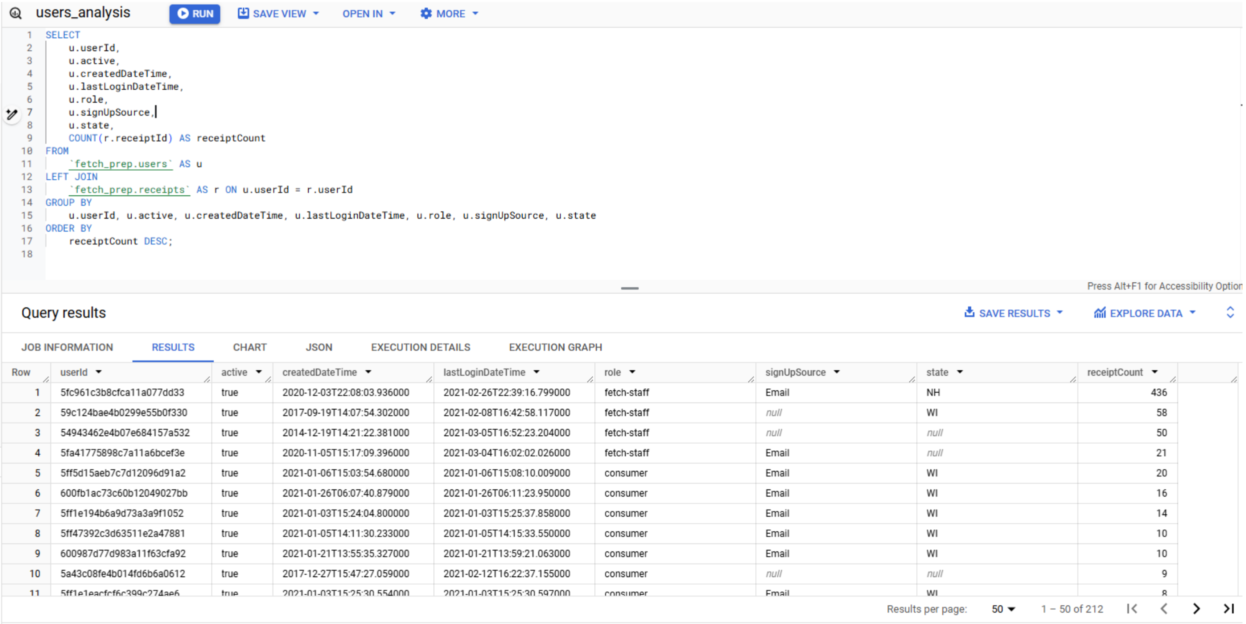
fetch :
https://console.cloud.google.com/bigquery?ws=!1m4!1m3!3m2!1sae-devraw!2sfetch
fetch_prep:
https://console.cloud.google.com/bigquery?ws=!1m4!1m3!3m2!1sae-devraw!2sfetch_prep
fetch_serve:
https://console.cloud.google.com/bigquery?ws=!1m4!1m3!3m2!1sae-devraw!2sfetch_prep

I have attached the data mapping excel file (named Mapping_Datatypes.xlsx) to the repository wherever datatype and other changes were done.

I have used the queries in analysis to draw insights in Serve dataset.

User Analysis : `SELECT`

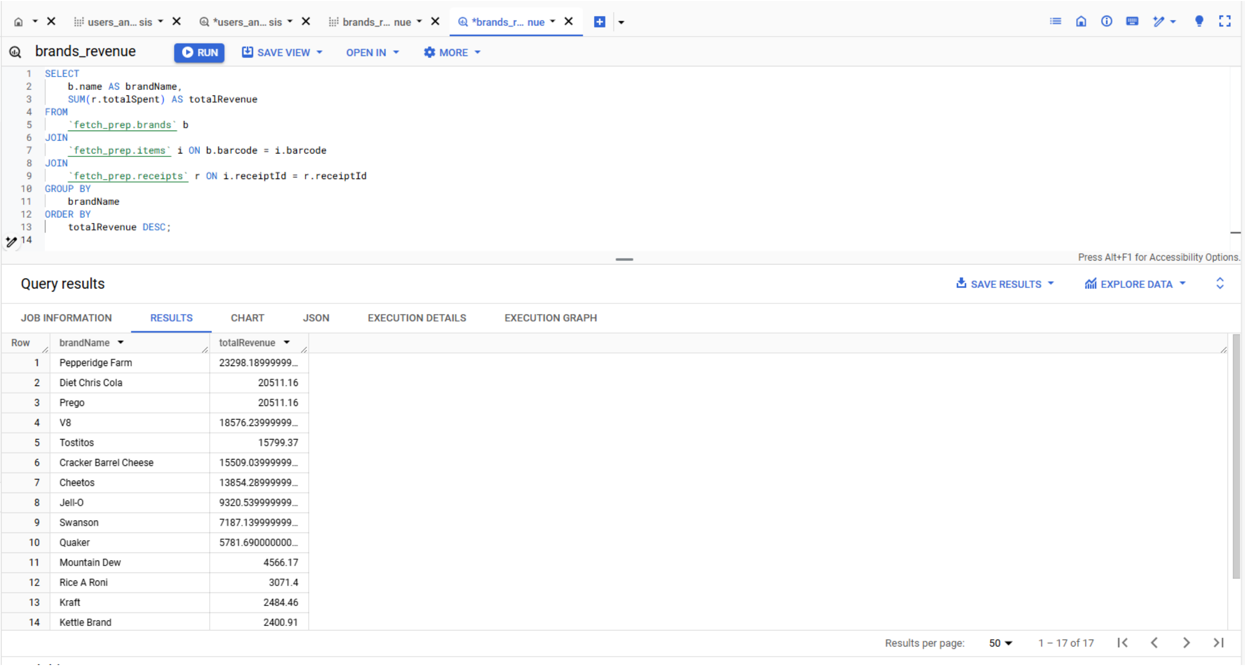
```
    u.userId,
    u.active,
    u.createdDateTime,
    u.lastLoginDateTime,
    u.role,
    u.signUpSource,
    u.state,
    COUNT(r.receiptId) AS receiptCount
FROM
    `fetch_prep.users` AS u
LEFT JOIN
    `fetch_prep.receipts` AS r ON u.userId = r.userId
GROUP BY
    u.userId, u.active, u.createdDateTime, u.lastLoginDateTime, u.role,
u.signUpSource, u.state
ORDER BY
    receiptCount DESC;
```



```
SELECT
    u.userId,
    u.active,
    u.createdDateTime,
    u.lastLoginDateTime,
    u.role,
    u.signUpSource,
    u.state,
    COUNT(r.receiptId) AS receiptCount
FROM
    `fetch_prep.users` AS u
LEFT JOIN
    `fetch_prep.receipts` AS r ON u.userId = r.userId
GROUP BY
    u.userId, u.active, u.createdDateTime, u.lastLoginDateTime, u.role, u.signUpSource, u.state
ORDER BY
    receiptCount DESC;
```

Row	userId	active	createdDateTime	lastLoginDateTime	role	signUpSource	state	receiptCount
1	5fc961c3b8cfca11a077dd33	true	2020-12-03T22:08:03.936000	2021-02-26T22:39:16.799000	fetch-staff	Email	NH	436
2	59c124bae4b0299e55b0f330	true	2017-09-19T14:07:54.302000	2021-02-08T16:42:58.117000	fetch-staff	*null*	WI	58
3	54943462e4b07e684157a532	true	2014-12-19T14:21:22.381000	2021-03-05T16:52:23.204000	fetch-staff	*null*	*null*	50
4	5fa41775898c7a11a6bcef3e	true	2020-11-05T15:17:09.396000	2021-03-04T16:02:02.026000	fetch-staff	Email	*null*	21
5	5ff5d15aeb7c7d12096d91a2	true	2021-01-06T15:03:54.680000	2021-01-06T15:08:10.009000	consumer	Email	WI	20
6	600fb1ac73c60b12049027bb	true	2021-01-26T06:07:40.879000	2021-01-26T06:11:23.950000	consumer	Email	WI	16
7	5ff1e194b6a9d73a3a9f1052	true	2021-01-03T15:24:04.800000	2021-01-03T15:25:37.858000	consumer	Email	WI	14
8	5ff47392c3d63511e2a47881	true	2021-01-05T14:11:30.233000	2021-01-05T14:15:33.550000	consumer	Email	WI	10
9	600987d77d983a11f63cfa92	true	2021-01-21T13:55:35.327000	2021-01-21T13:59:21.063000	consumer	Email	WI	10
10	5a43c08fe4b014fd6b6a0612	true	2017-12-27T15:47:27.059000	2021-02-12T16:22:37.155000	consumer	*null*	*null*	9
11	5ff1e1eacfcf6c399c274ae6	true	2021-01-03T15:25:30.554000	2021-01-03T15:25:30.597000	consumer	Email	WI	8

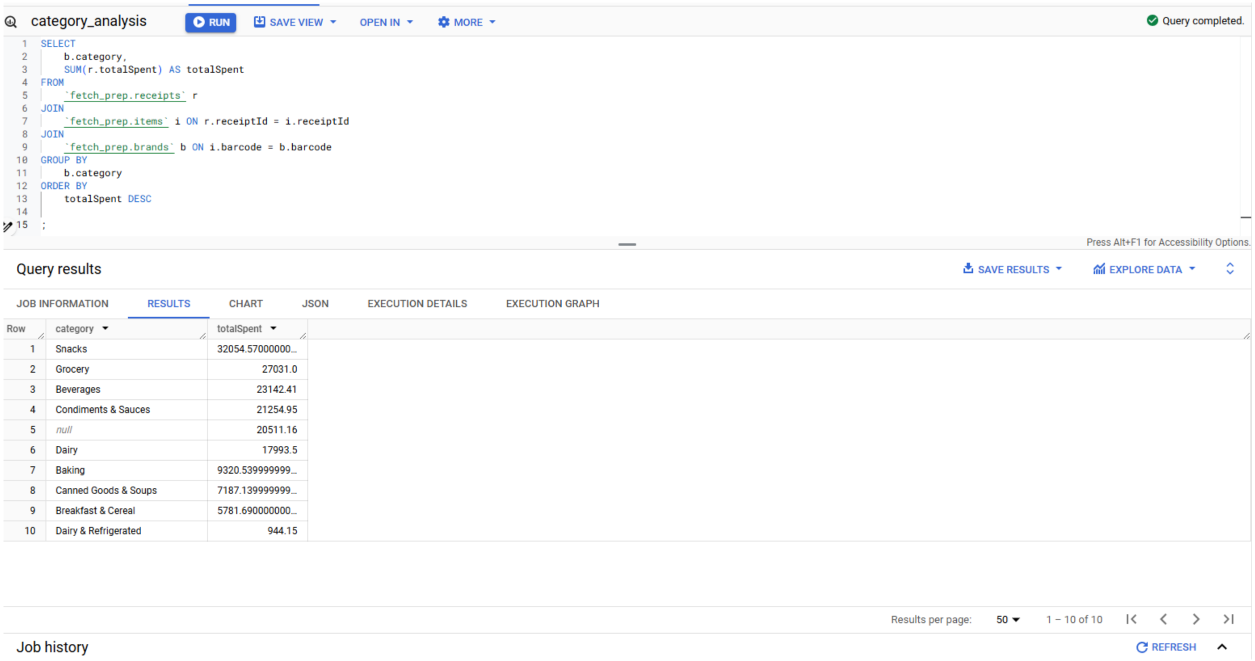
2)Brand Revenue



```sql
SELECT
    b.name AS brandName,
    SUM(r.totalSpent) AS totalRevenue
FROM
    `fetch_prep.brands` b
JOIN
    `fetch_prep.items` i ON b.barcode = i.barcode
JOIN
    `fetch_prep.receipts` r ON i.receiptId = r.receiptId
GROUP BY
    brandName
ORDER BY
    totalRevenue DESC;
```

Row	brandName	totalRevenue
1	Pepperidge Farm	23298.18999999...
2	Diet Chris Cola	20511.16
3	Prego	20511.16
4	V8	18576.23999999...
5	Tostitos	15799.37
6	Cracker Barrel Cheese	15509.03999999...
7	Cheetos	13854.28999999...
8	Jell-O	9320.539999999...
9	Swanson	7187.139999999...
10	Quaker	5781.690000000...
11	Mountain Dew	4566.17
12	Rice A Roni	3071.4
13	Kraft	2484.46
14	Kettle Brand	2400.91

3) Category Analysis

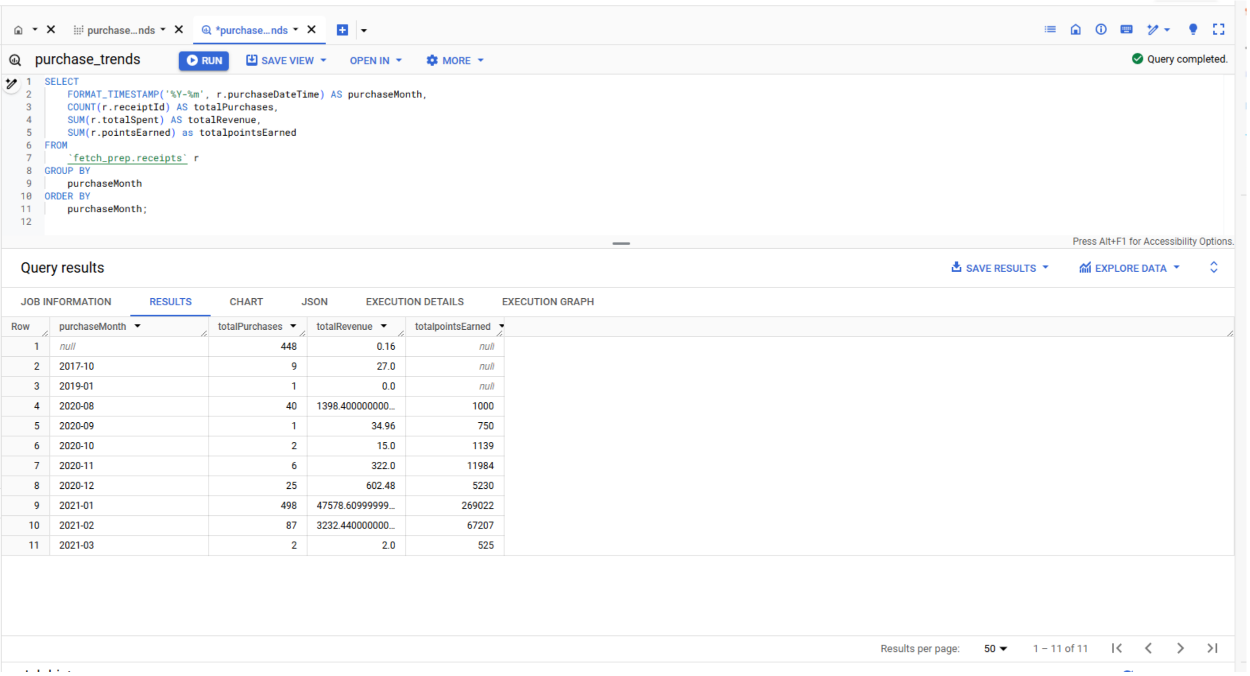


```sql
SELECT
    b.category,
    SUM(r.totalSpent) AS totalSpent
FROM
    `fetch_prep.receipts` r
JOIN
    `fetch_prep.items` i ON r.receiptId = i.receiptId
JOIN
    `fetch_prep.brands` b ON i.barcode = b.barcode
GROUP BY
    b.category
ORDER BY
    totalSpent DESC

;
```

Row	category	totalSpent
1	Snacks	32054.57000000...
2	Grocery	27031.0
3	Beverages	23142.41
4	Condiments & Sauces	21254.95
5	*null*	20511.16
6	Dairy	17993.5
7	Baking	9320.539999999...
8	Canned Goods & Soups	7187.139999999...
9	Breakfast & Cereal	5781.690000000...
10	Dairy & Refrigerated	944.15

4) Purchase Trends over time

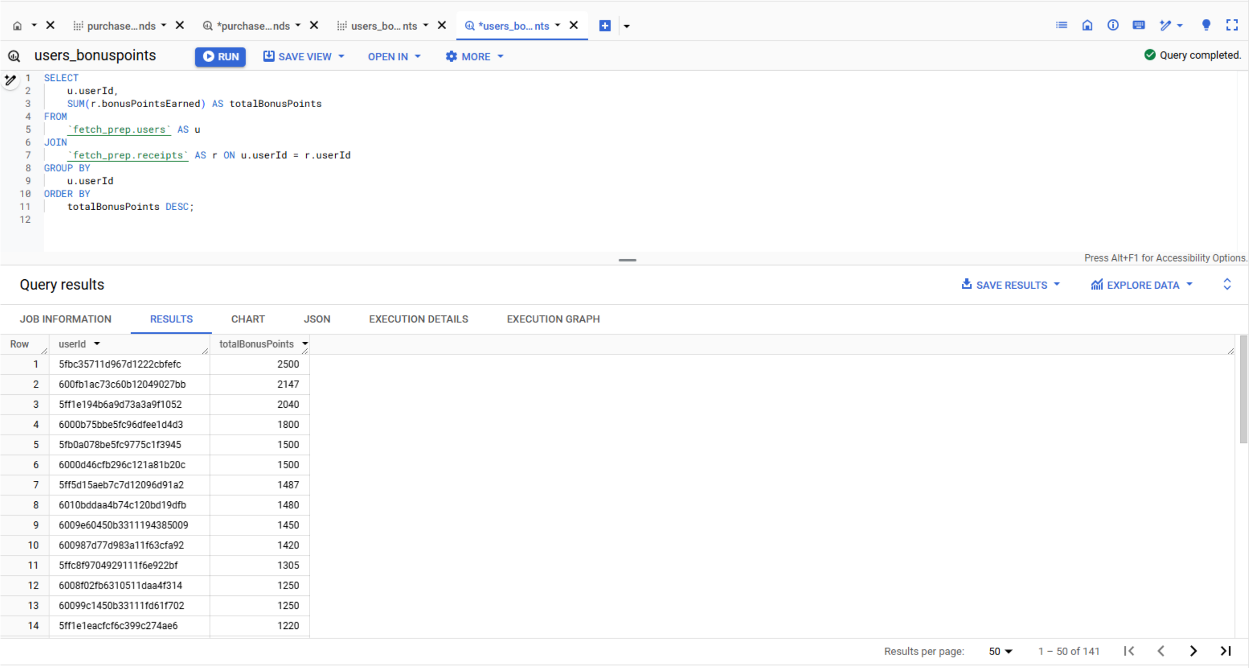


```sql
SELECT
    FORMAT_TIMESTAMP('%Y-%m', r.purchaseDateTime) AS purchaseMonth,
    COUNT(r.receiptId) AS totalPurchases,
    SUM(r.totalSpent) AS totalRevenue,
    SUM(r.pointsEarned) as totalpointsEarned
FROM
    `fetch_prep.receipts` r
GROUP BY
    purchaseMonth
ORDER BY
    purchaseMonth;
```

Row	purchaseMonth	totalPurchases	totalRevenue	totalpointsEarned
1	null	448	0.16	null
2	2017-10	9	27.0	null
3	2019-01	1	0.0	null
4	2020-08	40	1398.400000000...	1000
5	2020-09	1	34.96	750
6	2020-10	2	15.0	1139
7	2020-11	6	322.0	11984
8	2020-12	25	602.48	5230
9	2021-01	498	47578.60999999...	269022
10	2021-02	87	3232.440000000...	67207
11	2021-03	2	2.0	525

5)

User bonus points



```sql
SELECT
    u.userId,
    SUM(r.bonusPointsEarned) AS totalBonusPoints
FROM
    `fetch_prep.users` AS u
JOIN
    `fetch_prep.receipts` AS r ON u.userId = r.userId
GROUP BY
    u.userId
ORDER BY
    totalBonusPoints DESC;
```

Row	userId	totalBonusPoints
1	5fbc35711d967d1222cbfefc	2500
2	600fb1ac73c60b12049027bb	2147
3	5ff1e194b6a9d73a3a9f1052	2040
4	6000b75bbe5fc96dfee1d4d3	1800
5	5fb0a078be5fc9775c1f3945	1500
6	6000d46cfb296c121a81b20c	1500
7	5ff5d15aeb7c7d12096d91a2	1487
8	6010bddaa4b74c120bd19dfb	1480
9	6009e60450b3311194385009	1450
10	600987d77d983a11f63cfa92	1420
11	5ffc8f9704929111f6e922bf	1305
12	6008f02fb6310511daa4f314	1250
13	60099c1450b33111fd61f702	1250
14	5ff1e1eacfcf6c399c274ae6	1220

Further , these can be used in a Visualization platform like Power BI , Tableau and looker to show more insightful data , this attempt is just showing the overall process outline we usually follow.

Thank you very for your consideration !!!