

This document outlines the analysis on Amazon.Listings dataset for Sprint 2.

Data Summary:

```
SELECT count(*) FROM `bigqueryexport-183608.amazon.listings`;
```

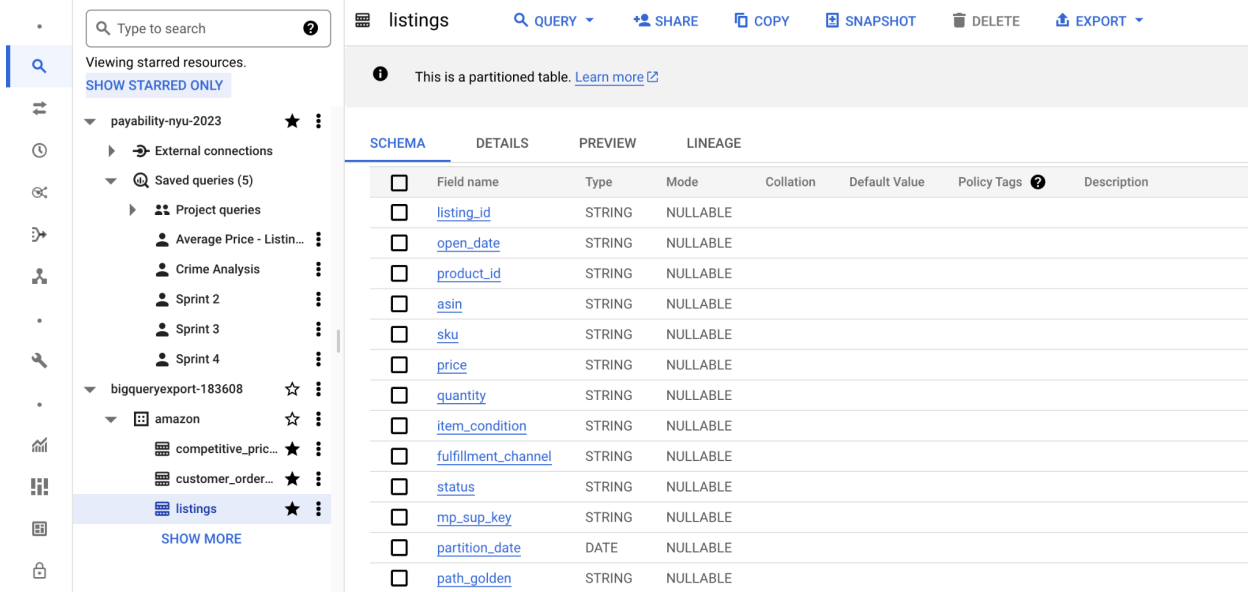
Total no. of observations: 2724431928

```
SELECT * FROM `bigqueryexport-183608.amazon.listings` LIMIT 20;
```

Total no. of features: 13

Key features:

No description provided on the table schema.



Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
listing_id	STRING	NULLABLE				
open_date	STRING	NULLABLE				
product_id	STRING	NULLABLE				
asin	STRING	NULLABLE				
sku	STRING	NULLABLE				
price	STRING	NULLABLE				
quantity	STRING	NULLABLE				
item_condition	STRING	NULLABLE				
fulfilment_channel	STRING	NULLABLE				
status	STRING	NULLABLE				
mp_sup_key	STRING	NULLABLE				
partition_date	DATE	NULLABLE				
path_golden	STRING	NULLABLE				

Based on my research, this should be the *data dictionary* for this table.

1. *listing_id* : This is a unique identifier for a listing. Each row is a unique listing.
2. *open_date*: This is the data a listing was posted on amazon platform by the seller.
3. *product_id*: This is a unique identifier for a product type.
4. *asin*: This is a unique identifier for a product type given by Amazon used for inventory, sales etc. Amazon Standard Identification Number is a 10-character alphanumeric code that is used to identify and track products in the Amazon catalog.
5. *sku*: Stock Keeping Unit is also a unique identifier code used by sellers to track inventory and manage product information.

6. *price*: This is the price of a product on a listing.
7. *quantity*: This is the quantity of a particular product (asin) on a listing.
8. *item_condition*: This is the condition of a product(asin) on a listing.
9. *fulfillment_channel*: This is the channel by which a seller fulfills shipping of the product on a particular listing.
10. *Status*: This is the status of a listing.
11. *mp_sup_key*: This is a unique identifier for a seller. There are 16399 unique amazon sellers in our table.
12. *partition_date*: This field is generated by bigquery.
13. *path_golden*: This is a unique url generated for a listing.

These features can be grouped into the following categories:

1. Product Identifiers: product_id, asin, sku
2. Listing Attributes: price, item_condition, quantity
3. Listing Delivery: fulfillment_channel, and status
4. Seller Identification: mp_sup_key
5. Dates: open_date, partition_date

Key questions to answer for this dataset:

1. What is the date range for the data in this dataset?
2. What are the different types of status for a listing? What are their proportions?
3. How many types of fulfillment channels exist for a seller based on their listings? What are their proportions in terms of usage?
4. Who are the top sellers based on the number of listings?
5. What is the inventory level of a product (asin) of a particular seller?

Date Range:

```
SELECT MAX(partition_date), MIN(partition_date)
FROM `bigqueryexport-183608.amazon.listings`;
```

Data exists from 2019-09-20 to 2023-03-01 based on the partition date.

```
SELECT MIN(SAFE_CAST(LEFT(open_date,10) AS DATE)) AS min_listing_date, MAX(SAFE_CAST(LEFT(open_date,10) AS
DATE)) AS max_listing_date
FROM `bigqueryexport-183608.amazon.listings`;
```

Based on the open_date field, listing data sellers exist from **2002-12-13 to 2023-03-01**.

This makes sense because we are looking at the listing data for an amazon seller. This is not related to whether they are a payability client. It is possible for a seller to have been a client for a period of time or not.

Distinct values in status column:

```
SELECT distinct status
FROM `bigqueryexport-183608.amazon.listings`
order by status;
```

Total distinct values: 19

Row	status
1	null
2	
3	0
4	ADAC Template
5	AMAZON_NA
6	Active
7	CA
8	DEFAULT
9	Default Amazon Template
10	Free Shipping US
11	Inactive
12	Incomplete
13	Migrated Template
14	Migrated Template-FBM
15	NO Prime
16	Tees Template
17	US Economy Template
18	US prime MIA CA MJ
19	wallpaper template

```
SELECT distinct status
FROM `bigqueryexport-183608.amazon.listings`
WHERE status NOT IN ('0','')
AND status IS NOT NULL
ORDER BY status;
```

Removed these values from Status column: 0, "" and null.

Total distinct values: 16

Row	status
1	ADAC Template
2	AMAZON_NA
3	Active
4	CA
5	DEFAULT
6	Default Amazon Template
7	Free Shipping US
8	Inactive
9	Incomplete
10	Migrated Template
11	Migrated Template-FBM
12	NO Prime
13	Tees Template
14	US Economy Template
15	US prime MIA CA MJ
16	wallpaper template

The proportions of each of these statuses are as follows:

```
SELECT status, count(listing_id)
FROM `bigqueryexport-183608.amazon.listings`
WHERE status NOT IN ('0','')
AND status IS NOT NULL
GROUP BY status
ORDER BY count(listing_id) desc;
```

Row	status	f0_
1	Active	1475885767
2	Inactive	1133551900
3	Incomplete	115005467
4	DEFAULT	78
5	Migrated Template	54
6	AMAZON_NA	11
7	Migrated Template-FBM	9
8	wallpaper template	7
9	Tees Template	3
10	ADAC Template	2
11	Free Shipping US	2
12	NO Prime	1
13	CA	1
14	US Economy Template	1
15	Default Amazon Template	1
16	US prime MIA CA MJ	1

Most of the listings are under 'Active', 'Inactive', and 'Incomplete' status. Therefore, I am going to focus further analysis on these three values.

Distinct values in fulfillment_channel column:

```
SELECT distinct fulfillment_channel
FROM `bigqueryexport-183608.amazon.listings`
ORDER BY fulfillment_channel;
```

Total distinct values: 12

Row	fulfillment_channel
1	null
2	
3	0
4	AMAZON_JP
5	AMAZON_NA
6	Active
7	DEFAULT
8	Default Template
9	FBM - Ali Sourcing
10	FREE SHIPPING
11	Incomplete
12	Migrated Template

Removed these values from fulfillment_channel column: {0,'} and null values

```
SELECT distinct fulfillment_channel
FROM `bigqueryexport-183608.amazon.listings`
WHERE fulfillment_channel NOT IN ('0','')
AND fulfillment_channel IS NOT NULL
ORDER BY fulfillment_channel;
```

Total distinct values: 9

Row	fulfillment_channel
1	AMAZON_JP
2	AMAZON_NA
3	Active
4	DEFAULT
5	Default Template
6	FBM - Ali Sourcing
7	FREE SHIPPING
8	Incomplete
9	Migrated Template

Based on our conversation with the Payability team, the expectation was that there could only be two channels for fulfillment - Amazon or Merchant. Therefore, it is unclear what some of these column values refer to. For example : DEFAULT, Migrated Template. Also, Incomplete seems like a status rather than a fulfillment channel.

```
SELECT fulfillment_channel, count(listing_id)
FROM `bigqueryexport-183608.amazon.listings`
WHERE fulfillment_channel NOT IN ('0','')
AND fulfillment_channel IS NOT NULL
GROUP BY fulfillment_channel
ORDER BY count(listing_id) desc;
```

Row	fulfillment_channel	f0_
1	DEFAULT	2287057634
2	AMAZON_NA	437385480
3	Migrated Template	215
4	Active	24
5	AMAZON_JP	20
6	FREE SHIPPING	8
7	Default Template	2
8	Incomplete	1
9	FBM - Ali Sourcing	1

DEFAULT and AMAZON_NA are the two main fulfillment channels used by sellers for their product listings.

Relationship between fulfillment channel and status feature:

For my analysis, I am going to take these three statuses into consideration:

Active: This means that the listing is currently active on Amazon marketplace.

Inactive: This means that the listing is currently inactive on Amazon marketplace.

Incomplete: This could mean that the listing was incomplete. I am not sure at what part of the fulfillment process a listing is marked incomplete.

There is no description about these values present on the schema.

Next step is to find the number of these statuses based on their fulfillment channels. Below query only looks at one particular seller id but the goal is to extend this to include all sellers. This was done to tackle lag issue for the looker studio dashboard.

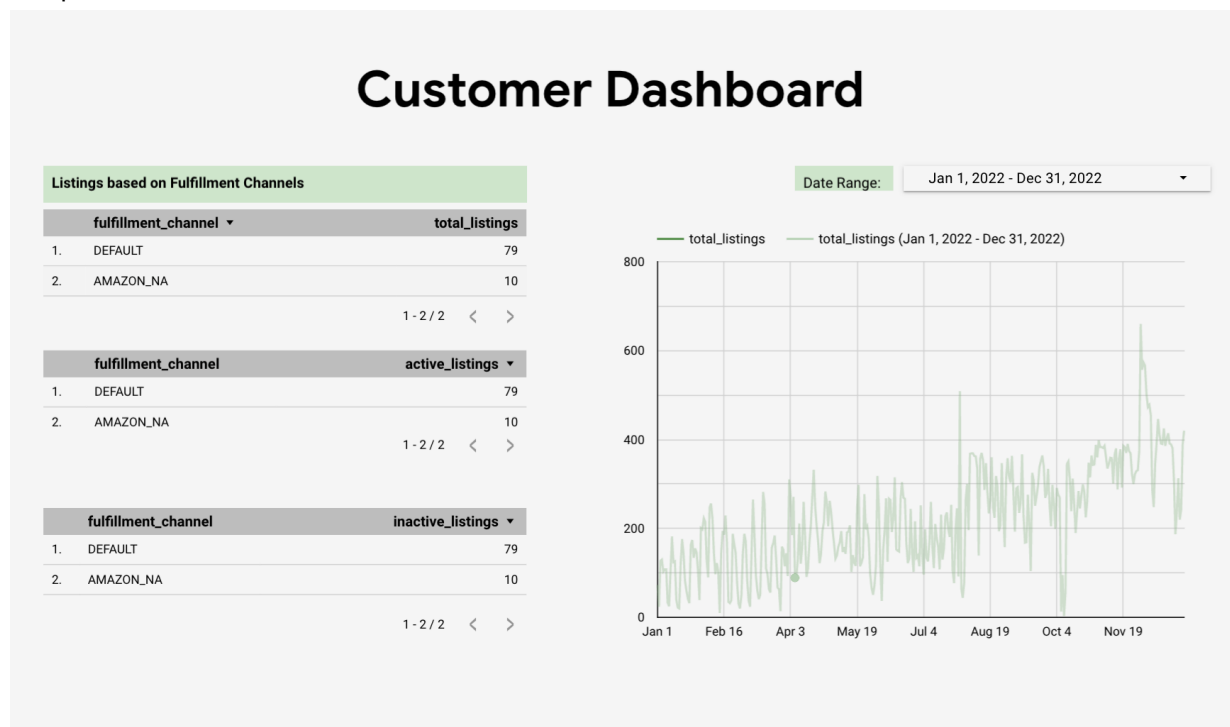
```

2 SELECT
3   open_date,
4   fulfillment_channel,
5   COUNT(*) as total_listings,
6   COUNT(IF(status='Active', 1, NULL)) as active_listings,
7   COUNT(IF(status='Inactive', 1, NULL)) as inactive_listings,
8   COUNT(IF(status='Incomplete', 1, NULL)) as incomplete_listings
9 FROM
10  `bigqueryexport-183608.amazon.listings`
11 WHERE
12   open_date IS NOT NULL
13   AND fulfillment_channel NOT IN ( '0', '' )
14   AND status NOT IN ( '0', '' )
15   AND mp_sup_key = '0f191dec-c666-4060-b001-a373f4899d72'
16 GROUP BY
17   open_date,
18   fulfillment_channel
19 ORDER BY
20   open_date DESC;

```

Row	open_date	fulfillment_channel	total_listings	active_listings	inactive_listings	incomplete_listings
1	2023-01-14 18:59:35 PST	DEFAULT	201	195	6	0
2	2023-01-14 18:54:59 PST	DEFAULT	239	235	4	0
3	2023-01-14 18:51:08 PST	DEFAULT	237	229	8	0
4	2023-01-14 18:46:03 PST	DEFAULT	207	203	4	0
5	2023-01-14 18:42:53 PST	DEFAULT	225	215	10	0
6	2023-01-14 18:38:31 PST	DEFAULT	221	215	6	0

Sample Dashboard for this client:



There is an issue with this dashboard because for all the different fulfillment_channels, it is displaying the same value which needs to be resolved. This will also be expanded to include all sellers in the next sprint.

Top seller analysis

Naive analysis: For all the seller ids, get the total number of listings based on the row count.

```
SELECT
    mp_sup_key,
    count(listing_id) AS NUMBER_OF_LISTINGS
FROM `bigqueryexport-183608.amazon.listings`
WHERE
    status NOT IN ('0','')
    AND status IS NOT NULL
    AND fulfillment_channel NOT IN ('0','')
    AND fulfillment_channel IS NOT NULL
    AND item_condition IS NOT NULL
    AND item_condition <> ''
GROUP BY mp_sup_key
ORDER BY COUNT(listing_id) DESC;
```

Row	mp_sup_key	NUMBER_OF_LISTINGS
1	0f191dec-c666-4060-b001-a373f4899d72	196894600
2	a2dc9fce-4b96-4deb-9cea-31604b2123d1	110169190
3	a9f4c082-2840-4904-a987-8d563dc37900	105593128
4	21e09a81-3a0d-4149-8474-827016716c85	90015143
5	0f8df38f-5334-4587-bfdb-59780e3e83c6	63374852
6	38d49093-a72a-448d-9c06-28f653ce1fb9	48030200
7	53520f0a-9f27-47fc-9389-d4c9a4a503c6	47885973
8	a10ae09e-eeb3-44d8-b7aa-1b281fe382a2	37598016
9	df0c1eab-4688-481c-9329-ec8a3d10261a	35582613
10	04bd940e-a95a-4cd9-ab91-7c2039e932e7	34119718
11	c09ef359-495c-4b13-acb0-532297804c51	28854795

The numbers look inflated so we have to verify the min and max opne_date values.

```
SELECT mp_sup_key, count(listing_id) AS NUMBER_OF_LISTINGS,
MIN(SAFE_CAST(LEFT(open_date,10) AS DATE)) AS min_listing_date,
MAX(SAFE_CAST(LEFT(open_date,10) AS DATE)) AS max_listing_date
FROM `bigqueryexport-183608.amazon.listings`
WHERE status NOT IN ('0','') AND status IS NOT NULL
AND fulfillment_channel NOT IN ('0','') AND fulfillment_channel IS NOT NULL
AND item_condition IS NOT NULL AND item_condition <> ''
GROUP BY mp_sup_key
ORDER BY COUNT(listing_id) DESC;
```


Row	mp_sup_key	NUMBER_OF_LISTINGS	min_listing_date	max_listing_date
1	0f191dec-c666-4060-b001-a373f4899d72	196894600	2017-02-23	2023-01-14
2	a2dc9fce-4b96-4deb-9cea-31604b2123d1	110169190	2011-02-03	2020-06-25
3	a9f4c082-2840-4904-a987-8d563dc37900	105593128	2014-07-15	2022-06-13
4	21e09a81-3a0d-4149-8474-827016716c85	90015143	2015-03-09	2023-03-22
5	0f8df38f-5334-4587-bfdb-59780e3e83c6	63374852	2015-02-13	2023-03-25
6	38d49093-a72a-448d-9c06-28f653ce1fb9	48030200	2021-02-14	2022-06-02
7	53520f0a-9f27-47fc-9389-d4c9a4a503c6	47885973	2018-09-25	2022-07-23
8	a10ae09e-eeb3-44d8-b7aa-1b281fe382a2	37598016	2020-03-20	2021-11-14
9	df0c1eab-4688-481c-9329-ec8a3d10261a	35582613	2018-01-22	2020-05-19
10	04bd940e-a95a-4cd9-ab91-7c2039e932e7	34119718	2020-03-09	2021-11-16
11	c09ef359-495c-4b13-acb0-532297804c51	28854795	2013-08-05	2022-10-17

One clear indicator is that in #6, this seller started listing items on amazon in 2021. It is impossible for them to have 48,030,200 listings in over 2 years.

Because there are a lot of duplicate values in this table, the next logical step is to group listings by asin and remove the partition_date field. We do so because this export could get the same data for different partition dates.

```
SELECT
  mp_sup_key,
  asin,
  count(listing_id) AS number_of_listings
FROM `bigqueryexport-183608.amazon.listings`
WHERE
  status NOT IN ('0','') AND status IS NOT NULL
  AND fulfillment_channel NOT IN ('0','') AND fulfillment_channel IS NOT NULL
  AND item_condition IS NOT NULL AND item_condition <> ''
  AND mp_sup_key IS NOT NULL
GROUP BY mp_sup_key, asin
ORDER BY count(listing_id) desc;
```


Row	mp_sup_key	asin	number_of_listings
1	2d9dfd12-172a-4b2c-b862-49480a2742e8	B0013FRNKG	34320
2	2d9dfd12-172a-4b2c-b862-49480a2742e8	B0047DVWLW	27360
3	53520f0a-9f27-47fc-9389-d4c9a4a503c6	B08XPRG8BK	23500
4	f853b8db-76ac-4d23-9b68-95872a06c038	B072LB9W8T	18502
5	2d9dfd12-172a-4b2c-b862-49480a2742e8	B001I907I2	17680
6	53520f0a-9f27-47fc-9389-d4c9a4a503c6	B09NJ52V7H	17506
7	41929434-5058-46db-a417-3d5e1e15c851	B09NVKZ1C9	17206
8	f853b8db-76ac-4d23-9b68-95872a06c038	B002RUJB3S	16731
9	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B00YB25ERC	15745
10	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B001CTN3C0	14982


When I looked up the first product 'B0013FRNKG' on amazon, it was unavailable.

amazon prime Deliver to Surabhi New York 10019 All B0013FRNKG


EN Hello, Surabhi Account & Lists Returns & Orders Cart

All Amazon Health Buy Again Smart Home Prime Video Groceries Coupons Health & Household Find a Gift Handmade Pet Supplies Home Improvement Shop women-owned businesses







Butane for Refillable Lighters - No Impurities
Shop Premium Butane 99.999% Pure



Colibri Premium Butane Fuel Refill for Lighters, 300ml (10.1 fl oz) Cans, Pack of 3, Butane Torc...
★★★★★ 1,437
✓prime



Colibri Premium Butane Fuel Refill for Lighters, 300ml (10.1 fl oz) Cans, Pack of 2, Butane Torc...
★★★★★ 2,435
✓prime



Colibri Premium Butane Fuel Refill for Lighters, 300ml (10.1 fl oz) Cans, Pack of 1, Butane Torc...
★★★★★ 875
✓prime

No results for B0013FRNKG.
Try checking your spelling or use more general terms

Sponsored

Row	mp_sup_key	asin	number_of_listings	min_listing_date	max_listing_date
1	2d9dfd12-172a-4b2c-b862-49480a2742e8	B0013FRNKG	34320	2011-06-29	2015-12-13
2	2d9dfd12-172a-4b2c-b862-49480a2742e8	B0047DVVLW	27360	2011-07-14	2015-12-21
3	53520f0a-9f27-47fc-9389-d4c9a4a503c6	B08XPRG8BK	23500	2021-02-28	2021-03-01
4	f853b8db-76ac-4d23-9b68-95872a06c038	B072LB9W8T	18502	2018-07-06	2020-06-12
5	2d9dfd12-172a-4b2c-b862-49480a2742e8	B001907I2	17680	2011-06-29	2015-11-29
6	53520f0a-9f27-47fc-9389-d4c9a4a503c6	B09NJ52V7H	17506	2021-12-12	2021-12-21
7	41929434-5058-46db-a417-3d5e1e15c851	B09NVKZ1C9	17206	2021-12-19	2022-01-22
8	f853b8db-76ac-4d23-9b68-95872a06c038	B002RUJB3S	16731	2017-09-18	2021-02-11
9	b80e94c4-c376-413a-88b4-e2a1dd980d9	B00YB25ERC	15745	2021-05-01	2023-02-07
10	b80e94c4-c376-413a-88b4-e2a1dd980d9	B001CTN3C0	14982	2019-07-05	2023-02-08

This makes sense because the listing was open between 2011 and 2015. To avoid this issue, we can order by the min date (descending) and then by the count.

Row	mp_sup_key	asin	number_of_listings
1	7858f7c2-fde6-429a-b36e-6967071bd0bb	B01DPA6OCO	1

Even though I was able to find the product on Amazon now, now it is difficult to get the total count.

amazon prime Deliver to Surabhi New York 10019 All B01DPA6OCO


EN Hello, Surabhi Account & Lists Returns & Orders Cart

All Amazon Health Buy Again Smart Home Prime Video Groceries Coupons Health & Household Find a Gift Handmade Pet Supplies Home Improvement Subscribe & Save Shop women-owned businesses


1 result for "B01DPA6OCO" Sort by: Featured

Department


- Online Learning
- Alexa Skills
- Amazon Devices & Accessories
- Appliances
- Apps & Games
- Arts, Crafts & Sewing
- Audible Books & Originals
- Automotive
- Baby Products
- Beauty & Personal Care
- Books
- CDs & Vinyl
- Cell Phones & Accessories
- Clothing, Shoes & Jewelry
- Collectibles & Fine Art
- Credit & Payment Cards
- Digital Educational Resources
- Digital Music
- Electronics
- Everything Else
- Gift Cards
- Grocery & Gourmet Food
- Handmade Products
- Health & Household
- Home & Business Services
- Home & Kitchen
- Industrial & Scientific
- Kindle Store
- Magazine Subscriptions
- Movies & TV
- Musical Instruments
- Office Products




We're the water experts so you don't have to be
Shop Brio >



Brio Self Cleaning Bottom Loading Water Cooler Water Dispenser - Limited Edition - 3 Temperat...
★★★★★ 5,144
✓prime



Brio Bottom Loading Water Cooler Water Dispenser - Essential Series - 3 Temperature Set...
★★★★★ 4,000
✓prime



L'Oréal Paris Colour Riche Eye Pocket Palette Eye Shadow, Voilet Amour, 0.1 oz.

Sponsored

Taking a look at data for last year:

```
SELECT
  mp_sup_key,
  asin,
  count(listing_id) AS number_of_listings,
  MIN(SAFE_CAST(LEFT(open_date,10) AS DATE)) AS min_listing_date
FROM `bigqueryexport-183608.amazon.listings`
WHERE
  status NOT IN ('0','') AND status IS NOT NULL
  AND fulfillment_channel NOT IN ('0','') AND fulfillment_channel IS NOT NULL
  AND item_condition IS NOT NULL AND item_condition <> ''
  AND mp_sup_key IS NOT NULL
  AND SAFE_CAST(LEFT(open_date,10) AS DATE) > '2021-12-31'
GROUP BY mp_sup_key, asin
ORDER BY count(listing_id) desc, MIN(SAFE_CAST(LEFT(open_date,10) AS DATE)) desc;
```

Row	mp_sup_key	asin	number_of_listings	min_listing_date
1	41929434-5058-46db-a417-3d5e1e15c851	B09NVKZ1C9	17201	2022-01-06
2	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B00YB25ERC	8456	2022-01-04
3	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B001CTN3C0	7299	2022-01-04
4	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B008LTMBVI	5567	2022-01-09
5	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B00FY10RMQ	5192	2022-01-09
6	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B002H3SEDI	5117	2022-01-05
7	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B000BTBUBS	4906	2022-01-04
8	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B0031TPX3Q	4443	2022-01-08
9	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B0065NGKJQ	4203	2022-01-04
10	cb860960-4bb7-4b07-a6fa-070a4d7cb3cb	B00005JM5E	3824	2022-03-16

I tried to get total listings based on seller id only but the numbers are inflated. This is something I have to look at next time.

```
SELECT DISTINCT * FROM (SELECT
  mp_sup_key,
  --asin,
  count(listing_id) AS number_of_listings,
  MIN(SAFE_CAST(LEFT(open_date,10) AS DATE)) AS min_listing_date
FROM `bigqueryexport-183608.amazon.listings`
WHERE
  status NOT IN ('0','') AND status IS NOT NULL
  AND fulfillment_channel NOT IN ('0','') AND fulfillment_channel IS NOT NULL
  AND item_condition IS NOT NULL AND item_condition <> ''
  AND mp_sup_key IS NOT NULL
  AND SAFE_CAST(LEFT(open_date,10) AS DATE) > '2021-12-31'
GROUP BY mp_sup_key
ORDER BY count(listing_id) desc, MIN(SAFE_CAST(LEFT(open_date,10) AS DATE)) desc)
```

```
ORDER BY number_of_listings desc;
```

Row	mp_sup_key	number_of_listings	min_listing_date
1	0f191dec-c666-4060-b001-a373f4899d72	101069869	2022-01-01
2	484866e3-e026-4963-a1ef-8edf561b25b4	14487910	2022-01-01
3	b558ad85-5d95-4ca6-9b25-e160ab04e733	14325460	2022-01-25
4	c09ef359-495c-4b13-acb0-532297804c51	7348048	2022-01-13
5	ed5d6081-5d64-4e0f-8f91-07f41bae41d1	6857001	2022-01-18
6	b80e94c4-c376-413a-88b4-e2a1ddd980d9	6326123	2022-01-01
7	21e09a81-3a0d-4149-8474-827016716c85	5963526	2022-01-01
8	028a1eaa-e6ba-47dc-8441-fc0bc4325f9e	5650225	2022-01-06
9	837ec6b3-8a03-47c4-8330-2f18b6f7c116	5284734	2022-01-02
10	6e4ec6f5-4cd7-421d-8663-3c7c8997f338	3420471	2022-01-04

Some things I tried that did not work for this sprint:

Calculating Inventory value:

```
SELECT
mp_sup_key,
asin,
COUNT(*) as total_listings,
(CAST('quantity' AS INT64)) as quantity
--COUNT(IF(status='Active', 1, NULL)) as active_listings,
--AVG(SAFE_CAST('price' AS FLOAT64)) as avg_price
FROM
`bigqueryexport-183608.amazon.listings`
WHERE
mp_sup_key IS NOT NULL
AND fulfillment_channel NOT IN ('0','')
AND status NOT IN ('0','')
AND price <> ''
GROUP BY
mp_sup_key,
asin,
(CAST('quantity' AS INT64))
ORDER BY
total_listings DESC, (CAST('quantity' AS INT64)) DESC;
```

The issue with this analysis was that a lot of listings are priced 99999999, or 999999989. This seems inappropriate. I have to find a way to ignore these numbers as they will inflate the average price for a listing.