This document outlines the analysis on Amazon.Listings and Amazon.Customer_Order_Metrics datasets for Sprint 4.

Customer Order Metrics:

Key features:

No description provided on the table schema.

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

1. *amazon_order_id* : This is a unique identifier for an order on Amazon Marketplace.

2. *order_status*: This is the status of an order.

3. *item_price*: This is the price of an order.

4. *quantity*: This is the quantity of an order.

5. *last_updated_date*: This is the date when this order was last updated by the seller.

6. *fulfillment_channel*: This is the fulfillment channel used in the order to send this amazon order to the customer.

7. *ship_service_level*: This is the service level of the shipping chosen for this order.

8. *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.

9. *sku*: Stock Keeping Unit is also a unique identifier code used by sellers to track inventory and manage product information.

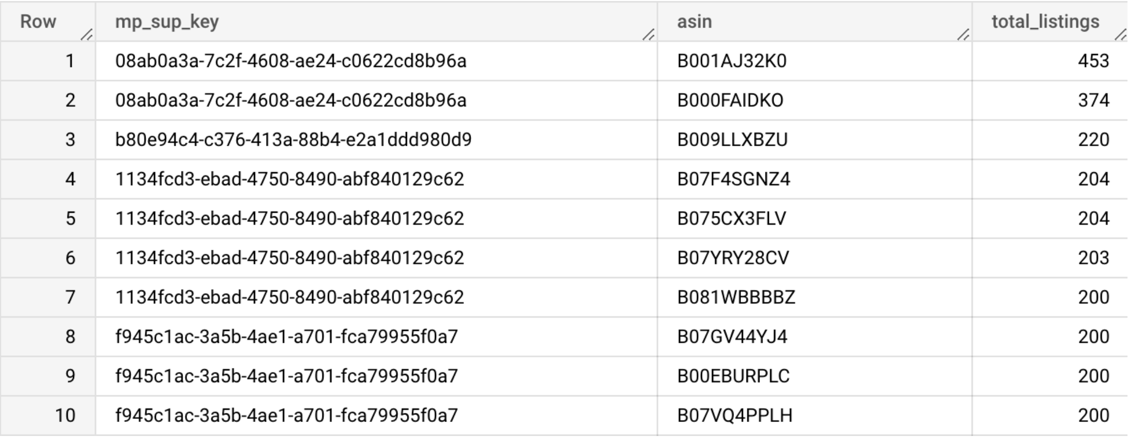
There are more features on this table but for this sprint my focus is on fulfillment_channel, ship_service_level and order_status and analyzes their relationship. This was done on looker studio and the code for my queries are on git hub. **Key findings are mentioned on the presentation deck for this week.**

Listings:

Question #1: Find total number of listings for all sellers.

How many records are present for a seller_id, ASIN combination?

```
SELECT mp_sup_key,asin,COUNT(*) as total_listings
FROM `bigqueryexport-183608.amazon.listings`
WHERE mp_sup_key IS NOT NULL AND asin IS NOT NULL
GROUP BY listing_id, mp_sup_key,asin
ORDER BY total_listings DESC;
```



Row	mp_sup_key	asin	total_listings
1	08ab0a3a-7c2f-4608-ae24-c0622cd8b96a	B001AJ32K0	453
2	08ab0a3a-7c2f-4608-ae24-c0622cd8b96a	B000FAIDKO	374
3	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B009LLXBZU	220
4	1134fcd3-ebad-4750-8490-abf840129c62	B07F4SGNZ4	204
5	1134fcd3-ebad-4750-8490-abf840129c62	B075CX3FLV	204
6	1134fcd3-ebad-4750-8490-abf840129c62	B07YRY28CV	203
7	1134fcd3-ebad-4750-8490-abf840129c62	B081WBBBBZ	200
8	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07GV44YJ4	200
9	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B00EBURPLC	200
10	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07VQ4PPLH	200

We are counting the number of listing_id for each combination and that is the total_listing.

Seller id: 08ab0a3a-7c2f-4608-ae24-c0622cd8b96a

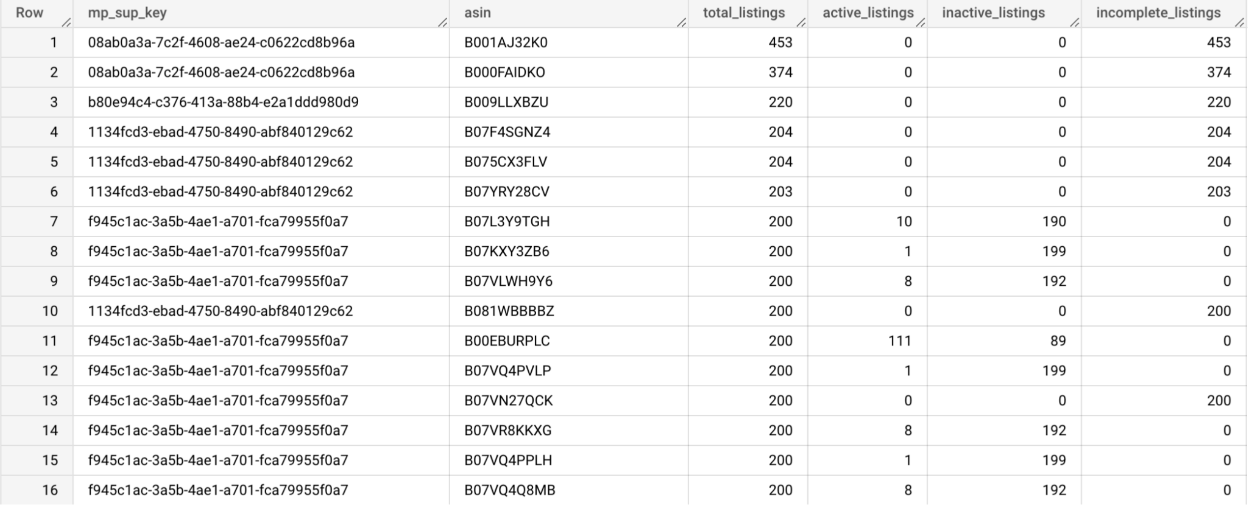
Asin 1: B001AJ32K0

Asin 2: B000FAIDKO

No current listing. Analysis done manually on google sheets.

The reason there is no data is because these listings could be inactive. Next step is to look at the number of active vs. inactive listings for these sellers and asin combination. This will allow to filter out the active records only.

```
SELECT
 mp_sup_key,
 asin,
 COUNT(*) as total_listings,
 COUNT(IF(status='Active', 1, NULL)) as active_listings,
 COUNT(IF(status='Inactive', 1, NULL)) as inactive_listings,
 COUNT(IF(status='Incomplete', 1, NULL)) as incomplete_listings,
FROM `bigqueryexport-183608.amazon.listings`
WHERE mp_sup_key IS NOT NULL AND asin IS NOT NULL
GROUP BY
 listing_id,
 mp_sup_key,
 asin
ORDER BY total_listings DESC;
```



Row	mp_sup_key	asin	total_listings	active_listings	inactive_listings	incomplete_listings
1	08ab0a3a-7c2f-4608-ae24-c0622cd8b96a	B001AJ32K0	453	0	0	453
2	08ab0a3a-7c2f-4608-ae24-c0622cd8b96a	B000FAIDKO	374	0	0	374
3	b80e94c4-c376-413a-88b4-e2a1ddd980d9	B009LLXBZU	220	0	0	220
4	1134fcd3-ebad-4750-8490-abf840129c62	B07F4SGNZ4	204	0	0	204
5	1134fcd3-ebad-4750-8490-abf840129c62	B075CX3FLV	204	0	0	204
6	1134fcd3-ebad-4750-8490-abf840129c62	B07YRY28CV	203	0	0	203
7	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07L3Y9TGH	200	10	190	0
8	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07KXY3ZB6	200	1	199	0
9	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07VLWH9Y6	200	8	192	0
10	1134fcd3-ebad-4750-8490-abf840129c62	B081WBBBBZ	200	0	0	200
11	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B00EBURPLC	200	111	89	0
12	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07VQ4PVLP	200	1	199	0
13	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07VN27QCK	200	0	0	200
14	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07VR8KKXG	200	8	192	0
15	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07VQ4PPLH	200	1	199	0
16	f945c1ac-3a5b-4ae1-a701-fca79955f0a7	B07VQ4Q8MB	200	8	192	0

CALCULATE AVERAGE NUMBER OF LISTINGS PER PRODUCT PER SELLER. This is a takeaway for the next Sprint. Is this a relevant question in terms of inventory analysis?

Based on the feedback from the Payability team, we only care about the total count of active listings. Also, we need to add SKU and status on the Inventory dashboard.

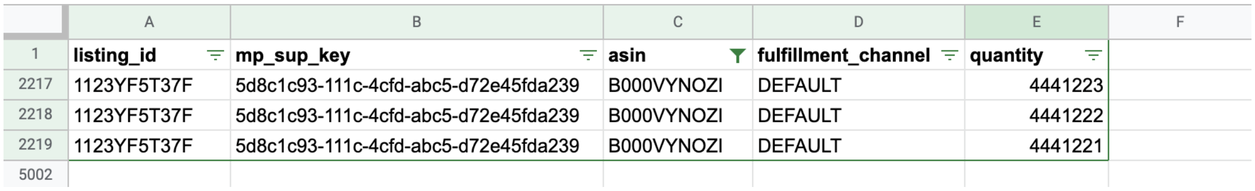
Question #2: Find the fulfillment_channel distribution for 2021 data

```
 SELECT
 listing_id,
 mp_sup_key,
 fulfillment_channel
FROM `bigqueryexport-183608.amazon.listings`
WHERE
 mp_sup_key IS NOT NULL
 AND asin IS NOT NULL
 AND listing_id IS NOT NULL AND listing_id <> ''
 AND LEFT(open_date, 4) = '2021'
GROUP BY
 listing_id,
 mp_sup_key,
 Fulfillment_channel;
```

The result was achieved using data visualization on looker studio and based on this analysis there are two main ones - default and amazon_na. I have confirmed that default means fulfilled by self (merchant). We are not interested in this type of data. The analysis will focus on Amazon fulfilled data for the next step.

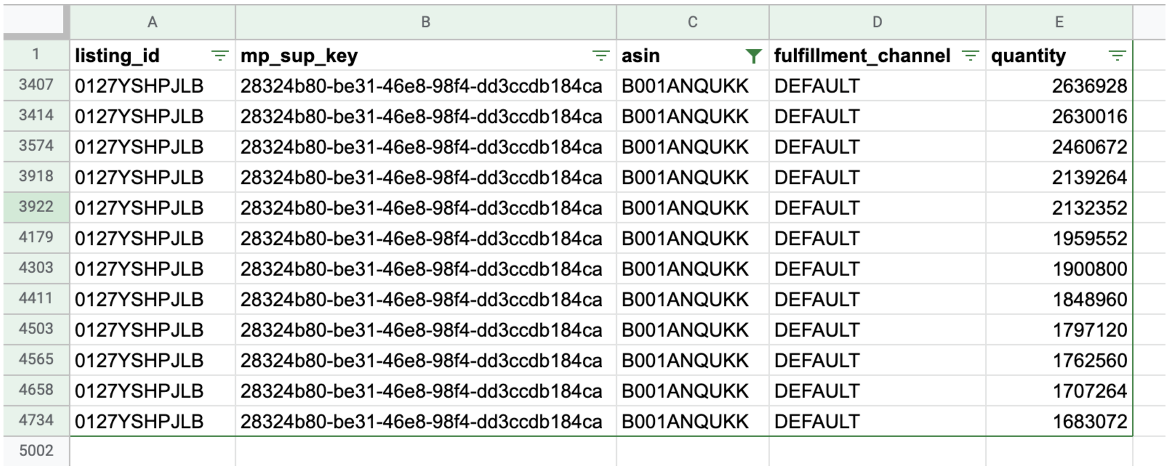
Step 2: Inventory

Why are there different quantities for the same listing_id?



	A	B	C	D	E	F
1	listing_id	mp_sup_key	asin	fulfillment_channel	quantity	
2217	1123YF5T37F	5d8c1c93-111c-4cfd-abc5-d72e45fda239	B000VYNOZI	DEFAULT	4441223	
2218	1123YF5T37F	5d8c1c93-111c-4cfd-abc5-d72e45fda239	B000VYNOZI	DEFAULT	4441222	
2219	1123YF5T37F	5d8c1c93-111c-4cfd-abc5-d72e45fda239	B000VYNOZI	DEFAULT	4441221	
5002						

This could be because the data gets updated on every partition date and it is reflecting a change in the quantity based on that. For example: an item could have been returned.



	A	B	C	D	E
1	listing_id	mp_sup_key	asin	fulfillment_channel	quantity
3407	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	2636928
3414	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	2630016
3574	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	2460672
3918	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	2139264
3922	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	2132352
4179	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	1959552
4303	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	1900800
4411	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	1848960
4503	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	1797120
4565	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	1762560
4658	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	1707264
4734	0127YSHPJLB	28324b80-be31-46e8-98f4-dd3ccdb184ca	B001ANQUKK	DEFAULT	1683072
5002					

There are a couple of issues with my query for Inventory calculation:

1. It still counts duplicate records.
2. I am not sure if taking the average for price and quantity makes sense for inventory calculation.