Hi All,

Jeremy Harper here, this is my attempt at partnering with everyone doing research on N3C so that we build up a repository of sql statements for areas of interest letting us all get to the analysis phase of our work. N3C is working hard to create and define codesets for disease states like Diabetes or Asthma make sure you pay attention to that work. If you can help by dropping your generic sql statements into a central repository and update this everyone else would appreciate it too.

Thanks,

Jeremy Harper

# Principles

1. Every Script you write should limit the number of values returned until you are sure you are doing it correctly and are ready to commit. This will both speed up your build time as well as avoid slowing down the system for all users. In SQL this is accomplished by adding "limit 10" at the end of your statement.

2. Every Script should be internally documented. That means both a header and calling out the decisions made for around logic of script

--Purpose - Add Hospitalization Information

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 10/30/2020

--Description - This takes a list of people with a specific event date (COVID19) and looks for hospitalization in the 15 days pre/post the date. Note that you might want to restructure this to look at the specific encounter that is listed from the original generated on C19 patients, here we only look at whether the encounter for what we define as inpatient has happened between plus/minus 15 days

3. If you structure a code-workbook where one transformation leads to the next you can rerun your entire query with a single click of a button and you do not need to export tables each step

# Setting up your environment Instructions

## If you follow these instructions you will be Creating a COVID19 Positive table with the following Columns

**Person\_ID** - Who the person is

**Age\_At\_C19**- Calculated Age at time of Covid19 based on 1/1/year

**age\_group** - Bucketing people by decade

**C19\_start\_date** - The earliest date documented in this identification workflow

**description** - What process allowed you to identify them as C19 Positive, EG problem list vs lab

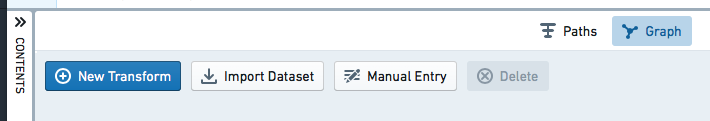
**C19\_identification\_visit\_id** - The visit associated with the record we are using to identify them as C19 Positive

**Hospitalization\_Visit\_ID** - The visit we matched with a hospitalization

**Source** – Whether it’s from problem list or lab

Step 1, Open a Code Workbook

Step 2, click "import dataset" (Gray button pictured below)



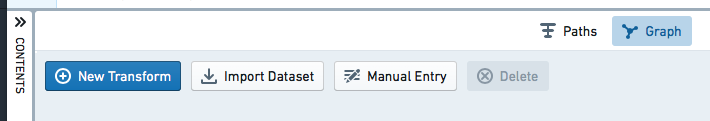
Step 3, Select All --> Data Catalog --> Safe Harbor Data

Step 4, select **Measurement** & **condition\_occurrence** tables

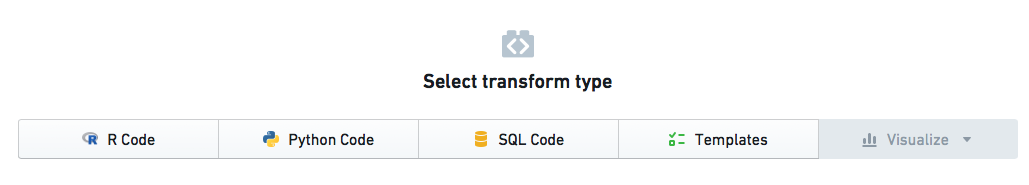
Step 5, navigate back Select All --> Data Catalog --> OMOP code sets

Step 6, select **concept\_set\_members**

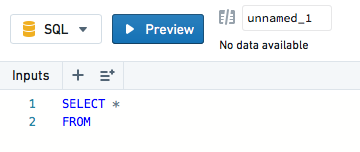
Step 7, click New Transform (Blue button)



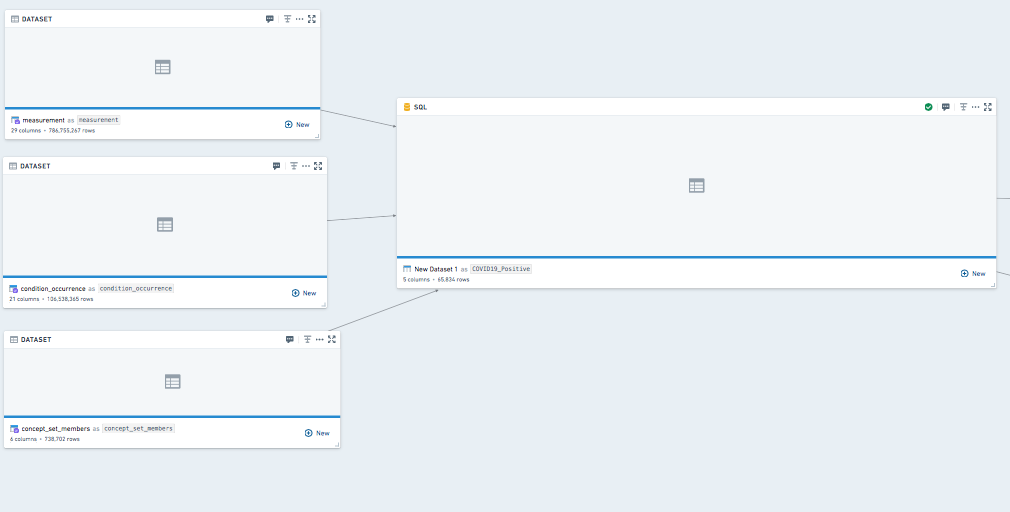
Step 8, Select Transform Type as “SQL Code”



Step 9, Click the small + button with ellipsis to add multiple tables (works with just plus as well but then you have to keep reclicking), Also click Unnamed and name this as “COVID19\_Positive\_Identification”



Step 10, Click on the other tables to add a dependency, It should look like what is pictured below (note you can drag tables all around the screen so it won’t match perfectly)



Step 11, Copy the SQL Code below and click run, ensure there are no errors in the preview.

--Purpose - pull together all covid19 positive patients into a single table

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 10/30/2020

--Description - This gets all people with a measurement or a problem list indicating C19. The main n3c site reports 170,125 ppl in the cohort, this grabbed 169350 unique people.

select measurement.person\_id, measurement.measurement\_date as start\_date, measurement.value\_as\_concept\_name as description, measurement.visit\_occurrence\_id as C19\_identification\_visit\_id, "concept\_651620200" as source --measurement.data\_partner\_id

from measurement

--At the time of writing, measurement\_source\_concept\_id got 64706, measurement concept id got 109526, both got 101093 it looks like measurement concept id is the best choice.

left join concept\_set\_members on measurement.measurement\_concept\_id=concept\_set\_members.concept\_id

--The concept grabs all current potential concept\_id's. The Value indicates it is positive result

where concept\_set\_members.codeset\_id in ('651620200') and (measurement.value\_as\_concept\_id in ('4126681', '45877985', '45884084', '9191'))

union all

--This grabs everyone with a problem list item

select person\_id, MIN(condition\_start\_datetime) as start\_date, first(condition\_concept\_name) as description, first(visit\_occurrence\_id) as C19\_identification\_visit\_id, "condition\_37311061" as source

from condition\_occurrence

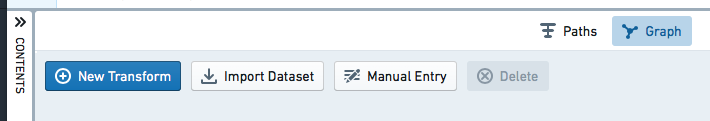
where condition\_occurrence.condition\_concept\_id=37311061

group by person\_id

--

Step 12, Create a new SQL Transform, (Blue Button), then link to the transform you just created

Step 13, click "import dataset" (Gray button pictured below)



Step 14, Select All --> Data Catalog --> Safe Harbor Data

Step 15, select **visit\_occurrence** table

Step 16, Link Visit table to new transform

Step 17 Copy the SQL Code below and click run, ensure there are no errors in the preview. Ensure you name this transform “C19\_Positive\_Hospitalization”

--Purpose - Add Hospitalization Information

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 10/30/2020

--Description - This takes a list of people with a specific event date (COVID19) and looks for hospitalization in the 15 days pre/post the date. Note that you might want to restructure this to look at the specific encounter that is listed from the original generated on C19 patients, here we only look at whether the encounter for what we define as inpatient has happened between plus/minus 15 days. Currently 86% result in null

SELECT COVID19\_Positive\_Identification.person\_id, COVID19\_Positive\_Identification.start\_date, COVID19\_Positive\_Identification.description, COVID19\_Positive\_Identification.C19\_identification\_visit\_id, COVID19\_Positive\_Identification.source,visit\_occurrence.visit\_occurrence\_id as Hospitalization\_visit\_id, if(visit\_occurrence.visit\_occurrence\_id is not Null, "Hospitalized", "Not\_Hospitalized") as hospitalization

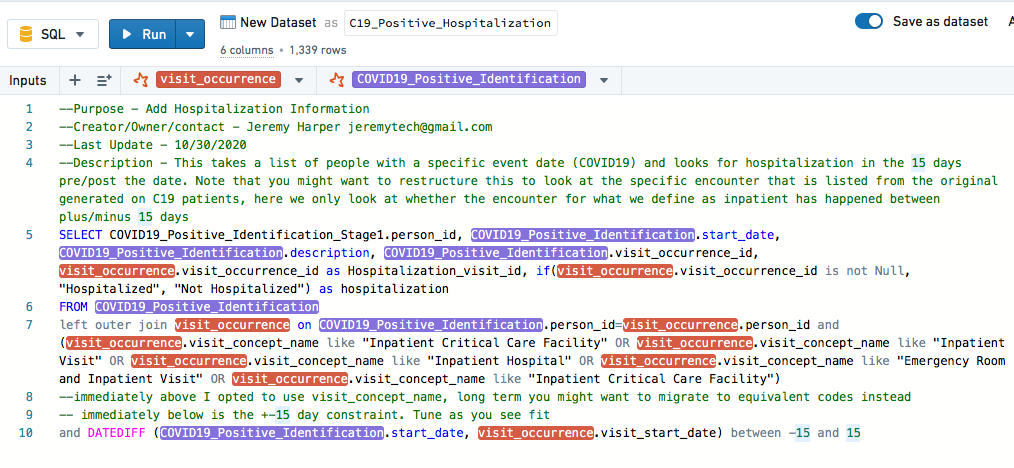
FROM COVID19\_Positive\_Identification

left outer join visit\_occurrence on COVID19\_Positive\_Identification.person\_id=visit\_occurrence.person\_id and (visit\_occurrence.visit\_concept\_name like "Inpatient Critical Care Facility" OR visit\_occurrence.visit\_concept\_name like "Inpatient Visit" OR visit\_occurrence.visit\_concept\_name like "Inpatient Hospital" OR visit\_occurrence.visit\_concept\_name like "Emergency Room and Inpatient Visit" OR visit\_occurrence.visit\_concept\_name like "Inpatient Critical Care Facility")

--immediately above I opted to use visit\_concept\_name, long term you might want to migrate to equivalent codes instead

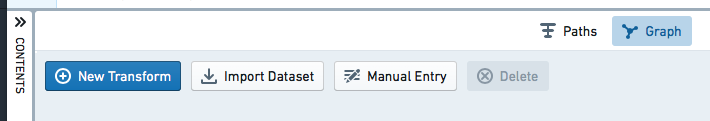
-- immediately below is the +-15 day constraint. Tune as you see fit

and DATEDIFF (COVID19\_Positive\_Identification.start\_date, visit\_occurrence.visit\_start\_date) between -15 and 15



Step 18, Create a new SQL Transform, (Blue Button), then link to the transform you just created

Step 19, click "import dataset" (Gray button pictured below)



Step 20, Select All --> Data Catalog --> Safe Harbor Data

Step 21, select **person** table

Step 22, Link person table to new transform

Step 23, Rename the dataset C19\_Positive\_All\_data

Step 24, Copy the SQL Code below and click run, ensure there are no errors in the preview.

--Purpose - Calculate age and decade grouping

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 10/30/2020

--Description - This takes everyone can creates an age column at time of C19 and a age\_group column. Note I chose to use an assumption of everyone being born 1/1/year. This is wrong, you could also list out years but that seems to create problems in 2021 etc.

select C19\_Positive\_Hospitalization.person\_id, C19\_Positive\_Hospitalization.start\_date, C19\_Positive\_Hospitalization.description, C19\_Positive\_Hospitalization.C19\_identification\_visit\_id, C19\_Positive\_Hospitalization.Hospitalization\_visit\_id, C19\_Positive\_Hospitalization.hospitalization, C19\_Positive\_Hospitalization.source, round(Age\_At\_C19\_temp, 2) as Age\_At\_C19, if(round(Age\_At\_C19\_temp, 1) between 0 and 9.9,"0-9",if(round(Age\_At\_C19\_temp, 1) between 10 and 19.9,"10-19",if(round(Age\_At\_C19\_temp, 1) between 20 and 29.9,"20-29",if(round(Age\_At\_C19\_temp, 1) between 30 and 39.9,"30-39",if(round(Age\_At\_C19\_temp, 1) between 40 and 49.9,"40-49",if(round(Age\_At\_C19\_temp, 1) between 50 and 59.9,"50-59",if(round(Age\_At\_C19\_temp, 1) between 60 and 69.9,"60-69",if(round(Age\_At\_C19\_temp, 1) between 70 and 79.9,"70-79",if(round(Age\_At\_C19\_temp, 1) between 80 and 89.9,"80-89",if(round(Age\_At\_C19\_temp, 1) between 90 and 99.9,"90-99",if(round(Age\_At\_C19\_temp, 1) between 100 and 109.9,"100-109",if(round(Age\_At\_C19\_temp, 1) between 110 and 119.9,"110-119","ERROR")))))))))))) as age\_group

from (

select C19\_Positive\_Hospitalization.person\_id, C19\_Positive\_Hospitalization.start\_date, C19\_Positive\_Hospitalization.description, C19\_Positive\_Hospitalization.C19\_identification\_visit\_id, C19\_Positive\_Hospitalization.Hospitalization\_visit\_id, C19\_Positive\_Hospitalization.hospitalization, C19\_Positive\_Hospitalization.source, datediff(C19\_Positive\_Hospitalization.start\_date,PPL.fake\_dob)/365 as Age\_At\_C19\_temp

from C19\_Positive\_Hospitalization

--If you just want a year without any other considerations get rid of date diff above and insert this as age\_at\_c19\_temp: (year(C19\_Positive\_Hospitalization.start\_date) - person.year\_of\_birth)

left outer join (select person.person\_id,

concat(if(person.year\_of\_birth is not null, person.year\_of\_birth, '01'),

'-', if(person.month\_of\_birth is not null, person.month\_of\_birth, '01'), '-', if(person.day\_of\_birth is not null, person.day\_of\_birth, '01')

) as fake\_dob

from person) PPL on C19\_Positive\_Hospitalization.person\_id=PPL.person\_id

) as C19\_Positive\_Hospitalization

Step 25, You may want to save as a Dataset (This will take a while as it now saves your dataset as a file under your folder and a folder called workbooks.

Step 26, Create another transform linking from C19\_Positive\_All\_data.

Step 27, Name this file as: C19\_positive\_persons (Don’t forget to save as a dataset)

Step 28, Use the following code to get a list of unique C19 patients.

--Purpose - Get a list of only C19 patients and what is the earliest source of C19 identification

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 10/30/2020

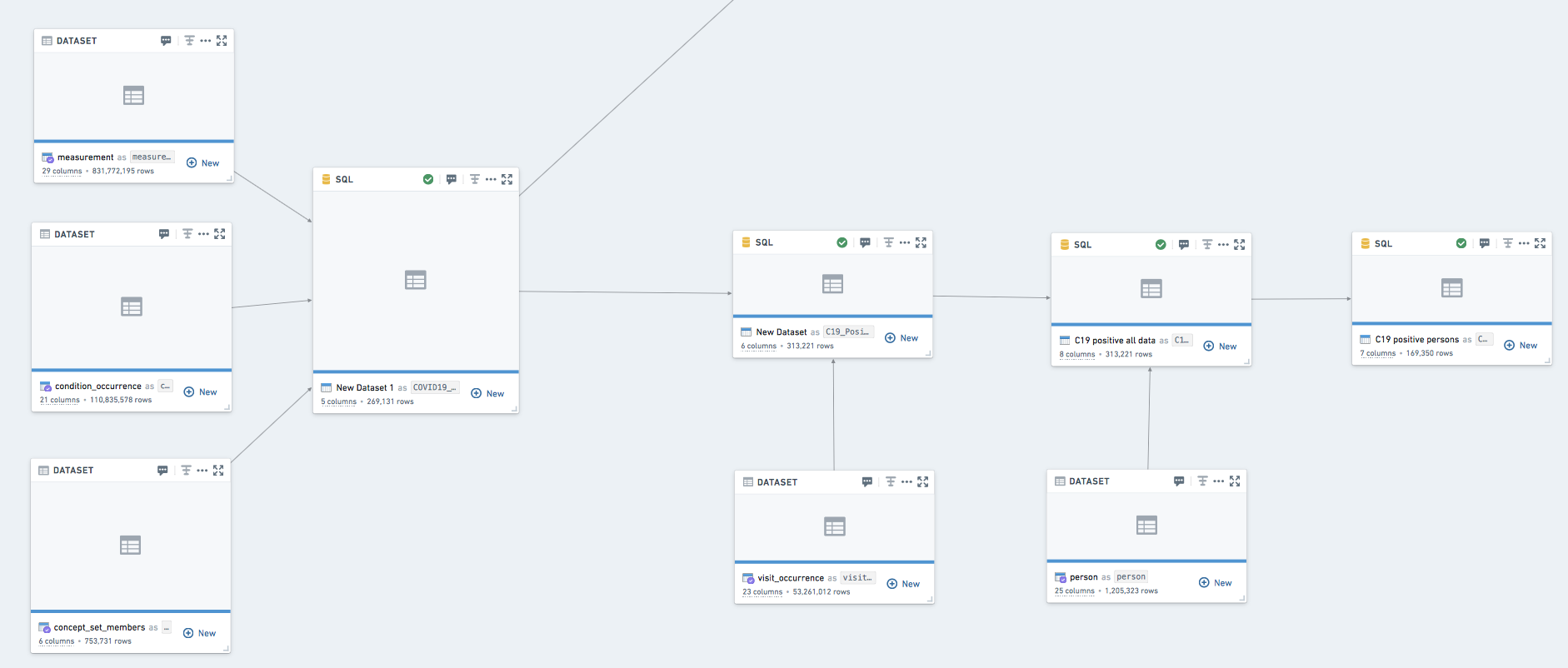
--Description - this provides a list with each person being unique

select person\_id, max(Age\_At\_C19) as Age\_At\_C19, max(age\_group) as age\_group, MIN(start\_date) as C19\_start\_date, first(description) as description, first(C19\_identification\_visit\_id) as C19\_identification\_visit\_id, first(Hospitalization\_visit\_id) as Hospitalization\_visit\_id, first(source) as source

from C19\_Positive\_All\_data

group by C19\_Positive\_All\_data.person\_id

Final diagram



# Assumptions made with dataset Generated:

1. You capture all patients who have a lab test or a problem list of C19 positive

2. Hospitalization that occurs inside a 15 day window is attributed to C19, you might be hospitalized and tested for C19 and the hospitalization have nothing to do with C19

3. Age, Since we have safe harbored to only give the year of birth this assumes all were born 1/1/year. Its a bad system but the good thing about it is that it works in 2021 as well as 2020. It is to allow for bucketing, you could also state each individual year group but then you would need to update it for future years.

# Tips:

When starting a new project create individualized folders for each project member, Palentir throws off a lot of extra files, you'll be happy to not have to move them into a project workspace until you're ready

Always add \_ to field names, Palentir does not like spaces

# Death Table:

Count how many people have passed away who had COVID19 (note this is not people who passed away from COVID19, just people who have passed.

1. Either create a new code workbook referencing the data you collected in the last exercise or extend the code workbook you have been using
2. Add the death table from the data catalogue

--Purpose - identify who has had covid19 and has died. this returns a distinct count.

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 11/10/2020

--Description - this is very simple

SELECT count(distinct(death.person\_id))

FROM death

left outer join C19\_positive\_persons on death.person\_id=C19\_positive\_persons.person\_id

where C19\_positive\_persons.person\_id is not null

1. Test that you were able to get a count. This will give you a good sense of how useful this data will be to our research
2. Replace the SQL with the code below. This will give you a list of patients who are deceased that presumably died while in the health system

--Purpose - identify who has had covid19 and has died. this returns a distinct count.

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 11/10/2020

--Description - this is very simple

SELECT \*

FROM death

left outer join C19\_positive\_persons on death.person\_id=C19\_positive\_persons.person\_id

--Death type concept ID 32510 is EHR record patient status "Deceased". It appears the best option to presume that someone died in the hospital. Note that workflows do not guarantee this is the case, I can log into the EHR while on the phone with a family member and mark them as deceased and it will appear the same way.

where C19\_positive\_persons.person\_id is not null and death.death\_type\_concept\_id=32510

# Condition\_Occurrence

1. Either create a new code workbook referencing the data you collected in the first exercise or extend the code workbook you have been using
2. Import dataset for condition occurrence
3. Create a new transformation and link condition\_occurrence and C19\_postive\_persons
4. You have the option of joining either by person or visit occurrence. Frankly you need to have a specific question you are asking in a specific way to make the visit linking appropriate and useful. The visit occurrence is likely most useful if you are looking for a new diagnosis related to covid19. You should closely examine how you are joining your hospitalization identification because it will make or break any visit relationship.

--Purpose - group conditions from the dataset and count number of people who have C19 and condition

--Creator/Owner/contact - Jeremy Harper jeremytech@gmail.com

--Last Update - 11/10/2020

--Description - this is very simple count

SELECT condition\_occurrence.condition\_type\_concept\_id, condition\_occurrence.condition\_source\_value, count(condition\_occurrence.person\_id) as condition\_count

FROM condition\_occurrence

left outer join C19\_positive\_persons on condition\_occurrence.person\_id=C19\_positive\_persons.person\_id

where C19\_positive\_persons.person\_id is not null

group by condition\_occurrence.condition\_type\_concept\_id, condition\_occurrence.condition\_source\_value

# GeoCoding

Geocoding in this dataset is somewhat limited. We have scrubbed zip codes to three digits and sometimes have included county information, state is indicated in the majority of situations.

Person table includes location ID (This in theory indicated where the person lives)

Care\_site includes location ID (This in theory indicates where a person seeks care)

Both of these link to the location table