OMOP Cohort Creation and Deidentification Guide

The following scripts are to be run on a site’s full OMOP dataset in order to prepare the relevant data for sharing with the VIRUS registry. Each script should be run on the same server as the OMOP data but can be customized to run on the preferred Database and Schema.

Instructions: Replace the database name and schema in each of these scripts with your own, then run the cohort creation and deidentification scripts in the following sequence:

01 – Cohort Creation ....(Filename: 01_CURE_ID_Cohort.sql)

02 – Generate CURE ID Tables ....(Filename: 02_CURE_ID_All_Tables.sql)

03 – Deidentify Rare Conditions ....(Filename: 03_CURE_ID_replace_rare_conditions_with_parents.sql)

04 – Generate OMOP Tables ....(Filename: 04_DE_ID_CDM_Table_ddl.sql)

05 – Remove Identifiers.....................(Filename: 05_DE_ID_script.sql)

06 – Run Data Quality Checks ....(Filename: 06_DE_ID_Quality_Checks.sql)

07-A – Profile Conditions ....(Filename: 07_A_condition_profile.sql)

07-B – Profile Measurements ....(Filename: 07_B_measurement_profile.sql)

07-C – Profile Drug Exposure ....(Filename: 07_C_drug_exposure_profile.sql)

07-D – Profile Unmapped Drugs ....(Filename: 07_D_review_unmapped_drugs.sql)

07-D – Profile Devices........................(Filename: 07_E_device_profile.sql)

OMOP Cohort Creation and Deidentification Process:

01 – Cohort Creation Script

Filename: 01_CURE_ID_Cohort.sql

Purpose: This script creates a cohort of patients for the CURE ID registry. The patient list is saved in the cohort table, along with other useful data elements.

Description: This SQL script creates a cohort of COVID-positive hospitalized patients based on specific criteria. The script performs several steps to identify and filter the patients before finally creating the cohort table. The script sets the context to use a specific database, but the actual name of the database is meant to be provided by the user.

Steps:

1) Create cohort table.
2) Identify patients (inpatient and outpatient) with covid positive lab results
    a. Use OMOP concepts that represent the LOINC codes for SARS-COV-2 nucleic acid test
    b. The concept ids here represent LOINC or SNOMED codes for standard ways to code a lab that is positive.
3) Identify the first positive covid test per patient (after January 1, 2020).
4) Limit to covid-positive patients with inpatient encounters.
5) Apply all inclusion/exclusion criteria to identify all patients hospitalized with symptomatic covid-19 up to 21 days after a positive SARS-CoV-2 test or up to 7 days prior to a positive SARS-CoV-2 test
6) Find the closest inpatient encounter to first positive SARS-COV-2 test (for patients hospitalized more than once)
7) Account for edge cases where patients have two hospitalizations same number of absolute days from SARS-COV-2 test (Ex: Patient hospitalized separately 3 days before and 3 days after SARS-COV-2 test)
8) Create the cohort by adding on birth date and death date

02 – CURE ID Tables Script

Filename: 02_CURE_ID_All_Tables.sql

Purpose: This script takes your OMOP dataset and generates a copy of key tables that have been filtered down to only include people and records related to the CURE ID registry.

Description: Creates CURE_ID tables from the generated CURE_ID cohort.

Dependencies: This script depends on CURE_ID_Cohort.sql, and must be run after that script completes

Steps:

1) Load Person table
2) Load Measurements table

3) Load Drug Exposure table
4) Load Death table
5) Load Observation data
6) Load Procedure Occurrence Table
7) Load Condition Occurrence Table
8) Load Visit Occurrence table
9) Load Device Exposure table

03 – Replace Rare Conditions Script

Filename: 03_CURE_ID_replace_rare_conditions_with_parents.sql

Purpose: Replace conditions occurring 10 or less times in the dataset with parent concepts that have at least 10 counts

Dependencies: This script requires the cohort table built from 01_CURE_ID_Cohort.sql, and the data loaded into the Condition Occurrence table built from 02_CURE_ID_All_Tables.sql.

Steps:

1) Condition roll up: concepts are mapped to their corresponding ancestor concept(s)
2) Create table that counts the ancestor concepts for each original concept
3) Create table that counts the original concepts
4) Filter to only include conditions that have more than 10 counts
5) Get just the most specific condition in the ancestor-descendent hierarchy

04 – Deidentified Data DDL Script

Filename: 04_DE_ID_CDM_Table_ddl.sql

Purpose: Generate the necessary tables for the de-identified version of the CURE ID Cohort

Dependencies: None

Customization: By default this script will create tables in a schema titled “deident,” however this can be set to whatever value you desire.

Steps:

1) Create the Person table
2) Create the Death table
3) Create the Visit Occurrence table
4) Create the Drug Exposure table
5) Create the Device Exposure table
6) Create the Condition Occurrence table

7) Create the Measurement table
8) Create the Observation table

05 – Deidentification Script

Filename: 05_DE_ID_script.sql

Purpose: This script creates a copy of the Cohort and removes identifying characteristics to prepare the data for sharing with the VIRUS registry.

Dependencies: This script requires the cohort table built from 01_CURE_ID_Cohort.sql, and the data loaded into all tables built from 02_CURE_ID_All_Tables.sql, rare conditions replace from 03_CURE_ID_replace_rare_conditions_with_parents.sql, and the de-identified OMOP CDM tables generated from 04_DE_ID_CDM_Table_ddl.sql.

Description: Run this file to generate a deidentified copy of your target data. Insert your data into the OMOP tables, and de-identify person_id, and date fields using date.shift. (*If a person is 90 years of age or older, assign a random age between 90-99 years.)

Steps:

1) Use find and replace to set source and target DB and Schema names
2) Load the OMOP Person table, and de-identify
3) Load the OMOP Visit Occurrence table, and de-identify
4) Load the OMOP Condition Occurrence table, and de-identify
5) Load the OMOP Procedure Occurrence table, and de-identify
6) Load the OMOP Drug Exposure table, and de-identify
7) Load the OMOP Observation table, and de-identify
8) Load the OMOP Death table, and de-identify
9) Load the OMOP Device Exposure table, and de-identify
10) Load the OMOP Measurement table, and de-identify

Reassignment of Person IDs:

- Person IDs are regenerated sequentially from a sorted copy of the Person table. These new Person IDs are carried throughout the CDM to all tables that reference it.

Date Shifting:

- Each person is assigned a random date shift value between -186 and +186 days. All dates for that person are then shifted shifted by that amount.
- Birthdays: After date shifting a person's birthday, the day is then set to the first of the new birth month. If the person would be > 89 years old then they are assigned a random birth year that would make them 90-99 years old.

Date Truncation:

- A user-defined Start and End date are used to exclude any date shifted data that falls outside of the target date range (E.G. Procedures, conditions occurrences, etc. Does not include Birthdates).

Removal of other identifiers:

- Other potentially identifying datapoints are removed from the dataset such as location_id, provider_id, and care_site_id

06 – Quality Checks Script (optional)

Filename: 06_DE_ID_Quality_Checks.sql

Purpose: This script checks basic metrics for each table in the deidentified dataset to ensure the previous scripts were successful. This does

Description: This script runs a number of summary level quality checks for each table to audit basic data counts and date ranges.

Dependencies: This script requires the populated deidentified OMOP tables generated from the sequence of running:

01_CURE_ID_Cohort.sql,

02_CURE_ID_All_Tables.sql,

03_CURE_ID_replace_rare_conditions_with_parents.sql,

04_DE_ID_CDM_Table_ddl.sql,

05_DE_ID_script.sql

Steps:

1) Count distinct person_ids and find the maximum and minimum birthdates in the OMOP Person table.
2) Count distinct person_ids in the OMOP Death table.
3) Count distinct person_ids, count number of records per observation_concept_id, and find the maximum and minimum observation dates for all records in the OMOP Observation table.
4) Count distinct person_ids, count number of records per procedure_concept_id, and find the maximum and minimum procedure dates for all records in the OMOP Procedure Occurrence table.
5) Count distinct person_ids, count number of records per condition_concept_id, and find the maximum and minimum condition dates for all records in the OMOP Condition Occurrence table.
6) Count distinct person_ids, count number of records per measurement_concept_id, and find the maximum and minimum measurement dates for all records in the OMOP Measurement table.
7) Count distinct person_ids, count number of records per device_concept_id, and find the maximum and minimum device exposure dates for all records in the OMOP Device Exposure table.
8) Count distinct person_ids, count number of records per drug_concept_id, and find the maximum and minimum drug exposure dates for all records in the OMOP Drug Exposure table.

07 – Cohort Profile Scripts

Dependencies: These scripts require the populated deidentified OMOP tables generated from the sequence of running scripts 1-5:

01_CURE_ID_Cohort.sql,

02_CURE_ID_All_Tables.sql,

03_CURE_ID_replace_rare_conditions_with_parents.sql,

04_DE_ID_CDM_Table_ddl.sql,

05_DE_ID_script.sql

07-A – Condition Profile

Filename: 07_A_condition_profile.sql

Purpose: Generate a profile of condition prevalence in the final cohort.

Description: Condition counts are calculated per patient and are aggregated by parent concepts for each condition concept present in the final OMOP Condition Occurrence table.

07-B – Measurement Profile

Filename: 07_B_measurement_profile.sql

Purpose: Generate a profile of measurement prevalence in the final cohort.

Description: Measurement counts are calculated per patient and are aggregated by parent concepts for each measurement concept present in the final OMOP Measurement table.

07-C – Drug Exposure Profile

Filename: 07_C_drug_exposure_profile.sql

Purpose: Generate a profile of drug prevalence in the final cohort.

Description: Drug counts are calculated per patient and are aggregated by ingredient for each drug concept present in the final OMOP Drug Exposure table.

07-D – Unmapped Drugs Profile

Filename: 07_D_review_unmapped_drugs.sql

Purpose: Generate a profile of drugs that are not mapped to drug_concept_ids in the final cohort.

Description: This file filters drugs that were unsuccessfully mapped to a drug_concept_id when running the 02_CURE_ID_All_Tables.sql script. Drug source values for which the drug_concept_id is “0” and have at least 20 instances in the final cohort are aggregated for manual review.

** Drug source values can contain PHI. Please review the output for PHI before sharing.

07-E – Device Profile

Filename: 07_E_device_profile.sql

Purpose: Generate a profile of device prevalence in the final cohort.

Description: Device counts are calculated per patient and are aggregated by parent concepts for each device concept present in the final OMOP Device Exposure table.