VA-OHDSI Cohort Creation Process
script = Process.R

Inputs:
Normal package inputs(
cohortDatabaseSchema,
outputFolder,
cohortTable)
Custom inputs(timeoutInSeconds,
outputFolder.cohortSql)

1. Identify all OHDSI cohort SQL files

run prepareCohortSql() from OHDSI package.

provide sql output folder as input to RunSQLCustom_bv.R. Should be the path to the outputFolder from prepareCohortSql() function.

Pull in all files in outputFolder/cohortsql/cte, identify if any are complete by checking if they exist in cohortTable

Exclude those that are already complete

2. Run SQL Customization process

Described in detail in next tab (Customize SQL)

Ensure that all .sql files in folder are ones that we want to run, otherwise they will run

Run Customize SQL process Looping through all files, running VAcustomSQL()

3. Build All Cohorts

1. Run SQLCohortsCreate() function with new folder path to customized SQL

2. Run with a pre-specified time-out period (e.g., 1200 seconds=20 minutes) to help ensure any issues don't result in infinite running scripts.

3. Ensure all SQL cohorts were created by running sql query: SELECT cohort_definition_ID, COUNT(*) as ct FROM [cohort_table] GROUP BY cohort_definition_ID

This function checks which cohorts are created in the cohort table and runs any sql files that do not have a row in the cohort table.

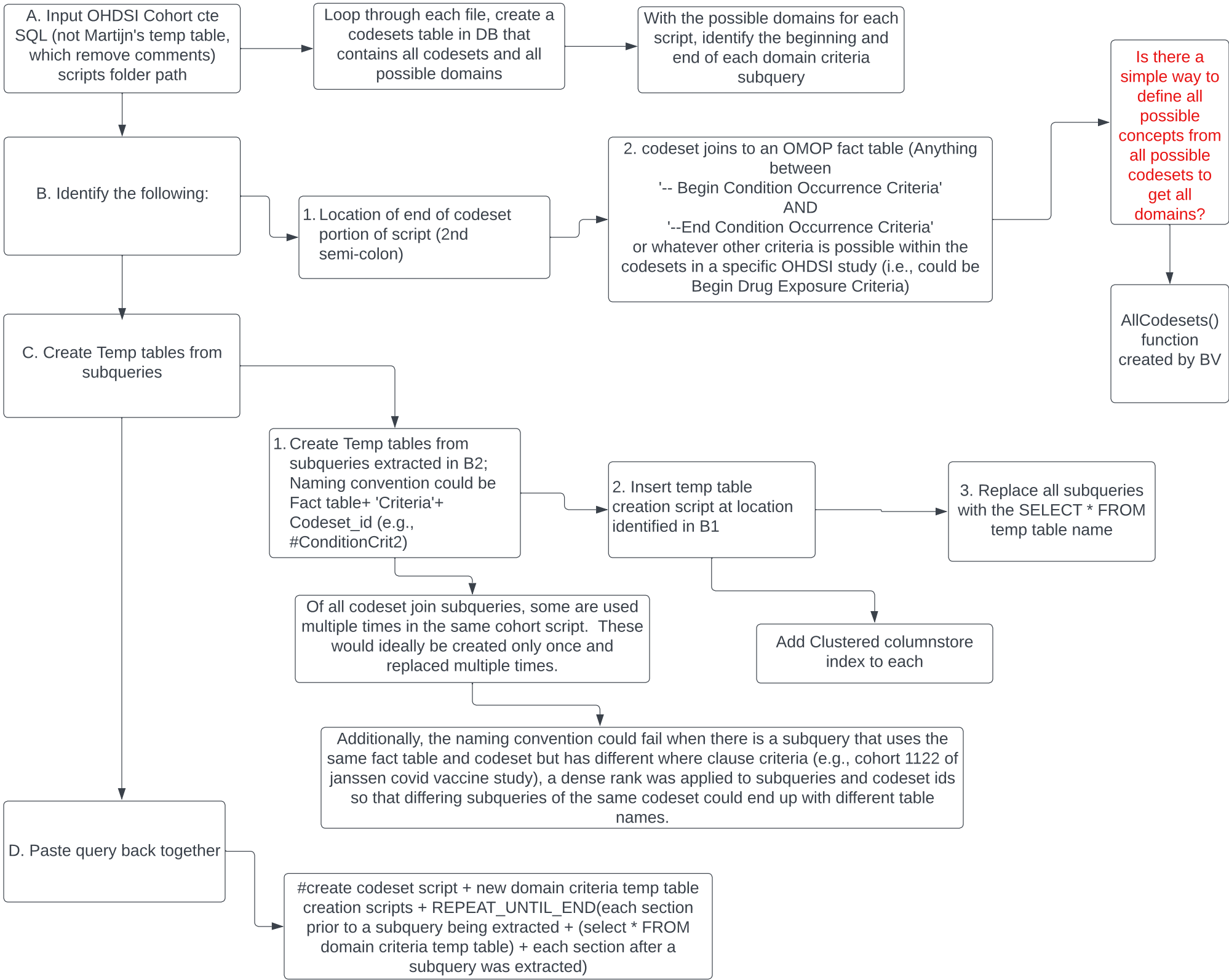Some bugs still remain with this part. Some scripts still appear to run infinitely from R, but those few scrips can be run manually in SQL easily.

This process is meant to improve performance of slow-running/complex OHDSI SQL cohort scripts by replacing subquery joins to fact tables with indexed temp tables.

Based on the assumptions:
(1)that all OHDSI cohort scripts start with the creation of a #codeset table,
(2) and only use the #codeset table in joins to an ohdsi domain table that can be identified from codes in the codeset table within the creation of the #qualifyingevents and #inclusion0 tables,
(3) and these joins can be extracted as subqueries between comments that look like
'-- Begin Condition Occurrence Criteria'
AND
'--End Condition Occurrence Criteria'

A. Input OHDSI Cohort cte SQL (not Martijn's temp table, which remove comments) scripts folder path

Loop through each file, create a codesets table in DB that contains all codesets and all possible domains

With the possible domains for each script, identify the beginning and end of each domain criteria subquery

Is there a simple way to define all possible concepts from all possible codesets to get all domains?

B. Identify the following:

1. Location of end of codeset portion of script (2nd semi-colon)

2. codeset joins to an OMOP fact table (Anything between
'-- Begin Condition Occurrence Criteria'
AND
'--End Condition Occurrence Criteria'
or whatever other criteria is possible within the codesets in a specific OHDSI study (i.e., could be Begin Drug Exposure Criteria)

AllCodesets() function created by BV

C. Create Temp tables from subqueries

1. Create Temp tables from subqueries extracted in B2; Naming convention could be Fact table+ 'Criteria'+ Codeset_id (e.g., #ConditionCrit2)

2. Insert temp table creation script at location identified in B1

3. Replace all subqueries with the SELECT * FROM temp table name

Of all codeset join subqueries, some are used multiple times in the same cohort script. These would ideally be created only once and replaced multiple times.

Add Clustered columnstore index to each

Additionally, the naming convention could fail when there is a subquery that uses the same fact table and codeset but has different where clause criteria (e.g., cohort 1122 of janssen covid vaccine study), a dense rank was applied to subqueries and codeset ids so that differing subqueries of the same codeset could end up with different table names.

D. Paste query back together

#create codeset script + new domain criteria temp table creation scripts + REPEAT_UNTIL_END(each section prior to a subquery being extracted + (select * FROM domain criteria temp table) + each section after a subquery was extracted)

Definitions

Domain criteria: In each cohort sql script, a codeset exists that identifies a list of codes found within a specific OMOP domain table. We refer to the subquery that contains the OMOP domain table join to a specific codeset as domain criteria.