VIReC

The Researcher's Notebook

Principles of Building a Data Cohort from CDW: An example of Veteran patients who visited the optometry clinic

Introduction

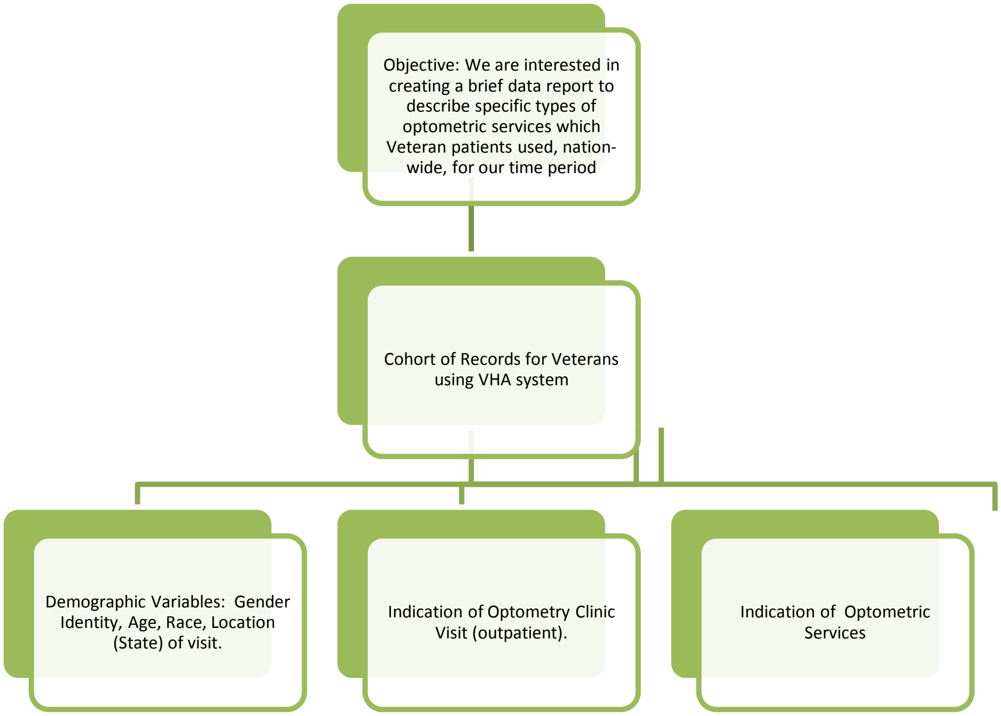
This notebook is intended to act as a guide to show the principles used to create a SQL data file using data fields from the VHA Corporate Data Warehouse (CDW) to create a SQL table for either a medical research or healthcare operational project using CPT codes and stop codes. We will use SQL (the data management programming language used with relational data) to collect our records and variables that can be exported as a CSV file.

For this exercise, our "operational purpose" is that we are interested in creating a brief data report to describe specific types of optometric services which Veteran patients used, nation-wide for a specific period of time. We need a dataset containing specific variables/columns from CDW to measure:

- Veterans who had at least one optometry clinic visit (outpatient), one optometric service--e.g., routine eye exam/procedure visit, or any indication of a "Comprehensive Eye Exam" etc.
- Veteran demographic variables: Gender, date of birth (so we can calculate age at time of visit), State where service visit occurred
- Indication of an Optometry Clinic Visit (outpatient) during the specified timeframe

The way in which we can think about constructing a dataset for operational analysis might take the form expressed in *Figure 1*. After this conceptualization of the problem, we must move to thinking about where data are located in the data warehouse and constructing a data table with specific data columns from the perspective a relational database. There are several steps to creating the necessary database file for export/import and statistical analysis. These steps are laid out in the remainder of this notebook.

Figure 1: Conceptualization of the Problem


Objective: We are interested in creating a brief data report to describe specific types of optometric services which Veteran patients used, nation-wide, for our time period
Cohort of Records for Veterans using VHA system
Demographic Variables: Gender Identity, Age, Race, Location (State) of visit.
Indication of Optometry Clinic Visit (outpatient).
Indication of Optometric Services

Step 1 | Identify Stop Codes and CPT Codes

Given our purpose, we need to identify specific stop codes and CPT codes.

1. **Stop Code**
   a. 408 (VHA Clinic Stop Code for Optometry Clinic Visits)
2. **CPT Codes used by Optometrists in VHA**
   a. Routine Eye Exams (RE_Exam) = 92002, 92004, 92012, and/or 92014
      i. These codes are used for "new and/or established ophthalmology or optometry patients that include the evaluation and management of a patient. These codes are appropriate when the level of services includes several routine optometric or ophthalmologic examination techniques that are integrated with and cannot be separated from the diagnostic evaluation."
   b. Refraction (Ref) = 92015
      i. Any time a refraction is performed, is should be reported as an additional code, 92015.
   c. Comprehensive Eye Exam = 99000.
      i. The 99 series comprehensive exam includes a complete optometric history, general medical observations, external and ophthalmoscopic examination, Gross Visual fields (CVF), basic sensorimotor examination (EOM, PUPILS), Slit-Lamp biomicroscopy, examination with cycloplegia or mydriasis and tonometry.

Step 2 | Identify Tables and Columns of Interest

Table 1 contains a list of tables and columns used to capture our research topic of interest. Note that the need to link to other tables necessitates the inclusion of primary and foreign keys (all ending in SID) that may not initially seem to have relevance to the construct being measured. Also, columns that are needed to "clean" the data (e.g., CDWPossibleTestPatientFlag or IneligibleDate) may not initially seem relevant to the research question, but they are essential part of making sure your analytic file fits your needs.

Table 1

Construct being measured	Table	Columns
Outpatient Visits	Outpat.Visit	VisitSID, VisitDateTime, PrimaryStopCodeSID, SecondaryStopCodeSID , PatientSID, DivisionSID, Sta3n
Vision tests and examinations	Outpat.VProcedure	VProcedureSID, VisitDateTime, EventDateTime, VisitSID, CPTSID
CPT Codes	Dim.CPT	CPTCode, CPTName, CPTSID
Stop Codes	Dim.StopCode	StopCode, StopCodeName, PrimaryStopCodeSID , SecondaryStopCodeSID
Location of Visit	Dim.Division Dim.Institution Dim.State	Sta6a, DivisionSID StateSID State, StateSID
Patient Information	SPatient.SPatient*	Gender, Age, State, PatientICN , CDWPossibleTestPatientFlag, VeteranFlag, IneligibleDate

*Note that with the new Patient 3.0 Domain, you will need to add SPatient.SPatientAddress to retrieve State, OrdinalNumber, AddressType, and PatientSID; in your query, you would likely want to limit these records to circumstances when OrdinalNumber is either 1 or 13 or when AddressType is either 'Patient' or 'Legal Residence'.

Step 3 | SQL Code

Step 3.1 Reducing Large Fact Tables

```
/********************************************************************************************
Step one reduces the large fact tables by timeframe and by number of columns.

The first query builds a small visit file for the chosen timeframe with only the selected
columns in order to reduce the size of the future queries. The SELECT statement lists each selected
column and uses a comma to separate them.  The INTO statement names a temporary table (using the #
sign) as the place to store the results of the query.  The FROM statement tells the processor that
the listed columns can be found in a table called Outpat.Visit.  The WHERE statement reduces the
data to only those records where the date of the visit is within the desired timeframe;
simultaneously, the date values are being converted to a new format (DATETIME2) to increase the
efficiency of the query.

The second query repeats the process for the Outpat.VProcedure table.
```

```
If you are a researcher, you would be asking for research extracts for your IRB approved time
period; then, you can reduce the number of columns. Also, you will put the name of your research
folder after USE rather than CDWWork.

**********************************************************************************************/

USE CDWWork
GO

SELECT VisitSID, VisitDateTime, Sta3n,
      PrimaryStopCodeSID, SecondaryStopCodeSID , PatientSID, DivisionSID
INTO #Visit
FROM Outpat.Visit
WHERE (VisitDateTime >= CONVERT (DATETIME2(0), '2014-01-01 00:00:01')
and VisitDateTime <= CONVERT (DATETIME2(0), '2014-01-06 00:00:00')) ;



SELECT VProcedureSID, VisitDateTime, EventDateTime, VisitSID, CPTSID
INTO #Procedure
FROM Outpat.VProcedure
WHERE (VisitDateTime >= CONVERT (DATETIME2(0), '2014-01-01 00:00:01')
and VisitDateTime <= CONVERT (DATETIME2(0), '2014-01-06 00:00:00')) ;
```

Step 3.2 Selecting the Correct Stop Code and CPT Codes

```
/********************************************************
*
The third query selects a combination of CPT and stop codes to narrow visit and procedure records
to only those that represent optometry care; simultaneously, it collects location information
(sta6a and State) from the Dim.Division and Dim.State tables respectively.  The SELECT statement
pulls columns from: the Outpat.Visit extract (aliased with “v.”), the Outpat.VProcedure extract
(aliased with “p.”), the Dim.StopCode table (aliased with “s.” for primary stop codes and “s2.” for
secondary stop codes).  The series of FROM and INNER JOIN statements list the tables in which the
columns are to be found.  The WHERE statement is used to select only those visits that are
associated with the correct stop code or CPT code.
*
*******************************************************/


SELECT v.VisitSID, v.VisitDateTime, v.PatientSID, v.Sta3n,
       st.State as VisitState,
       p.VProcedureSID, p.EventDateTime, d.Sta6a,
        c.CPTCode, c.CPTName,
       s.StopCode as PrimaryStopCode , s.StopCodeName as PrimaryStopName,
       s2.StopCode as SecondaryStopCode , s2.StopCodeName as SecondaryStopName
INTO #OptVisit
FROM #Visit as v
INNER JOIN #Procedure as p
      ON v.VisitSID = p.VisitSID
INNER JOIN Dim.CPT as c
```

```
      ON p.CPTSID = c.CPTSID
INNER JOIN Dim.StopCode as s
      ON v.PrimaryStopCodeSID = s.StopCodeSID
INNER JOIN Dim.StopCode as s2
      ON v.SecondaryStopCodeSID = s2.StopCodeSID
INNER JOIN Dim.Division as d
      ON v.DivisionSID = d.DivisionSID
INNER JOIN Dim.Institution as i
      ON d.InstitutionSID = i.InstitutionSID
INNER JOIN Dim.State as st
      ON i.StateSID = st.StateSID
WHERE (s.StopCode = 408 or s2.StopCode = 408)
      or CPTCode IN ('92002', '92004', '92012', '92014', '92015', '92000') ;
```

Step 3.3 Connecting Patient Information to Optometry Visits

```
/****************************************************************************
The fourth query joins patient data to the selected visit and procedure data.  The SELECT lists
takes everything from the temporary table created in the third query (aliased with “op.”) and then
collects columns from SPatient.SPatient (aliased “p.”).  The CONVERT function is used to change the
DATETIME fields to DATE fields, and the number 103 is used to specify the style MM/DD/YYYYY for
ease of use in SPSS software.  An inner join is used because we only want visits and procedures
that were for real veteran patients. The WHERE statement is used to eliminate any marked test
patients and non-Veterans; guidelines set forth by the Data Quality reports were used to develop
this logic.

The INTO statement creates a temporary table called #OptometryCareVisits, but you might eliminate
the line altogether to see your results come directly into the results window in SSMS or you might
create a permanent table by using the “dflt.” schema in the place of the # sign.

******************************************************************************/

Select op.VisitSID, CONVERT (DATE, op.VisitDateTime, 103) as VisitDate,
       op.PatientSID, op.Sta3n as VisitSta3n, VisitState,
       VProcedureSID, CONVERT (DATE, op.EventDateTime, 103) as EventDate,
       op.Sta6a, op.CPTCode, op.CPTName,
       op.PrimaryStopCode , op.PrimaryStopName,
       op.SecondaryStopCode , op.SecondaryStopName  ,
       p.Gender, p.Age, p.State , p.PatientICN
INTO #OptometryCareVisits  -- see options explained above
FROM #OptVisit as op
INNER JOIN SPatient.SPatient as p
      ON op.PatientSID = p.PatientSID
WHERE CDWPossibleTestPatientFlag <> 'Y' AND
      --a decision about ex/including null values needs to be made
      (VeteranFlag IS NULL OR VeteranFlag = 'Y')
      AND IneligibleDate IS NULL AND IneligibleReason IS NULL;
```

Step 3.4 Examining Your Output

```
/****************************************************************************
If you selected to create a temporary table (#OptometryCareVisits) or a permanent table
(dflt.OptometryCareVisit) in Step 3.3, you will not see your output until you query that table.

A simple query would look at the first 1000 rows of everything (*) in table.

*****************************************************************************/

SELECT TOP 1000 *
FROM #OptometryCareVisits;
```

Step 3.5 Save SQL Data View/Table for export to a statistical software package

The SQL database file, now, includes the data fields and row values that are needed to address the “operational purpose” previously identified. Save the SQL data view/table in a specific location in a format that can be imported by your statistical data management software of choice (e.g., SPSS, SAS, Stata, etc.). One possibility is to save the data prepared using SQL in a CSV format and then export it to or import it from that statistical software package for further data management.

Notes:

- It is important to have the data field names included in the first-row of the SQL database file before “saving” in the CSV format. To include the data field names in the “first-row” of the “open” SQL database file, go to the Query menu and select “Query Options.” Find “Results/Grid.” Choose “Include column headers when copying or saving results.” Then, “Okay.” Now, “Save” the SQL database file as a CSV format.

- There will be a row for each episode of optometry care. It may be necessary to restructure (pivot) the data using the selected statistical software package to put all patient information into a single record. The PatientICN is the ideal identifier in the CDW for accurate identification of unique patients. During the process of pivoting, there may be inconsistent demographic values for a patient that will have to resolved based on your specific goals and aims.

Step 3.6 Cleaning Up

```
/****************************************************************************
Clean up space on the servers by getting rid of tables that you don’t need.
*****************************************************************************/

drop table #visit;
drop table #OptVisit;
drop table #Procedure;
drop table #OptometryCareVisits;
```

Conclusion

We described the basic principles needed to create a cohort of data from sources in the VHA's CDW. Then, we shared the SQL programming code needed to construct a database file that can be exported to/imported by a statistical software package for further data management and statistical data analysis.

Resources

VHA Data Quality Patient Identification Reports
http://vaww.vhadataportal.med.va.gov/Resources/DataReports.aspx

VIReC CDW Domain Factbooks
http://vaww.virec.research.va.gov/CDW/Documentation.htm#Factbooks

Acknowledgements
We appreciate the encouragement of Dr. Erick Ishii, Epidemiologist, Patient Care Services, Post-Deployment Health, Epidemiology Program, VHA Central Office, Washington, DC, in the preparation of this Researcher's Notebook. The authors of this notebook are Margaret Gonsoulin, PhD, Sociologist, VIReC, Hines, IL and Gregory Fant, PhD, MSHS, MPA, Epidemiologist, Patient Care Services, Population Health Services, VHA Central Office, Washington, DC.

Suggested Citation
Gonsoulin, M. & Fant, G. Principles of Building a Data Cohort from CDW: An example of Veteran patients who visited the optometry clinic. *The Researcher's Notebook*; no. 8. Hines, IL: VA Information Resource Center; 2017.

Disclaimer
The material found in this Researcher's Notebook represents one potential set of logic that was appropriate to the context in which the author(s) were working at that point in time. Each reader should assess the appropriateness of this logic for their specific context, question and time of study.

VA Information Resource Center
Edward Hines, Jr. VA Hospital
5000 South 5th Avenue
Hines, IL 60141-5000
virec@va.gov
vaww.virec.research.va.gov
VIReC is supported by HSR&D grant SDR 98-004.