

# RCHC Analytics Academy: Can't Wait for the SQL!

---

BY BENJAMIN FOUTS, MPH

MAY 12, 2022



# Agenda

---

12:00pm – **Part 1: Understanding and Using Standard Value Sets in Your SQL Code**

1:00pm – 15 minute break

1:15pm – **Part 2: Utilizing SQL Code from Other Health Centers**

2:05pm – Final Questions and Answers

2:10pm – Complete survey

2:15pm – End

# For Intermediate Users

---

So, you should:

- Be able to write a SQL report in DataGrip or Relevant
- Be familiar with the most common tables in Relevant and the fields used to join them
- Be familiar with basic SQL coding to display columns, join tables and narrow results

*Handouts: one with tables and SQL examples sent out already. Another handout with the exercise answers (along with the presentation slides) will be made available in the RCHC website after this live presentation*

# Part 1: Understanding and Using Standard Value Sets in Your SQL Code

---

FOR INTERMEDIATE USERS



# Clinical Data Evolution

---

- In the past, quality assessment of clinical services relied on billing codes, abstracting charts, manual review of written notes, and communication between performance improvement personnel and clinicians to clarify information. This gave an incomplete (and sometimes biased) picture of performance.
- As clinical data began to be placed in electronic format, systematic and standardized methods of storing and extracting summary results evolved
- Clinical operations and information were encoded in different ways

# Clinical Data Evolution: Encoding

---

This approach has the following advantages:

- Information can be transmitted from organization to organization
- Everybody understands the same concept by a common definition
- Enhanced electronic methods to extract specific data elements
- More complex ways to evaluate clinical performance and other measures

# Code Authorities

---

- Different organizations began developing code systems to describe aspects of clinical operations. These are called Code Authorities
- For example, SNOMED International manages the code system SNOMED
- You may see a copyright registration mark after some code systems, like SNOMED® and LOINC®
- The CPT Code Authority often does not allow their standard code descriptions to be published in useful ways. This makes it sometimes difficult to know what a code means without extra steps

# Standard Code Systems

---

**International Classification of Diseases (ICD).** Diagnosis classification system developed by the Centers for Disease Control and Prevention (CDC) National Center for Health Statistics (NCHS) for use in healthcare treatment settings. The current code set we use is called ICD-10-CM (note: ICD-10-PCS is used in inpatient hospital settings)

**Current Procedural Terminology (CPT).** A registered trademark of the American Medical Association (AMA), these codes are a listing of descriptive terms and identifying codes for accurately describing and reporting medical, surgical, and diagnostic services and procedures.

*Source of descriptions: CMS Measures Management System Blueprint Version 17.0, Supplemental Material*

# Standard Code Systems, Continued

---

**RxNorm.** Produced by the National Library of Medicine is a normalized naming system for generic and branded drugs. The goal of RxNorm is to allow computer systems to communicate drug-related information efficiently and unambiguously.

**National Drug Codes (NDC).** A numeric code composed of a labeler code (the drug manufacturer, repackager, or distributor), a product code (specific strength or dosage) and a package code (package size or type).

# Standard Code Systems, Continued

---

**Logical Observation Identifier Names and Codes (LOINC).** This code set is for clinical and laboratory observations, healthcare screening/survey instruments, and document type identifiers. Health centers almost exclusively use it to identify labs.

**Codes for Vaccine Administered (CVX).** CDC's National Center of Immunization and Respiratory Diseases (NCIRD) developed and maintains this code set. It includes both active and inactive vaccines available in the US.

# Standard Code Systems, Continued

---

**SNOMED.** Contains healthcare concepts with unique meanings and formal logic-based definitions organized into hierarchies. The codes represent clinical concepts across many domains, which includes conditions, diagnoses, symptoms, and signs, all of which are a type of finding. SNOMED also represents procedures, observations, and some laboratory tests, drugs, and devices. RCHC health centers and Relevant do not normally use these codes.

**Code on Dental Procedures and Nomenclature (CDT).** These codes assure consistency in documenting dental treatment and are commonly used for billing.

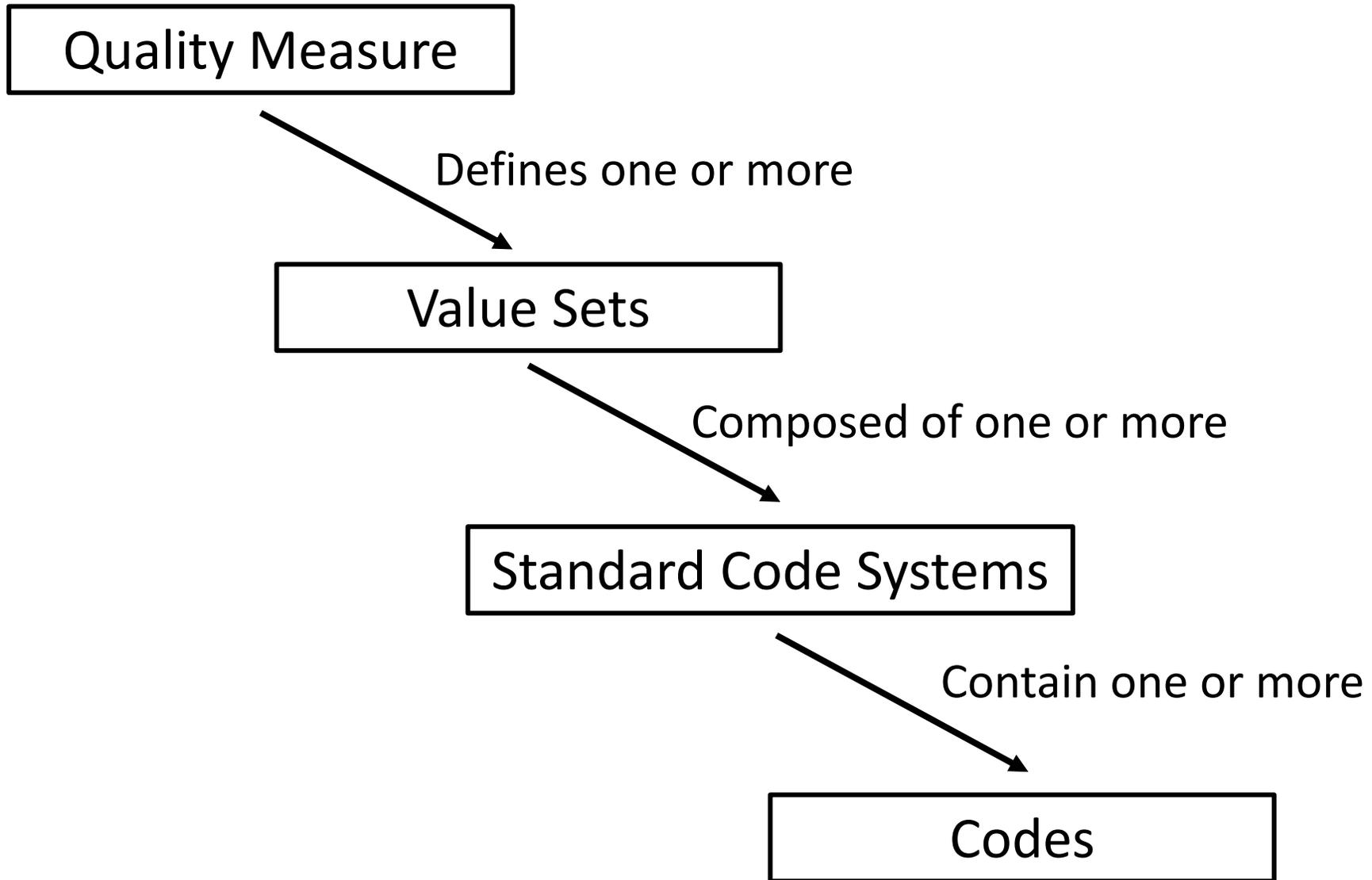
# Value Set Definitions

---

From the National Library of Medicine Value Set Authority Center:

“Value sets are lists of codes and corresponding terms, from NLM-hosted standard clinical vocabularies (such as SNOMED<sup>®</sup>, RxNorm, LOINC<sup>®</sup> and others), that define clinical concepts to support effective and interoperable health information exchange.”

(source: <https://vsac.nlm.nih.gov/>)



## PROGRAM ASSISTANCE LETTER

DOCUMENT NUMBER: 2021-05

DOCUMENT TITLE: Approved Uniform Data System Changes for Calendar Year 2022

1. Childhood Immunization Status has been revised to align with [CMS117v10](#).
2. Cervical Cancer Screening has been revised to align with [CMS124v10](#).
3. Breast Cancer Screening has been revised to align with [CMS125v10](#).
4. Weight Assessment and Counseling for Nutrition and Physical Activity for Children and Adolescents has been revised to align with [CMS155v10](#).
5. Preventive Care and Screening: Body Mass Index (BMI) Screening and Follow-Up Plan has been revised to align with [CMS69v10](#).
6. Preventive Care and Screening: Tobacco Use: Screening and Cessation Intervention has been revised to align with [CMS138v10](#).
7. Statin Therapy for the Prevention and Treatment of Cardiovascular Disease has been revised to align with [CMS347v5](#).
8. Colorectal Cancer Screening has been revised to align with [CMS130v10](#).
9. HIV Screening has been revised to align with [CMS349v4](#).
10. Preventive Care and Screening: Screening for Depression and Follow-Up Plan has been revised to align with [CMS2v11](#).
11. Depression Remission at Twelve Months has been revised to align with [CMS159v10](#).
12. Controlling High Blood Pressure has been revised to align with [CMS165v10](#).
13. Diabetes: Hemoglobin A1c (HbA1c) Poor Control (> 9%) has been revised to align with [CMS122v10](#).

# Find UDS Value Sets Through Documentation

Click on the hyperlink

# Childhood Immunization Status

Measure Information   **Specifications and Data Elements**   Release Notes

Click on this tab

## Specifications

Attachment	Size
 <a href="#">CMS117v10.html</a>	144.3 KB
 <a href="#">CMS117v10.zip (ZIP)</a>	156.47 KB
 <a href="#">CMS117v10-TRN.xlsx (Excel)</a>	20.59 KB

## Data Element Repository

[Data Elements contained within CMS117v10](#)

### Value Sets

[Value Sets to be used with CMS117v10](#)

Click on the hyperlink

Welcome

Search Value Sets

Download

Q Browse Code Systems

Help

Search the NLM Value Set Repository.

Program: All

Expansion Version: eCQM Update 2021-05-06

Refine by:

Steward

Code System

EP  
EH  
CMS eCQM ID (NQF Number)

Quality Data Model Category

Query: CMS117v10

Q Search

Clear

Search Results

API Resource

Results for All : eCQM Update 2021-05-06 : "CMS117v10"

Export Search Results

Select a hyperlinked OID to see its value set details.

OID

Matched Value Sets

Download View Toggle Clear

Page 1 of 4 20 View 1 - 20 of 62

<input type="checkbox"/>	Name	Code System	Definition Type	Steward	OID	Code Count
<input type="checkbox"/>	Anaphylactic Reaction to Common Baker's Yeast	SNOMEDCT	Grouping	NCQA	<a href="#">2.16.840.1.113883.3.464.1003.199.12.1032</a>	2
<input type="checkbox"/>	Anaphylactic Reaction to DTaP Vaccine	SNOMEDCT	Grouping	NCQA	<a href="#">2.16.840.1.113883.3.464.1003.199.12.1031</a>	4
<input type="checkbox"/>	Anti Hepatitis A IgG Antigen Test	LOINC	Grouping	NCQA	<a href="#">2.16.840.1.113883.3.464.1003.198.12.1033</a>	3
<input type="checkbox"/>	Anti Hepatitis B Virus Surface Ab	LOINC	Grouping	NCQA	<a href="#">2.16.840.1.113883.3.464.1003.198.12.1073</a>	6
<input type="checkbox"/>	Disorders of the Immune System	ICD10CM SNOMEDCT	Grouping	NCQA	<a href="#">2.16.840.1.113883.3.464.1003.120.12.1001</a>	149
<input type="checkbox"/>	DTaP Vaccine	CVX	Grouping	NCQA	<a href="#">2.16.840.1.113883.3.464.1003.196.12.1214</a>	6
<input type="checkbox"/>	DTaP Vaccine Administered	CPT SNOMEDCT	Grouping	NCQA	<a href="#">2.16.840.1.113883.3.464.1003.110.12.1022</a>	43

# Childhood Immunization Status

Measure Information   **Specifications and Data Elements**   Release Notes

Click on this tab

## Specifications

Attachment	Size
 <a href="#">CMS117v10.html</a>	144.3 KB
 <a href="#">CMS117v10.zip (ZIP)</a>	156.47 KB
 <a href="#">CMS117v10-TRN.xlsx (Excel)</a>	20.59 KB

Click on the hyperlink

## Data Element Repository

[Data Elements contained within CMS117v10](#)

## Value Sets

[Value Sets to be used with CMS117v10](#)

eCQM Title	Childhood Immunization Status		
eCQM Identifier (Measure Authoring Tool)	117	eCQM Version Number	10.0.000
NQF Number	Not Applicable	GUID	b2802b7a-3580-4be8-9458-921aea62b78c
Measurement Period	January 1, 20XX through December 31, 20XX		



**Scroll down to the bottom**

**Data Criteria (QDM Data Elements)**

- "Diagnosis: Neomycin adverse reaction (disorder)" using "Neomycin adverse reaction (disorder) (SNOMEDCT Code 292927007)"
- "Diagnosis: Anaphylactic Reaction to Common Baker's Yeast" using "Anaphylactic Reaction to Common Baker's Yeast (2.16.840.1.113883.3.464.1003.199.12.1032)"
- "Diagnosis: Anaphylactic Reaction to DTaP Vaccine" using "Anaphylactic Reaction to DTaP Vaccine (2.16.840.1.113883.3.464.1003.199.12.1031)"
- "Diagnosis: Encephalopathy due to Childhood Vaccination" using "Encephalopathy due to Childhood Vaccination (2.16.840.1.113883.3.464.1003.114.12.1007)"
- "Diagnosis: Hepatitis A" using "Hepatitis A (2.16.840.1.113883.3.464.1003.110.12.1024)"
- "Diagnosis: Hepatitis B" using "Hepatitis B (2.16.840.1.113883.3.464.1003.110.12.1025)"
- "Diagnosis: HIV" using "HIV (2.16.840.1.113883.3.464.1003.120.12.1003)"
- "Diagnosis: Intussusception" using "Intussusception (2.16.840.1.113883.3.464.1003.199.12.1056)"

**OID**

# Finding Value Sets for Quality Measures

---

RCHC: “Instructions for Using the Relevant Validation Report Set (2021)”

*Includes UDS and QIP Quality Measures*

## **Appendix C: List of Value Sets Used in Quality Measures**

2021 Quality Measure	eCQM Value Set Description	eCQM OID	Value Set Type
Breast Cancer Screening	History of bilateral mastectomy	2.16.840.1.113883.3.464.1003.198.12.1068	Diagnosis
	Status Post Left Mastectomy	2.16.840.1.113883.3.464.1003.198.12.1069	Diagnosis
	Status Post Right Mastectomy	2.16.840.1.113883.3.464.1003.198.12.1070	Diagnosis
	Unilateral Mastectomy, Unspecified Laterality	2.16.840.1.113883.3.464.1003.198.12.1071	Diagnosis
Cervical Cancer Screening	HPV Test	2.16.840.1.113883.3.464.1003.110.12.1059	Labs
	Pap Test	2.16.840.1.113883.3.464.1003.108.12.1017	Labs
	Hysterectomy with No Residual Cervix	2.16.840.1.113883.3.464.1003.198.12.1014	Diagnosis
	Congenital or Acquired Absence of Cervix	2.16.840.1.113883.3.464.1003.111.12.1016	Diagnosis
Colorectal Cancer Screening	Malignant Neoplasm of Colon	2.16.840.1.113883.3.464.1003.108.12.1001	Diagnosis
	Fecal Occult Blood Test (FOBT)	2.16.840.1.113883.3.464.1003.198.12.1011	Labs
	FIT DNA	2.16.840.1.113883.3.464.1003.108.12.1039	Labs

# Finding Value Sets for Quality Measures

---

RCHC: “QIP 2021 Quality Measure and Report Set Instructions (2021)”

*Includes only QIP Quality Measures – HEDIS and eCQM Value Sets*

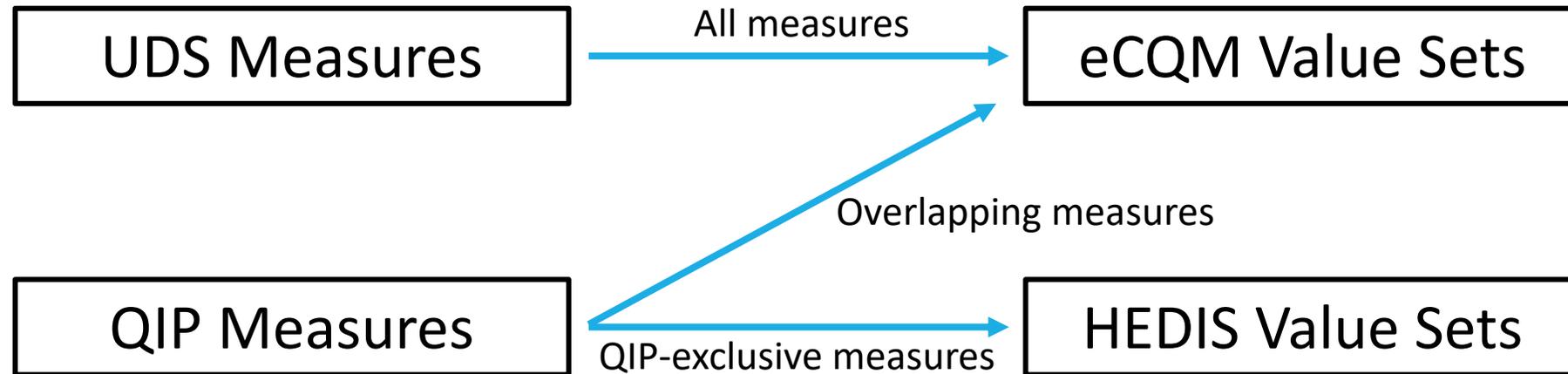
## **Appendix: Value Sets Associated With the QIP Measures**

### **Current QIP Measures Attached to HEDIS Value Sets**

Quality Measure Name	Importer	HEDIS Value Set Name	HEDIS OID	Value Set Type
Well-Child Visits in the First 15 Months of Life (QIP 2021)	well_child_interventions	Well-Care (see Note #7)	2.16.840.1.113883.3.464.1004.1262	Procedures and Diagnosis
Child and Adolescent Well-Care Visits (QIP 2021)	well_child_interventions	Well-Care (see Note #7)	2.16.840.1.113883.3.464.1004.1262	Procedures and Diagnosis
Counseling for Nutrition for Children/Adolescents (QIP 2021)	nutrition_counselings	Nutrition Counseling (see Note #9)	2.16.840.1.113883.3.464.1004.1190	Procedures and Diagnosis

# Relationship of Measures to Value Sets in Relevant

---



# How to Find Value Set Codes in Relevant

---

Relevant report name “RCHC List of QM Value Set Codes”

- Displays all Value Set codes associated with the UDS and QIP Quality Measures, whether or not they are associated with codes ever used in the EHR.
- Shows the most recent (i.e., latest = TRUE) Value Set codes for diagnosis (ICD-9 and ICD-10), labs (LOINC), medications (RXNORM), procedures (CPT), and vaccines (CVX).

# How to Find Value Set Codes in Relevant

---

- The report “RCHC List of QM Value Set Codes” displays all codes, but there are also individual reports for labs (RCHC List of QM Lab Names and Attributes), medications (RCHC List of QM Medications) and vaccines (RCHC List of QM Vaccines)
- These individual reports also show the number of times the code was used (if at all) and the last date it was used

# Value Sets Present in Relevant

---

## Table Names of **Current** Value Sets

- `cqm_value_set_codes` (Staging and Production)
- `hedis_value_set_codes` (Staging only)

## Table Names of **Non-Current** Data Sets

- `cms_value_set_codes` (Staging and Production). Latest values are from 2020.
- `relevant_qip_2019_value_set` (Staging only). Latest values are from 2020. The HEDIS table replaced this one.

# Value Sets Present in Relevant

---

- The “current” Value Set tables are updated every year
- The update happens in the Summer or when the new Quality Measures are released
- The newest records are appended to the table with the older records. There is a column named “Latest” that is set to TRUE for the most recent Value Set records
- No changes to Transformers, Data Elements, Reports, etc. that use the Value Sets are necessary

# Note on “relevant\_qip\_2019\_value\_set”

## Data Analytics and Governance

### PROGRAM

#### Data Workgroup and Report Documentation:

This is RCHC's longest running data peer group and historically is the forum where RCHC has shared measure documentation, how to use new standard reports, validation practices, benchmarking and assistance with questions around standard report sets such as the Uniform Data System (UDS Federal Reporting) and Office of Statewide Health Planning and Development (OSHPD State of California Reporting), as well as other shared standardized reports.



Additional Resources and Companion Documents

If you need a password for the locked content please contact us.

- Partnership QIP Reporting
  - Switching\_Value\_Sets\_Programmers(slides 11/2021)
  - Recording 2021 QIP Report Set (12/2021)
  - Relevant 2021 QIP Report Set (slides, 5/2021)
  - Relevant\_QIP\_Report\_Set\_v2(12/ 2021)
  - Reporting 2018 QIP Measures v1 (1/2019)

Scroll  
down  
to get  
to this  
section

- The table “hedis\_value\_set\_codes” replaced the 2019 table
- The 2019 Value Sets should really not be used in Relevant
- This presentation describes the switch in more detail

# Characteristics of Current Value Set Tables Present in Relevant

---

*These are the fields present on both tables (also see the accompanying handout)*

cqm_value_set_codes	hedis_value_set_codes
id	id
value_set_oid	value_set_oid
value_set_name	value_set_name
value_set_version	value_set_version
code_value	code_value
code_description	code_description
code_system_name	code_system_name
code_system_version	code_system_version
code_system_oid	code_system_oid
	med_category
	med_route
purpose	
latest	latest

# Characteristics of Current Value Set Tables Present in Relevant

*These code types are present in the "latest" Value Sets*

cqm_value_set_codes	hedis_value_set_codes	Relevant Use Case
AdministrativeGender		--
CDCREC		--
CDT		Dental procedures
CPT	CPT	Medical procedures
	CPT-CAT-II	--
CVX	CVX	Immunizations
HCPCS Level II	HCPCS	Equipment
HSLOC		--
ICD10CM	ICD10CM	Diagnosis
ICD10PCS	ICD10PCS	--
ICD9CM	ICD9CM	Diagnosis (old version)
	ICD9PCS	--
LOINC	LOINC	Used for labs
	Modifier	--
	NDC	Medications
RXNORM		Medications
	POS	--
SNOMEDCT	SNOMED CT US Edition	--
	UBREV	--
	UBTOB	--
SOP		--
1187	420	Total Value Sets

# How Value Sets Are Used in Relevant

---

Value Sets can be used whenever you need to define a particular concept. Therefore, they can be used in:

- Transformers (probably the most common place)
- Importers
- Reports
- Care Gaps
- Ad hoc use (research, QI projects, etc.)

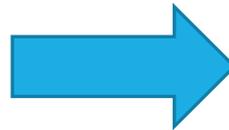
# How to Find a Browse Value Sets

---

Example: you are planning new Transformers for the Partnership ECDS HEDIS measure “Alcohol Screening and Follow-up”

Generally, look for any Value Sets with ‘alcohol’ in the Value Set name

```
SELECT DISTINCT value_set_name
FROM hedis_value_set_codes
WHERE latest = 'TRUE'
  AND value_set_name ILIKE '%Alcohol%'
ORDER BY value_set_name
```



	value_set_name
1	Alcohol Abuse and Dependence
2	Alcohol Counseling or Other Follow Up Care
3	Alcohol Disorders
4	Alcohol Use Disorder
5	Alcohol Withdrawal

# Exercise #1

---

A HEDIS measure you are working on has an exclusion for patients with alcohol use disorder. This is a diagnosis. How could you use SQL code to view all of the ICD-10 diagnosis codes in this Value Set?

Here are the fields on `hedis_value_set_codes`:

(Hint: which fields should go in the SELECT statement and which fields should go in the WHERE statement?)

id
value_set_name
value_set_oid
value_set_version
code_value
code_description
code_system_name
code_system_oid
code_system_version
med_category
med_route
latest

# Answer to Exercise #1

```
SELECT DISTINCT value_set_name,  
               code_system_name,  
               code_value,  
               code_description  
FROM hedis_value_set_codes  
WHERE latest = 'TRUE'  
      AND value_set_name = 'Alcohol Use Disorder'  
      AND code_system_name = 'ICD10CM'  
ORDER BY value_set_name, code_system_name,  
         code_value, code_description
```



	value_set_name	code_system_name	code_value	code_description
1	Alcohol Use Disorder	ICD10CM	F10.10	[F10.10] Alcohol abuse, uncomplicated
2	Alcohol Use Disorder	ICD10CM	F10.120	[F10.120] Alcohol abuse with intoxication, uncomplicated
3	Alcohol Use Disorder	ICD10CM	F10.121	[F10.121] Alcohol abuse with intoxication delirium

# How Value Sets Are Used in Relevant

---

We will look at examples of how to join the Value Set tables to other Relevant tables. The code systems we will cover are:

- ICD
- CPT/HCPCS
- LOINC
- CVX
- RxNorm/NDC

# Key Fields in the Value Set Tables

---

Value Set Tables: **cqm\_value\_set\_codes** and **hedis\_value\_set\_codes**

- Field **code\_value**: contains the specific codes. Join to this field.
- Field **value\_set\_oid**: use to identify the specific code set.
- Field **code\_system\_name**: use to specify the code system (optional)

# General Approach to JOINing

---

What kind of data is in the Relevant table that you are intending to join to?

- A field with the raw code (e.g., the actual ICD code itself)
- An ID field that connects to another table which features a field that contains the raw code
- A Value Set OID field

# Connecting to Diagnosis Codes: Problem List

---

If your health center has the standard Transformer **relevant\_cases**, you should be able to JOIN through the Value Set OID

Example uses the eCQM Value Set for essential hypertension

```
SELECT DISTINCT  
  patientid  
FROM relevant_cases  
  INNER JOIN cqm_value_set_codes ON cqm_value_set_codes.value_set_oid = relevant_cases.value_set_oid  
    AND cqm_value_set_codes.latest = TRUE  
    AND cqm_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1003.104.12.1011'
```

# Connecting to Diagnosis Codes: Problem List (eCW)

---

*Example for Alcohol Use Disorder Using the HEDIS Value Set*

```
SELECT DISTINCT
    problemlist.patientid
FROM problemlist
    INNER JOIN itemdetail ON itemdetail.itemID = problemlist.asmtid AND propID = 13
    INNER JOIN hedis_value_set_codes
        ON hedis_value_set_codes.code_value = itemdetail.value
        AND hedis_value_set_codes.latest = 'TRUE'
        AND hedis_value_set_codes.value_set_oid =
            '2.16.840.1.113883.3.464.1004.1339'
```

In other words, itemdetail.value contains the diagnosis code when  
itemdetail.propID = 13

# Connecting to Diagnosis Codes: Problem List (NextGen)

---

**SELECT DISTINCT**

patient\_diagnosis.person\_id

**FROM** patient\_diagnosis

**INNER JOIN** person\_surrogate\_key **ON** person\_surrogate\_key.person\_id = patient\_diagnosis.person\_id

**INNER JOIN** hedis\_value\_set\_codes

**ON** hedis\_value\_set\_codes.code\_value = patient\_diagnosis.diagnosis\_code\_id

**AND** hedis\_value\_set\_codes.latest = 'TRUE'

**AND** hedis\_value\_set\_codes.value\_set\_oid = '2.16.840.1.113883.3.464.1004.1339'

**SELECT DISTINCT**

patient\_problems.person\_id

**FROM** patient\_problems

**INNER JOIN** hedis\_value\_set\_codes

**ON** hedis\_value\_set\_codes.code\_value = patient\_problems.concept\_id

**AND** hedis\_value\_set\_codes.latest = 'TRUE'

**AND** hedis\_value\_set\_codes.value\_set\_oid = '2.16.840.1.113883.3.464.1004.1339'

*NOTE that concept\_id is a SNOMED code*

# Connecting to Diagnosis Codes: Assessments (eCW)

---

**SELECT DISTINCT**

relevant\_visits.patient\_id

**FROM** relevant\_visits

**INNER JOIN** diagnosis **ON** diagnosis.EncounterId = relevant\_visits.id

**INNER JOIN** itemdetail **ON** itemdetail.itemid = diagnosis.itemid **AND** propID = 13

**INNER JOIN** hedis\_value\_set\_codes

**ON** hedis\_value\_set\_codes.code\_value = itemdetail.value

**AND** hedis\_value\_set\_codes.latest = 'TRUE'

**AND** hedis\_value\_set\_codes.value\_set\_oid = '2.16.840.1.113883.3.464.1004.1339'

If your instance of Relevant has it, you might also be able to use the Transformer  
relevant\_visit\_diagnosis\_codes

# Connecting to Diagnosis Codes: Claim Diagnosis (eCW)

---

**SELECT DISTINCT**

edi\_invoice.**patientid**

**FROM** edi\_invoice

**INNER JOIN** edi\_inv\_diagnosis **ON** edi\_inv\_diagnosis.**invoiceid** = edi\_invoice.**id**

**INNER JOIN** hedis\_value\_set\_codes **ON** hedis\_value\_set\_codes.**code\_value** = edi\_inv\_diagnosis.**code**

**AND** hedis\_value\_set\_codes.**latest** = **'TRUE'**

**AND** hedis\_value\_set\_codes.**value\_set\_oid** = **'2.16.840.1.113883.3.464.1004.1339'**

**WHERE** edi\_invoice.**deleteflag** = 0

**AND** edi\_invoice.**voidflag** = 0

# Connecting to Diagnosis Codes: Encounter Diagnosis (NextGen)

---

**SELECT DISTINCT**

    encounter\_diags.**person\_id**

**FROM** encounter\_diags

**INNER JOIN** hedis\_value\_set\_codes

**ON** hedis\_value\_set\_codes.**code\_value** = encounter\_diags.**icd9cm\_code\_id**

**AND** hedis\_value\_set\_codes.**latest** = 'TRUE'

**AND** hedis\_value\_set\_codes.**value\_set\_oid** = '2.16.840.1.113883.3.464.1004.1339'

# Connecting to Procedure Codes

---

If your health center has the standard Transformer **relevant\_visit\_billing\_codes**, you should be able to JOIN through the code field

Example using HEDIS Value Set for Well Child Visits

Note that CPT and HCPCS codes can appear on a claim

## **SELECT DISTINCT**

relevant\_visits.**patient\_id**

**FROM** relevant\_visit\_billing\_codes

**INNER JOIN** relevant\_visits **ON** relevant\_visits.**id** = relevant\_visit\_billing\_codes.**visit\_id**

**INNER JOIN** hedis\_value\_set\_codes

**ON** hedis\_value\_set\_codes.**code\_value** = *LEFT*(relevant\_visit\_billing\_codes.**code**, 5)

**AND** hedis\_value\_set\_codes.**latest** = 'TRUE'

**AND** hedis\_value\_set\_codes.**value\_set\_oid** = '2.16.840.1.113883.3.464.1004.1262'

# Connecting to Procedure Codes (eCW)

---

## SELECT DISTINCT

relevant\_patients.id AS patient\_id

FROM relevant\_patients

INNER JOIN edi\_invoice ON edi\_invoice.patientid = relevant\_patients.id

AND edi\_invoice.deleteflag = 0

AND edi\_invoice.voidflag = 0

INNER JOIN edi\_inv\_cpt ON edi\_inv\_cpt.invoiceid = edi\_invoice.id

INNER JOIN hedis\_value\_set\_codes

ON hedis\_value\_set\_codes.code\_value = LEFT(edi\_inv\_cpt.code, 5)

AND hedis\_value\_set\_codes.latest = 'TRUE'

AND hedis\_value\_set\_codes.value\_set\_oid = '2.16.840.1.113883.3.464.1004.1262'

# Connecting to Procedure Codes (NextGen)

---

```
SELECT DISTINCT
    claims.person_id
FROM claims
    INNER JOIN charges ON charges.source_id = claims.enc_id
    INNER JOIN hedis_value_set_codes
        ON hedis_value_set_codes.code_value = LEFT(charges.cpt4_code_id, 5)
        AND hedis_value_set_codes.latest = 'TRUE'
        AND hedis_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1004.1262'
```

# Connecting to Lab Codes

---

If your health center has the standard Transformer **relevant\_labs**, you should be able to JOIN through the loinc\_code field

Example using eCQM Value Set for HIV Laboratory Test Codes (Ab and Ag)

**SELECT DISTINCT**

relevant\_labs.person\_id

**FROM** relevant\_labs

**INNER JOIN** cqm\_value\_set\_codes **ON** cqm\_value\_set\_codes.code\_value = relevant\_labs.loinc\_code

**AND** cqm\_value\_set\_codes.latest = **TRUE**

**AND** cqm\_value\_set\_codes.value\_set\_oid = '2.16.840.1.113762.1.4.1056.50'

# Connecting to Lab Codes (eCW)

---

**SELECT DISTINCT**

relevant\_visits.patient\_id

**FROM** labdata

**INNER JOIN** relevant\_visits **ON** relevant\_visits.id = labdata.encounterid

**INNER JOIN** labdatadetail **AS** ldd **ON** ldd.reportid = labdata.reportid

**INNER JOIN** labloinccodes **ON** labloinccodes.itemid = ldd.propid **AND** labloinccodes.deleteflag = 0

**INNER JOIN** cqm\_value\_set\_codes **ON** cqm\_value\_set\_codes.code\_value = labloinccodes.code

**AND** cqm\_value\_set\_codes.latest = **TRUE**

**AND** cqm\_value\_set\_codes.value\_set\_oid = '2.16.840.1.113762.1.4.1056.50'

# Connecting to Lab Codes (NextGen)

---

A bit more complex because of more than one table with lab codes. Refer to the standard you use at your health center. JOIN the Value Set table through the LOINC field

- `lab_order_tests.loinc_code`
- `lab_results_obr_p.loinc_code`
- `lab_results_obx.loinc_code`

It is probably better to just use `relevant_labs` (see previous slide)

# Connecting to Immunization Codes

---

If your health center has the standard Transformer **relevant\_immunizations**, you should be able to JOIN through the **cvx\_code** field

Example using eCQM Value Set for DTaP Vaccines

**SELECT**

relevant\_immunizations.**patient\_id**

**FROM** relevant\_immunizations

**INNER JOIN** cqm\_value\_set\_codes **ON** cqm\_value\_set\_codes.**code\_value** =

relevant\_immunizations.**cvx\_code** **:: VARCHAR**

**AND** cqm\_value\_set\_codes.**latest** = **TRUE**

**AND** cqm\_value\_set\_codes.**value\_set\_oid** = **'2.16.840.1.113883.3.464.1003.196.12.1214'**

# Connecting to Immunization Codes (eCW)

---

**SELECT DISTINCT**

**patientid**

**FROM** immunizations

**INNER JOIN** cqm\_value\_set\_codes **ON** cqm\_value\_set\_codes.**code\_value** = immunizations.**cvx\_code**

**AND** cqm\_value\_set\_codes.**latest** = **TRUE**

**AND** cqm\_value\_set\_codes.**value\_set\_oid** = '2.16.840.1.113883.3.464.1003.196.12.1214'

# Connecting to Immunization Codes (NextGen)

---

```
SELECT DISTINCT
    imm_nor.person_id
FROM imm_order_vaccines
    INNER JOIN imm_nor ON imm_order_vaccines.order_num = imm_nor.order_num
    INNER JOIN cqm_value_set_codes
        ON cqm_value_set_codes.code_value = imm_order_vaccines.cvx_code :: VARCHAR
        AND cqm_value_set_codes.latest = TRUE
        AND cqm_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1003.196.12.1214'
```

# Connecting to Medication Codes

---

- It is difficult to JOIN to raw medications tables directly
- Use the table relevant\_medications and the field value\_set\_oids
- Therefore, you just need the eCQM OID (and not the eCQM Value Set table)
- Note that the value\_set\_oids field can have more than one Value Set (in other words, it is aggregated)

```
value_set_oids
```

```
{2.16.840.1.113883.3.526.1577,2.16.840.1.113883.3.526.3.1139,2.16.840.1.113883.3.600.1476}
```

```
{2.16.840.1.113762.1.4.1045.5,2.16.840.1.113883.3.464.1003.196.12.1001,2.16.840.1.113883.3.666.5.840}
```

```
{2.16.840.1.113883.3.464.1003.196.12.1253,2.16.840.1.113883.3.464.1003.196.12.1372}
```

# Connecting to Medication Codes

---

```
SELECT DISTINCT
    patient_id
FROM relevant_medications
WHERE value_set_oids && '{2.16.840.1.113883.3.526.3.1190}'
```

You can add more than one Value Set by adding a comma between the set OIDs.  
For example,

```
WHERE value_set_oids && '{2.16.840.1.113883.3.526.3.1572,
    2.16.840.1.113883.3.526.3.1574,
    2.16.840.1.113883.3.526.3.1575}'
```

# Practice writing SQL using Value Sets:

## Exercise #2

---

- A provider wants to get some data, so you will write an Ad Hoc report. The request is for a list of pneumococcal vaccine names and the number of times each was administered from May 1, 2021 to April 30, 2022
- The provider did not give many details about the vaccine beyond referring to it as “a type of pneumococcal vaccine” and “PPSV23”
- When you research it on the Internet, you find out that this type of vaccine is sometimes referred to as “Pneumococcal Polysaccharide 23 Vaccine” or “Pneumovax”

# Answer to Exercise #2

---

**Step #1:** Find the Value Set(s) referring to pneumococcal vaccines.

- You can decide to look at the HEDIS Value Set codes or the eCQM Value Set codes
- You can decide to see all of the code types or only CVX codes (you will eventually JOIN by CVX code, but you might be curious what else is in the Value Set)

# Answer to Exercise #2 (Step #1)

---

## HEDIS

```
SELECT DISTINCT value_set_name,  
               value_set_oid,  
                code_system_name,  
                code_value,  
                code_description  
FROM hedis_value_set_codes  
WHERE latest = 'TRUE'  
      AND value_set_name ILIKE '%pneumococc%'  
      AND code_system_name = 'CVX'  
ORDER BY value_set_name, code_system_name,  
         code_value, code_description
```

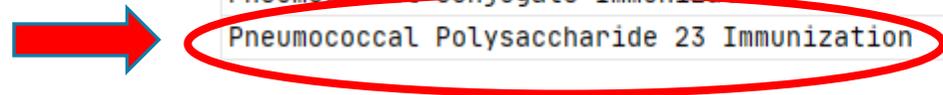
## eCQM (example without CVX restriction)

```
SELECT DISTINCT value_set_name,  
               value_set_oid,  
                code_system_name,  
                code_value,  
                code_description  
FROM cqm_value_set_codes  
WHERE latest = 'TRUE'  
      AND value_set_name ILIKE '%pneumococc%'  
ORDER BY value_set_name, code_system_name,  
         code_value, code_description
```

# Answer to Exercise #2

---

Let's use the HEDIS Value Set OIDs for the rest of the example. The approach is the same as using the eCQM Value Set OID



value_set_name	value_set_oid	code_system_name
<del>Pneumococcal Conjugate Immunization</del>	2.16.840.1.113883.3.464.1004.1781	CVX
Pneumococcal Polysaccharide 23 Immunization	2.16.840.1.113883.3.464.1004.1921	CVX

```
hedis_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1004.1921'
```

# Answer to Exercise #2

---

- **Step #2:** Use the Value Set OID to identify the appropriate vaccines in your system
- Write the rest of the query. Remember, you need to list the vaccine names and count the number of times they were used
- You also need to restrict the data to a date range (May 1, 2021 to April 30, 2022)

# Answer to Exercise #2 (eCW)

---

**SELECT DISTINCT**

**vaccinename,**  
**count(patientid) AS count**

**FROM** immunizations

**INNER JOIN** hedis\_value\_set\_codes **ON**

hedis\_value\_set\_codes.**code\_value** = immunizations.**cvx\_code**

**AND** hedis\_value\_set\_codes.**latest** = **'TRUE'**

**AND** hedis\_value\_set\_codes.**value\_set\_oid** =  
**'2.16.840.1.113883.3.464.1004.1921'**

**WHERE** **givendate** **BETWEEN** **'2021-05-01'** **AND** **'2022-04-30'**

**GROUP BY** **vaccinename**

# Answer to Exercise #2 (NextGen – but using relevant\_immunizations)

---

**SELECT DISTINCT**

**vaccine\_name,**  
**count(patient\_id) AS count**

**FROM** relevant\_immunizations

**INNER JOIN** hedis\_value\_set\_codes **ON** hedis\_value\_set\_codes.**code\_value** =  
relevant\_immunizations.**cvx\_code** :: **VARCHAR**

**AND** hedis\_value\_set\_codes.**latest** = **'TRUE'**

**AND** hedis\_value\_set\_codes.**value\_set\_oid** =  
**'2.16.840.1.113883.3.464.1004.1921'**

**WHERE** **applied\_on** **BETWEEN** **'2021-05-01'** **AND** **'2022-04-30'**

**GROUP BY** **vaccine\_name**

# Practice writing SQL using Value Sets: Exercise #3 (Time Permitting)

---

- You need a new Transformer for Alcohol Counseling
- Your initial search showed that there is a HEDIS Value Set called “Alcohol Counseling or Other Follow Up Care”
- The Transformer should display the following fields:
  1. patientid
  2. performed\_on (date format)

# Answer to Exercise #3

---

**Step #1:** Find and explore the Value Set

```
SELECT DISTINCT value_set_name,  
               value_set_oid,  
               code_system_name,  
               code_value,  
               code_description  
FROM hedis_value_set_codes  
WHERE latest = 'TRUE'  
      AND value_set_name = 'Alcohol Counseling or Other Follow Up Care'  
ORDER BY value_set_name, code_system_name, code_value, code_description
```

- value\_set\_oid = '2.16.840.1.113883.3.464.1004.1437'
- This Value Set contains CPT, HCPCS, ICD-10, and SNOMED codes
- Think: what tables in Relevant would be joined to these codes?

# Answer to Exercise #3 (eCW)

---

**Step #2:** Link the Value Set codes to billed CPT codes and assessment ICD-10 codes

**SELECT**

relevant\_patients.**id** **AS** patient\_id,  
**servicdt** **:: DATE AS** performed\_on

**FROM** relevant\_patients

**INNER JOIN** edi\_invoice **ON** edi\_invoice.**patientid** = relevant\_patients.**id**

**AND** edi\_invoice.**deleteflag** = 0

**AND** edi\_invoice.**voidflag** = 0

**INNER JOIN** edi\_inv\_cpt **ON** edi\_inv\_cpt.**invoiceid** = edi\_invoice.**id**

**INNER JOIN** hedis\_value\_set\_codes

**ON** hedis\_value\_set\_codes.**code\_value** = *LEFT*(edi\_inv\_cpt.**code**, 5)

**AND** hedis\_value\_set\_codes.**latest** = 'TRUE'

**AND** hedis\_value\_set\_codes.**value\_set\_oid** = '2.16.840.1.113883.3.464.1004.1437'

**AND** hedis\_value\_set\_codes.**code\_system\_name** = 'CPT'

Billed CPT codes



Restricting for CPT is optional. There could theoretically be HCPCS codes as well (?)

# Answer to Exercise #3 (eCW)

---

**SELECT**

relevant\_visits.patient\_id,  
relevant\_visits.visit\_date :: DATE AS performed\_on

Assessment ICD-10 codes

**FROM** relevant\_visits

**INNER JOIN** diagnosis **ON** diagnosis.EncounterId = relevant\_visits.id

**INNER JOIN** itemdetail **ON** itemdetail.itemid = diagnosis.itemid **AND** propID = 13

**INNER JOIN** hedis\_value\_set\_codes

**ON** hedis\_value\_set\_codes.code\_value = itemdetail.value

**AND** hedis\_value\_set\_codes.latest = 'TRUE'

**AND** hedis\_value\_set\_codes.value\_set\_oid = '2.16.840.1.113883.3.464.1004.1437'

**AND** hedis\_value\_set\_codes.code\_system\_name = 'ICD10CM'

# Answer to Exercise #3 (eCW)- Together

```
SELECT DISTINCT *
FROM (SELECT
  relevant_patients.id AS patient_id,
  servicedt :: DATE AS performed_on
FROM relevant_patients
  INNER JOIN edi_invoice ON edi_invoice.patientid = relevant_patients.id
  AND edi_invoice.deleteflag = 0
  AND edi_invoice.voidflag = 0
  INNER JOIN edi_inv_cpt ON edi_inv_cpt.invoiceid = edi_invoice.id
  INNER JOIN hedis_value_set_codes
  ON hedis_value_set_codes.code_value = LEFT(edi_inv_cpt.code, 5)
  AND hedis_value_set_codes.latest = 'TRUE'
  AND hedis_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1004.1437'
  AND hedis_value_set_codes.code_system_name = 'CPT'
UNION
SELECT
  relevant_visits.patient_id,
  relevant_visits.visit_date :: DATE AS performed_on
FROM relevant_visits
  INNER JOIN diagnosis ON diagnosis.EncounterId = relevant_visits.id
  INNER JOIN itemdetail ON itemdetail.itemid = diagnosis.itemid AND propID = 13
  INNER JOIN hedis_value_set_codes
  ON hedis_value_set_codes.code_value = itemdetail.value
  AND hedis_value_set_codes.Tatest = 'TRUE'
  AND hedis_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1004.1437'
  AND hedis_value_set_codes.code_system_name = 'ICD10CM') AS alc_counsel_temp
```

# Answer to Exercise #3 (NextGen)

```
SELECT DISTINCT *
FROM(SELECT
  person_surrogate_key.person_key AS patient_id,
  patient_encounter.enc_timestamp :: DATE AS performed_on
FROM patient_encounter
INNER JOIN charges ON charges.source_id = patient_encounter.enc_id
INNER JOIN person_surrogate_key ON person_surrogate_key.person_id = patient_encounter.person_id
INNER JOIN hedis_value_set_codes
  ON hedis_value_set_codes.code_value = charges.service_item_id
  AND hedis_value_set_codes.latest = 'TRUE'
  AND hedis_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1004.1437'
  AND hedis_value_set_codes.code_system_name = 'CPT'
UNION
SELECT
  person_surrogate_key.person_key AS patient_id,
  patient_encounter.enc_timestamp :: DATE AS performed_on
FROM encounter_diags
INNER JOIN person_surrogate_key ON person_surrogate_key.person_id = encounter_diags.person_id
INNER JOIN patient_encounter ON patient_encounter.enc_id = encounter_diags.enc_id
INNER JOIN hedis_value_set_codes
  ON hedis_value_set_codes.code_value = encounter_diags.icd9cm_code_id
  AND hedis_value_set_codes.latest = 'TRUE'
  AND hedis_value_set_codes.value_set_oid = '2.16.840.1.113883.3.464.1004.1437'
  AND hedis_value_set_codes.code_system_name = 'ICD10CM') AS alc_counsel_temp
```

# Questions?

---

# Part 2: Utilizing SQL Code from Other Health Centers

---

FOR INTERMEDIATE USERS

# Resources Available

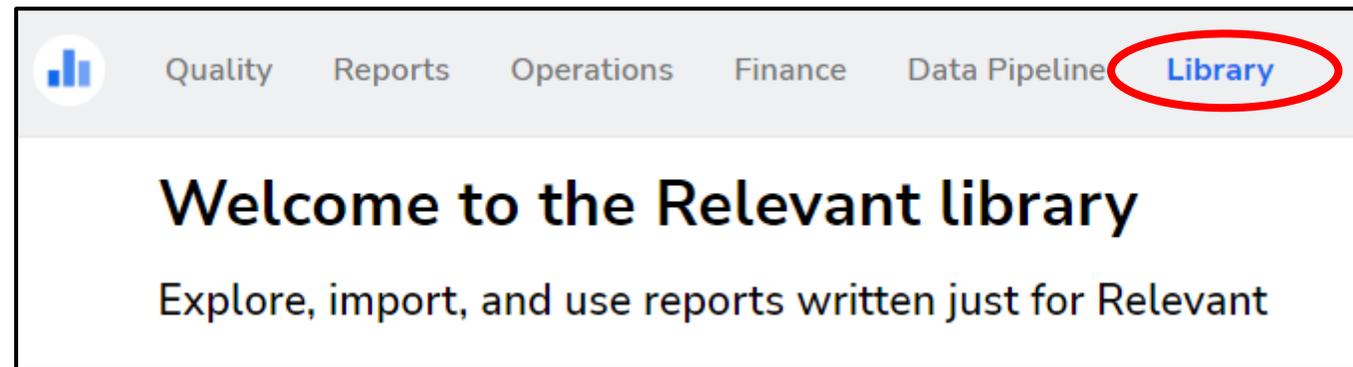
---

- The Relevant Library contains reports that have been curated by the Relevant team. Many seem to be developed by Relevant, but some might have been originally developed (or at least requested) by Relevant users nationally
- The RCHC Aggregate contains SQL for many types of objects that exist on member instances. They have been developed by the health center programmers or requested from Relevant. In either case, you may need to modify the code and then validate it

# How to Find Reports: Relevant Library

---

## The Relevant Library



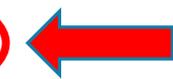
Click here



## Scroll or search for key words

Search reports

Imported status  All  Imported  Not imported



# Import From the Relevant Library

---

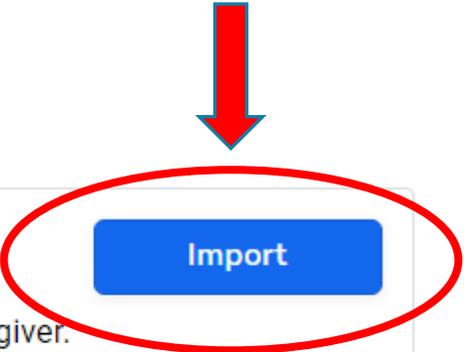
- Import the one you want from the library
- You may need to modify it

Patient Visits with Clinical Team (PCMH) ✓

Relevant database

Percentage of medical visits (using the UDS medical visit definition) in the selected date range that are with the patient's primary care giver.

If Provider Teams are mapped, also displays the percentage of said visits that are with the patient's primary care giver's team.



Import

# How to Find More in the RCHC Aggregate

---

- You need a password from RCHC to get in
- Health centers have a limited number of users who can access it (i.e., 10 users)
- If you think you will need to access the RCHC Aggregate frequently, let your supervisor know (or whoever is managing Relevant data access at your health center)
- Requests can be directed to [rchcanalytics@rchc.net](mailto:rchcanalytics@rchc.net)
- Otherwise, for rare access, you can ask someone who has access to the aggregate to search for you

# Health Center SQL Available For

---

- Reports\*
- Care Gaps\*
- Custom Quality Measures\*
- Populations
- Transformers
- Importers (AKA, Data Elements)

\* These topics come in a pair of reports. The other two are single reports

# “Pair of Reports” Approach

---

- There are dozens or perhaps hundreds of reports, care gaps and measures
- The first report of the pair helps to identify the object you are interested in. The data is organized by health center
- The second report displays the (semi-formatted) SQL code of the particular item you need on a tab named “SQL (copy and paste)”
- Examples are displayed on the following slides

# “Report on Reports”

---

- Lists report name and description, by health center
- You can use the browser find function (e.g., Ctrl-F in Windows) to search for key terms
- Does not show the SQL
- Copy the report names you are interested in on one tab and then go to the next tab

AVH	AVHC	CHA	CommuniCare	LVHC	Marin City	MCC	Ole	Petaluma	Ritter	Santa Rosa
name		description								
ACP - Advance care plan report		Most recent ACP statuses for all patients.								
Acupuncture Patient Report (DRAFT)		Patients with 2 or more visits at LVHC Acupuncture facility in the last 3 years								

# “SQL from Health Center Report(s)”

---

Steps:

1. Search for and select the name of the report you want in the Parameters

Parameters

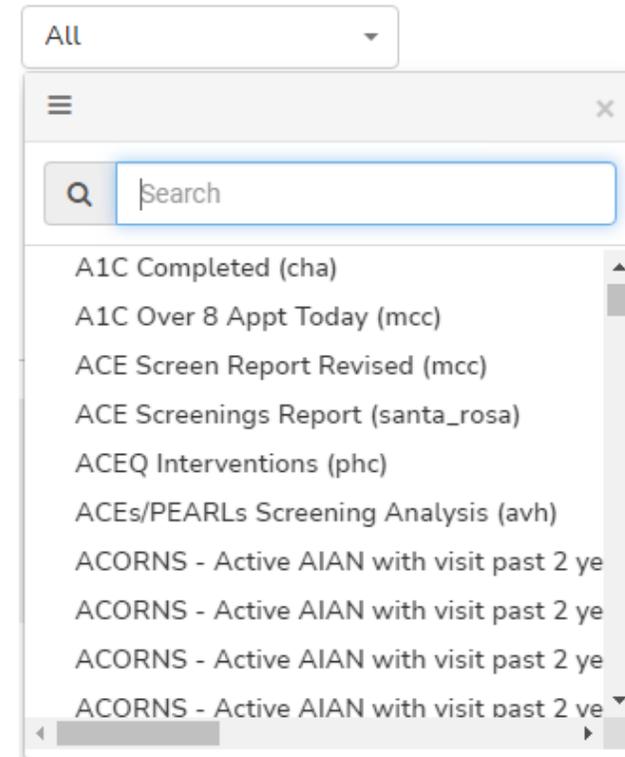
Report

All ▾

☰ ×

🔍 Search

- A1C Completed (cha)
- A1C Over 8 Appt Today (mcc)
- ACE Screen Report Revised (mcc)
- ACE Screenings Report (santa\_rosa)
- ACEQ Interventions (phc)
- ACEs/PEARLs Screening Analysis (avh)
- ACORNS - Active AIAN with visit past 2 ye
- ACORNS - Active AIAN with visit past 2 ye
- ACORNS - Active AIAN with visit past 2 ye
- ACORNS - Active AIAN with visit past 2 ye

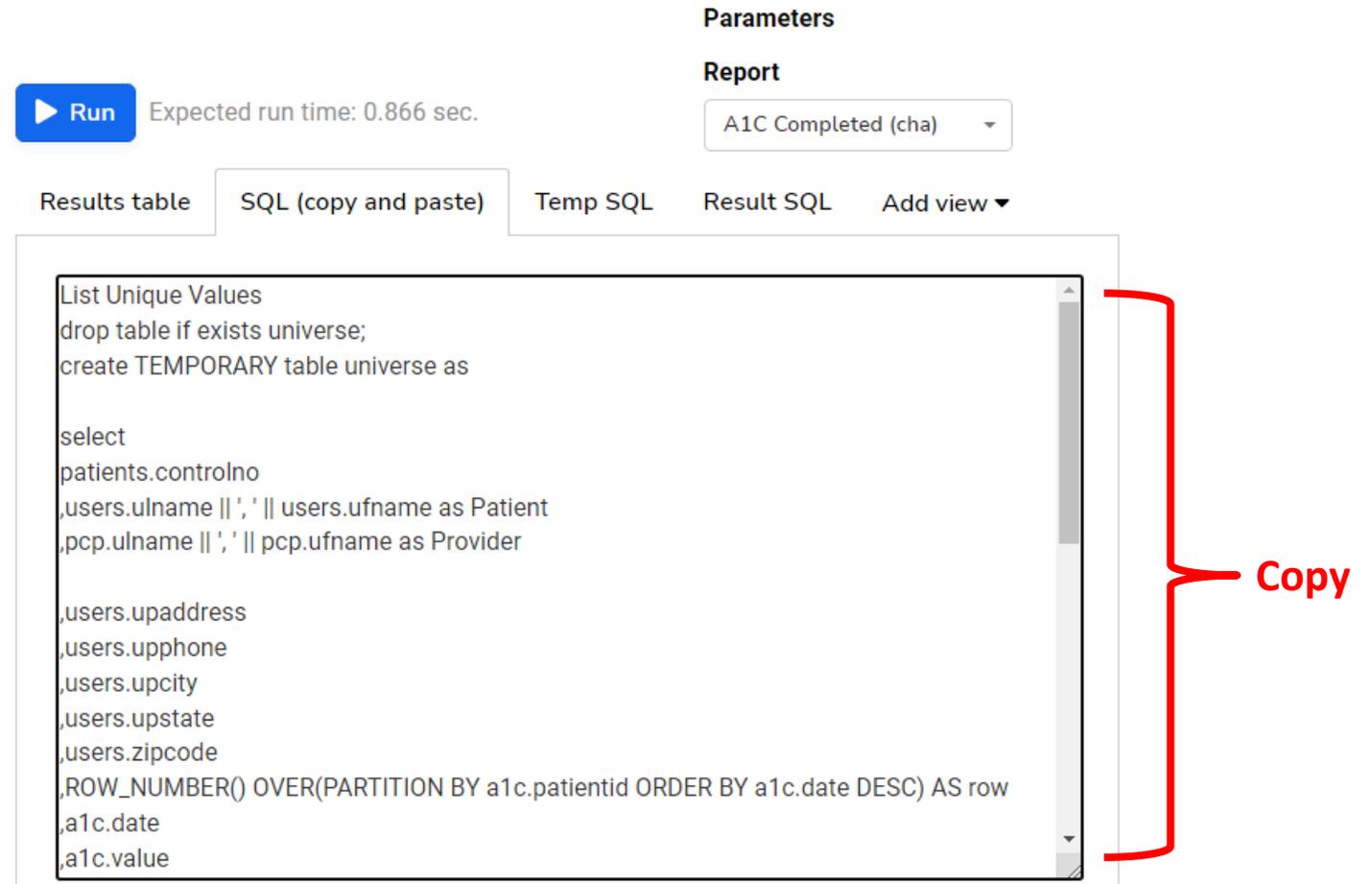


# “SQL from Health Center Report(s)”

2. Go to the “SQL (copy and paste)” tab and copy the SQL code text.

3. Create a new blank report in your instance of Relevant and copy the SQL code text to it

4. Customize the SQL (see next section of this presentation)



The screenshot displays a web-based SQL query editor interface. At the top right, there is a 'Parameters' section with a 'Report' dropdown menu set to 'A1C Completed (cha)'. Below this, a 'Run' button is visible next to the text 'Expected run time: 0.866 sec.'. The main area of the interface is divided into tabs: 'Results table', 'SQL (copy and paste)', 'Temp SQL', 'Result SQL', and 'Add view'. The 'SQL (copy and paste)' tab is currently selected, showing a SQL query. The query text is as follows:

```
List Unique Values
drop table if exists universe;
create TEMPORARY table universe as

select
patients.controlno
,users.ulname || ',' || users.ufname as Patient
,pcp.ulname || ',' || pcp.ufname as Provider

,users.upaddress
,users.upphone
,users.upcity
,users.upstate
,users.zipcode
,ROW_NUMBER() OVER(PARTITION BY a1c.patientid ORDER BY a1c.date DESC) AS row
,a1c.date
,a1c.value
```

A red bracket on the right side of the query text points to a red 'Copy' button, indicating that the SQL code should be copied.

- “Care Gaps at Member Centers”
  - “SQL from Health Center Care Gap(s)”
- 

- This pair uses the same approach as for Reports
- Columns on the report “Care Gaps at Member Centers” are:
  1. Name (of the Care Gap)
  2. Intervention. What is entered for the Care Gap. You can copy it from this report
  3. SQL. This is the SQL of the Care Gap, but it is in text form with no breaks or return characters. It is better to copy the SQL code text from the “SQL (copy and paste)” tab on the report “SQL from Health Center Care Gap(s)”

- “Custom Measures at Member Health Centers”
  - “SQL from Health Center Custom Measure(s)”
- 

- This pair uses the same approach as the one for Reports
- Columns on the report “Custom Measures at Member Centers” are:
  1. Name (of the measure)
  2. Database (Staging or Production)
  3. Description. What is entered for the measure
  4. SQL. This is the SQL of the measure, but it is easier to copy the SQL code text from the tab “SQL (copy and paste)” tab on the report “SQL from Health Center Custom Measure(s)”

# SQL for Populations, Transformers and Importers

---

- “Populations”
  - “Transformer SQL from health centers”
  - “Importer SQL from health centers”
- 
- These reports are “single” reports (not pairs like the previous descriptions)
  - They simply display the name of the Population/Transformer/Importer and the SQL code it contains
  - Organized by Health Center. See tabs along the top

# Exercise #4

---

*If you have access to RCHC Aggregate... log-in*

## Description of your task

- You have a request from a provider working in your health center's substance abuse medication-assisted treatment (MAT) clinic
- The provider wants a report that lists all patients using the medications Suboxone and Buprenorphine
- Search and copy the SQL of one or two potential report candidates

# Answer to Exercise #4

---

**Step #1:** Use the “Report on Reports” to search for candidates

- [AVH] Suboxone/Buprenorphine Active Patient List: Alive and active patients with a visit within the past year with suboxone and/or buprenorphine on their medication list.
- [CHA] Patients on Suboxone/Buprenorphine: Patients on Suboxone/Buprenorphine

It is probably a good idea to look at the code from both sources. The provider said the report should list all patients using the medications Suboxone and Buprenorphine. Does that mean either/or –or– both?

# Answer to Exercise #4

---

**Step #2:** Use the “SQL from Health Center Report(s)” to copy the SQL

The image shows a screenshot of a report interface with several annotations. On the left, a red arrow points to a dropdown menu labeled "Report" with a warning icon, which currently displays "3 of 2413". Above this menu, the text "Parameters" is visible, and a red circle highlights a "Reset to defaults" link with a red arrow pointing to it. Below the dropdown is a search bar containing the text "Suboxone". A list of report titles is shown below the search bar, with two items checked: "Patients on Suboxone/Buprenorphine (cha)" and "Suboxone/Buprenorphine Active Patient Lis".

On the right side of the interface, a red arrow points down to a blue "Refresh" button with a play icon, which is circled in red. To the right of the button, the text "Expected run time: 0.865 sec." is displayed. Below the button is a "Results table" section, and a red circle highlights the text "SQL (copy and paste)" with a red arrow pointing to it.

# Customizing Imported SQL Code

---

- Customization applies to code based on the Staging Database
- Code from the Production Database is already standardized in terms of field and table names.
- However, keep in-mind that not every Data Element (Importer) is activated in every instance of Relevant.
- On the Data Elements page in Relevant, the symbol  represents a Data Element that is not mapped/enabled
- Also note that you should search for and download Staging Database SQL code from health centers that use the same EHR

# Ideas for Staging Database SQL Code

---

- On the Staging Database level, there are differences in table names, field names and approaches to using the data among health centers
- Example of a smaller difference: field names like, patientid vs patient\_id
- Example of larger differences:
  1. A reference to a whole Transformer that exists in one instance but not another (you would have to consider constructing it)
  2. A reference to a Transformer with a field that exists in one instance but not another (you would have to consider constructing it)
  3. Some health centers use relevant\_medications, relevant\_cases, etc. and some do not

# How to Modify/Customize the SQL Code

---

- Once you have located some Staging Database code using the methods in the last section, copy it to DataGrip or a blank report in Relevant
- The first step is to go through the code, line-by-line, to identify tables or fields that have different names, or that do not exist
- DataGrip is helpful because of text color-coding

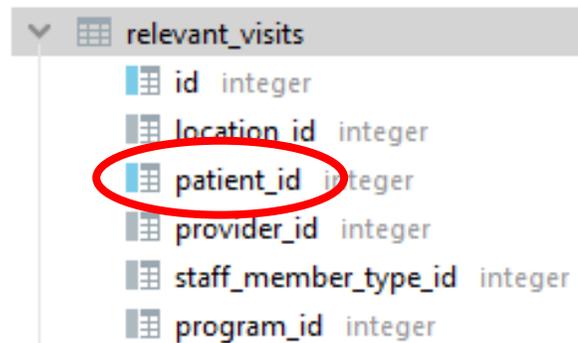
# Fields With Different Names (DataGrip)

---

The good thing about DataGrip is that it highlights (in red) fields that do not exist on the referenced table

```
SELECT
    patientid,
    visit_date
FROM relevant_visits
```

In DataGrip, keep the field list for the specific table open for quick reference



# Fields With Different Names (Relevant)

- The Relevant report writer will show the error once you run the report. In some cases, it gives a hint.

## Query Error

```
ERROR: column "patientid" does not exist
LINE 4:   patientid,
          ^
HINT: Perhaps you meant to reference the column "relevant_visits.patient_id".
```

- Keep the Schema Browser open to see the field names and formats

### Schema browser

Browse tables	
relevant_visits	2191700
billable_universe	boolean
id	integer
location_id	integer
patient_id	integer
primary_payer_id	integer
program_id	integer

# Tables With Different Names

---

- If the table is not recognized by DataGrip, the table name and associated fields will be highlighted

```
SELECT
    patient_id,
    visit_date
FROM relevant_visit
```

Missing an "s"  
as in relevant\_visits

- In the Relevant report writer, there is not a color scheme, but it will give an error. Use the Schema Browser to look at table names or look for suggestions

```
1 SELECT
2     patient_id,
3     visit_date
4 FROM relevant_visit
    relevant_visit_billing_codes
    relevant_visit_diagnosis_codes
    relevant_visits
    relevant_visits_hbns
    relevant_visits_temp
    relevant_visit_type
```

# Field Formatting

---

- Normally, field formatting does not make much of a difference unless it is being used for a JOIN or used in a calculation
- Examples of errors in DataGrip

```
[42883] ERROR: operator does not exist: character varying = integer
```

```
Hint: No operator matches the given name and argument type(s). You might need to add explicit type casts.
```

```
[42883] ERROR: operator does not exist: character varying + interval
```

```
Hint: No operator matches the given name and argument type(s). You might need to add explicit type casts.
```

- In the Relevant report writer, an error comes up but there is no description of it to help the programmer

# Defining Field Formatting

Relevant report writer

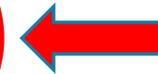
## Schema browser

Browse tables	
relevant_visits	2191700
billable_universe	boolean
id	integer
location_id	integer
patient_id	integer
primary_payer_id	integer
program_id	integer



DataGrip

relevant_visits	
id	integer
location_id	integer
patient_id	integer
provider_id	integer
staff_member_type_id	integer
program_id	integer



# Casting Fields

---

- Casting is done to change the field format
- Use in the SELECT statement or other SQL statement
- Can be done in different ways. The easiest is [expression :: type]

## Common examples

visit\_date :: DATE

ldl\_value :: INTEGER

uds\_visit :: BOOLEAN

visit\_type :: VARCHAR

# Completely Missing Tables

---

- These are references to tables that exist in the imported code but completely do not exist in your instance of Relevant (that is, do not exist under a different name)
- Ask yourself how important that table is to your query. Is it absolutely necessary? Is there a work-around?
- Is it a completely new concept for your instance of Relevant? For example, a lab or structured data element that has never been specifically defined before?

# Finding the Original Transformer

---

- Example: say that you copied a report from Petaluma Health Center that referenced a table named `rchc_dialysis_treatments`, but this table was not in your instance of Relevant. What do you do?

```
DROP TABLE IF EXISTS exclusion_dialysis;  
CREATE TEMPORARY TABLE exclusion_dialysis AS  
SELECT DISTINCT patient_id  
FROM rchc_dialysis_treatments  
WHERE ended_on IS NULL;
```

- First, search your own system for any Transformers with similar names



A search input field with a light gray border and rounded corners. The text 'dialysis' is entered in the field. To the right of the text is a vertical line separator, followed by the character count '0/0'. Further right are three icons: an upward-pointing caret (^), a downward-pointing caret (v), and a close button (x).

# Finding the Original Transformer

- Follow the procedure we examined previously to search for and copy the SQL
- Go to the RCHC Aggregate and run the report: [Transformer SQL from health centers](#)
- Find the health center tab (for PHC) and search for a key word (e.g., dialysis)

Results table		AVH	CHA	CommuniCare	LVHC	Marin City	MCC	Ole	PHC	Ritter	Santa Rosa	dialysis	1/15	^	v	x
name	sql	health_center_key										phc	Totals			
Build rchc_dialysis_treatments	<pre> /***** RCHC Transformer: Build rchc_dialysis_treatments Code Written By: Ben Fouts Description: Displays all patients with dialysis treatments noted in Surgical History or Past Medical History. RCHC Started: December 2018 Version Date: December 19, 2018 Revision History: First version Other Notes: This transformer originally written for the RCHC HTN PHASE report. It was based on code from importer "Dialysis Treatments" *****/ DROP TABLE IF EXISTS rchc_dialysis_treatments; CREATE TABLE rchc_dialysis_treatments AS SELECT enc.patientid patient_id, MIN(CASE WHEN surgicalhistory.date ~* '\d\d\d\d\$' THEN make_date(surgicalhistory.date :: INT, 12, 31) WHEN surgicalhistory.date ~* '\d+\d+\d+\$' THEN surgicalhistory.date :: DATE WHEN surgicalhistory.date ~* '\d\d\d\d\d\d\$' THEN make_date(RIGHT(surgicalhistory.date, 4) :: INT, LEFT(surgicalhistory.date, 2) :: INT, 1) WHEN surgicalhistory.date ~* '\d\d\d\d\d\d\$' THEN make_date(RIGHT(surgicalhistory.date, 4) :: INT, LEFT(surgicalhistory.date, 1) :: INT, 1) ELSE enc.date :: DATE END) AS started_on, MAX(CASE WHEN surgicalhistory.date ~* '\d\d\d\d\$' THEN make_date(surgicalhistory.date :: INT, 12, 31) WHEN surgicalhistory.date ~* '\d+\d+\d+\$' THEN surgicalhistory.date :: DATE WHEN surgicalhistory.date ~* '\d\d\d\d\d\d\$' THEN make_date(RIGHT(surgicalhistory.date, 4) :: INT, LEFT(surgicalhistory.date, 2) :: INT, 1) WHEN surgicalhistory.date ~* '\d\d\d\d\d\d\$' THEN make_date(RIGHT(surgicalhistory.date, 4) :: INT, LEFT(surgicalhistory.date, 1) :: INT, 1) ELSE enc.date :: DATE END) AS ended_on FROM surgicalhistory INNER JOIN enc ON enc.encounterid = surgicalhistory.encounterid WHERE (surgicalhistory.reason ILIKE '%dialysis%') AND enc.deleteflag = 0 GROUP BY enc.patientid UNION SELECT enc.patientid patient_id, MIN(enc.date) :: DATE AS started_on, MAX(enc.date) :: DATE AS ended_on FROM encounterdata INNER JOIN enc ON enc.encounterid = encounterdata.encounterid WHERE (encounterdata.pasthistory ILIKE '%dialysis%') AND enc.deleteflag = 0 GROUP BY enc.patientid; </pre>	1	1													

# Finding the Original Transformer

---

- Copy the code to your own DataGrip (or test in Relevant report writer)
- Once you customize it, you can decide to save it as its own Transformer or add it to the report as a **Temporary Table** in the body of the report

```
DROP TABLE IF EXISTS rchc_dialysis_treatments;  
CREATE TABLE rchc_dialysis_treatments AS  
SELECT enc.patientid patient_id,
```



```
DROP TABLE IF EXISTS rchc_dialysis_treatments;  
CREATE TEMPORARY TABLE rchc_dialysis_treatments AS
```

```
MIN(CASE WHEN surgicalhistory.date ~* '^d\d\d\d$'  
  THEN make_date(surgicalhistory.date :: INT, 12, 31)  
  WHEN surgicalhistory.date ~* '^d+\/\d+\/\d+$'  
  THEN surgicalhistory.date :: DATE  
  WHEN surgicalhistory.date ~* '^d\d\/\d\d\d\d$'  
  THEN make_date(RIGHT(surgicalhistory.date, 4) :: INT, LEFT(surgicalhistory.date, 2) :: INT, 1)  
  WHEN surgicalhistory.date ~* '^d\/\d\d\d\d$'  
  THEN make_date(RIGHT(surgicalhistory.date, 4) :: INT, LEFT(surgicalhistory.date, 1) :: INT, 1)  
  ELSE enc.date :: DATE END) AS started_on,
```

# Completely Missing Fields

---

- These are fields that exist in the imported code that completely do not exist on the corresponding Transformers in your instance of Relevant (that is, do not exist under a different name)
- Ask yourself how important that field is to your query. Is it absolutely necessary? Is there a work-around?

# Finding the Original (Missing) Transformer

---

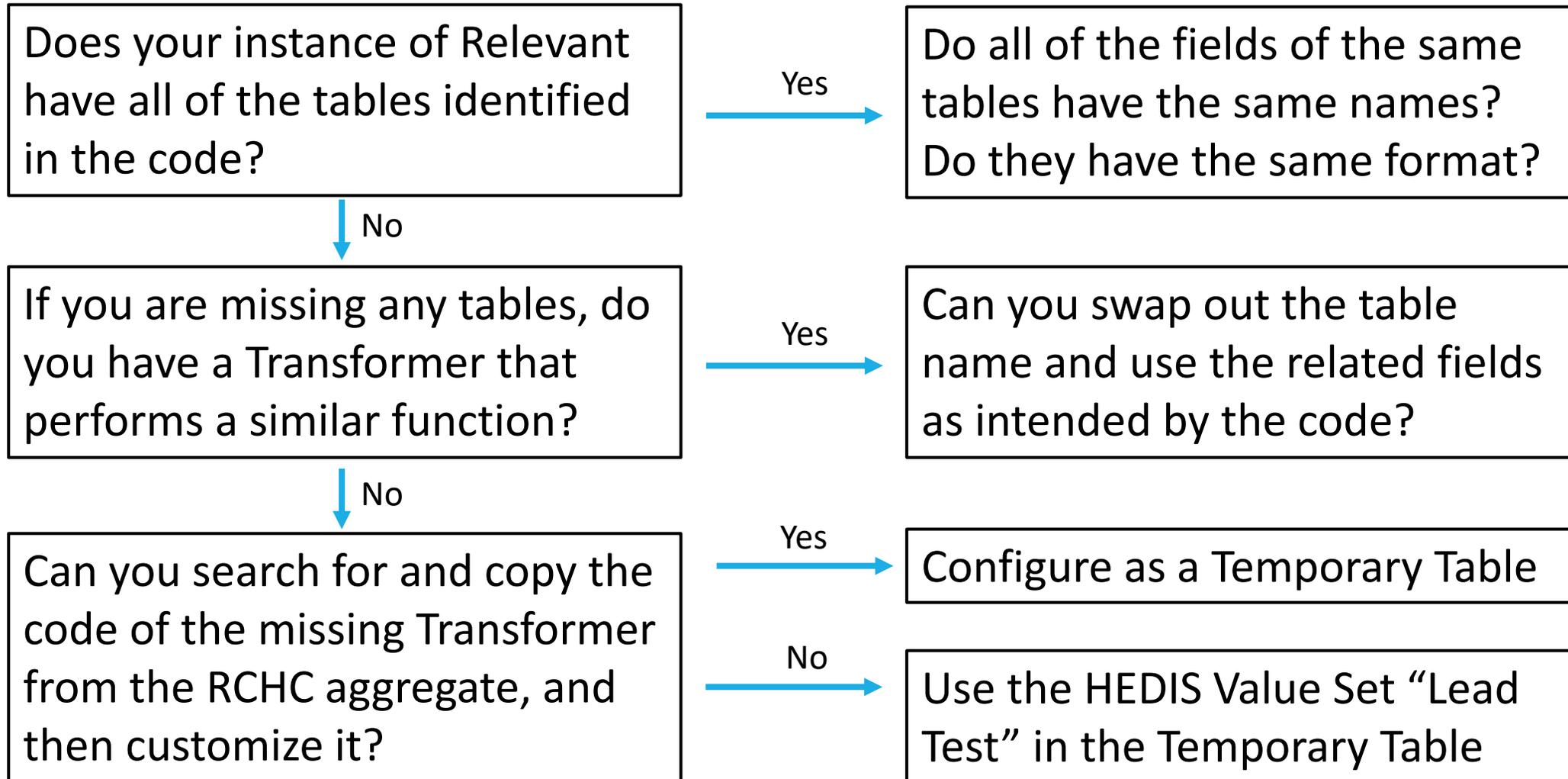
- If you need just a field within a Transformer you already have, look at the SQL code from the original health center to see if it is something you can add to the report as a Temporary Table
- You can copy the SQL code from the original health center Transformer in the same way as we just covered

# Exercise #5

---

- Your Medical Director informed you that a group of health centers received a grant targeting children with certain risk factors
- RCHC already developed a report called “Sunny Day Grant” and put it on the RCHC Aggregate. The Medical Director asked you to import the report for use at your health center.
- The grant covers any child under 12 years of age seen for at least one medical visit between 1/1/2022 and 4/30/2022 and with at least one of the three risk factors. These criteria have been hard-coded in the report.
- If you have access to the RCHC Aggregate, import the report from there. Those without access can copy the code on the handout

# Exercise #5: Tips



# Exercise #5: Answers (part 1)

---

## Lines in the code that likely need to be changed:

```
--Defines the universe
DROP TABLE IF EXISTS universe;
CREATE TEMPORARY TABLE universe AS
SELECT DISTINCT
  relevant_patients.id,
  relevant_patients.mrn,
  relevant_patients.last_name,
  relevant_patients.first_name,
  relevant_patients.date_of_birth,
  extract(YEAR FROM age('2022-01-01', relevant_patients.date_of_birth)) AS age
FROM relevant_patients
WHERE extract(YEAR FROM age('2022-01-01', relevant_patients.date_of_birth)) <= 12
AND relevant_patients.inactive = 'f'
AND relevant_patients.deceased = 'f'
AND EXISTS(SELECT FROM relevant_visits
  WHERE relevant_patients.id = relevant_visits.patient_id
  AND uds_medical IS TRUE
  AND visit_date BETWEEN '2022-01-01' AND '2022-04-30');
```

**This may be BOOLEAN in your instance**

# Exercise #5: Answers (part 2)

---

```
--Risk factor #1 is persistent asthma diagnosis
DROP TABLE IF EXISTS relevant_asthma_temp;
CREATE TEMPORARY TABLE relevant_asthma_temp AS
SELECT DISTINCT patient_id
FROM relevant_asthma
WHERE persistent ILIKE '%TRUE%'
AND started_on <= '2022-04-30';
```

**Patient\_id may be patientid in your system**

**If the 'persistent' field is BOOLEAN, use the IS TRUE expression**

```
--Risk factor #2 is BMI Percentile above 75
DROP TABLE IF EXISTS relevant_bmi_percentile_temp;
CREATE TEMPORARY TABLE relevant_bmi_percentile_temp AS
SELECT patient_id
FROM (SELECT DISTINCT ON (patient_id)
      patient_id,
      value,
      date
      FROM relevant_bmi_percentile
      WHERE date <= '2022-04-30'
      ORDER BY patient_id, date DESC) AS last_bmi
WHERE value > '75';
```

**Patient\_id may be patientid in your system**

**If the 'value' field is Numeric, you do not necessarily need quotes around the number**

# Exercise #5: Answers (part 3)

---

```
--Risk factor #3 is ever had positive blood lead lab  
DROP TABLE IF EXISTS lead_lab_temp;  
CREATE TEMPORARY TABLE lead_lab_temp AS  
SELECT DISTINCT patient_id  
FROM srhc_lead_labs  
WHERE lab_result ILIKE '%abnormal%';
```

**An existing Transformer for lead labs is uncommon (but a couple of health centers already have it)**

# Exercise #5: Answers (part 4)

---

```
SELECT
  universe.mrn,
  universe.last_name,
  universe.first_name,
  universe.date_of_birth,
  universe.age,
  CASE WHEN relevant_asthma_temp.patient_id IS NULL
        THEN FALSE ELSE TRUE END AS has_asthma,
  CASE WHEN relevant_bmi_percentile_temp.patient_id IS NULL
        THEN FALSE ELSE TRUE END AS has_high_bmi,
  CASE WHEN lead_lab_temp.patient_id IS NULL
        THEN FALSE ELSE TRUE END AS has_positive_lead_lab
FROM universe
  LEFT JOIN relevant_asthma_temp ON relevant_asthma_temp.patient_id = universe.id
  LEFT JOIN relevant_bmi_percentile_temp ON relevant_bmi_percentile_temp.patient_id = universe.id
  LEFT JOIN lead_lab_temp ON lead_lab_temp.patient_id = universe.id
WHERE relevant_asthma_temp.patient_id IS NOT NULL
  OR relevant_bmi_percentile_temp.patient_id IS NOT NULL
  OR lead_lab_temp.patient_id IS NOT NULL;
```

**Patient\_id (from previous Temporary Tables) may be patientid in your system**

# Exercise #5: Example of Lead Lab Transformer (eCW)

---

```
SELECT DISTINCT
  enc.patientid AS patient_id,
  CASE WHEN Id.colldate IS NOT NULL AND
EXTRACT(YEAR FROM Id.colldate) > 1902
    THEN Id.colldate :: DATE
    WHEN Id.resultdate IS NOT NULL AND
EXTRACT(YEAR FROM Id.resultdate) > 1902
    THEN Id.resultdate :: DATE
  END AS performed_on,
  Id.result AS lab_result,
  Idd.value AS lab_value
FROM labdata AS Id
  INNER JOIN labdatadetail AS Idd ON Idd.reportid =
Id.reportid
  INNER JOIN enc ON enc.encounterid = Id.encounterid
  INNER JOIN labloinccodes ON labloinccodes.itemid =
Idd.propid AND labloinccodes.deleteflag = 0
```

```
  INNER JOIN hedis_value_set_codes ON
hedis_value_set_codes.code_value = labloinccodes.code
  AND hedis_value_set_codes.latest = 'TRUE'
  AND hedis_value_set_codes.value_set_oid =
'2.16.840.1.113883.3.464.1004.1147'
WHERE COALESCE(Idd.value, Id.result) IS NOT NULL
  AND ((Id.colldate IS NOT NULL AND EXTRACT(YEAR
FROM Id.colldate) > 1902)
  OR (Id.resultdate IS NOT NULL AND
EXTRACT(YEAR FROM Id.resultdate) > 1902))
  AND Id.received = 1
  AND Id.status = 1
  AND Id.deleteflag = 0
  AND enc.deleteflag = 0
  AND Id.cancelled = 0
  AND (Id.result IS NULL OR Id.result !~*
'(*cancelled.*|*error.*|*never done.*)');
```

# Questions?

---

My e-mail is: [bfouts@rchc.net](mailto:bfouts@rchc.net)