OMOP Common Data Model Extract, Transform & Load Tutorial

Clair Blacketer

Mui van Zandt

Erica A. Voss

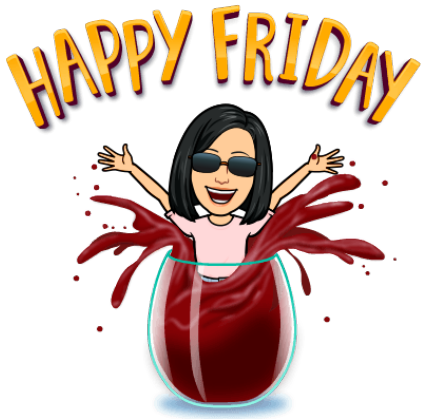
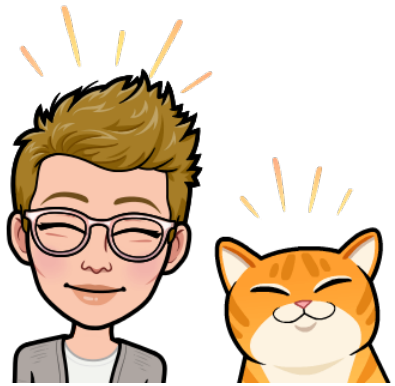
What this tutorial will provide . . .

- Suggested process for developing a CDM ETL
- OHDSI ETL tools: White Rabbit, Rabbit-In-A-Hat, and Usagi
- Resources like the CDM Wiki and THEMIS
- Generation of a simple ETL

Agenda

Time	Agenda Item
9:00 – 9:30	Overview • VM Set Up • What is OHDSI / CDM? • What is the ETL Process?
9:30 – 10:45	ETL Step 1 – Design Your ETL
10:45 – 11:15	Coffee
11:15 – 12:30	ETL Step 2 – Mapping to the Vocabulary
12:30 – 13:30	Lunch
13:30 – 14:30	ETL Step 3 – Develop ETL
14:30 – 15:30	ETL Step 4 – Quality Control
15:30 – 16:00	Coffee Break
16:00 – 17:00	ETL Pain Points & Conclusions
17:00 – 18:00	Beer, Wine, & Snacks

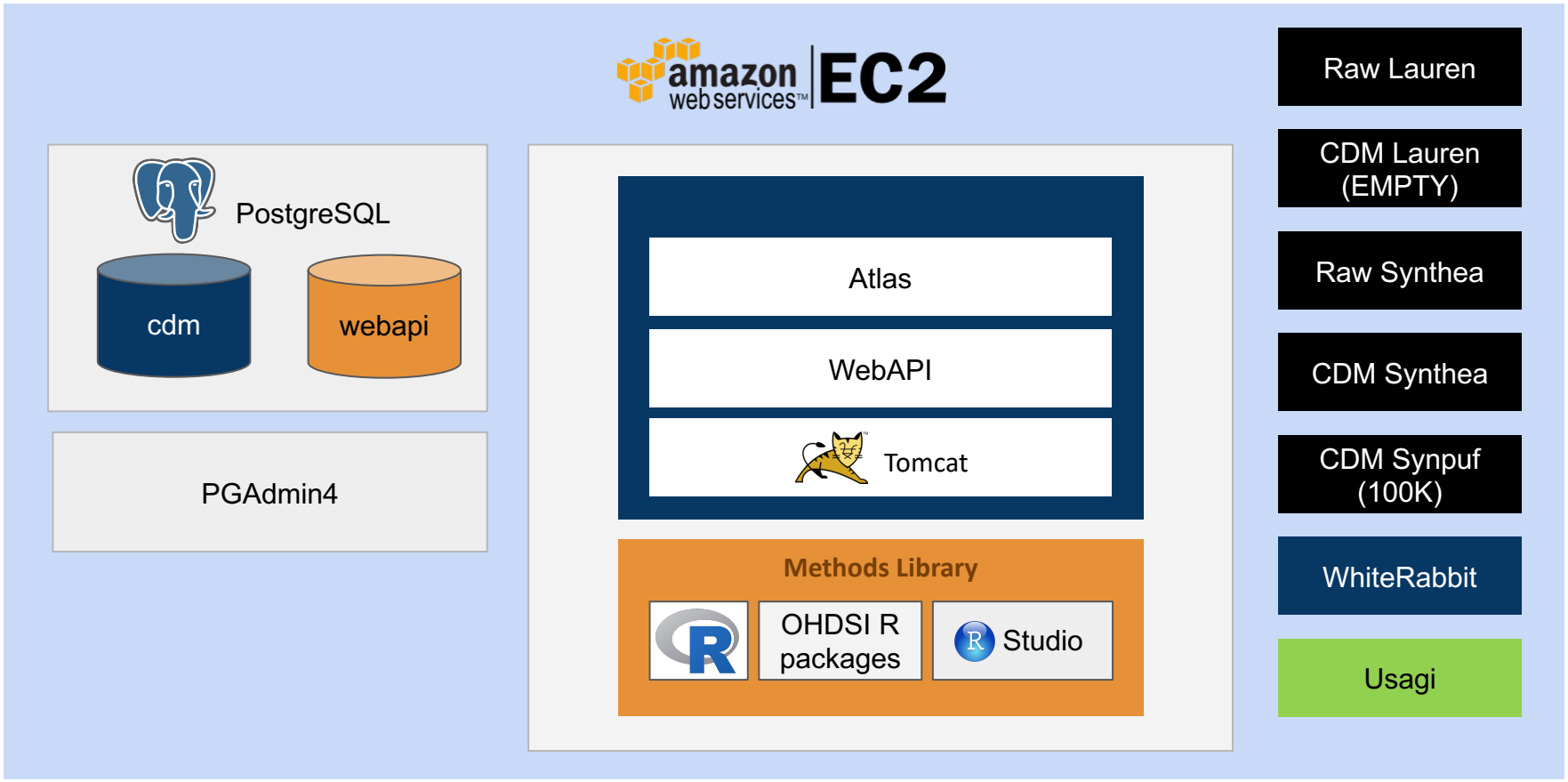
Instructors

Clair Blacketer	Mui van Zandt
	
Erica Voss	
	

Ground Rules

- We are recording this presentation for future use
- We may take some questions off-line if too specific

OHDSI in a Box



Directions for Accessing a VM

LINK TBD

- Pick one of the rows and put your name on the second column

- Go to c:/windows/system32 and click mstc.exe

- Username: erasmusmc
- Password: 123beter

OHDSI's Mission & Vision

To improve health by empowering a community to collaboratively generate the evidence that promotes better health decisions and better care.

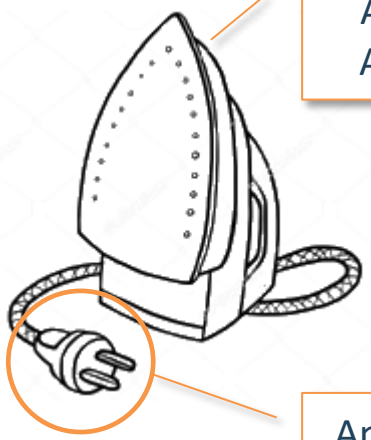
A world in which observational research produces a comprehensive understanding of health and disease.

Join us on the journey

http://ohdsi.org

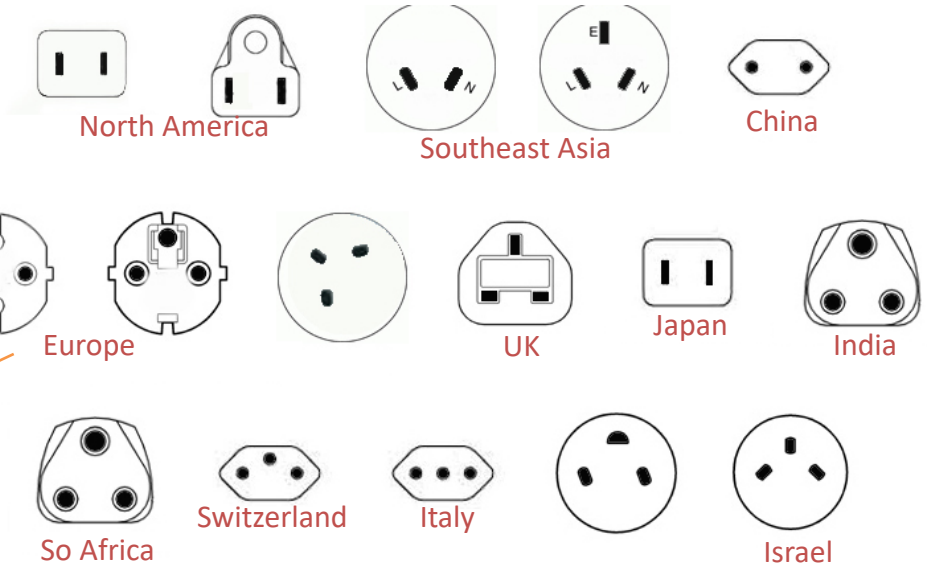
Current Approach: “One Study – One Script“

"What's the adherence to my drug in the data assets I own?"

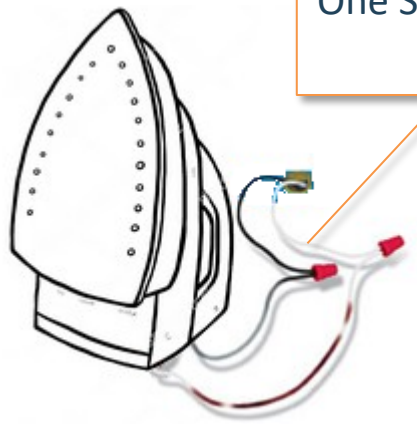


Analytical method: Adherence to Drug

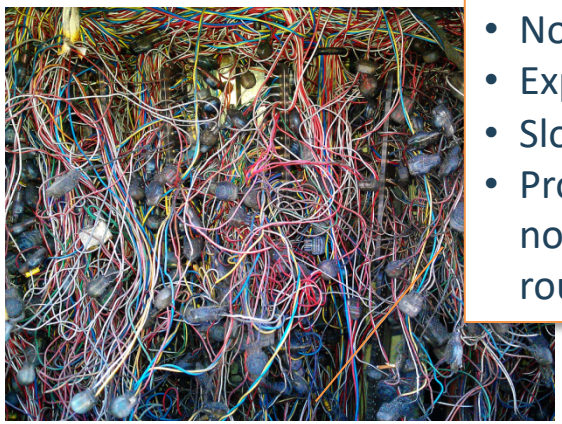
Application to data



Current solution:

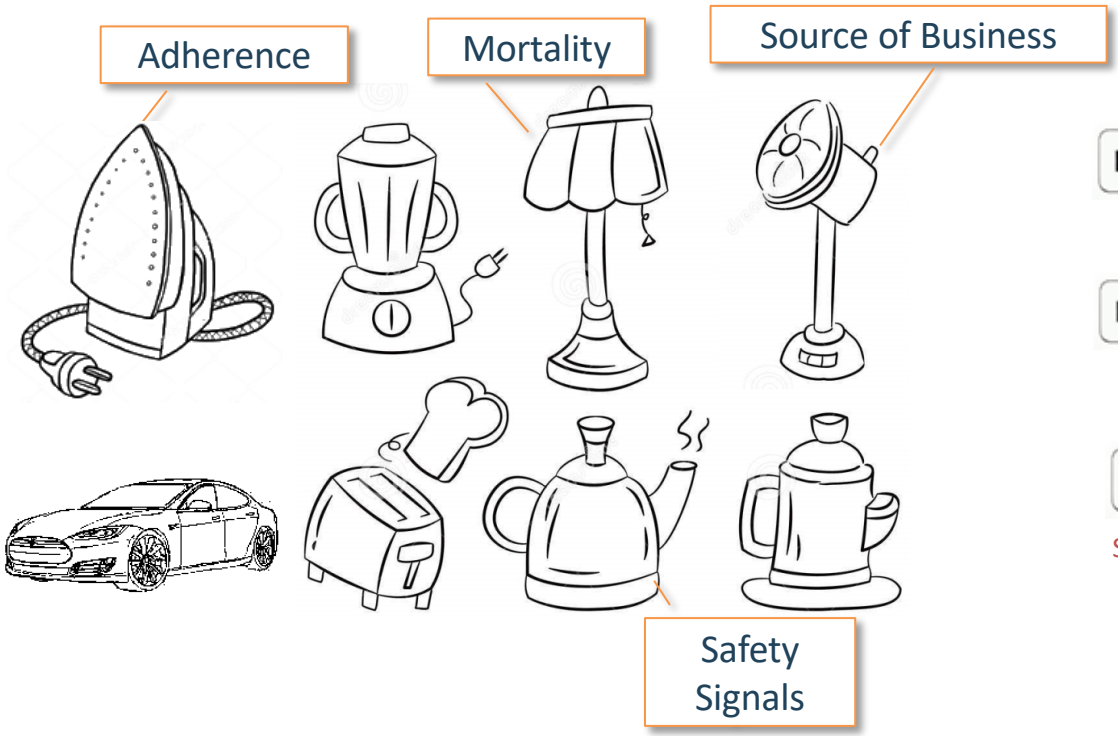


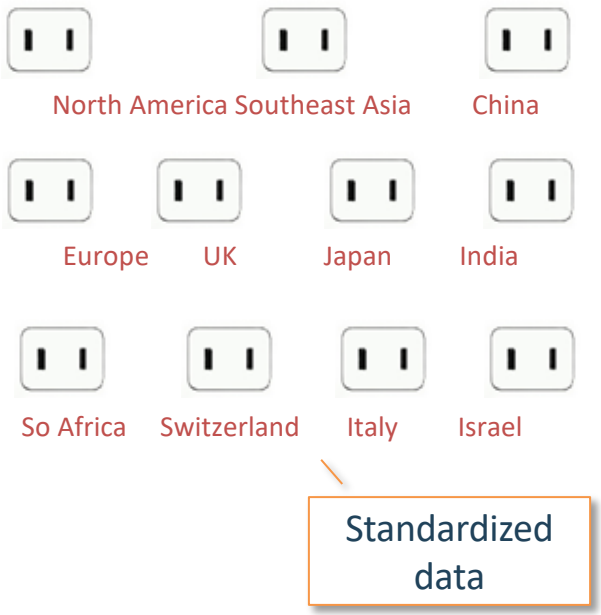
One SAS or R script for each study



- Not scalable
- Not transparent
- Expensive
- Slow
- Prohibitive to non-expert routine use

Solution: Data Standardization Enables Systematic Research

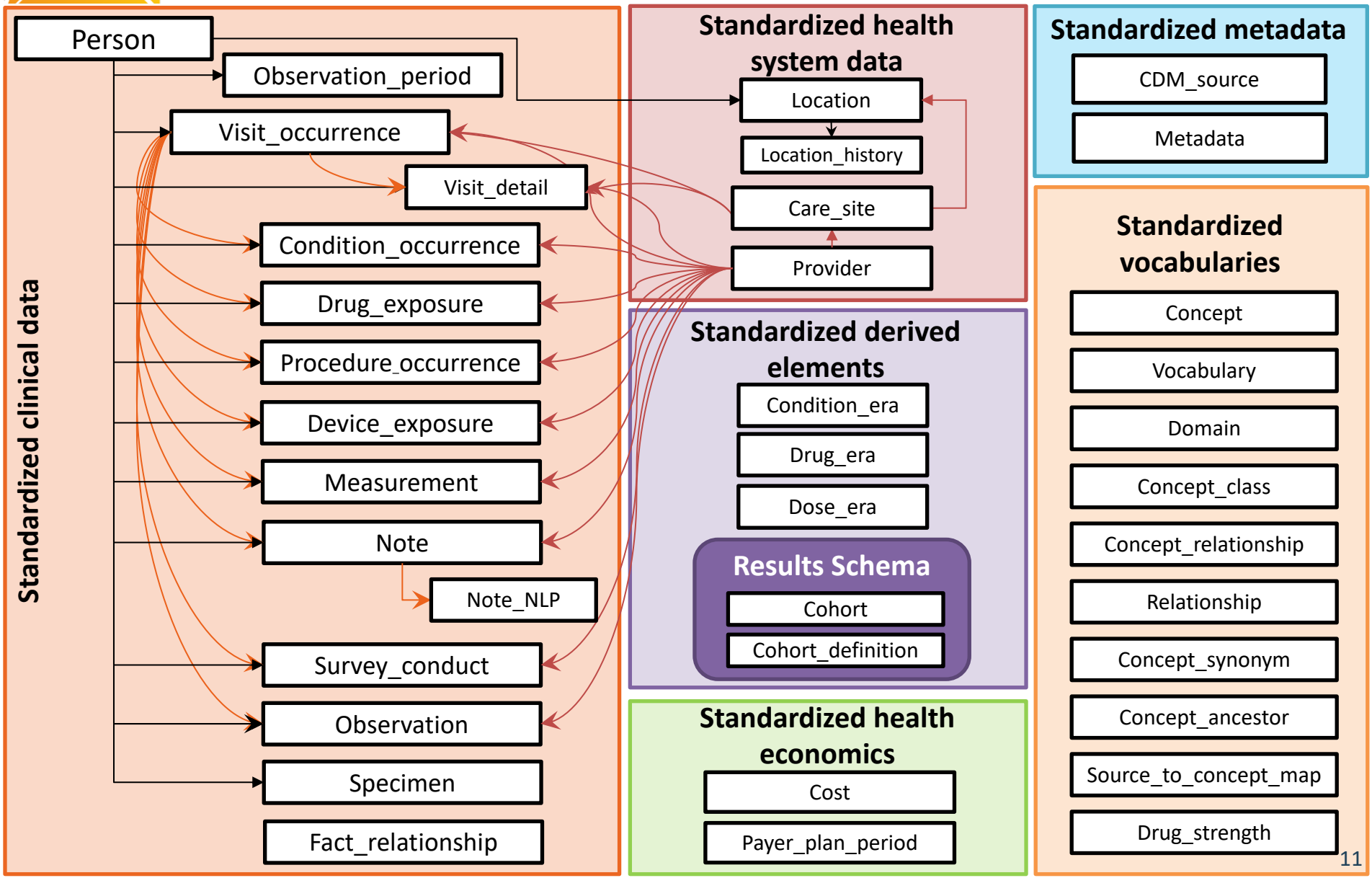




OHDSI Tools

OMOP CDM

CDM Version 6 Key Domains



Why the CDM?

Ability to pursue **cross-institutional collaborations**

Write **one program** to run on multiple data assets

OMOP Vocabularies has greatly increased our **ability to find relevant codes**

You truly **know your data** if you convert it to the CDM

If you know a problem with your data, you can use the **ETL to address it**

Whole community of researchers across diverse organizations and countries

You can use **standardized tools** developed by OHDSI like ATLAS and the Patient Level Prediction Package

The CDM brings **consistency** to observational research through standardization of many of its components

Buy vs Build: leverage an entire community of technical and scientific capability for **"free"**

Takes observational research towards **open science**

Why the CDM?

Regulator	Data Source Owners	Small to Medium Enterprises
• Increased capacity to carry out **studies with big geographical coverage** • Increased capacity to **look at patients holistically** across health systems • Easier assessment of **data quality**	• Facilitates **scientific collaboration** by becoming part of a thriving network • **Increased analysis capability** thanks to a host of open source tools to use • **Faster performing studies**, more studies in less time.	• Opportunity to **change paradigm benefitting health of citizens** • **Expand your existing market** • Open source **community boosts opportunities**

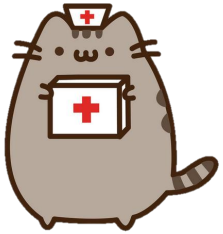
ETL

- Extract, Transform, Load
- In order to get from our native/raw data into the OMOP CDM we need to design and develop and ETL process

- Goal in ETLing is standardize the format and terminology
- This tutorial
  - Will teach you best practices around designing an ETL and CDM maintenance
  - Will not teach you how to program an ETL

ETL Process

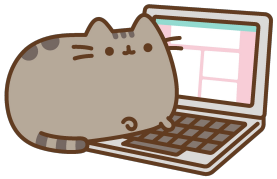
Data experts and CDM experts together design the ETL



People with medical knowledge create the code mappings

ETL Documentation

All are involved in quality control



A technical person implements the ETL

OHDSI Tools

White Rabbit

Rabbit In a Hat

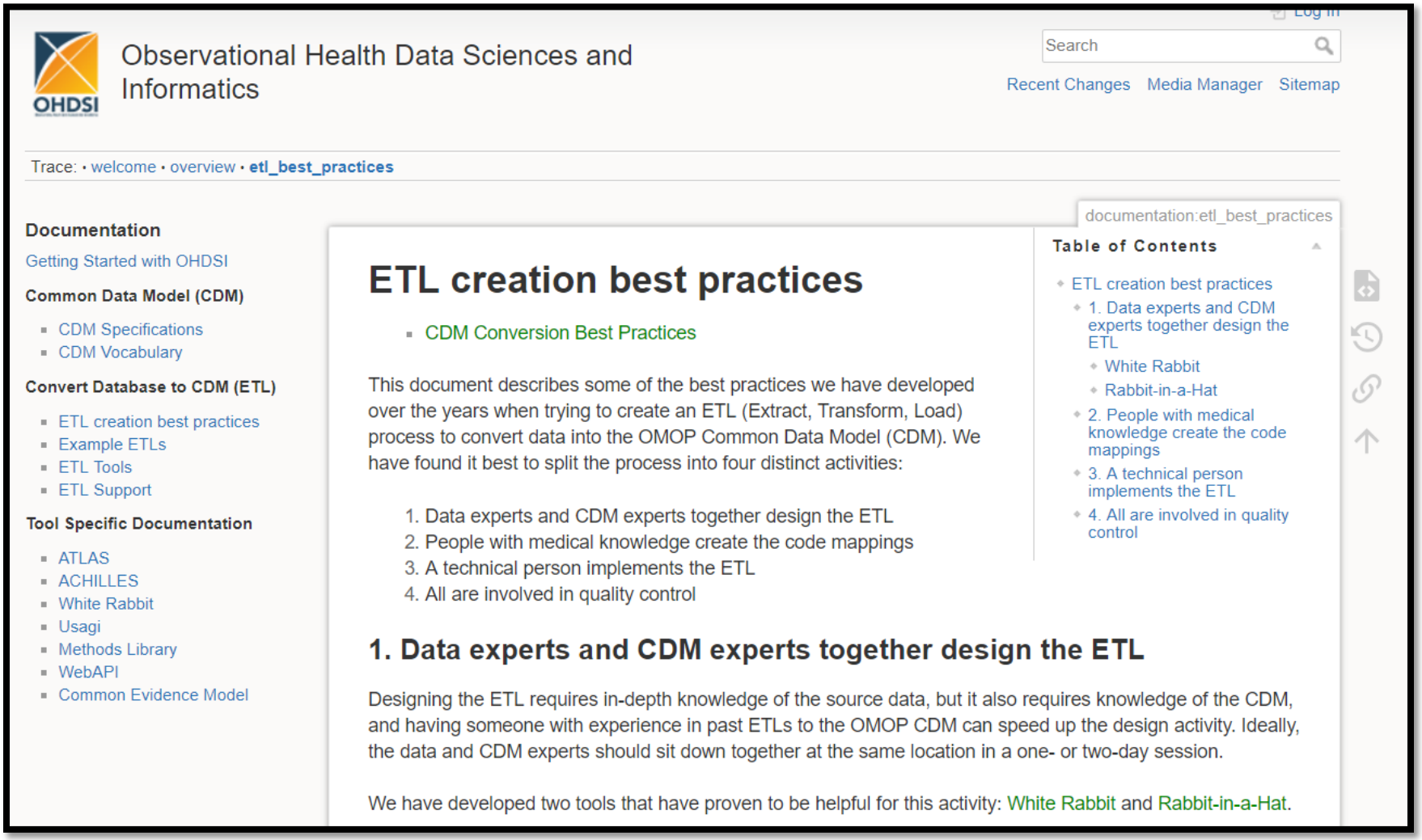
Usagi

White Rabbit

ACHILLES

Rabbit In a Hat

ETL Process



http://www.ohdsi.org/web/wiki/doku.php?id=documentation:etl_best_practices

Hands On Exercises for Today

- Scan a database with White Rabbit
- Build a ETL document with Rabbit in a Hat
- Mapping Source Codes by with the OMOP Vocabulary and USAGI
- SQL to build an ETL

Data experts and CDM experts together design the ETL

People with medical knowledge create the code mappings

ETL Documentation

All are involved in quality control

A technical person implements the ETL

OHDSI

OBSERVATIONAL HEALTH DATA SCIENCES AND INFORMATICS

A Patient's Story: Lauren

Lauren's story

endometriosis uk

"Every step of this painful journey I've had to convince everyone how much pain I was in."

"My first surgery taught me that I had to be very patient with my recovery and very patient with myself in general."

https://www.endometriosis-uk.org/laurens-story

What data do we have?

dysmenorrhea

abdominal pain

missed work

missed work

acetaminop hen

acetaminop hen

acetaminop hen

GP visit

ultrasound

pelvic exam

cyst of ovary

Hospital Visit

severe pain

temp 103°F

CT Scan

ambulance

Bloated abdomen

ascites

surgery

endometrioma

Endometriosis

Lauren's Timeline

-3 Years *-2 Years* *-1 Years* *-2 Weeks* *-3 Days* *Day 0*

Data Format

- Synthea™ is a Synthetic Patient Population Simulator. The goal is to output synthetic, realistic (but not real), patient data and associated health records in a variety of formats.

- The resulting data is free from cost, privacy, and security restrictions. It can be used without restriction for a variety of secondary uses in academia, research, industry, and government (although a citation would be appreciated).

- https://github.com/synthetichealth/synthea

Walonoski J, Kramer M, Nichols J, Quina A, Moesel C, Hall D, Duffett C, Dube K, Gallagher T, McLachlan S. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. J Am Med Inform Assoc. 2017 Aug 30. doi: 10.1093/jamia/ocx079. [Epub ahead of print] PubMed PMID: 29025144.

Synthea Tables

File	Description
allergies.csv	Patient allergy data.
careplans.csv	Patient care plan data, including goals.
conditions.csv	Patient conditions or diagnoses.
encounters.csv	Patient encounter data.
imaging_studies.csv	Patient imaging metadata.
immunizations.csv	Patient immunization data.
medications.csv	Patient medication data.
observations.csv	Patient observations including vital signs and lab reports.
organizations.csv	Provider organizations including hospitals.
patients.csv	Patient demographic data.
procedures.csv	Patient procedure data including surgeries.
providers	Clinicians that provide patient care.

Raw Data

1 Patient

Lauren Data

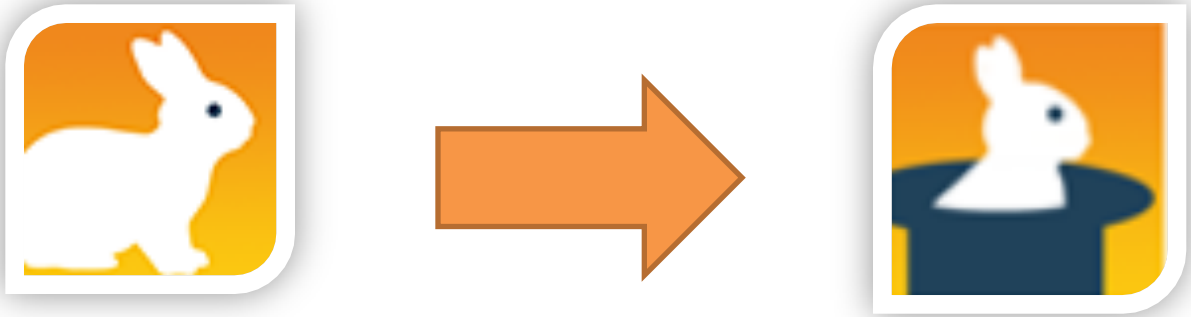
Synthea Format

1000 Patient

Synthetic Data

Synthea Format

Tools help us get started . . .



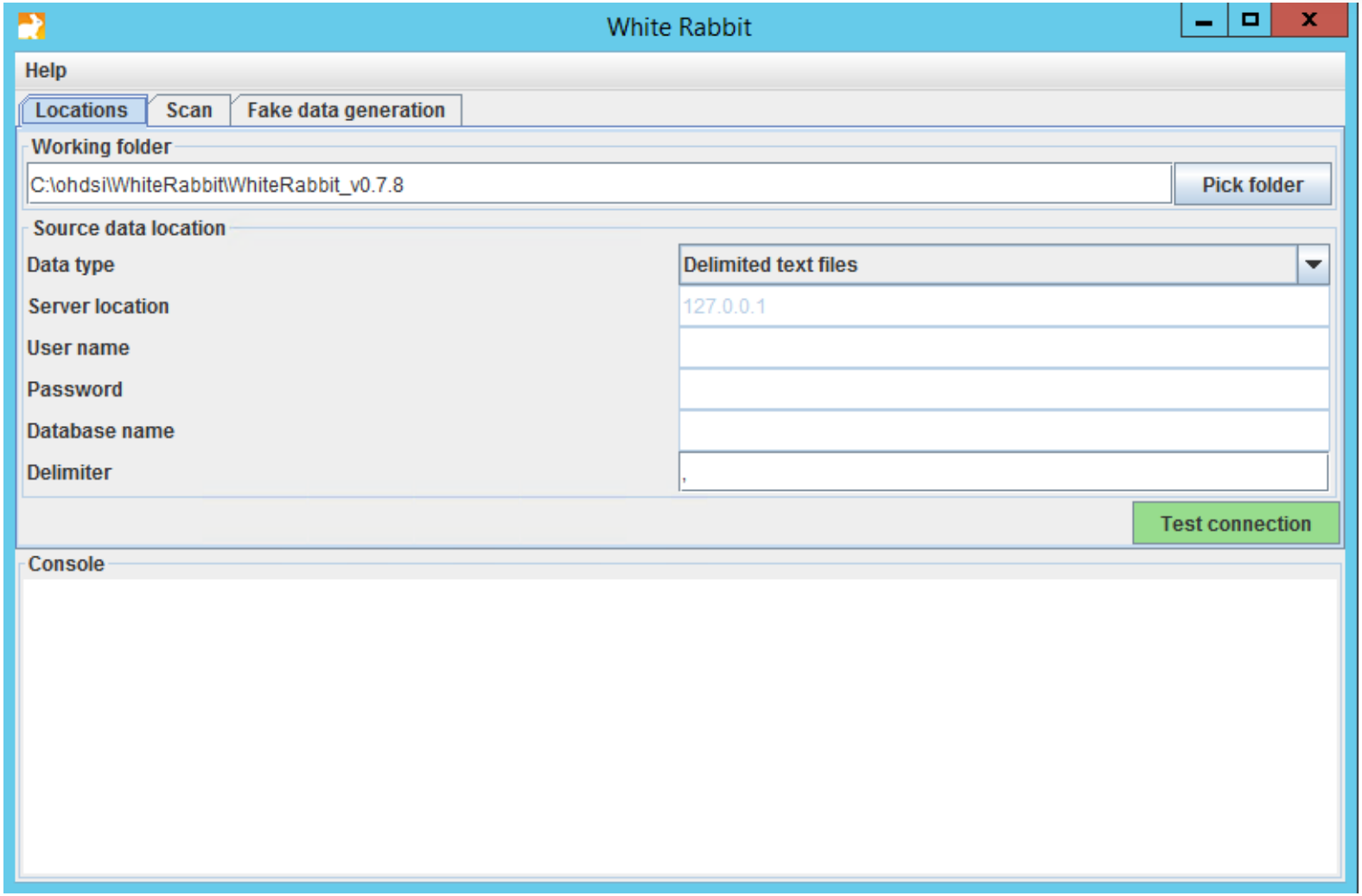
White Rabbit

- performs a scan of the source data, providing detailed information on the tables, fields, and values that appear in a field

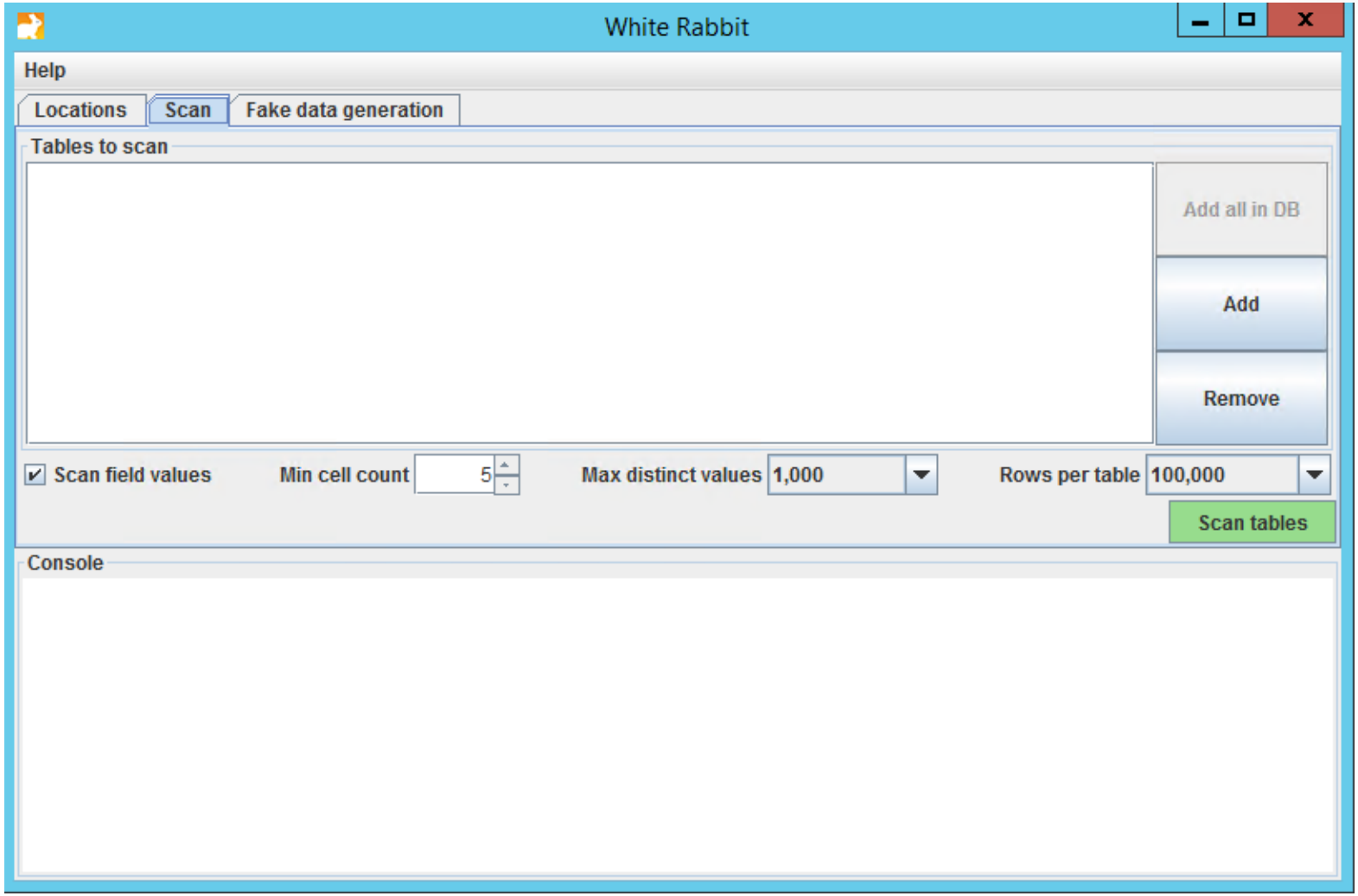
Rabbit In a Hat

- Uses White Rabbit scan to provide a graphical user interface to help build an ETL document
- Does not generate code

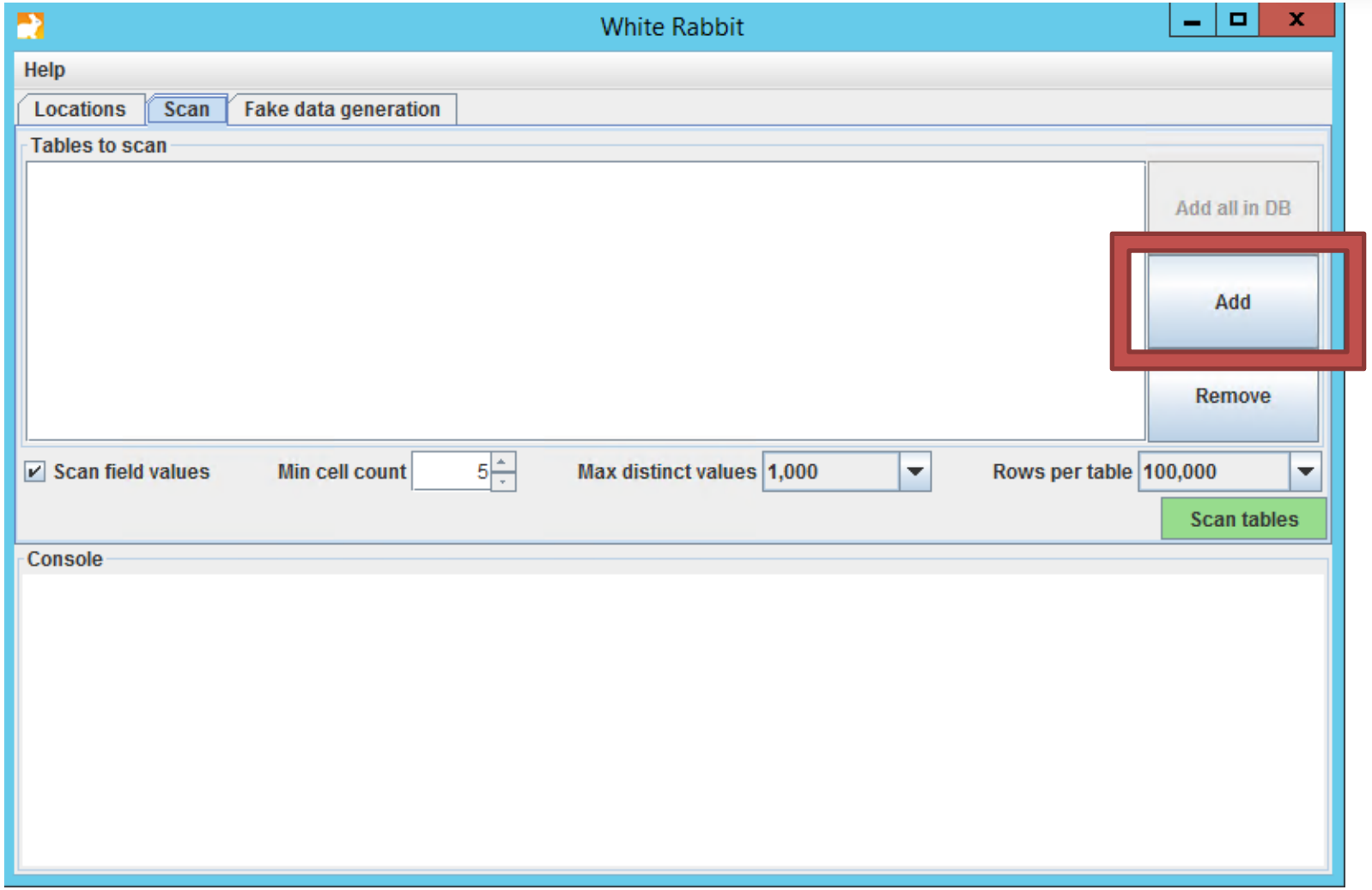
White Rabbit - Location



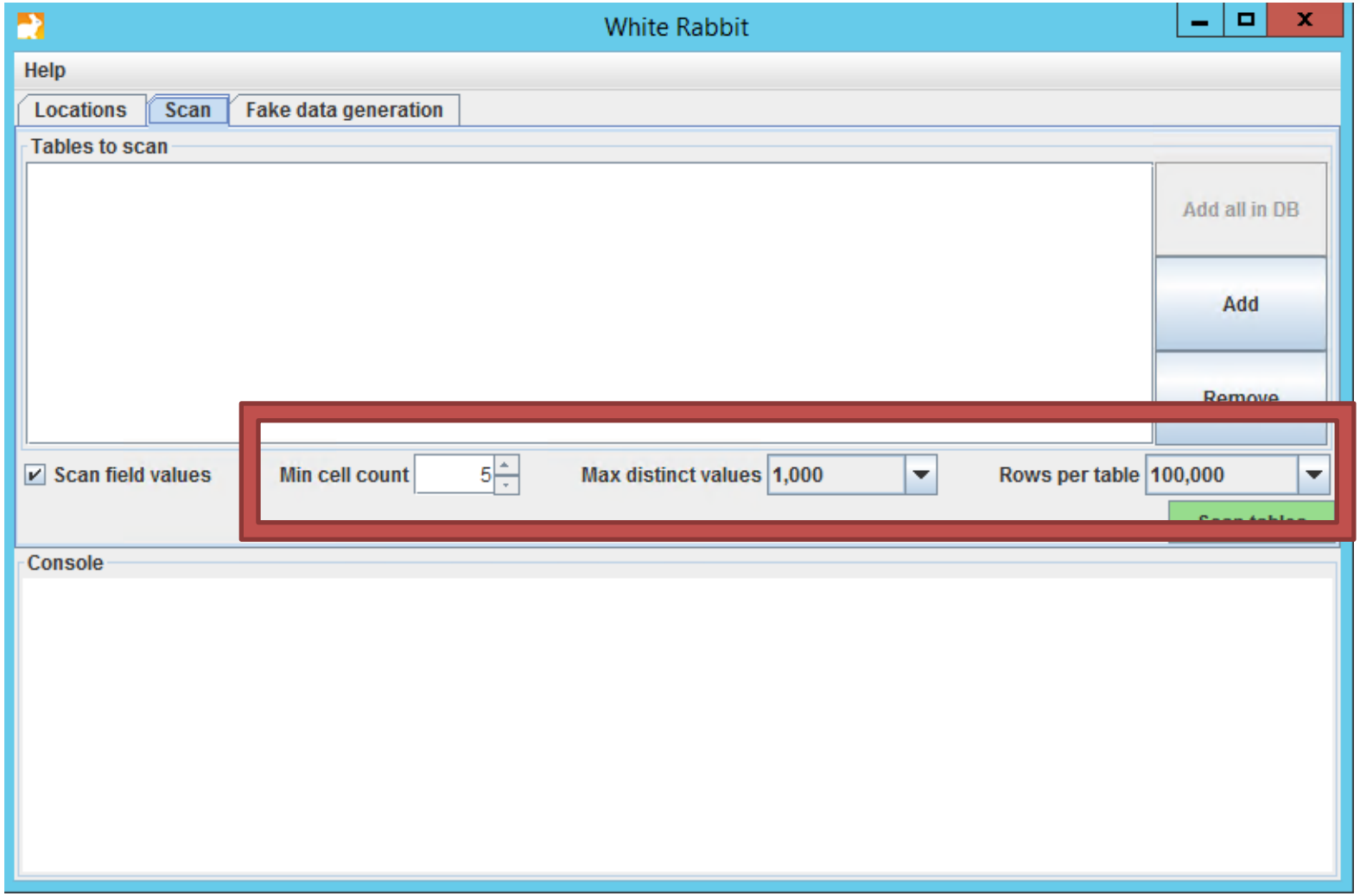
White Rabbit - Scan



White Rabbit - Scan



White Rabbit - Scan



White Rabbit – Scan Report

- We already ran the scan on raw_synthea
- To open the scan while we review:
  - https://github.com/OHDSI/Tutorial-ETL
  - Materials → WhiteRabbit → ScanReport_raw_synthea.xlsx
  - Click “View Raw” to download the XLSX

White Rabbit – Scan Report: raw_synthea

Table	Field	Type	Max lengt	N rows	N rows ch	Fraction empty
allergies	start	date	10	619	619	0
allergies	stop	date	10	619	619	0.904685
allergies	patient	character	36	619	619	0
allergies	encounter	character	36	619	619	0
allergies	code	character	9	619	619	0
allergies	descriptio	character	24	619	619	0
careplans	id	character	36	2939	2939	0
careplans	start	date	10	2939	2939	0
careplans	stop	date	10	2939	2939	0.380061
careplans	patient	character	36	2939	2939	0
careplans	encounter	character	36	2939	2939	0
careplans	code	character	15	2939	2939	0
careplans	descriptio	character	62	2939	2939	0
careplans	reason_cc	character	14	2939	2939	0.090507
careplans	reason_de	character	69	2939	2939	0.090507
condition:	start	date	10	7898	7898	0
condition:	stop	date	10	7898	7898	0.458091
condition:	patient	character	36	7898	7898	0
condition:	encounter	character	36	7898	7898	0
condition:	code	character	15	7898	7898	0
condition:	descriptio	character	80	7898	7898	0
encounter	id	character	36	34262	34262	0
encounter	start	date	10	34262	34262	0
encounter	stop	date	10	34262	34262	0
encounter	patient	character	36	34262	34262	0
encounter	provider	character	36	34262	34262	0.006275
encounter	encounter	character	10	34262	34262	0
encounter	code	character	9	34262	34262	0
encounter	descriptio	character	59	34262	34262	0
encounter	cost	numeric	6	34262	34262	0
encounter	reasoncoc	character	15	34262	34262	0.663155
encounter	reasonde:	character	69	34262	34262	0.663155
imaging_s	id	character	0	0	0	0
imaging_s	date	date	0	0	0	0

Overview | allergies | careplans | conditions | encounters | imaging_studies | immunizations | medication

Overview Tab

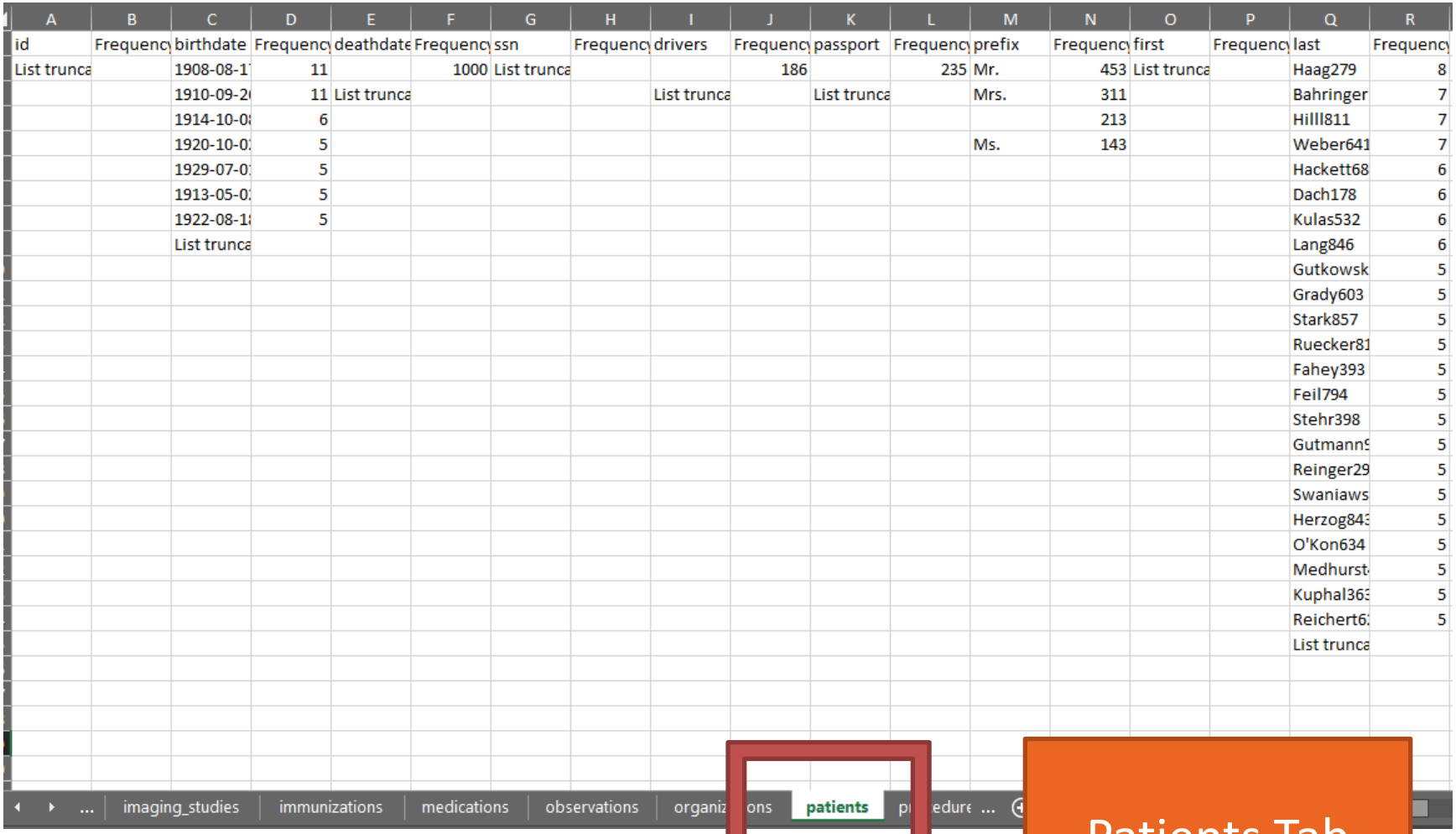
White Rabbit – Scan Report: raw_synthea

patients	id	character varying	36	1120	1120	0
patients	birthdate	date	10	1120	1120	0
patients	deathdate	date	10	1120	1120	0.892857143
patients	ssn	character varying	11	1120	1120	0
patients	drivers	character varying	9	1120	1120	0.166071429
patients	passport	character varying	10	1120	1120	0.209821429
patients	prefix	character varying	4	1120	1120	0.190178571
patients	first	character varying	15	1120	1120	0
patients	last	character varying	16	1120	1120	0
patients	suffix	character varying	3	1120	1120	0.991964286
patients	maiden	character varying	16	1120	1120	0.722321429
patients	marital	character varying	1	1120	1120	0.307142857
patients	race	character varying	8	1120	1120	0
patients	ethnicity	character varying	16	1120	1120	0
patients	gender	character varying	1	1120	1120	0
patients	birthplace	character varying	21	1120	1120	0
patients	address	character varying	36	1120	1120	0
patients	city	character varying	21	1120	1120	0
patients	state	character varying	13	1120	1120	0
patients	zip	character varying	5	1120	1120	0.014285714
procedures	date	date	10	29471		
procedures	patient	character varying	36	29471		

Overview | allergies | careplans | conditions | encounters | ima

Overview Tab

White Rabbit – Scan Report: raw_synthea



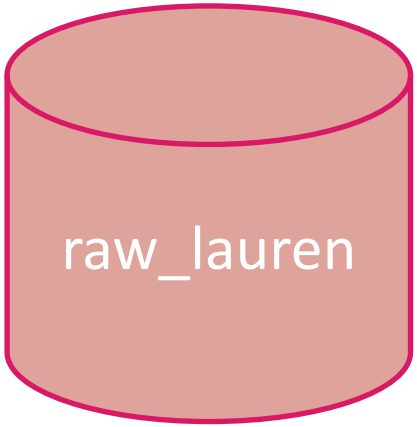
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
id	Frequency	birthdate	Frequency	deathdate	Frequency	ssn	Frequency	drivers	Frequency	passport	Frequency	prefix	Frequency	first	Frequency	last	Frequency
List trunca		1908-08-1	11		1000	List trunca			186		235	Mr.	453	List trunca		Haag279	8
		1910-09-2	11	List trunca				List trunca		List trunca		Mrs.	311			Bahringer	7
		1914-10-0	6										213			Hilll811	7
		1920-10-0	5									Ms.	143			Weber641	7
		1929-07-0	5													Hackett68	6
		1913-05-0	5													Dach178	6
		1922-08-1	5													Kulas532	6
		List trunca														Lang846	6
																Gutkowsk	5
																Grady603	5
																Stark857	5
																Ruecker81	5
																Fahey393	5
																Feil794	5
																Stehr398	5
																Gutmann9	5
																Reinger29	5
																Swaniaws	5
																Herzog843	5
																O'Kon634	5
																Medhurst	5
																Kuphal363	5
																Reichert6	5
																List trunca	

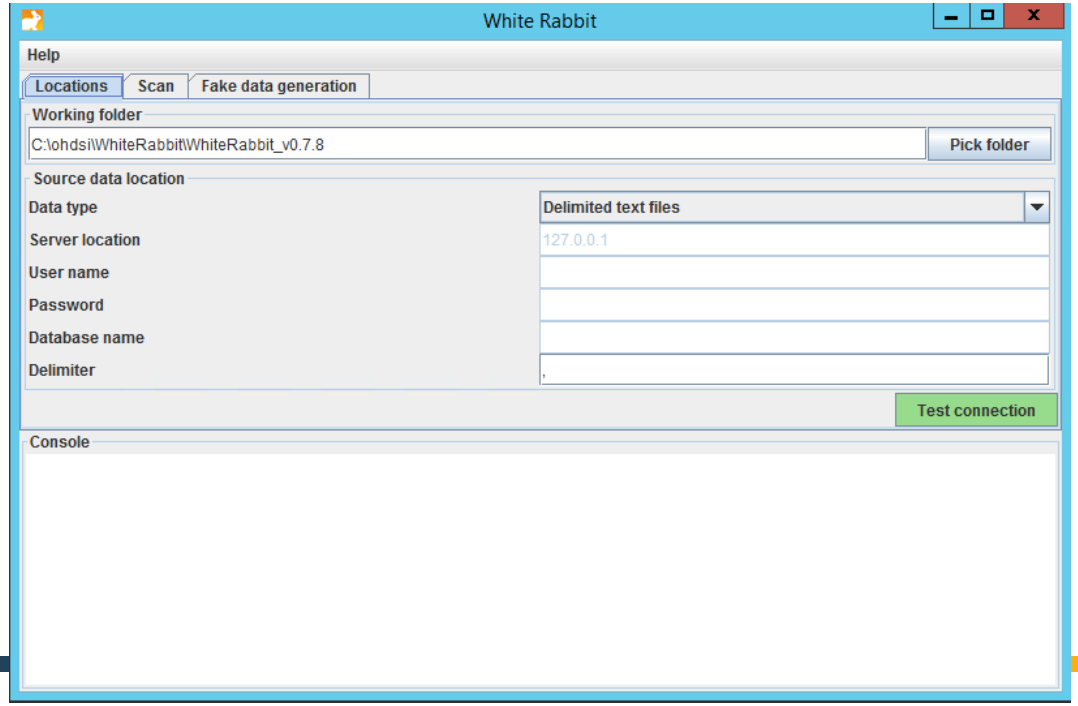
... | imaging_studies | immunizations | medications | observations | organiz ons | patients | pr edure ...

Patients Tab

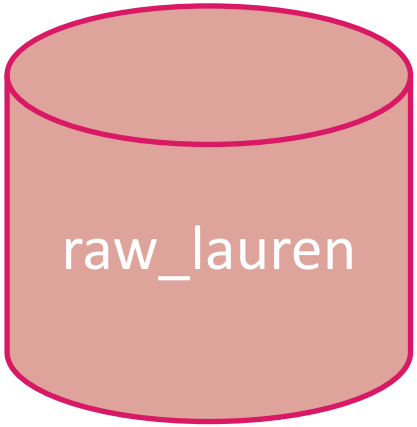
Now Your Turn: Scan Lauren's Data

- Click on WhiteRabbit shortcut
- Go into the WhiteRabbit folder
- Open WhiteRabbit.jar

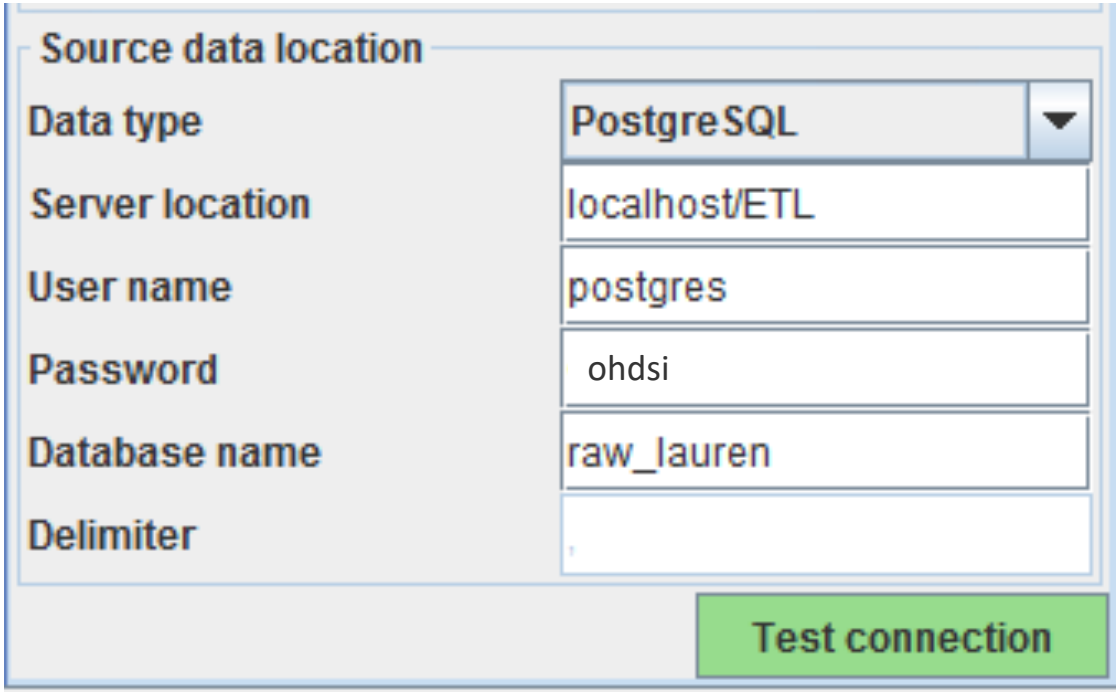




Now Your Turn: Scan Lauren's Data

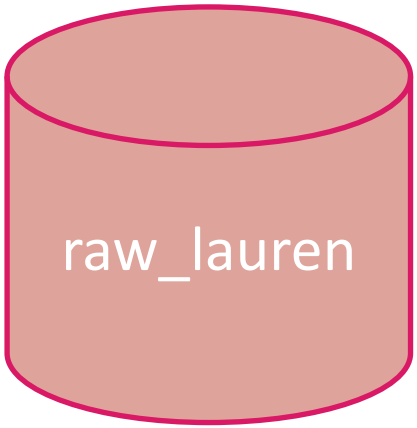


- Connect to Lauren's Data



- Test connection

Now Your Turn: Scan Lauren's Data



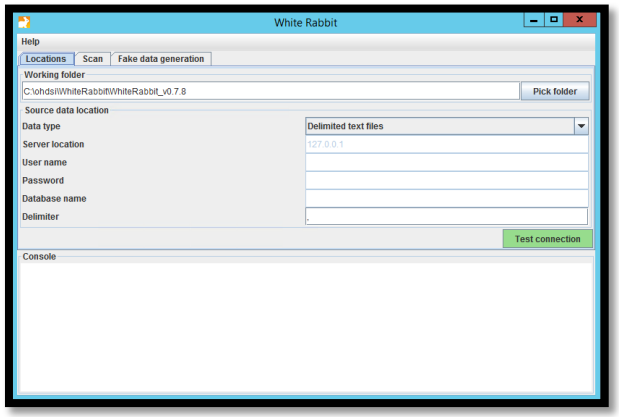
- Go to the "Scan" tab
- Press "Add all in DB" button, set "Min cell count" to 0, and then "Scan tables"

Console

Mar 12, 2019 8:59:06 PM	Scanning table allergies
Mar 12, 2019 8:59:06 PM	Scanning table careplans
Mar 12, 2019 8:59:06 PM	Scanning table conditions
Mar 12, 2019 8:59:06 PM	Scanning table encounters
Mar 12, 2019 8:59:06 PM	Scanning table imaging_studies
Mar 12, 2019 8:59:06 PM	Scanning table immunizations
Mar 12, 2019 8:59:06 PM	Scanning table medications
Mar 12, 2019 8:59:06 PM	Scanning table observations
Mar 12, 2019 8:59:06 PM	Scanning table organizations
Mar 12, 2019 8:59:06 PM	Scanning table patients
Mar 12, 2019 8:59:06 PM	Scanning table procedures
Generating scan report	
Mar 12, 2019 8:59:07 PM	Scan report generated: C:\ohdsi\WhiteRabbit\WhiteRabbit_v0.7.8/ScanReport.xlsx

- Open ScanReport.xlsx

White Rabbit

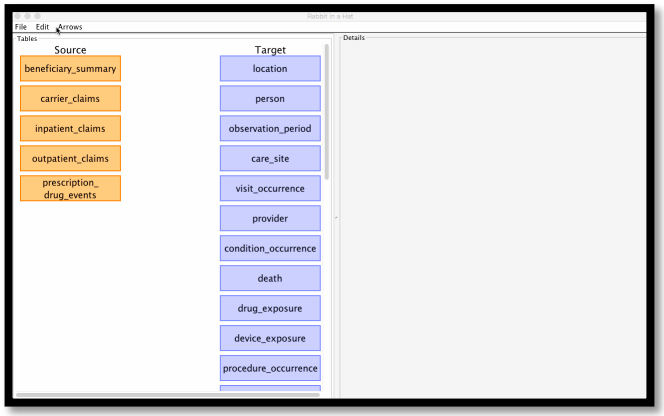


- White Rabbit creates an export of information about the source data
- The scan can be used to:
  - Learn about your source data
  - Needed for Rabbit In a Hat

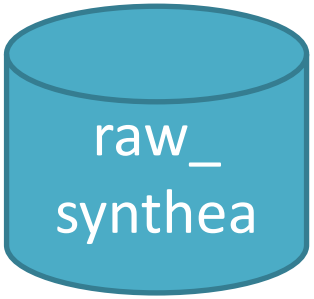
Rabbit in a Hat

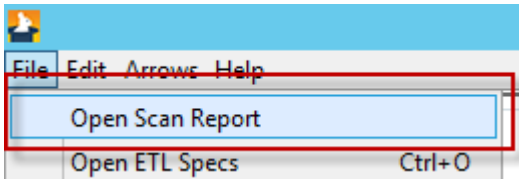
- Can read and display a White Rabbit scan document

- Provides a graphical interface to allow a user to connect source data to tables



Rabbit in a Hat

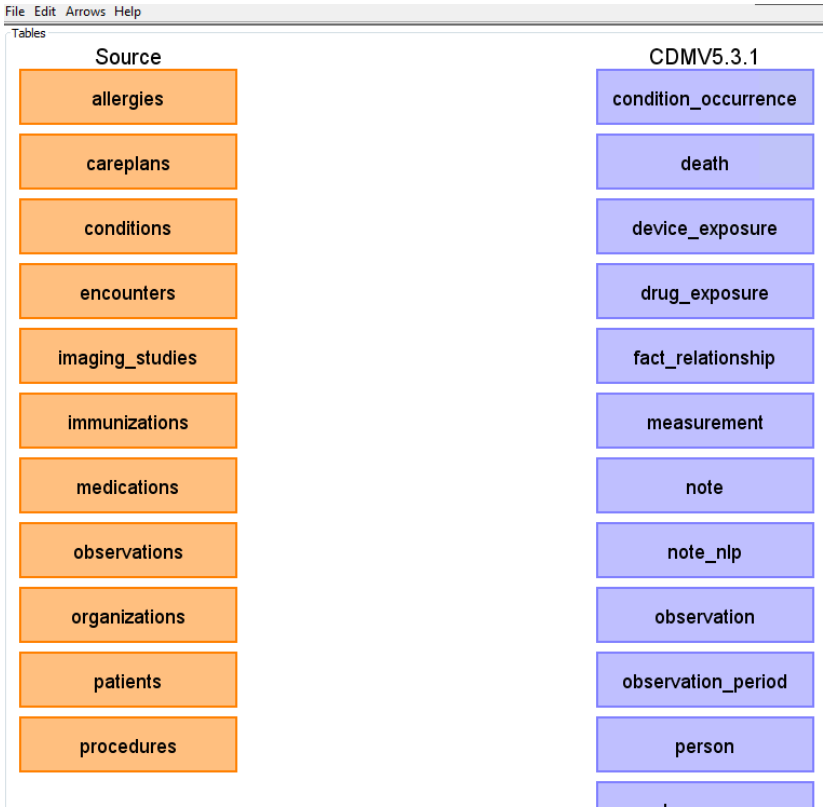




- We will use the ScanReport_raw_synthea.xlsx for this:
  - https://github.com/OHDSI/Tutorial-ETL
  - Materials → WhiteRabbit → ScanReport_raw_synthea.xlsx
  - Click “View Raw” to download the XLSX
  - Save it to the desktop
  - Open in Rabbit in a Hat

Rabbit in a Hat

- The scan tells Rabbit in a Hat what is in the raw database
  - Orange Tables = Raw
  - Blue Tables = CDM



Rabbit in a Hat

Together

person

observation_period

condition_occurrence

On your Own

drug_exposure

Generate document

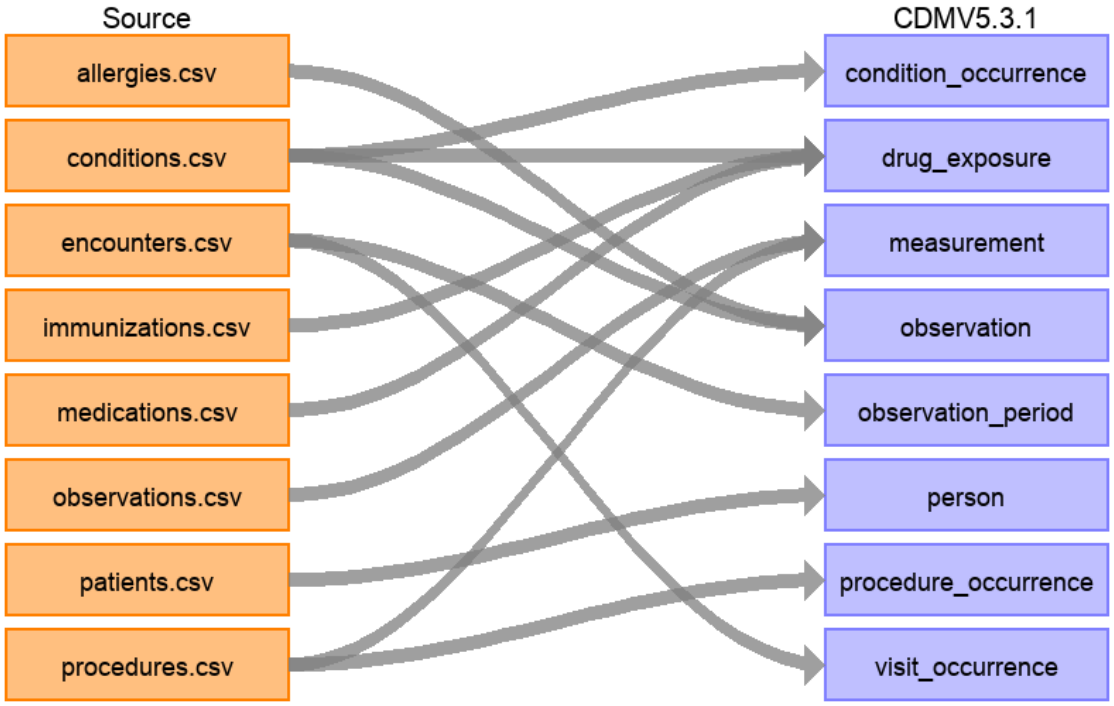
Resources

- Important links to keep in mind when working on an ETL:
  - CDM Wiki
    https://github.com/OHDSI/CommonDataModel/wiki
    Information about the CDM structure and conventions to follow can be found here
  - OHDSI Forums
    http://forums.ohdsi.org/
    http://forums.ohdsi.org/c/cdm-builders
    OHDSI is an active community, your questions may have already been asked on the forum however if not do not be afraid to ask it yourself!
  - CDM Examples
    http://www.ohdsi.org/web/wiki/doku.php?id=resources:2018_data_network
    About 100 CDMs currently exist
  - THEMIS Working Group
    https://github.com/OHDSI/Themis

THEMIS

Rabbit in a Hat

- The full ETL document: https://ohdsi.github.io/ETL-Synthea/



Some Parting Thoughts On ETL

- Vocabulary will tell a source record where to go.
  - Example, just because it is a condition code and in a condition table does not mean it will end up in CONDITION_OCCURRENCE

    ICD9 781.1 - Abnormal weight gain

- STEM Table in Rabbit In a Hat

OHDSI

OBSERVATIONAL HEALTH DATA SCIENCES AND INFORMATICS

Data experts and CDM experts together design the ETL

People with medical knowledge create the code mappings

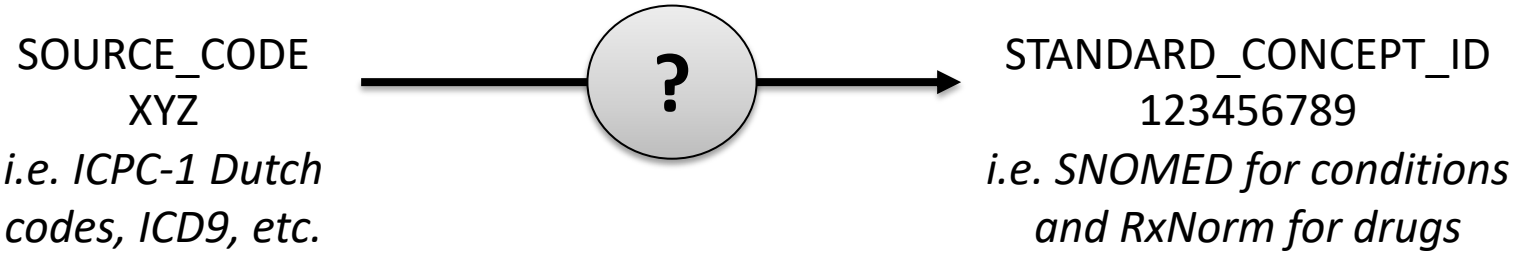
ETL Documentation

ETL

A technical person implements the ETL

All are involved in quality control

Standardizing Terminologies



- What is standardize:
  1. TABLE_CONCEPT_ID
     standard concept the source code maps to, **used for analysis**
  2. TABLE_SOURCE_CONCEPT_ID
     concept representation of the source code, **helps maintain tie to raw data**

- Ways to get a source code to standard code:
  1. OMOP Vocabulary
  2. USAGI

OMOP Vocab

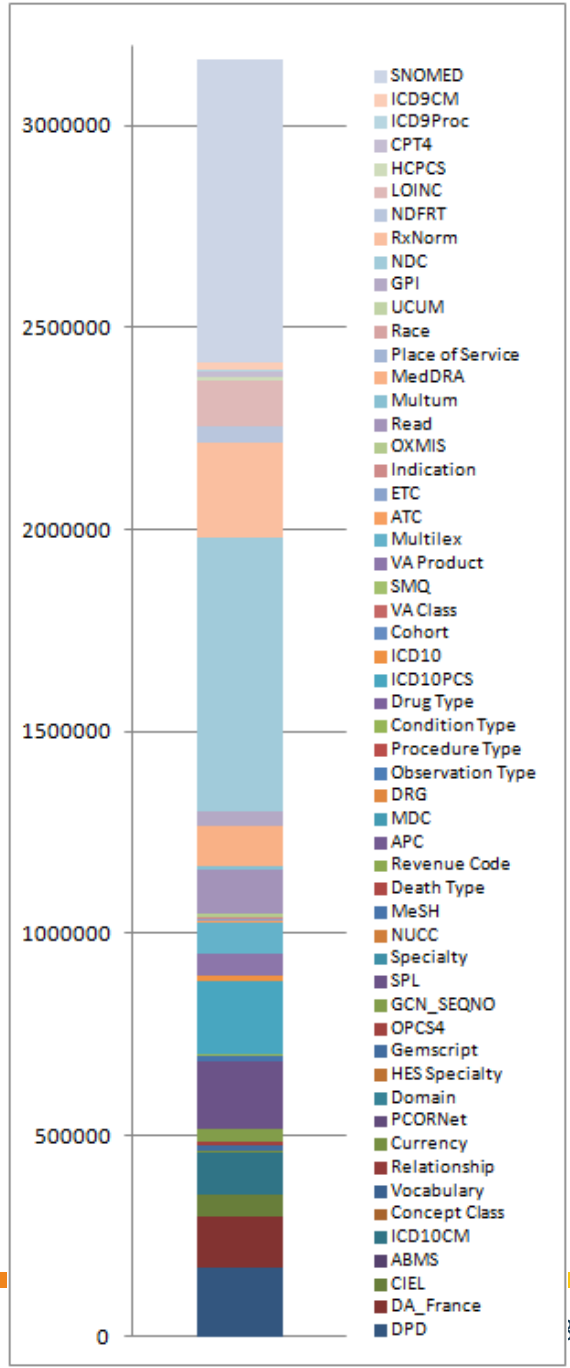
- There are two standard queries to help us use the OMOP Vocabulary:
  - `SOURCE_TO_STANDARD.sql`
  - `SOURCE_TO_SOURCE.sql`

- https://github.com/OHDSI/Tutorial-ETL
  - Materials → Queries

OMOP Vocab

- If your source data's codes are in the OMOP Vocab you can use it to translate to a standard

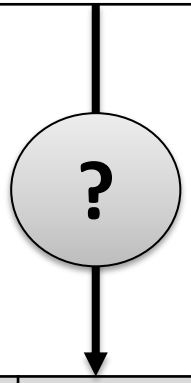
- Synthea already speaks standard terminology:
  - Conditions = SNOMED
  - Drugs = RxNorm
  - Procedures = SNOMED
  - Observations = SNOMED



Mapping a Lauren Row to CONCEPT_ID

```
SELECT *
FROM RAW_LAUREN.CONDITIONS
WHERE ENCOUNTER = '70'
```

START	STOP	PATIENT	ENCOUNTER	CODE	DESCRIPTION
1/6/2010		1	70	266599000	Dysmenorrhea



CONDITION_CONCEPT_ID	CONDITION_SOURCE_CONCEPT_ID

Source to Standard

```
WITH CTE_VOCAB_MAP AS (
      SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS SOURCE_CONCEPT_ID, c.concept_name AS SOURCE_CODE_DESCRIPTION,
       c.vocabulary_id AS SOURCE_VOCABULARY_ID, c.domain_id AS SOURCE_DOMAIN_ID, c.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,
       c.VALID_START_DATE AS SOURCE_VALID_START_DATE, c.VALID_END_DATE AS SOURCE_VALID_END_DATE, c.INVALID_REASON AS SOURCE_INVALID_REASON,
       c1.concept_id AS TARGET_CONCEPT_ID, c1.concept_name AS TARGET_CONCEPT_NAME, c1.VOCABULARY_ID AS TARGET_VOCABUALRY_ID,
       c1.domain_id AS TARGET_DOMAIN_ID, c1.concept_class_id AS TARGET_CONCEPT_CLASS_ID, c1.INVALID_REASON AS TARGET_INVALID_REASON,
       c1.standard_concept AS TARGET_STANDARD_CONCEPT
       FROM CONCEPT C
             JOIN CONCEPT_RELATIONSHIP CR
                       ON C.CONCEPT_ID = CR.CONCEPT_ID_1
                       AND CR.invalid_reason IS NULL
                       AND cr.relationship_id = 'Maps to'
              JOIN CONCEPT C1
                       ON CR.CONCEPT_ID_2 = C1.CONCEPT_ID
                       AND C1.INVALID_REASON IS NULL
       UNION

       SELECT source_code, SOURCE_CONCEPT_ID, SOURCE_CODE_DESCRIPTION, source_vocabulary_id, c1.domain_id AS SOURCE_DOMAIN_ID,
       c2.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,c1.VALID_START_DATE AS SOURCE_VALID_START_DATE,
       c1.VALID_END_DATE AS SOURCE_VALID_END_DATE, stcm.INVALID_REASON AS SOURCE_INVALID_REASON,target_concept_id,
       c2.CONCEPT_NAME AS TARGET_CONCEPT_NAME, target_vocabulary_id, c2.domain_id AS TARGET_DOMAIN_ID,
       c2.concept_class_id AS TARGET_CONCEPT_CLASS_ID, c2.INVALID_REASON AS TARGET_INVALID_REASON,
       c2.standard_concept AS TARGET_STANDARD_CONCEPT
       FROM source_to_concept_map stcm
              LEFT OUTER JOIN CONCEPT c1
                     ON c1.concept_id = stcm.source_concept_id
              LEFT OUTER JOIN CONCEPT c2
                     ON c2.CONCEPT_ID = stcm.target_concept_id
       WHERE stcm.INVALID_REASON IS NULL
)
SELECT TARGET_CONCEPT_ID, TARGET_CONCEPT_NAME, TARGET_DOMAIN_ID
FROM CTE_VOCAB_MAP
WHERE SOURCE_CODE = '266599000'
AND TARGET_STANDARD_CONCEPT = 'S'
```

Source to Standard

```
WITH CTE_VOCAB_MAP AS (
 SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS SOURCE_CONCEPT_ID, c.concept_name AS SOURCE_CODE_DESCRIPTION,
 c.vocabulary_id AS SOURCE_VOCABULARY_ID, c.domain_id AS SOURCE_DOMAIN_ID, c.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,
 c.VALID_START_DATE AS SOURCE_VALID_START_DATE, c.VALID_END_DATE AS SOURCE_VALID_END_DATE, c.INVALID_REASON AS SOURCE_INVALID_REASON,
 c1.concept_id AS TARGET_CONCEPT_ID, c1.concept_name AS TARGET_CONCEPT_NAME, c1.VOCABULARY_ID AS TARGET_VOCABUALRY_ID,
 c1.domain_id AS TARGET_DOMAIN_ID, c1.concept_class_id AS TARGET_CONCEPT_CLASS_ID, c1.INVALID_REASON AS TARGET_INVALID_REASON,
 c1.standard_concept AS TARGET_STANDARD_CONCEPT
  FROM CONCEPT C
        JOIN CONCEPT_RELATIONSHIP CR
                  ON C.CONCEPT_ID = CR.CONCEPT_ID_1
                  AND CR.invalid_reason IS NULL
                  AND cr.relationship_id = 'Maps to'
         JOIN CONCEPT C1
                  ON CR.CONCEPT_ID_2 = C1.CONCEPT_ID
                  AND C1.INVALID_REASON IS NULL
 UNION

        SELECT source_code, SOURCE_CONCEPT_ID, SO                                    id AS SOURCE_DOMAIN_ID,
        c2.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CL
        c1.VALID_END_DATE AS SOURCE_VALID_END_DA                                     oncept_id,
        c2.CONCEPT_NAME AS TARGET_CONCEPT_NAME,
        c2.concept_class_id AS TARGET_CONCEPT_CL
        c2.standard_concept AS TARGET_STANDARD_C
         FROM source_to_concept_map stcm
                LEFT OUTER JOIN CONCEPT c1
                        ON c1.concept_id = st
                LEFT OUTER JOIN CONCEPT c2
                        ON c2.CONCEPT_ID = stcm.target_concept_id
         WHERE stcm.INVALID_REASON IS NULL
)
SELECT TARGET_CONCEPT_ID, TARGET_CONCEPT_NAME, TARGET_DOMAIN_ID
FROM CTE_VOCAB_MAP
WHERE SOURCE_CODE = '266599000'
AND TARGET_STANDARD_CONCEPT = 'S'
```

Look in the
OMOP Vocabulary for a map

Source to Standard

```
WITH CTE_VOCAB_MAP AS (
      SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS SOURCE_CONCEPT_ID, c.concept_name AS SOURCE_CODE_DESCRIPTION,
       c.vocabulary_id AS SOURCE_VOCABULARY_ID, c.domain_id AS SOURCE_DOMAIN_ID, c.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,
       c.VALID_START_DATE AS SOURCE_VALID_START_DATE, c.VALID_END_DATE AS SOURCE_VALID_END_DATE, c.INVALID_REASON AS SOURCE_INVALID_REASON,
       c1.concept_id AS TARGET_CONCEPT_ID, c1.concept_name AS TARGET_CONCEPT_NAME, c1.VOCABULARY_ID AS TARGET_VOCABUALRY_ID,
       c1.domain_id AS TARGET_DOMAIN_ID, c1.co                                                    EASON AS TARGET_INVALID_REASON,
       c1.standard_concept AS TARGET_STANDARD_
        FROM CONCEPT C
              JOIN CONCEPT_RELATIONSHIP C
                        ON C.CONCEPT_ID
                        AND CR.invalid_r
                        AND cr.relations
               JOIN CONCEPT C1
                        ON CR.CONCEPT_ID
                        AND C1.INVALID_REA
       UNION
    SELECT source_code, SOURCE_CONCEPT_ID, SOURCE_CODE_DESCRIPTION, source_vocabulary_id, c1.domain_id AS SOURCE_DOMAIN_ID,
    c2.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,c1.VALID_START_DATE AS SOURCE_VALID_START_DATE,
    c1.VALID_END_DATE AS SOURCE_VALID_END_DATE, stcm.INVALID_REASON AS SOURCE_INVALID_REASON,target_concept_id,
    c2.CONCEPT_NAME AS TARGET_CONCEPT_NAME, target_vocabulary_id, c2.domain_id AS TARGET_DOMAIN_ID,
    c2.concept_class_id AS TARGET_CONCEPT_CLASS_ID, c2.INVALID_REASON AS TARGET_INVALID_REASON,
    c2.standard_concept AS TARGET_STANDARD_CONCEPT
     FROM source_to_concept_map stcm
            LEFT OUTER JOIN CONCEPT c1
                  ON c1.concept_id = stcm.source_concept_id
            LEFT OUTER JOIN CONCEPT c2
                  ON c2.CONCEPT_ID = stcm.target_concept_id
     WHERE stcm.INVALID_REASON IS NULL
)
SELECT TARGET_CONCEPT_ID, TARGET_CONCEPT_NAME, TARGET_DOMAIN_ID
FROM CTE_VOCAB_MAP
WHERE SOURCE_CODE = '266599000'
AND TARGET_STANDARD_CONCEPT = 'S'
```

Look in the Source to Concept Map table for a map

Source to Standard

```
WITH CTE_VOCAB_MAP AS (
      SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS SOURCE_CONCEPT_ID, c.concept_name AS SOURCE_CODE_DESCRIPTION,
       c.vocabulary_id AS SOURCE_VOCABULARY_ID, c.domain_id AS SOURCE_DOMAIN_ID, c.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,
       c.VALID_START_DATE AS SOURCE_VALID_START_DATE, c.VALID_END_DATE AS SOURCE_VALID_END_DATE, c.INVALID_REASON AS SOURCE_INVALID_REASON,
       c1.concept_id AS TARGET_CONCEPT_ID, c1.concept_name AS TARGET_CONCEPT_NAME, c1.VOCABULARY_ID AS TARGET_VOCABUALRY_ID,
       c1.domain_id AS TARGET_DOMAIN_ID, c1.concept_class_id AS TARGET_CONCEPT_CLASS_ID, c1.INVALID_REASON AS TARGET_INVALID_REASON,
       c1.standard_concept AS TARGET_STANDARD_CONCEPT
        FROM CONCEPT C
              JOIN CONCEPT_RELATIONSHIP CR
                        ON C.CONCEPT_ID = CR.CONCEPT_ID_1
                        AND CR.invalid_reason IS NULL
                        AND cr.relationship_id = 'Maps to'
               JOIN CONCEPT C1
                        ON CR.CONCEPT_ID_2 = C1.CONCEPT_ID
                        AND C1.INVALID_REASON IS NULL
       UNION

      SELECT source_code, SOURCE_CONCEPT_ID, SOURCE_CODE_DESCRIPTION, source_vocabulary_id, c1.domain_id AS SOURCE_DOMAIN_ID,
                                                                    LID_START_DATE AS SOURCE_VALID_START_DATE,
                                                                    LID_REASON AS SOURCE_INVALID_REASON,target_concept_id,
                                                                    lary_id, c2.domain_id AS TARGET_DOMAIN_ID,
                                                                    VALID_REASON AS TARGET_INVALID_REASON,


                                                                    concept_id

                                                                    _concept_id
       WHERE stcm.INVALID_REASON IS NULL
)
SELECT TARGET_CONCEPT_ID, TARGET_CONCEPT_NAME, TARGET_DOMAIN_ID
FROM CTE_VOCAB_MAP
WHERE SOURCE_CODE = '266599000'
AND TARGET_STANDARD_CONCEPT = 'S'
```

Look up your source Code here

Mapping a Lauren Row to CONCEPT_ID: Source to Standard

START	STOP	PATIENT	ENCOUNTER	CODE	DESCRIPTION
1/6/2010		1	70	266599000	Dysmenorrhea

TARGET_ CONCEPT_ID	TARGET_ CONCEPT_NAME	TARGET_ DOMAIN_ID
194696	Dysmenorrhea	Condition

CONDITION_CONCEPT_ID	CONDITION_SOURCE_CONCEPT_ID
194696	

Source to Source

```
WITH CTE_VOCAB_MAP AS (
	SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS SOURCE_CONCEPT_ID,
		c.CONCEPT_NAME AS SOURCE_CODE_DESCRIPTION, c.vocabulary_id AS SOURCE_VOCABULARY_ID,
		c.domain_id AS SOURCE_DOMAIN_ID, c.concept_class_id AS SOURCE_CONCEPT_CLASS_ID,
		c.VALID_START_DATE AS SOURCE_VALID_START_DATE, c.VALID_END_DATE AS SOURCE_VALID_END_DATE,
		c.invalid_reason AS SOURCE_INVALID_REASON, c.concept_ID as TARGET_CONCEPT_ID,
		c.concept_name AS TARGET_CONCEPT_NAME, c.vocabulary_id AS TARGET_VOCABULARY_ID,
		c.domain_id AS TARGET_DOMAIN_ID, c.concept_class_id AS TARGET_CONCEPT_CLASS_ID,
		c.INVALID_REASON AS TARGET_INVALID_REASON, c.STANDARD_CONCEPT AS TARGET_STANDARD_CONCEPT
	FROM CONCEPT c

	UNION

	SELECT source_code, SOURCE_CONCEPT_ID, SOURCE_CODE_DESCRIPTION, source_vocabulary_id,
		c1.domain_id AS SOURCE_DOMAIN_ID, c2.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,
		c1.VALID_START_DATE AS SOURCE_VALID_START_DATE, c1.VALID_END_DATE AS SOURCE_VALID_END_DATE,
		stcm.INVALID_REASON AS SOURCE_INVALID_REASON,target_concept_id,
		c2.CONCEPT_NAME AS TARGET_CONCEPT_NAME, target_vocabulary_id, c2.domain_id AS TARGET_DOMAIN_ID,
		c2.concept_class_id AS TARGET_CONCEPT_CLASS_ID, c2.INVALID_REASON AS TARGET_INVALID_REASON,
		c2.standard_concept AS TARGET_STANDARD_CONCEPT
	FROM source_to_concept_map stcm
		LEFT OUTER JOIN CONCEPT c1
			ON c1.concept_id = stcm.source_concept_id
		LEFT OUTER JOIN CONCEPT c2
			ON c2.CONCEPT_ID = stcm.target_concept_id
	WHERE stcm.INVALID_REASON IS NULL
)
SELECT *
FROM CTE_VOCAB_MAP
/*EXAMPLE FILTERS*/
WHERE SOURCE_CODE = '266599000'
AND SOURCE_VOCABULARY_ID = 'SNOMED'
```

Source to Source

```
WITH CTE_VOCAB_MAP AS (
	SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS SOURCE_CONCEPT_ID,
		c.CONCEPT_NAME AS SOURCE_CODE_DESCRIPTION, c.vocabulary_id AS SOURCE_VOCABULARY_ID,
		c.domain_id AS SOURCE_DOMAIN_ID, c.concept_class_id AS SOURCE_CONCEPT_CLASS_ID,
		c.VALID_START_DATE AS SOURCE_VALID_START_DATE, c.VALID_END_DATE AS SOURCE_VALID_END_DATE,
		c.invalid_reason AS SOURCE_INVALID_REASON, c.concept_ID as TARGET_CONCEPT_ID,
		c.concept_name AS TARGET_CONCEPT_NAME, c.vocabulary_id AS TARGET_VOCABULARY_ID,
		c.domain_id AS TARGET_DOMAIN_ID, c.concept_class_id AS TARGET_CONCEPT_CLASS_ID,
		c.INVALID_REASON AS TARGET_INVALID_REASON, c.STANDARD_CONCEPT AS TARGET_STANDARD_CONCEPT
	FROM CONCEPT c

	UNION

	SELECT source_code, SOURCE_CONCEPT_ID, SOURCE_CODE_DESCRIPTION, source_vocabulary_id,
		c1.domain_id AS SOURCE_DOMAIN_ID, c2.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID,
		c1.VALID_START_DATE AS SOURCE_VALID_START_DATE, c1.VALID_END_DATE AS SOURCE_VALID_END_DATE,
		                                    ALID_REASON,target_concept_id,
		                                    AME, target_vocabulary_id, c2.domain_id AS TARGET_DOMAIN_ID,
		                                    PT_CLASS_ID, c2.INVALID_REASON AS TARGET_INVALID_REASON,
		                                    ARD_CONCEPT

		                                    rce_concept_id

		                                    rget_concept_id
	WHERE stcm.INVALID_REASON IS NULL
)
SELECT *
FROM CTE_VOCAB_MAP
/*EXAMPLE FILTERS*/
WHERE SOURCE_CODE = '266599000'
AND SOURCE_VOCABULARY_ID = 'SNOMED'
```

Look up your source Code here

Mapping a Lauren Row to CONCEPT_ID: Source to Source

START	STOP	PATIENT	ENCOUNTER	CODE	DESCRIPTION
1/6/2010		1	70	266599000	Dysmenorrhea

TARGET_ CONCEPT_ID	TARGET_ CONCEPT_NAME	TARGET_ DOMAIN_ID
194696	Dysmenorrhea	Condition

CONDITION_CONCEPT_ID	CONDITION_SOURCE_CONCEPT_ID
194696	194696

Mapping a Lauren Row to CONCEPT_ID: Source to Source

START	STOP	PATIENT	ENCOUNTER	CODE	DESCRIPTION
1/6/2010		1	70	266599000	Dysmenorrhea

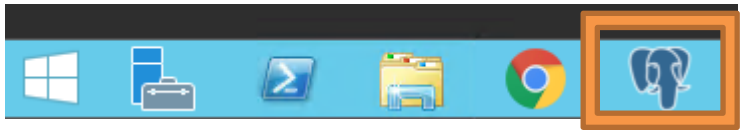
TARGET_ CONCEPT_ID	TARGET_ CONCEPT_NAME	TARGET_ DOMAIN_ID
194696	Dysmenorrhea	Condition

CONDITION_CONCEPT_ID	CONDITION_SOURCE_CONCEPT_ID
194696	194696

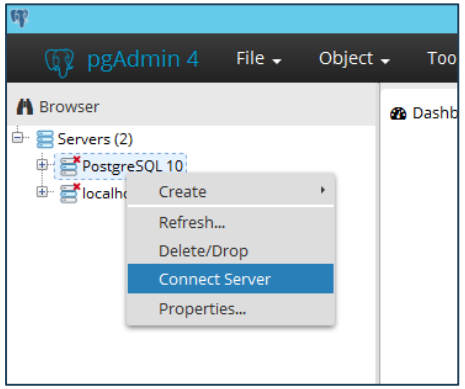
They are the same CONCEPT_IDs because the source codes in Synthea are also standard codes!

Mapping Source Codes – Your turn

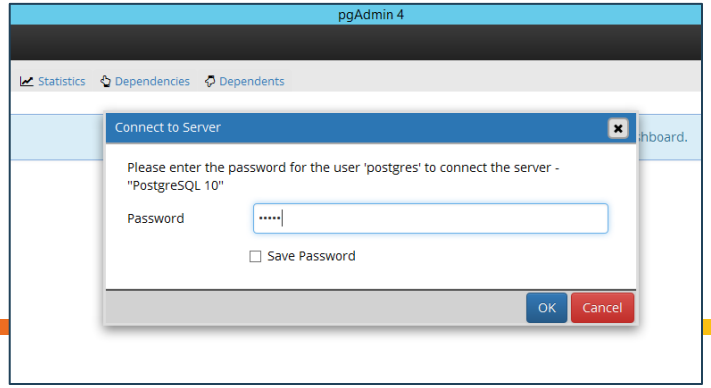
- Let's open PostgreSQL
  - Open up pgAdmin4 using the icon on the task bar



  - Expand the server list and right-click on PostgreSql 10 and choose Connect Server from the drop-down menu



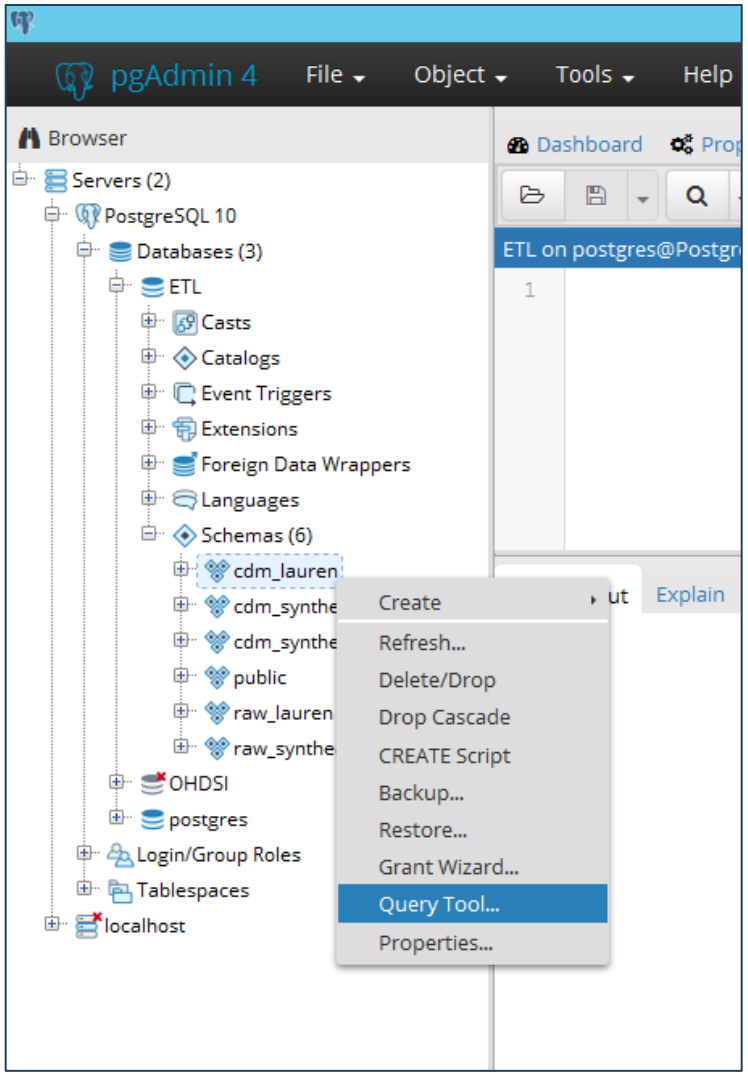
  - When it asks for a password, type in ohdsi



Mapping Source Codes – Your turn

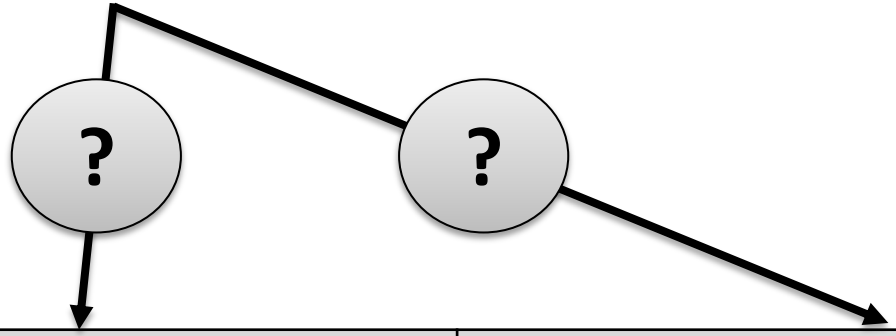
- Expand:
  a. Databases
  b. ETL
  c. Schemas

- Right click on cdm_synthea and choose Query Tool from the drop-down menu

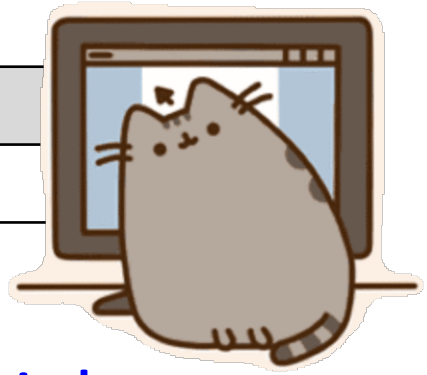


Mapping Source Codes – Your turn

CODE	DESCRIPTION	CODE TYPE
C83.3	Diffuse large B-cell lymphoma	ICD10 (not ICD10CM)



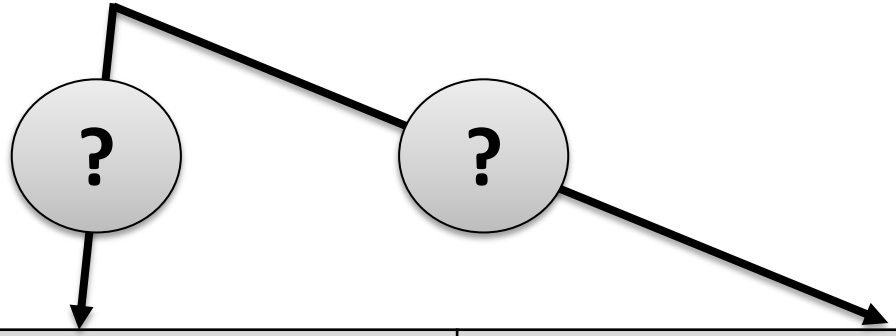
CONDITION_CONCEPT_ID	CONDITION_SOURCE_CONCEPT_ID



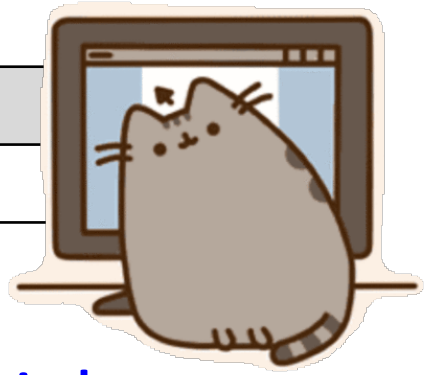
https://github.com/OHDSI/Tutorial-ETL/tree/master/materials/Queries

Mapping Source Codes – Your turn

CODE	DESCRIPTION	CODE TYPE
C83.3	Diffuse large B-cell lymphoma	ICD10 (not ICD10CM)



CONDITION_CONCEPT_ID	CONDITION_SOURCE_CONCEPT_ID
44808122	45600549



https://github.com/OHDSI/Tutorial-ETL/tree/master/materials/Queries

Usagi

- When the Vocabulary does not have your source codes you will need to create a map to OMOP Vocabulary Concepts
- Usagi is Japanese for rabbit and was named after the first mapping exercise it was used for; mapping source codes used in a Japanese dataset into OMOP Vocabulary concepts
- Usagi software tool to help with process of mapping source codes to OMOP concepts
- Usagi performs text similarity between your source codes and what is in the OMOP Vocabulary

Usagi Process

1. Get a copy of the **Vocabulary** from ATHENA
2. Download **Usagi**
3. Have Usagi **build an index** on the Vocabulary
4. **Load your source codes** and let Usagi process them
5. **Review and update suggest mappings** with someone who has medical knowledge
6. **Export codes** into the SOURCE_TO_CONCEPT_MAP

Usagi Process

*1. Get a copy of the **Vocabulary** from ATHENA*

http://athena.ohdsi.org

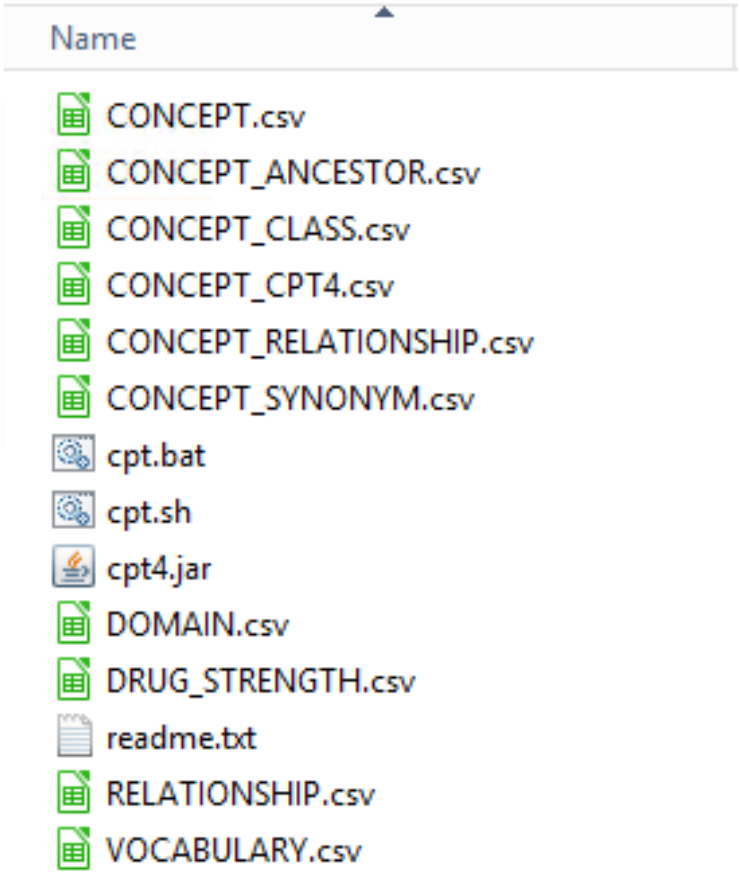
ATHENA | SEARCH | DOWNLOAD | Erica Voss

Show all | SHOW HISTORY | DOWNLOAD VOCABULARIES

ID (CDM V4.5)	CODE (CDM V5)	NAME	REQUIRED	LATEST UPDATE
1	SNOMED	Systematic Nomenclature of Medicine - Clinical Terms (IHTSDO)		31-Jan-2019
2	ICD9CM	International Classification of Diseases, Ninth Revision, Clinical Modification, Volume 1 and 2 (NCHS)		01-Oct-2014
3	ICD9Proc	International Classification of Diseases, Ninth Revision, Clinical Modification, Volume 3 (NCHS)		01-Oct-2014
4	CPT4	Current Procedural Terminology version 4 (AMA)	EULA required	05-Nov-2018

Usagi Process

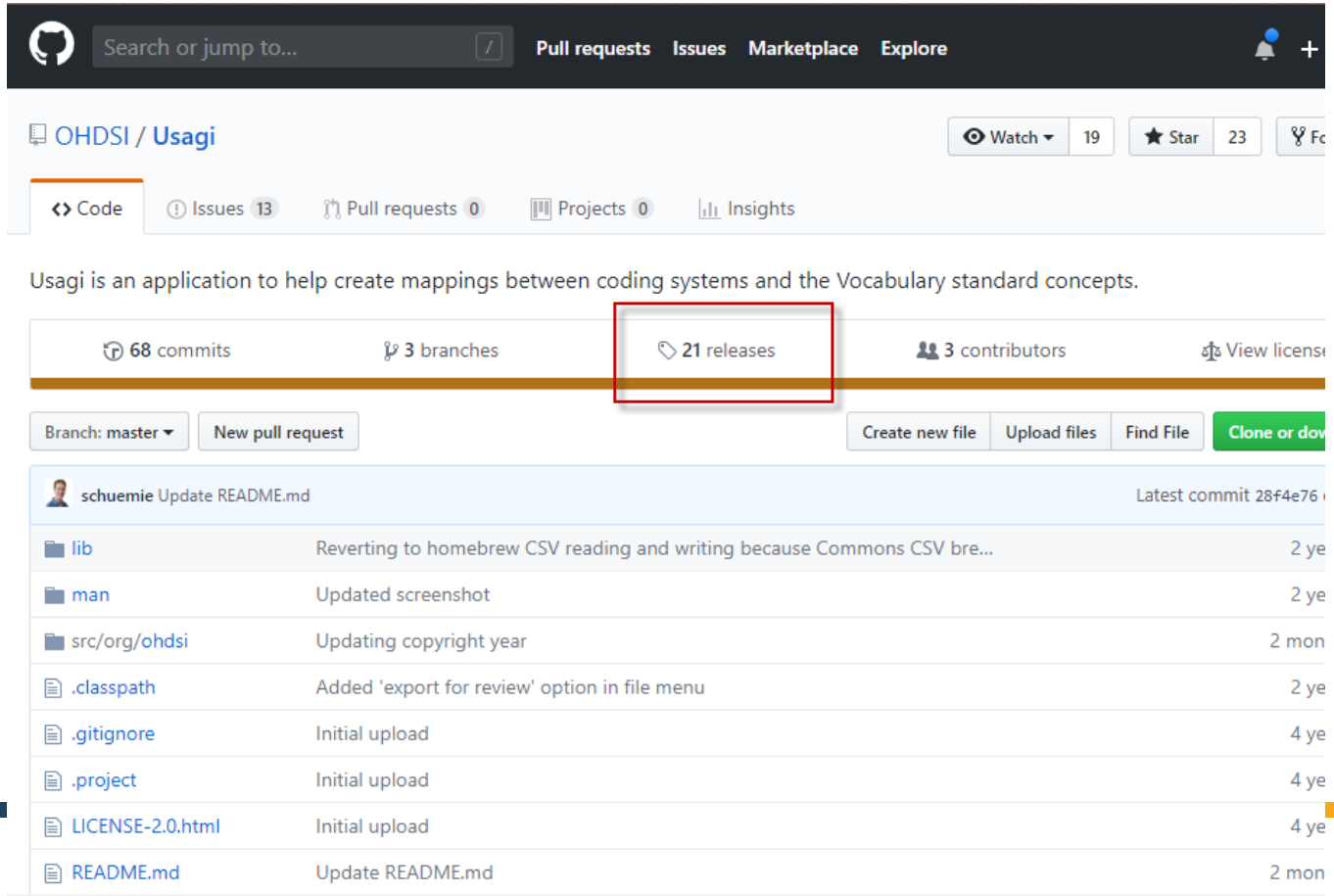
*1. Get a copy of the **Vocabulary** from ATHENA*



Usagi Process

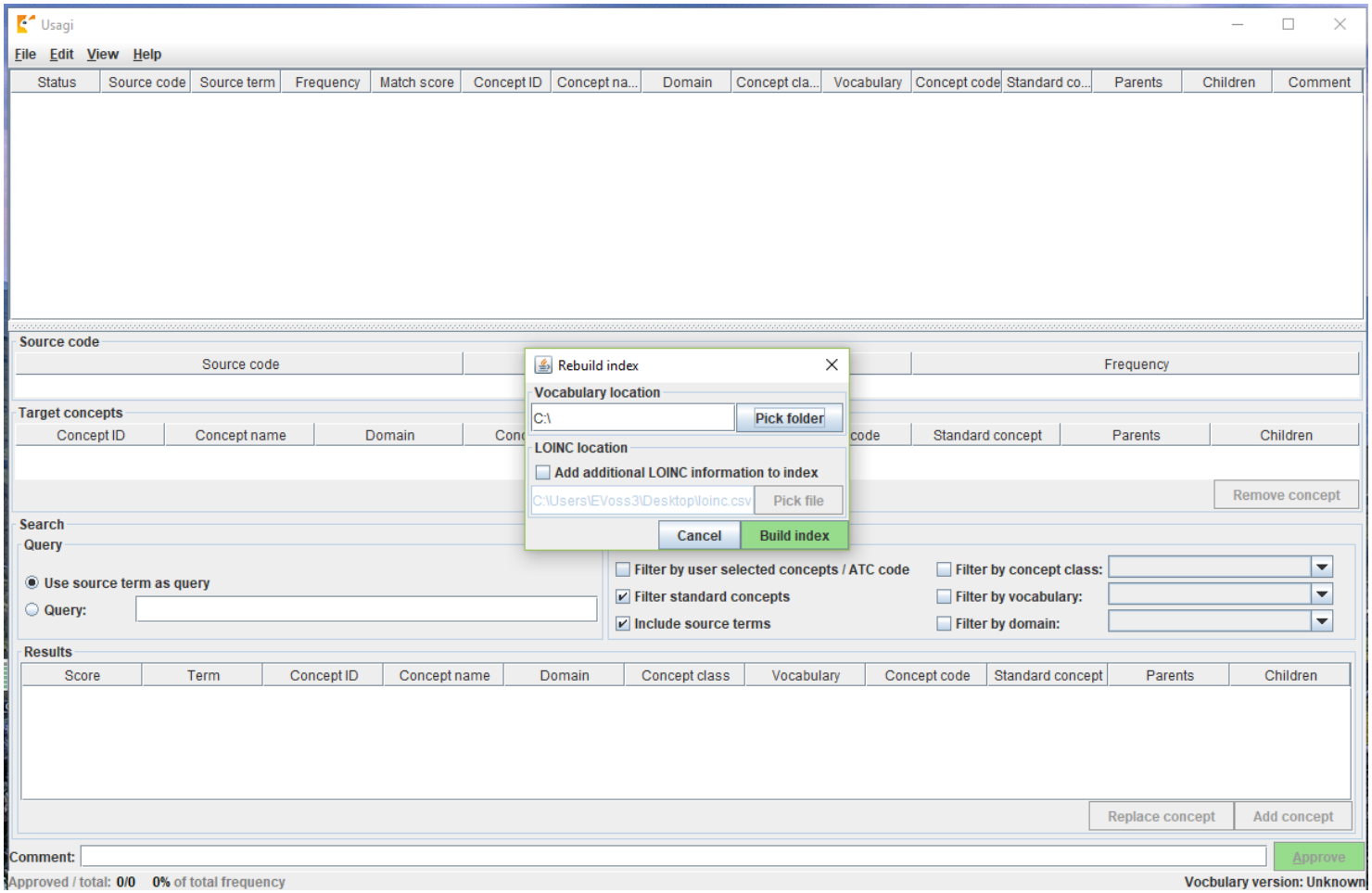
*2. Download **Usagi***

https://github.com/OHDSI/Usagi



Usagi Process

*3. Have Usagi **build an index** on the Vocabulary*



Usagi Process

*4. **Load your source codes**, let Usagi process them*

- Generate an XLSX of **distinct codes** from source system with descriptions and frequency

- If the codes are not in English, use Google Translate to convert

ICPC_CODE	ICPC_DESCRIPTION_DUTCH	FREQUENCY
R74	Acute infectie bovenste luchtwegen	800000
R44	Immunisatie/preventieve medicatie	1000000
R05	Hoesten	880000
A97	Geen ziekte	500000
S74	Dermatomycose(n)	100000
U71	Cystitis/urineweginfecties	500000
L99	Andere ziekte(n) bewegingsapparaat	100000
R74.02	Acute pharyngitis	800000
R78.00	Acute bronchitis/bronchiolitis	300000
W78.00	Zwangerschap (bevestigd)	100000
T83.0	overgewicht	100000
R65.00	episode op initiatief derde	1

Usagi Process

*4. **Load your source codes**, let Usagi process them*

- Import the codes into Usagi

Import codes from DUTCH_ICPC_CONDITION_CODES_TO_MAP.xlsx

ICPC_CODE	ICPC_DESCRIPTION_DUTCH	ICPC_DESCRIPTION_ENGLIGH	FREQUENCY
R74	Acute infectie bovenste luchtwegen	Acute upper respiratory tract infection	800000
R44	Immunisatie/preventieve medicatie	Immunization / preventive medication	1000000
R05	Hoesten	Cough	880000
A97	Geen ziekte	No illness	500000
S74	Dermatomycose(n)	Dermatomycosis (s)	100000
U71	Cystitis/urineweginfecties	Cystitis / urinary tract infections	500000
L99	Andere ziekte(n) bewegingsapparaat	Other disease (s) musculoskeletal system	100000
R74.02	Acute pharyngitis	Acute pharyngitis	800000
R78.00	Acute bronchitis/bronchiolitis	Acute bronchitis / bronchiolitis	300000
W78.00	Zwangerschap (bevestigd)	Pregnancy (confirmed)	100000
T83.0	overgewicht	overweight	100000
R65.00	episode op initiatief derde	episode on the initiative third	1

Importing codes...

Column mapping

Source code column	ICPC_CODE
Source name column	ICPC_DESCRIPTION_ENGLIGH
Source frequency column	FREQUENCY
Auto concept ID column	
Additional info column	ICPC_DESCRIPTION_DUTCH

Filters

- [ ] Filter by user selected concepts
- [x] Filter standard concepts
- [x] Include source terms
- [ ] Filter by concept class:
- [ ] Filter by vocabulary:
- [ ] Filter by domain:

Cancel Import

Usagi Process

*5. **Review and update suggest mappings** with someone who has medical knowledge*

Usagi

File Edit View Help

Status	Source code	Source term	Frequency	ICPC_DES...	Match score	Concept ID	Concept na...	Domain	Concept cl...	Vocabulary	Concept co...	Standard c...	Parents	Children	Comment
Unchecked	A97	No illness	500000	Geen ziekte	0.82	4192174	Illness	Condition	Clinical Fin...	SNOMED	39104002	S	1	3	
Unchecked	S74	Dermatomy...	100000	Dermatomy...	0.81	135473	Dermatoph...	Condition	Clinical Fin...	SNOMED	47382004	S	4	25	
Unchecked	L99	Other disea...	100000	Andere ziek...	0.77	4244662	Disorder of ...	Condition	Clinical Fin...	SNOMED	928000	S	3	84	
Unchecked	R74.02	Acute phary...	800000	Acute phary...	1.00	25297	Acute phary...	Condition	Clinical Fin...	SNOMED	363746003	S	6	10	
Unchecked	U71	Cystitis / uri...	500000	Cystitis/urin...	0.71	81902	Urinary trac...	Condition	Clinical Fin...	SNOMED	68566005	S	5	17	
Unchecked	R78.00	Acute bronc...	300000	Acute bronc...	0.84	260125	Acute bronc...	Condition	Clinical Fin...	SNOMED	5505005	S	5	4	
Unchecked	W78.00	Pregnancy ...	100000	Zwangersc...	0.84	4299535	Pregnant	Condition	Clinical Fin...	SNOMED	77386006	S	2	17	
Unchecked	T83.0	overweight	100000	overgewicht	1.00	437525	Overweight	Observation	Clinical Fin...	SNOMED	238131007	S	2	5	
Unchecked	R74	Acute uppe...	800000	Acute infect...	1.00	257011	Acute uppe...	Condition	Clinical Fin...	SNOMED	54398005	S	6	22	
Unchecked	R65.00	episode on...	1	episode op...	0.35	444406	Acute sube...	Condition	Clinical Fin...	SNOMED	70422006	S	4	0	
Unchecked	R44	Immunizati...	1000000	Immunisati...	0.70	4144375	Active imm...	Procedure	Procedure	SNOMED	33879002	S	2	19	
Unchecked	R05	Cough	880000	Hoesten	1.00	254761	Cough	Condition	Clinical Fin...	SNOMED	49727002	S	2	38	

Source code

Source code	Source term	Frequency	ICPC_DESCRIPTION_DUTCH
A97	No illness	500000	Geen ziekte

Target concepts

Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3

Remove concept

Search

Query

Use source term as query

Query:

Filters

Filter by user selected concepts

Filter standard concepts

Include source terms

Filter by concept class:

Filter by vocabulary:

Filter by domain:

Results

Score	Term	Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
0.82	Illness	4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3
0.80	Mental illness	4214703	Mental illness	Observation	Qualifier Value	SNOMED	394816006	S	1	0
0.80	Mental illness	432586	Mental disorder	Condition	Clinical Finding	SNOMED	74732009	S	2	41
0.78	Viral illness	440029	Viral disease	Condition	Clinical Finding	SNOMED	34014006	S	3	31
0.77	Mass illness	45883959	Mass illness	Meas Value	Answer	LOINC	LA18096-0	S	0	0
0.75	Stillness	4092256	Stillness	Condition	Clinical Finding	SNOMED	247902008	S	3	1

Replace concept | Add concept

Comment:

Approve

Approved / total: **0 / 12** **0.0%** of total frequency

Vocbulary version: v5.0 19-NOV-18

Usagi Process

*5. **Review and update suggest mappings** with someone who has medical knowledge*

Usagi

File Edit View Help

Status	Source code	Source term	Frequency	ICPC_	S...	Match score	oncept ID	Concept na...	Domain	Concept cl...	Vocabulary	Concept co...	Standard c...	Parents	Children	Comment
Unchecked	A97	No illness	500000	Geen z	te	0.82	92174	Illness	Condition	Clinical Fin...	SNOMED	39104002	S	1	3	
Unchecked	S74	Dermatomy...	100000	Derma	ny...	0.81	5473	Dermatoph...	Condition	Clinical Fin...	SNOMED	47382004	S	4	25	
Unchecked	L99	Other disea...	100000	Andere	ek...	0.77	44662	Disorder of ...	Condition	Clinical Fin...	SNOMED	928000	S	3	84	
Unchecked	R74.02	Acute phary...	800000	Acute	ry...	1.00	297	Acute phary...	Condition	Clinical Fin...	SNOMED	363746003	S	6	10	
Unchecked	U71	Cystitis / uri...	500000	Cystitis	in...	0.71	902	Urinary trac...	Condition	Clinical Fin...	SNOMED	68566005	S	5	17	
Unchecked	R78.00	Acute bronc...	300000	Acute	nc...	0.84	0125	Acute bronc...	Condition	Clinical Fin...	SNOMED	5505005	S	5	4	
Unchecked	W78.00	Pregnancy ...	100000	Zwang	c...	0.84	99535	Pregnant	Condition	Clinical Fin...	SNOMED	77386006	S	2	17	
Unchecked	T83.0	overweight	100000	overge	ht	1.00	7525	Overweight	Observation	Clinical Fin...	SNOMED	238131007	S	2	5	
Unchecked	R74	Acute uppe...	800000	Acute i	ct...	1.00	7011	Acute uppe...	Condition	Clinical Fin...	SNOMED	54398005	S	6	22	
Unchecked	R65.00	episode on...	1	episod	p...	0.35	4406	Acute sube...	Condition	Clinical Fin...	SNOMED	70422006	S	4	0	
Unchecked	R44	Immunizati...	1000000	Immun	ti...	0.70	44375	Active imm...	Procedure	Procedure	SNOMED	33879002	S	2	19	
Unchecked	R05	Cough	880000	Hoest		1.00	4761	Cough	Condition	Clinical Fin...	SNOMED	49727002	S	2	38	

Source code

Source code	Source term	Frequency	ICPC_DESCRIPTION_DUTCH
A97	No illness	500000	Geen ziekte

Target concepts

Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3

Remove concept

Search

Query

- Use source term as query
- Query:

Filters

- Filter by user selected concepts
- Filter standard concepts
- Include source terms
- Filter by concept class:
- Filter by vocabulary:
- Filter by domain:

Results

Score	Term	Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
0.82	Illness	4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3
0.80	Mental illness	4214703	Mental illness	Observation	Qualifier Value	SNOMED	394816006	S	1	0
0.80	Mental illness	432586	Mental disorder	Condition	Clinical Finding	SNOMED	74732009	S	2	41
0.78	Viral illness	440029	Viral disease	Condition	Clinical Finding	SNOMED	34014006	S	3	31
0.77	Mass illness	45883959	Mass illness	Meas Value	Answer	LOINC	LA18096-0	S	0	0
0.75	Stillness	4092256	Stillness	Condition	Clinical Finding	SNOMED	247902008	S	3	1

Replace concept | Add concept

Comment: Approve

Approved / total: **0 / 12** **0.0%** of total frequency

Vocbulary version: v5.0 19-NOV-18

Usagi Process

*5. **Review and update suggest mappings** with someone who has medical knowledge*

Usagi

File Edit View Help

Status	Source code	Source term	Frequency	ICPC_DES...	Match score	Concept ID	Concept na...	Domain	Concept cl...	Vocabulary	Concept co...	Standard c...	Parents	Children	Comment
Unchecked	A97	No illness	500000	Geen ziekte	0.82	4192174	Illness	Condition	Clinical Fin...	SNOMED	39104002	S	1	3	
Unchecked	S74	Dermatomy...	100000	Dermatomy...	0.81	135473	Dermatoph...	Condition	Clinical Fin...	SNOMED	47382004	S	4	25	
Unchecked	L99	Other disea...	100000	Andere ziek...	0.77	4244662	Disorder of ...	Condition	Clinical Fin...	SNOMED	928000	S	3	84	
Unchecked	R74.02	Acute phary...	800000	Acute phary...	1.00	25297	Acute phary...	Condition	Clinical Fin...	SNOMED	363746003	S	6	10	
Unchecked	U71	Cystitis / uri...	500000	Cystitis/urin...	0.71	81902	Urinary trac...	Condition	Clinical Fin...	SNOMED	68566005	S	5	17	
Unchecked	R78.00	Acute bronc...	300000	Acute bronc...	0.84	260125	Acute bronc...	Condition	Clinical Fin...	SNOMED	5505005	S	5	4	
Unchecked	W78.00	Pregnancy ...	100000	Zwangersc...	0.84	4299535	Pregnant	Condition	Clinical Fin...	SNOMED	77386006	S	2	17	
Unchecked	T83.0	overweight	100000	overgewicht	1.00	437525	Overweight	Observation	Clinical Fin...	SNOMED	238131007	S	2	5	
Unchecked	R74	Acute uppe...	800000	Acute infect...	1.00	257011	Acute uppe...	Condition	Clinical Fin...	SNOMED	54398005	S	6	22	
Unchecked	R65.00	episode on...	1	episode op...	0.35	444406	Acute sube...	Condition	Clinical Fin...	SNOMED	70422006	S	4	0	
Unchecked	R44	Immunizati...	1000000	Immunisati...	0.70	4144375	Active imm...	Procedure	Procedure	SNOMED	33879002	S	2	19	

Source code

Source code	Source term	Frequency	ICPC_DESCRIPTION_DUTCH
A97	No illness	500000	Geen ziekte

Target concepts

Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3

Remove concept

Query

Use source term as query

Query:

Filters

Filter by user selected concepts

Filter standard concepts

Include source terms

Filter by concept class:

Filter by vocabulary:

Filter by domain:

Results

Score	Term	Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
0.82	Illness	4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3
0.80	Mental illness	4214703	Mental illness	Observation	Qualifier Value	SNOMED	394816006	S	1	0
0.80	Mental illness	432586	Mental disorder	Condition	Clinical Finding	SNOMED	74732009	S	2	41
0.78	Viral illness	440029	Viral disease	Condition	Clinical Finding	SNOMED	34014006	S	3	31
0.77	Mass illness	45883959	Mass illness	Meas Value	Answer	LOINC	LA18096-0	S	0	0
0.75	Stillness	4092256	Stillness	Condition	Clinical Finding	SNOMED	247902008	S	3	1

Replace concept | Add concept

Comment:

Approve

Approved / total: 0 / 12 0.0% of total frequency

Vocbulary version: v5.0 19-NOV-18

Usagi Process

5. ***Review and update suggest mappings*** *with someone who has medical knowledge*

Usagi

File Edit View Help

Status	Source code	Source term	Frequency	ICPC_DES...	Match score	Concept ID	Concept na...	Domain	Concept cl...	Vocabulary	Concept co...	Standard c...	Parents	Children	Comment
Unchecked	A97	No illness	500000	Geen ziekte	0.82	4192174	Illness	Condition	Clinical Fin...	SNOMED	39104002	S	1	3	
Unchecked	S74	Dermatomy...	100000	Dermatomy...	0.81	135473	Dermatoph...	Condition	Clinical Fin...	SNOMED	47382004	S	4	25	
Unchecked	L99	Other disea...	100000	Andere ziek...	0.77	4244662	Disorder of ...	Condition	Clinical Fin...	SNOMED	928000	S	3	84	
Unchecked	R74.02	Acute phary...	800000	Acute phary...	1.00	25297	Acute phary...	Condition	Clinical Fin...	SNOMED	363746003	S	6	10	
Unchecked	U71	Cystitis / uri...	500000	Cystitis/urin...	0.71	81902	Urinary trac...	Condition	Clinical Fin...	SNOMED	68566005	S	5	17	
Unchecked	R78.00	Acute bronc...	300000	Acute bronc...	0.84	260125	Acute bronc...	Condition	Clinical Fin...	SNOMED	5505005	S	5	4	
Unchecked	W78.00	Pregnancy ...	100000	Zwangersc...	0.84	4299535	Pregnant	Condition	Clinical Fin...	SNOMED	77386006	S	2	17	
Unchecked	T83.0	overweight	100000	overgewicht	1.00	437525	Overweight	Observation	Clinical Fin...	SNOMED	238131007	S	2	5	
Unchecked	R74	Acute uppe...	800000	Acute infect...	1.00	257011	Acute uppe...	Condition	Clinical Fin...	SNOMED	54398005	S	6	22	
Unchecked	R65.00	episode on...	1	episode op...	0.35	444406	Acute sube...	Condition	Clinical Fin...	SNOMED	70422006	S	4	0	
Unchecked	R44	Immunizati...	1000000	Immunisati...	0.70	4144375	Active imm...	Procedure	Procedure	SNOMED	33879002	S	2	19	
Unchecked	R05	Cough	880000	Hoesten	1.00	254761	Cough	Condition	Clinical Fin...	SNOMED	49727002	S	2	38	

Source code

Source code	Source term	Frequency	ICPC_DESCRIPTION_DUTCH
A97	No illness	500000	Geen ziekte

Target concepts

Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3

Search

Query

- (•) Use source term as query
- ( ) Query:

Filters

- [ ] Filter by user selected concepts
- [x] Filter standard concepts
- [x] Include source terms
- [ ] Filter by concept class:
- [ ] Filter by vocabulary:
- [ ] Filter by domain:

Results

Score	Term	Concept ID	Concept name	Domain	Concept class	Vocabulary	Concept code	Standard concept	Parents	Children
0.82	Illness	4192174	Illness	Condition	Clinical Finding	SNOMED	39104002	S	1	3
0.80	Mental illness	4214703	Mental illness	Observation	Qualifier Value	SNOMED	394816006	S	1	0
0.80	Mental illness	432586	Mental disorder	Condition	Clinical Finding	SNOMED	74732009	S	2	41
0.78	Viral illness	440029	Viral disease	Condition	Clinical Finding	SNOMED	34014006	S	3	31
0.77	Mass illness	45883959	Mass illness	Meas Value	Answer	LOINC	LA18096-0	S	0	0
0.75	Stillness	4092256	Stillness	Condition	Clinical Finding	SNOMED	247902008	S	3	1

Replace concept | Add concept

Comment: | Approve

Usagi Process

*5. **Review and update suggest mappings** with someone who has medical knowledge*

- It is okay to map to zero or 0 – “No matching concept”
- A source code might end up being mapped to two concepts
- You might have what the system considers one domain but the OMOP Vocabulary lumps into another domain

Usagi Process

*6. **Export codes** into the SOURCE_TO_CONCEPT_MAP*

- After you have completed, you will export the relationships

- When exporting you will give a Vocabulary ID, for example JNJ_JMDC_PROVIDERS would signify a Johnson & Johnson map for the database JMDC for provider codes.

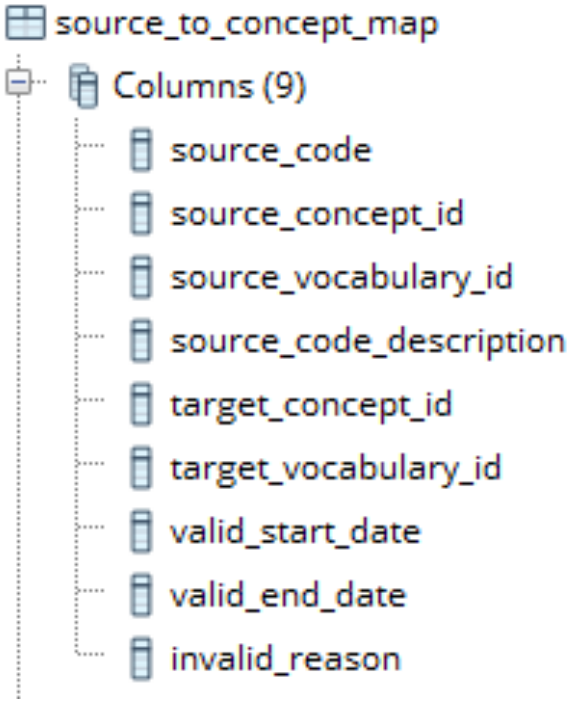
source_ code	source_ concept_id	source_ vocab_id	source_ code_ description	target_ concept_ id	target_ vocab_id	valid_ start_date	valid_ end_ date	invalid_ reason
R74.02	0	TEST_ VOCAB	Acute pharyngitis	25297	SNOMED	1/1/1970	12/31/2099	

R74.02 - Acute pharyngitis = 25297 - Acute pharyngitis

Usagi Process

*6. **Export codes** into the SOURCE_TO_CONCEPT_MAP*

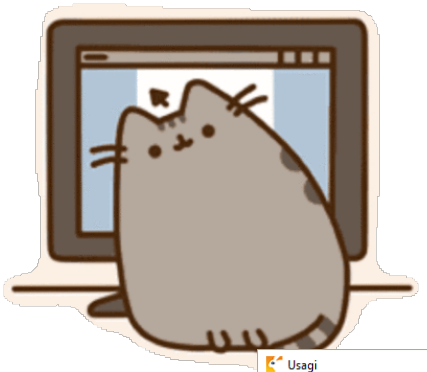
- You then load your generated maps into the empty Vocabulary table.



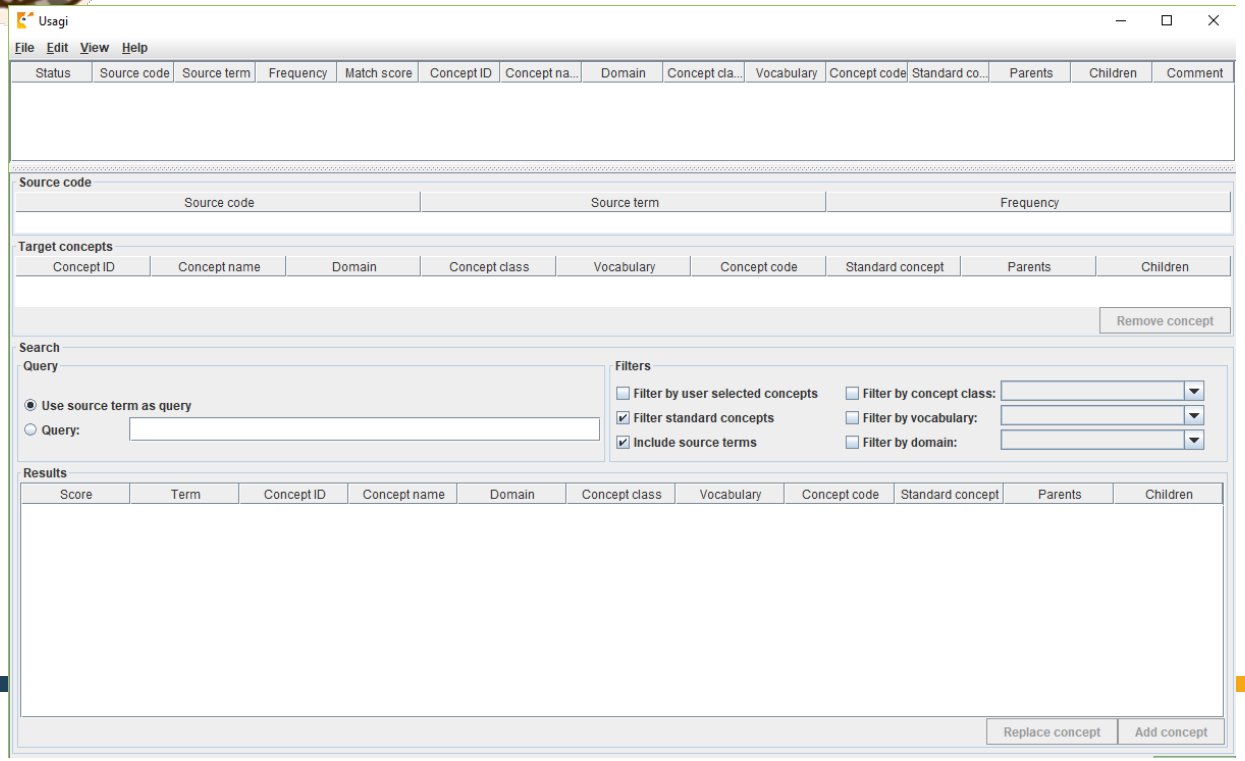
Usagi – Your Turn

1. ✓ Get a copy of the **Vocabulary** from ATHENA
2. ✓ Download **Usagi**
3. ✓ Have Usagi **build an index** on the Vocabulary
4. **Load your source codes** and let Usagi process them
5. **Review and update suggest mappings** with someone who has medical knowledge
6. **Export codes** into the SOURCE_TO_CONCEPT_MAP

Now Your Turn: Open Usagi



- Click on Usagi shortcut
- Go into the Usagi-1.1.6 folder
- Open Usagi.jar



Usagi – Your Turn

- We have provided a small subset of codes to try to map

https://github.com/OHDSI/Tutorial-ETL/

-> Materials -> Usagi -> DUTCH_ICPC_CONDITION_CODES_TO_MAP.xlsx

- These condition codes are in Dutch ICPC codes and need to be mapped to standard concepts

Usagi – Your Turn

- Your mission:
  - Download the codes to map
  - Translate codes to English
  - Import codes into Usagi
  - Map to standard concepts
  - Export SOURCE_TO_CONCEPT_MAP table

- For help review the User Guide:
  - http://www.ohdsi.org/web/wiki/doku.php?id=documentation:software:usagi

OHDSI
OBSERVATIONAL HEALTH DATA SCIENCES AND INFORMATICS

Data experts and CDM experts together design the ETL

People with medical knowledge create the code mappings

ETL Documentation

All are involved in quality control

A technical person implements the ETL

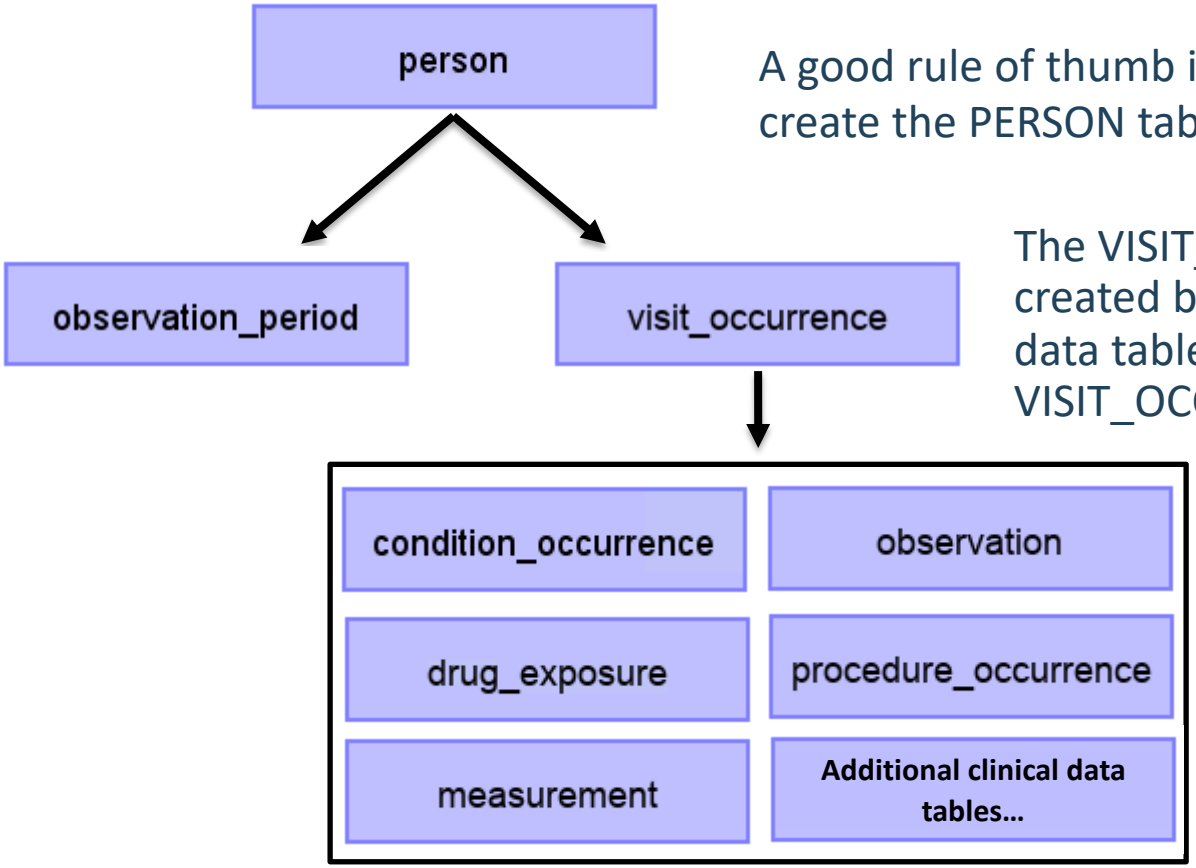
ETL Implementation

There are multiple tools available to implement your ETL

In this example we created a builder using SQL and R, though your choice will largely depend on the size and complexity of the ETL design

ETL Implementation

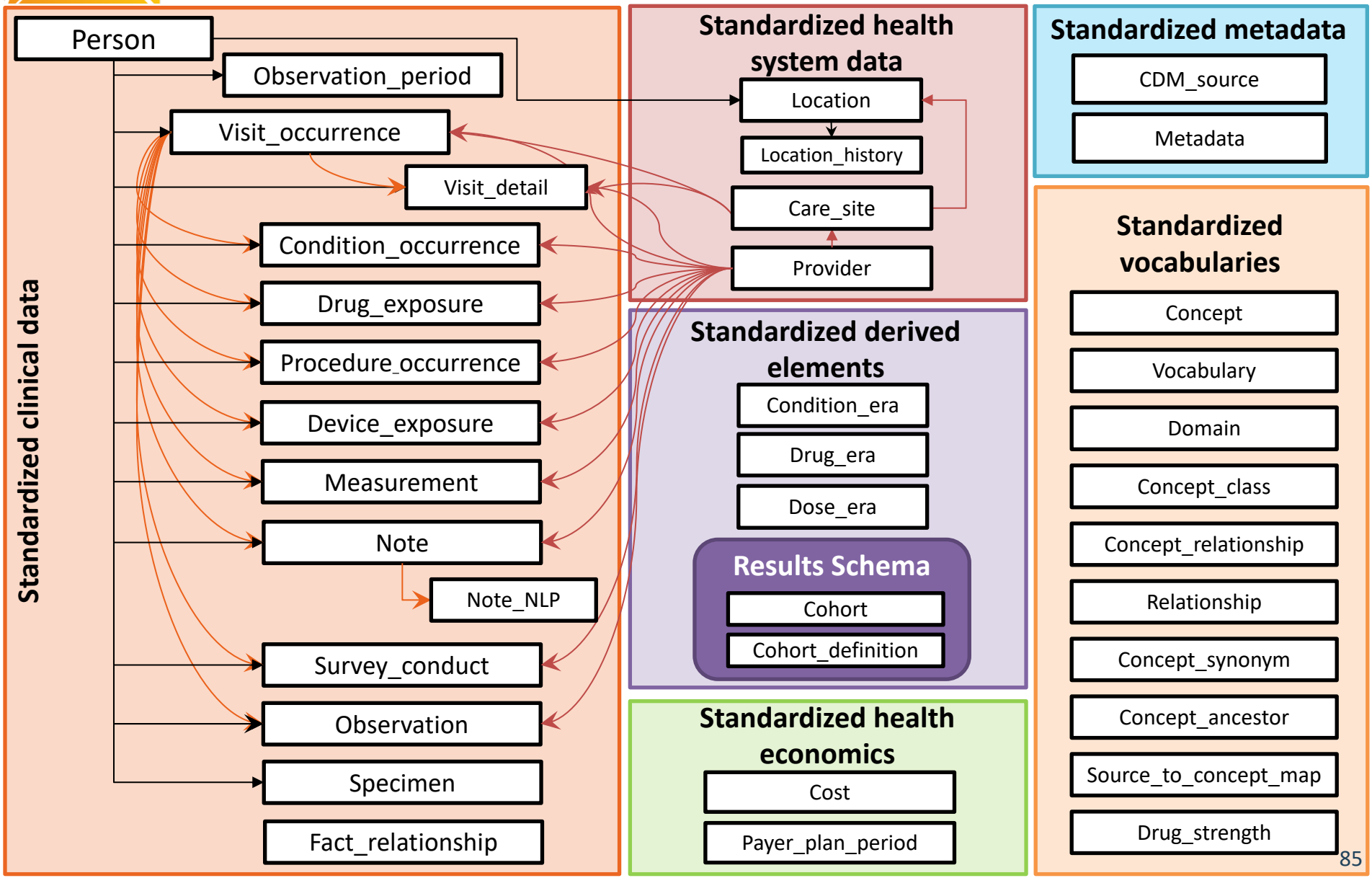
General Flow of Implementation



A good rule of thumb is to always create the PERSON table first

The VISIT_OCCURRENCE table must be created before the standardized clinical data tables as they all refer to the VISIT_OCCURRENCE_ID

CDM Version 6 Key Domains



ETL Implementation

Together

person

condition_occurrence

On your Own

observation_period

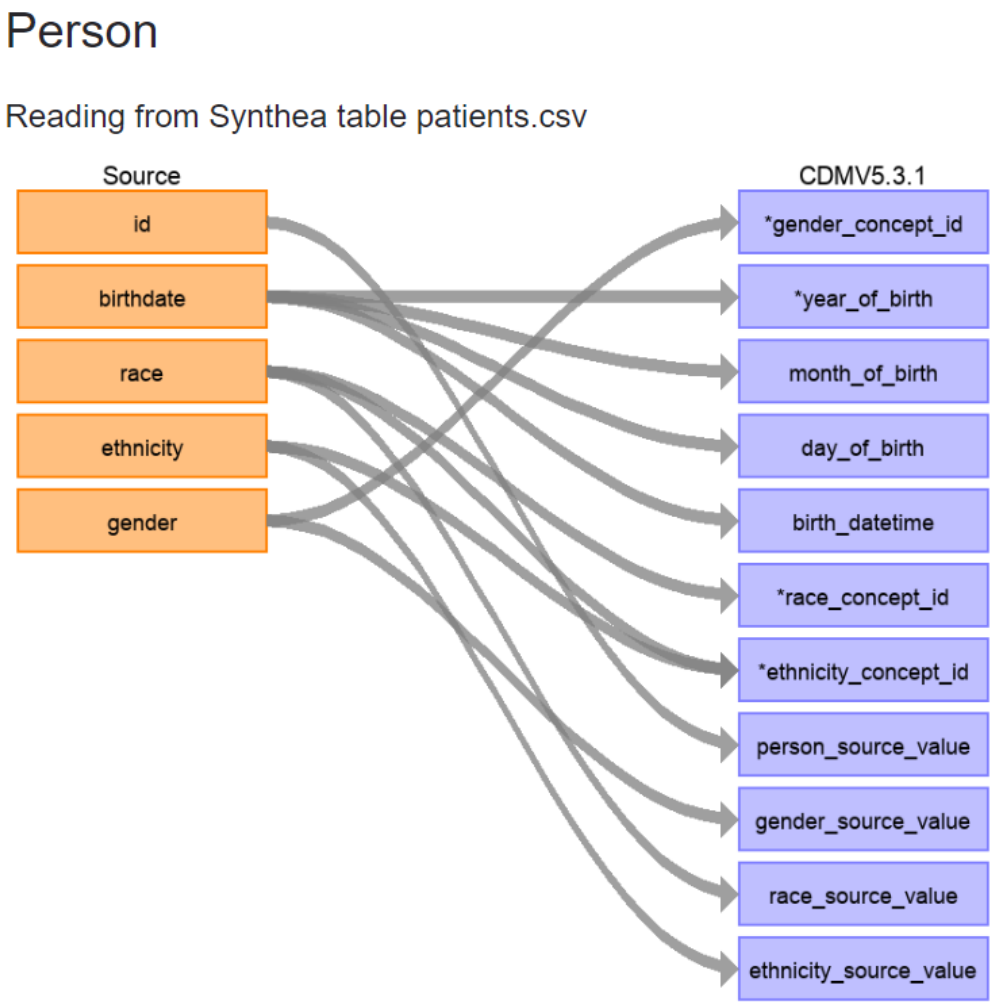
In this example we will not go over the VISIT_OCCURRENCE creation, though a link to how that was done will be provided later in the presentation.

ETL Implementation

person

First, let's review the logic we decided on for how the PERSON table should be created.

Navigate in your browser to: https://ohdsi.github.io/ETL-Synthea/Person.html



ETL Implementation

person

First, let's review the logic we decided on for how the PERSON table should be created.

Gender:

gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.

Birthdate:

year_of_birth	birthdate	Take year from birthdate	
month_of_birth	birthdate	Take month from birthdate	
day_of_birth	birthdate	Take day from birthdate	
birth_datetime	birthdate	With midnight as time 00:00:00	

Race:

race_concept_id	race	When race = 'WHITE' then set as 8527, when race = 'BLACK' then set as 8516, when race = 'ASIAN' then set as 8515, otherwise set as 0	

Ethnicity:

ethnicity_concept_id	race ethnicity	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0	

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

To open the query while we review:

https://github.com/OHDSI/Tutorial-ETL

Materials → Implementation→ Insert_Person_Lauren.sql

You can either view it directly in GitHub or download it and open it in pgAdmin4

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    row_number()over(order by p.id) as person_id,
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    date_part('year', p.birthdate) as year_of_birth,
    date_part('month', p.birthdate) as month_of_birth,
    date_part('day', p.birthdate) as day_of_birth,
    p.birthdate as birth_datetime,
    case upper(p.race)
        when 'WHITE' then 8527
        when 'BLACK' then 8516
        when 'ASIAN' then 8515
    else 0
    end as race_concept_id,
    case
        when upper(p.race) = 'HISPANIC'
        then 38003563 else 0
    end as ethnicity_concept_id,
    ...
```

ETL Implementation

person

Let's review the logic we decided on for how the PERSON table should be created.

Gender:	gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.
Birthdate:	year_of_birth	birthdate	Take year from birthdate	
	month_of_birth	birthdate	Take month from birthdate	
	day_of_birth	birthdate	Take day from birthdate	
	birth_datetime	birthdate	With midnight as time 00:00:00	
Race:	race_concept_id	race	When race = 'WHITE' then set as 8527, when race = 'BLACK' then set as 8516, when race = 'ASIAN' then set as 8515, otherwise set as 0	
Ethnicity:	ethnicity_concept_id	race ethnicity	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0	

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    date_part('year', p.birthdate) as year_of_birth,
        when 'ASIAN' then 8515
    else 0
    end as race_concept_id,
    case
        when upper(p.race) = 'HISPANIC'
        then 38003563 else 0
    end as ethnicity_concept_id,
    ...
```

Gender

gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    date_part('year', p.birthdate) as year_of_birth,
        when 'ASIAN' then 8515
    else 0
    end as race_concept_id,
    case
        when upper(p.race) = 'HISPANIC'
        then 38003563 else 0
    end as ethnicity_concept_id,
    ...
```

Gender

gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.

??

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
11      ...
12      end as gender_concept_id,
13      date_part('year', p.birthdate) as year_of_birth,
14      date_part('month', p.birthdate) as month_of_birth,
15      date_part('day', p.birthdate) as day_of_birth,
16      p.birthdate as birth_datetime,
17      case upper(p.race)
18          when 'WHITE' then 8527
19          when 'BLACK' then 8516
20          when 'ASIAN' then 8515
21      else 0
22      end as race_concept_id,
23      case
24          when upper(p.race) = 'HISPANIC'
25
26
27
28
29
30      p.id as person_source_value,
31      p.gender as gender_source_value,
32      0 as gender_source_concept_id,
33      p.race as race_source_value,
34      0 as race_source_concept_id,
35      p.ethnicity as ethnicity_source_value,
36
37  from raw_lauren.patients p
38  where p.gender is not null;
```

gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.

Gender

??

ETL Implementation

person

Let's review the logic we decided on for how the PERSON table should be created.

Gender:	gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.
Birthdate:	year_of_birth	birthdate	Take year from birthdate	
	month_of_birth	birthdate	Take month from birthdate	
	day_of_birth	birthdate	Take day from birthdate	
	birth_datetime	birthdate	With midnight as time 00:00:00	
Race:	race_concept_id	race	When race = 'WHITE' then set as 8527, when race = 'BLACK' then set as 8516, when race = 'ASIAN' then set as 8515, otherwise set as 0	
Ethnicity:	ethnicity_concept_id	race ethnicity	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0	

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    row_number()over(order by p.id) as person_id,
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    date_part('year', p.birthdate) as year_of_birth,
    date_part('month', p.birthdate) as month_of_birth,
    date_part('day', p.birthdate) as day_of_birth,
    p.birthdate as birth_datetime,
    case upper(p.race)
        then 38003563 else 0
    end as ethnicity_concept_id,
    ...
```

Birthdate

year_of_birth	birthdate	Take year from birthdate
month_of_birth	birthdate	Take month from birthdate
day_of_birth	birthdate	Take day from birthdate
birth_datetime	birthdate	With midnight as time 00:00:00

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    row_number()over(order by p.id) as person_id,
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    date_part('year', p.birthdate) as year_of_birth,
    date_part('month', p.birthdate) as month_of_birth,
    date_part('day', p.birthdate) as day_of_birth,
    p.birthdate as birth_datetime,
    case upper(p.race)
        then 38003563 else 0
    end as ethnicity_concept_id,
    ...
```

year_of_birth	birthdate	Take year from birthdate
month_of_birth	birthdate	Take month from birthdate
day_of_birth	birthdate	Take day from birthdate
birth_datetime	birthdate	With midnight as time 00:00:00

Birthdate

??

ETL Implementation

person

Let's review the logic we decided on for how the PERSON table should be created.

Gender:	gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.
Birthdate:	year_of_birth	birthdate	Take year from birthdate	
	month_of_birth	birthdate	Take month from birthdate	
	day_of_birth	birthdate	Take day from birthdate	
	birth_datetime	birthdate	With midnight as time 00:00:00	
Race:	race_concept_id	race	When race = 'WHITE' then set as 8527, when race = 'BLACK' then set as 8516, when race = 'ASIAN' then set as 8515, otherwise set as 0	
Ethnicity:	ethnicity_concept_id	race ethnicity	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0	

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    row_number()over(order by p.id) as person_id,
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    ...
    case upper(p.race)
        when 'WHITE' then 8527
        when 'BLACK' then 8516
        when 'ASIAN' then 8515
    else 0
    end as race_concept_id,
    case
        when upper(p.race) = 'HISPANIC'
        then 38003563 else 0
    end as ethnicity_concept_id,
    ...
```

race_concept_id	race	When race = 'WHITE' then set as 8527, when race = 'BLACK' then set as 8516, when race = 'ASIAN' then set as 8515, otherwise set as 0

Race

ETL Implementation

person

Let's review the logic we decided on for how the PERSON table should be created.

Gender:	gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.
Birthdate:	year_of_birth	birthdate	Take year from birthdate	
	month_of_birth	birthdate	Take month from birthdate	
	day_of_birth	birthdate	Take day from birthdate	
	birth_datetime	birthdate	With midnight as time 00:00:00	
Race:	race_concept_id	race	When race = 'WHITE' then set as 8527, when race = 'BLACK' then set as 8516, when race = 'ASIAN' then set as 8515, otherwise set as 0	
Ethnicity:	ethnicity_concept_id	race ethnicity	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0	

ETL Implementation

person

How should the PERSON table logic be implemented in SQL?

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    row_number()over(order by p.id) as person_id,
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    date_part('year', p.birthdate) as year_of_birth,
    date_part('month', p.birthdate) as month_of_birth,
    date_part('day', p.birthdate) as day_of_birth,
```

Ethnicity

ethnicity_concept_id	race ethnicity	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0

```
    case
        when upper(p.race) = 'HISPANIC'
        then 38003563 else 0
    end as ethnicity_concept_id,
```

??

ETL Implementation

person

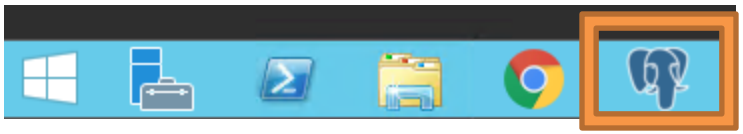
Now let's run the code and create the PERSON table in the cdm_lauren schema

1. Download the query from:

   https://github.com/OHDSI/Tutorial-ETL

   Materials → Implementation→ Insert_Person_Lauren.sql

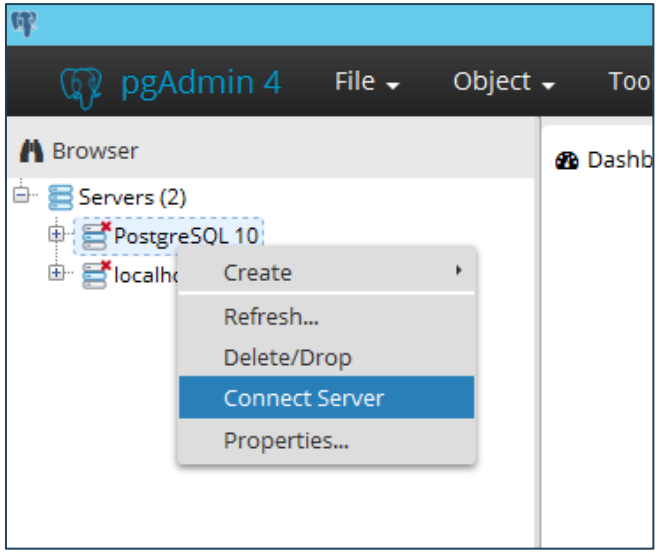
2. Open up pgAdmin4 using the icon on the task bar



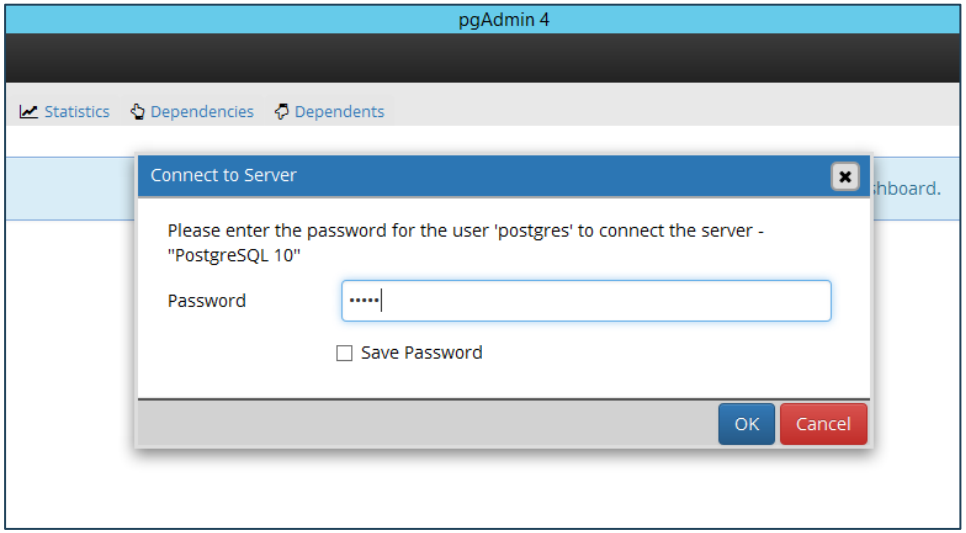
ETL Implementation

person

3. Expand the server list and right-click on PostgreSql 10 and choose Connect Server from the drop-down menu



4. When it asks for a password, type in ohdsi

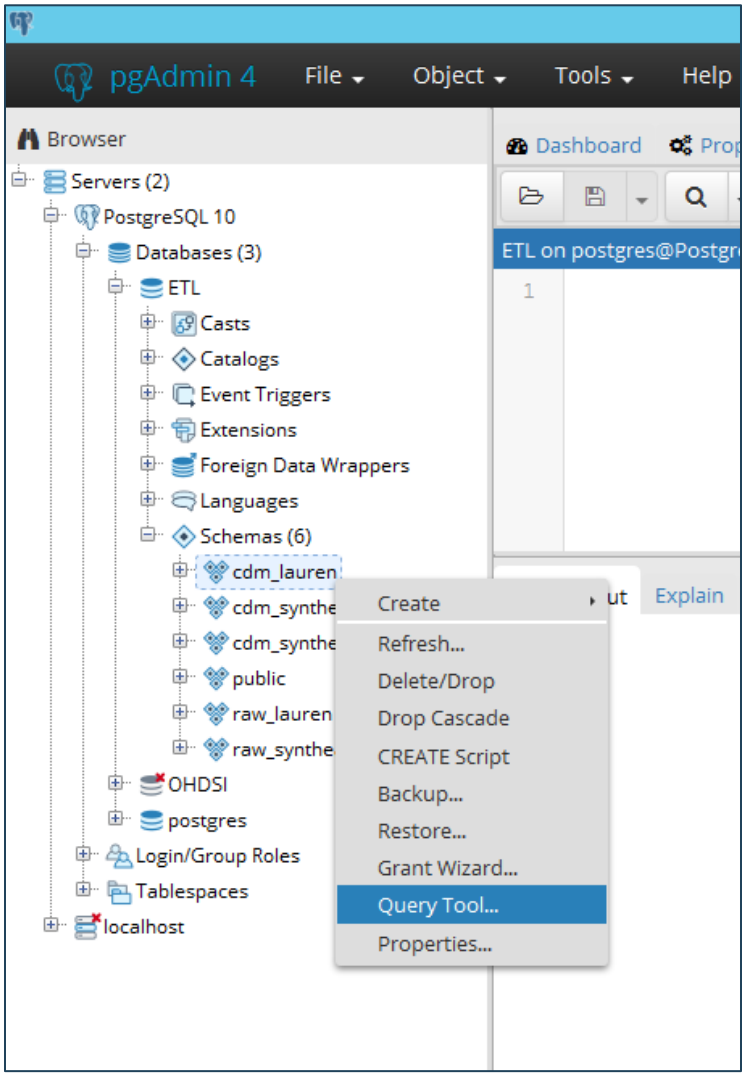


ETL Implementation

person

5. Expand:
   a. Databases
   b. ETL
   c. Schemas

6. Right click on cdm_lauren and choose Query Tool from the drop-down menu



ETL Implementation

person

7. Paste the sql code to create the PERSON table into the query window and press F5 or 

NOTE:

- The ‘truncate’ statement at the beginning deletes anything that is in the table already without deleting the table itself (helpful if you make a mistake)

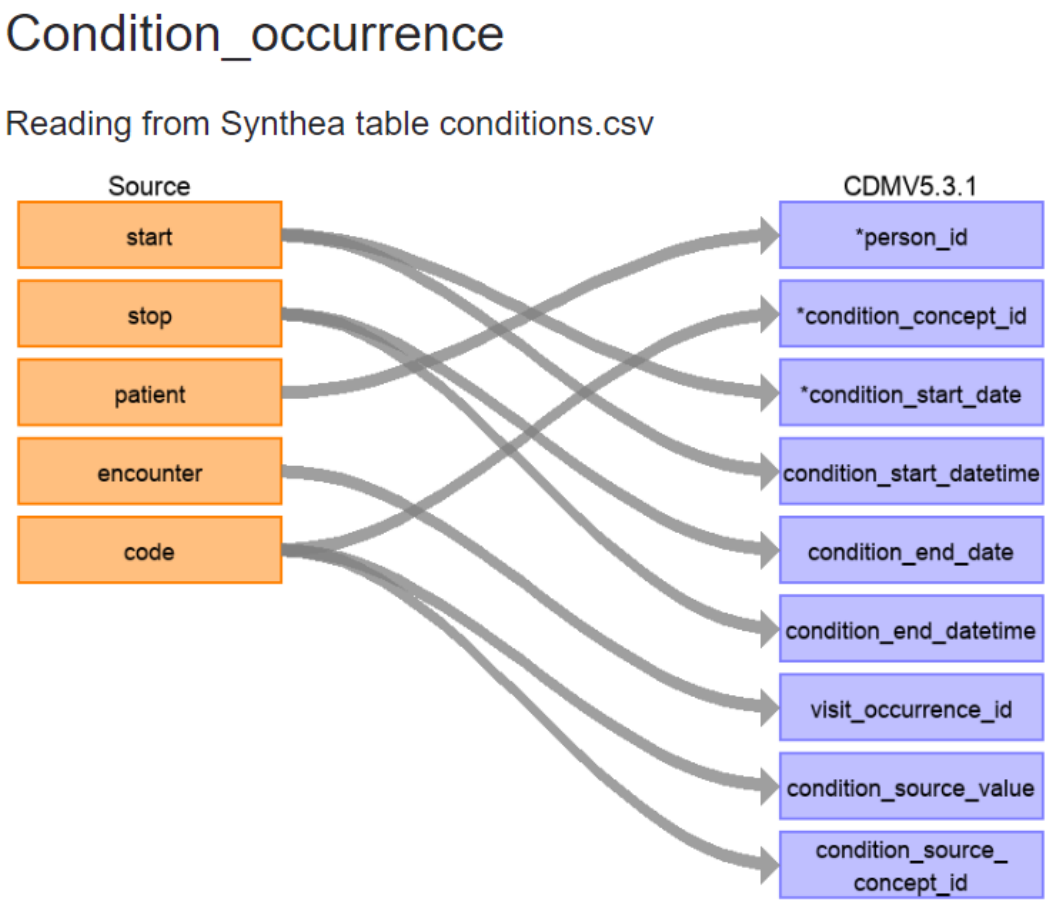
How would you check that your PERSON table was created?

ETL Implementation

condition_occurrence

First, let's review the logic we decided on for how the CONDITION_OCCURRENCE table should be created.

Navigate in your browser to: https://ohdsi.github.io/ETL-Synthea/Condition_occurrence.html



ETL Implementation

condition_occurrence

First, let's review the logic we decided on for how the CONDITION_OCCURRENCE table should be created.

Person:	person_id	patient	Map by mapping person.person_source_value to patient. Find person.person_id by mapping encouters.patient to person.person_source_value.	
Condition Concept:	condition_concept_id	code	Use code to lookup target_concept_id in CTE_TARGET_VOCAB_MAP: select ctvm.target_concept_id from conditions c join cte_target_vocab_map ctvm on ctvm.source_code = c.code and ctvm.target_domain_id = 'Condition' and ctvm.target_vocabulary_id = 'SNOMED'	
Condition Source Concept:	condition_source_concept_id	code	Use code to lookup source_concept_id in CTE_SOURCE_VOCAB_MAP: select csvm.source_concept_id from cte_source_vocab_map csvm join conditions c on csvm.source_code = c.code and csvm.source_vocabulary_id = 'SNOMED'	

ETL Implementation

condition_occurrence

How should the CONDITION_OCCURRENCE logic be implemented in SQL?

To open the query while we review:

https://github.com/OHDSI/Tutorial-ETL

Materials → Implementation→ Insert_Condition_Occurrence_Lauren.sql

You can either view it directly in GitHub or download it and open it in pgAdmin4

ETL Implementation

condition_occurrence

```
truncate cdm_lauren.condition_occurrence;

insert into cdm_lauren.condition_occurrence (
            condition_status_concept_id
)
select
        row_number()over(order by p.person_id) as condition_occurrence_id,
        p.person_id,
        case when srctostdvm.target_concept_id is null
             then 0
             else srctostdvm.target_concept_id
        end as target_concept_id,
        c.start as condition_start_date,
        c.start as condition_start_datetime,
        c.stop as condition_end_date,
        c.stop as condition_end_datetime,
        32020 as condition_type_concept_id,
        cast(null as varchar) as stop_reason,
        cast(null as integer) as provider_id,
        1 as visit_occurrence_id,
        0 as visit_detail_id,
        c.code as condition_source_value,
        (
            select case when source_concept_id
                             is null then 0 else source_concept_id end as source_concept_id
            from (
                                select srctosrcvm.source_concept_id
                               from cdm_synthea.source_to_source_vocab_map srctosrcvm
                               where srctosrcvm.source_code = c.code
                                 and srctosrcvm.source_vocabulary_id  = 'SNOMED'
                    ) a
        ) as condition_source_concept_id,
        NULL condition_status_source_value,
        0 as condition_status_concept_id
from raw_lauren.conditions c
left join cdm_synthea.source_to_standard_vocab_map srctostdvm
      on srctostdvm.source_code             = c.code
     and srctostdvm.target_domain_id        = 'Condition'
     and srctostdvm.target_vocabulary_id    = 'SNOMED'
     and srctostdvm.target_standard_concept = 'S'
     and srctostdvm.target_invalid_reason IS NULL
join cdm_lauren.person p
     on c.patient = p.person_source_value;
```

ETL Implementation

condition_occurrence

How should the CONDITION_OCCURRENCE logic be implemented in SQL?

```
21  select
22      row_number()over(order by p.person_id) as condition_occurrence_id,
23      p.person_id,
24      case when srctostdvm.target_concept_id is null
25          then 0
26          else srctostdvm.target_concept_id
27
50
51
52
53      and srctostdvm.target_domain_id      = 'Condition'
54      and srctostdvm.target_vocabulary_id  = 'SNOMED'
55      and srctostdvm.target_standard_concept = 'S'
56      and srctostdvm.target_invalid_reason IS NULL
57  join cdm_lauren.person p
58      on c.patient = p.person_source_value;
```

person_id	patient	Map by mapping person.person_source_value to patient. Find person.person_id by mapping encouters.patient to person.person_source_value.

ETL Implementation

condition_occurrence

How should the CONDITION_OCCURRENCE logic be implemented in SQL?

```
21  select
22          row_number()over(order by p.person_id) as condition_occurrence_id,
23          p.person_id,
24          case when srctostdvm.target_concept_id is null
25               then 0
26               else srctostdvm.target_concept_id
27          end as target_concept_id,
```

??

```
50  from raw_layer.conditions c
51  left join cdm_synthea.source_to_standard_vocab_map srctostdvm
52        on srctostdvm.source_code             = c.code
53       and srctostdvm.target_domain_id        = 'Condition'
54       and srctostdvm.target_vocabulary_id    = 'SNOMED'
55       and srctostdvm.target_standard_concept = 'S'
56       and srctostdvm.target_invalid_reason IS NULL
57  join cdm_layer.person p
58
```

condition_concept_id	code	Use code to lookup target_concept_id in CTE_TARGET_VOCAB_MAP: select ctvm.target_concept_id from conditions c join cte_target_vocab_map ctvm on ctvm.source_code = c.code and ctvm.target_domain_id = 'Condition' and ctvm.target_vocabulary_id = 'SNOMED'

ETL Implementation

condition_occurrence

How should the CONDITION_OCCURRENCE logic be implemented in SQL?

```
21  select
22          row_number()over(order by p.person_id) as condition_occurrence_id,
23          p.person_id,

37          c.code as condition_source_value,
38          (
39              select case when source_concept_id
40                          is null then 0 else source_concept_id end as source_concept_id
41              from (
42                              select srctosrcvm.source_concept_id
43                             from cdm_synthea.source_to_source_vocab_map srctosrcvm
44                            where srctosrcvm.source_code = c.code
45                              and srctosrcvm.source_vocabulary_id  = 'SNOMED'
46                      ) a
47          ) as condition_source_concept_id,
48          NULL condition_status_source_value
49
50  fro
```

condition_source_concept_id	code	Use code to lookup source_concept_id in CTE_SOURCE_VOCAB_MAP: select csvm.source_concept_id from cte_source_vocab_map csvm join conditions c on csvm.source_code = c.code and csvm.source_vocabulary_id = 'SNOMED'

ETL Implementation

condition_occurrence

Now let's run the code and create the CONDITION_OCCURRENCE table in the cdm_lauren schema

Download the query from:

https://github.com/OHDSI/Tutorial-ETL

Materials → Implementation→ Insert_Condition_Occurrence_Lauren.sql

NOTE: Make sure you have created the PERSON table in the cdm_lauren schema or this sql script will not work

ETL Implementation – Your Turn

observation_period

1. Review the OBSERVATION_PERIOD logic
   a. https://ohdsi.github.io/ETL-Synthea/Observation_period.html

2. Think through how that could be represented using SQL

Note: If you are not a SQL programmer don't worry! Feel free to use this time to explore the Achilles tool through the browser at http://localhost/achilles/#/Synthea/dashboard

ETL Implementation – Your Turn

observation_period

```sql
truncate table cdm_lauren.observation_period;

insert into cdm_lauren.observation_period (
observation_period_id,
person_id,
observation_period_start_date,
observation_period_end_date,
period_type_concept_id
)
select
        1 as observation_period_id,
        p.person_id,
        min(e.start) as observation_period_start_date,
        max(e.stop) as observation_period_end_date,
        44814724 as period_type_concept_id
from cdm_lauren.person p
join raw_lauren.encounters e
   on p.person_source_value = e.patient
group by p.person_id;
```

https://github.com/OHDSI/Tutorial-ETL

Materials → Implementation→ Insert_Observation_Period_Lauren.sql

ETL Implementation – Your Turn

observation_period

```
truncate table cdm_lauren.observation_period;

insert into cdm_lauren.observation_period (
observation_period_id,
person_id,
observation_period_start_date,
observation_period_end_date,
period_type_concept_id
)
select
        1 as observation_period_id,
        p.person_id,
        min(e.start) as observation_period_start_date,
        max(e.stop) as observation_period_end_date,
        [illegible] as period_type_concept_id
from cdm_lauren.person p
join raw_lauren.encounters e
    on p.person_source_value = e.patient
group by p.person_id;
```

https://github.com/OHDSI/Tutorial-ETL

Materials → Implementation→ Insert_Observation_Period_Lauren.sql

Resources

- The full Synthea builder can be found here:
  https://github.com/OHDSI/ETL-Synthea
- Another example of a R/SQL builder for a much larger database:
  https://github.com/OHDSI/ETL-HealthVerityBuilder
- A builder created using .NET:
  https://github.com/OHDSI/ETL-CDMBuilder
- A builder created using the AWS lambda functionality:
  https://github.com/OHDSI/ETL-lambdabuilder
  (in development)

Data experts and CDM experts together design the ETL

People with medical knowledge create the code mappings

ETL Documentation

ETL

All are involved in quality control

A technical person implements the ETL

Quality

What tools are available to check that the CDM logic was implemented correctly?

Achilles HEEL

Rabbit-in-a-Hat Test Case Framework

Achilles

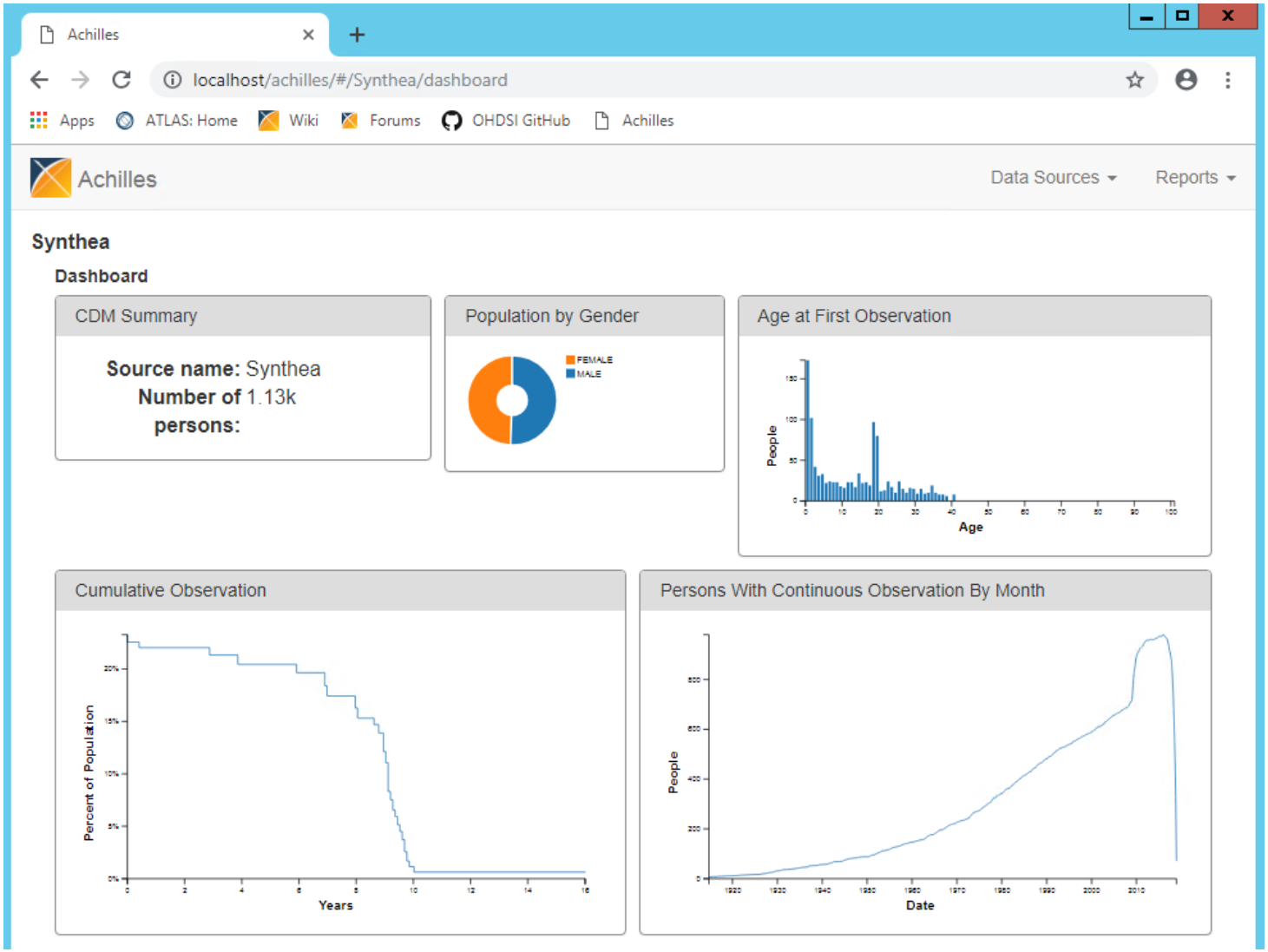
Achilles is a data characterization and quality tool available for download here:

https://github.com/OHDSI/Achilles

For an example of how it was run for our sample data, that R script is located here:

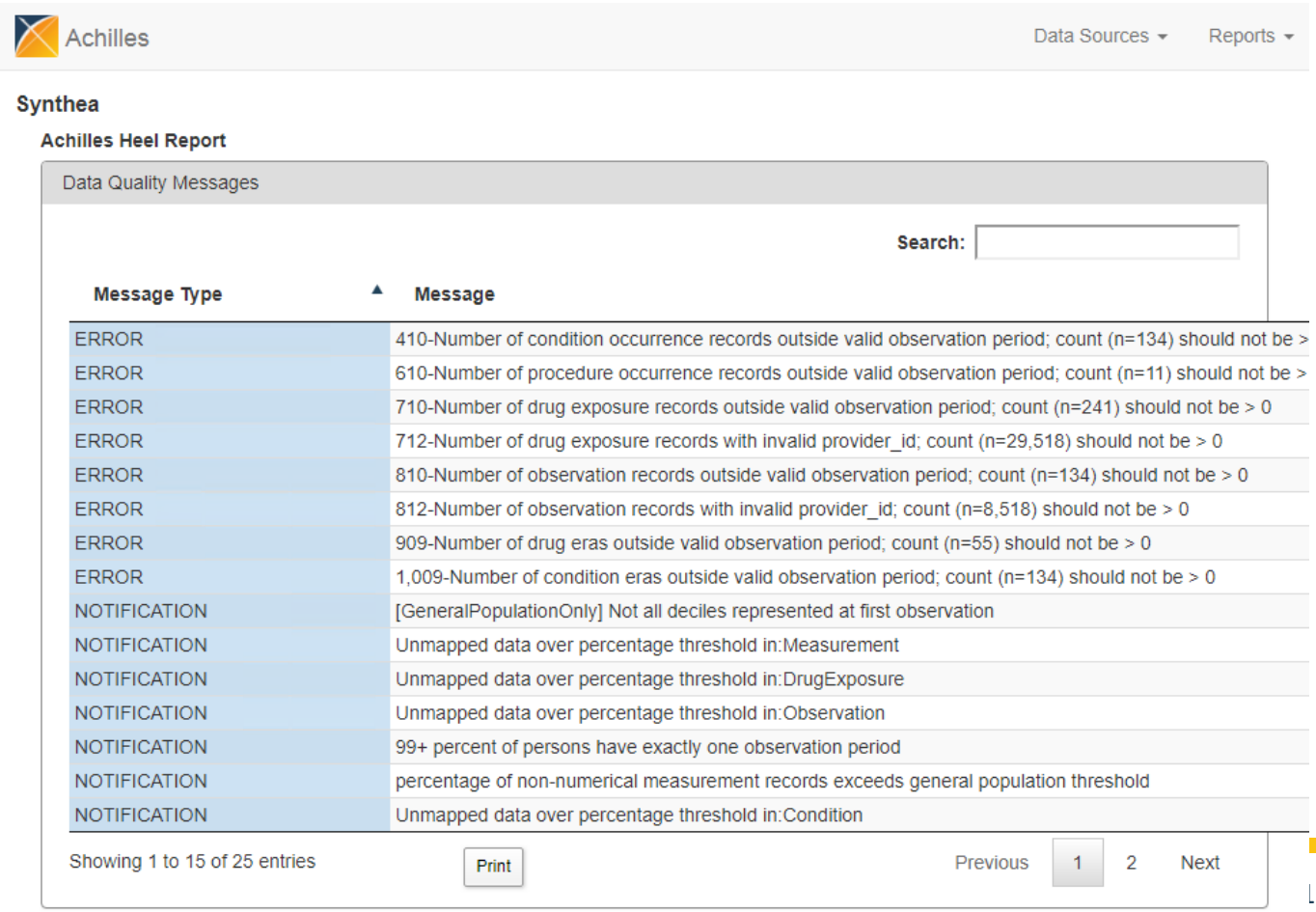
https://github.com/OHDSI/Tutorial-ETL/blob/master/materials/Achilles/achillesRun.R

Achilles


Achilles
localhost/achilles/#/Synthea/dashboard
Apps
ATLAS: Home
Wiki
Forums
OHDSI GitHub
Achilles
Achilles
Data Sources
Reports
Synthea
Dashboard
CDM Summary
Source name: Synthea
Number of persons: 1.13k
Population by Gender
FEMALE
MALE
Age at First Observation
People
Age
Cumulative Observation
Percent of Population
Years
Persons With Continuous Observation By Month
People
Date

Achilles Heel

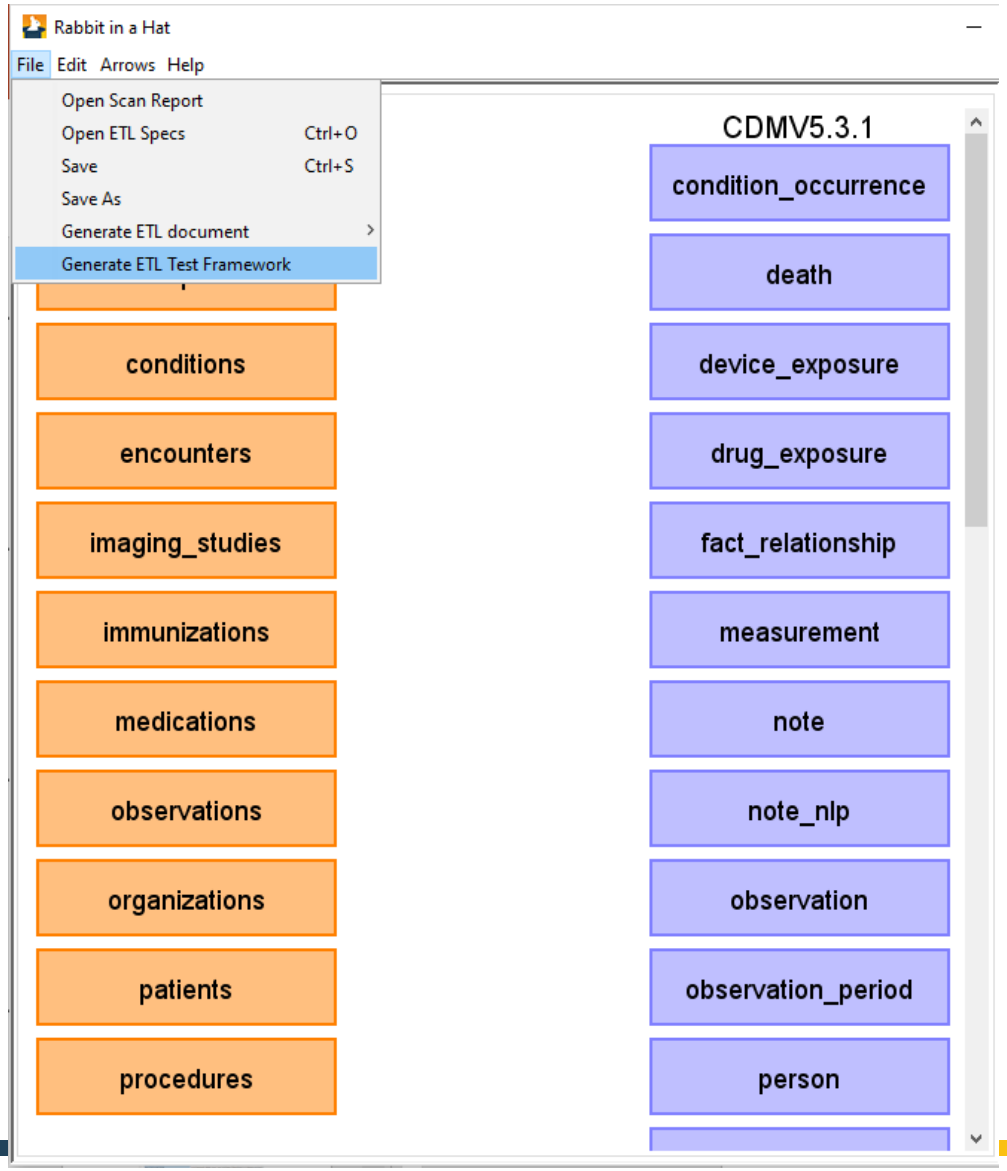
Achilles heel is a report generated by the Achilles application that will run a series of data quality checks on the CDM



Unit Test Cases

Rabbit-in-a-Hat

The application has a feature called **‘Generate ETL Test Framework’.** This feature allows you to create ‘fake’ people as a way to test your ETL logic.



Unit Test Cases

The test framework creates a series of R functions that enables you to specify your ‘fake’ people and records in the same structure as your source data using the scan report as a guide.

```
add_patients <- function(id, birthdate, deathdate, ssn, drivers, passport, prefix, first, last, suffix, maiden, marital,
state, zip) {
  defaults <- get('patients', envir = frameworkContext$defaultValues)
  fields <- c()
  values <- c()
  if (missing(id)) {
    id <- defaults$id
  }
  if (!is.null(id)) {
    fields <- c(fields, "id")
    values <- c(values, if (is.null(id)) "NULL" else if (is(id, "subQuery")) paste0("(", as.character(id), ")") else pas
  }

  if (missing(birthdate)) {
    birthdate <- defaults$birthdate
  }
  if (!is.null(birthdate)) {
    fields <- c(fields, "birthdate")
    values <- c(values, if (is.null(birthdate)) "NULL" else if (is(birthdate, "subQuery")) paste0("(", as.character(birt
    , "'"))
  }

  if (missing(deathdate)) {
    deathdate <- defaults$deathdate
  }
  if (!is.null(deathdate)) {
    fields <- c(fields, "deathdate")
    values <- c(values, if (is.null(deathdate)) "NULL" else if (is(deathdate, "subQuery")) paste0("(", as.character(deat
    , "'"))
  }
```

Unit Test Cases

An example of how this was done for the Synthea data is available from:

https://github.com/OHDSI/Tutorial-ETL/tree/master/materials/Unit%20Tests

The file that creates the test cases as a series of insert statement is **RunSyntheaTestCases.r**

Unit Test Cases

Let's revisit the PERSON table logic:

gender_concept_id	gender	When gender = 'M' then set gender_concept_id to 8507, when gender = 'F' then set to 8532	Drop any rows with missing/unknown gender.

How could we create a test case for this?

```
2  createPersonTests <- function () {
3
4      patient <- createPatient()
5      declareTest(id = patient$id, description = "Drop patients with no gender, id is PERSON_SOURCE_VALUE")
6      add_patients(id = patient$id, gender = NULL)
7      expect_no_person(person_source_value = patient$id)
```

```
11 -- 1: Drop patients with no gender, id is PERSON_SOURCE_VALUE
12 INSERT INTO synthea_test.[patients](id, birthdate, ssn, prefix, first, last, marital, race, ethnicity, birthplace, address, city,
   state, zip) VALUES ('1', '1926-02-23', '999-41-5589', 'Mr.', 'Benito209', 'Marks830', 'M', 'white', 'irish', 'Boston', '192
   MacGyver Dam', 'Boston', 'Massachusetts', '02108');
```

ETL Maintenance

Changed or Updated Raw Data?

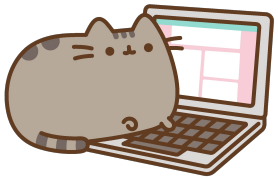
Bug Found?

New Vocab?

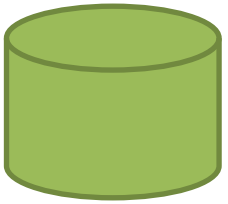
CDM Update?

ETL Documentation

All are involved in quality control



A technical person implements the ETL



Updated CDM

ETL Maintenance

Let's Revisit Ethnicity

```
truncate cdm_lauren.person;
insert into cdm_lauren.person (
            person_id,
            ...
            ethnicity_source_concept_id
)
select
    row_number()over(order by p.id) as person_id,
    case upper(p.gender)
        when 'M' then 8507
        when 'F' then 8532
    end as gender_concept_id,
    date_part('year', p.birthdate) as year_of_birth,
    date_part('month', p.birthdate) as month_of_birth,
    date_part('day', p.birthdate) as day_of_birth
```

ethnicity_concept_id	race ethnicity	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0

```
    case
        when upper(p.race) = 'HISPANIC'
        then 38003563 else 0
    end as ethnicity_concept_id,
```

??

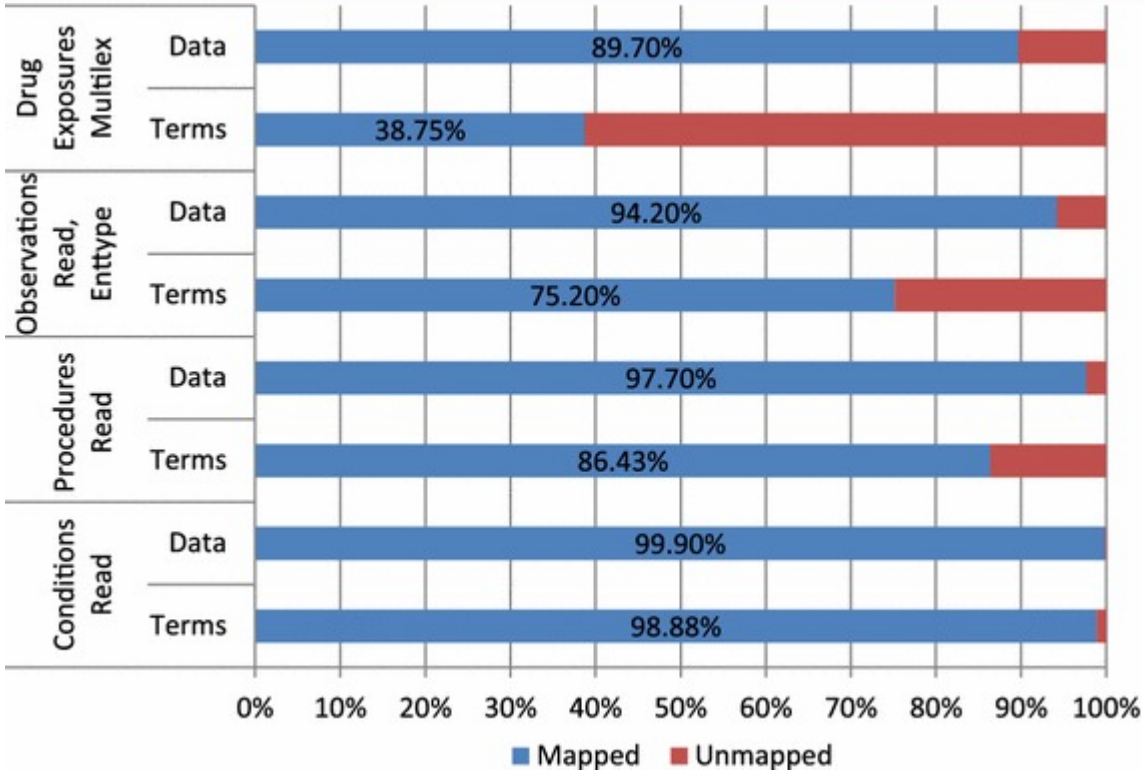
OHDSI
OBSERVATIONAL HEALTH DATA SCIENCES AND INFORMATICS

Final Hard Lessons Learned

80/20 Rule

Drug Safety
November 2014, Volume 37, Issue 11, pp 945–959 | Cite as

Fidelity Assessment of a Clinical Practice Research Datalink Conversion to the OMOP Common Data Model



You don't need to map all terms to get good data coverage!

Comfort with Data Loss

- If there is data that is not of research quality or there are methods to adjust, use the ETL to standardize that

Example Patient Drop Counts from a CDM Build	
Reason to Drop Someone	**Person Count**
Unknown gender	23,592
Implausible year of birth - past	749
Implausible year of birth - post earliest observation period	3,836
Gender changes	2

Thank you!

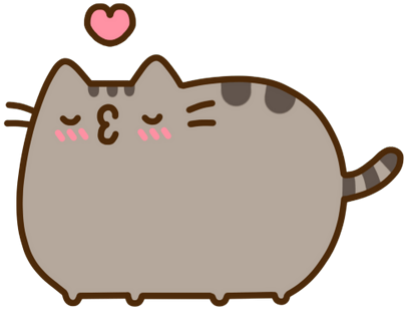
This tutorial would not have been possible without the contribution of many collaborators in the OHDSI Community

We like to thank Amazon Web Services for their valuable technical support and resources

Acknowledgements

Anthony Molinaro who wrote the Synthea CDM Builder

James Wiggins who helps us prepare an AWS instance for use today



Pusheen the Cat

http://pusheen.com/

Second Annual

EUROPEAN OHDSI SYMPOSIUM

March 29th 2019

Tutorials 30th and 31st

The Journey from Data to Evidence

Erasmus MC Rotterdam The Netherlands

www.ohdsi-europe.org