



🗨 DATABASE & METHODS CYBERSEMINAR SERIES

FY24 Session 9:

VHA Corporate Data Warehouse (CDW):
Using CDW Data to Conduct a Research Study for
SQL Beginners

June 3, 2024

Hosted by 

Anne E. Hines, PhD

Sr. Technical Analyst, VIReC



DATABASE & METHODS CYBERSEMINAR SERIES

Informational seminars to help VA researchers access and use VA databases.

Sessions cover...

- VA data sources & data access systems
- Application of VA data to research and quality improvement questions
- Limitations of secondary data use
- Resources to support VA data use

UPCOMING DATABASE & METHODS SESSIONS

First Monday of the month | 1:00pm-2:00pm ET

Date	Topic
7/1/24	Research Applications for JLV (Joint Longitudinal Viewer)
7/15/24	Text searching pre-configured widgets in JLV (Joint Longitudinal Viewer)
9/9/24	Automated Reporting of Large Database Research Methods for Studying Treatment Effects and Side Effects

Visit the [VIReC Database & Methods Cyberseminar](#) page for more information & registration links.

Visit [HSR's VIReC Cyberseminar Archive](#) page to watch previous sessions.

Where can I
download a
copy of the
slides?



SAMPLE EMAIL

A Practical Approach to Working with VA-Purchased Community Care Data

Thursday, October 13, 2022

2:00 PM | (UTC-04:00) Eastern Time (US & Canada) | 1 hr

Please download today's slides

~~Please click here for today's live captions~~

Join webinar

More ways to join:

Join from the webinar link

<https://veteransaffairs.webex.com/veteransaffairs/j.php?>

DATABASE & METHODS CYBERSEMINAR SERIES

FY24 Session 9:

VHA Corporate Data Warehouse (CDW): Using CDW Data to Conduct a Research Study for SQL Beginners

June 3, 2024

Hosted by 

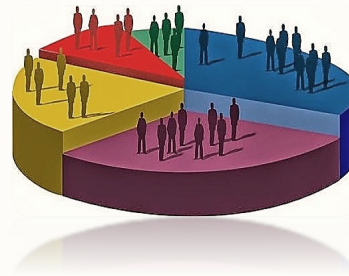
Anne E. Hines, PhD

Sr. Technical Analyst, VIReC

Poll #1:

*What is your primary **role** in projects using VA data?*

- Investigator, PI, Co-I
- Statistician, methodologist, biostatistician
- Data manager, analyst, or programmer
- Project coordinator
- Other – please describe via the chat function



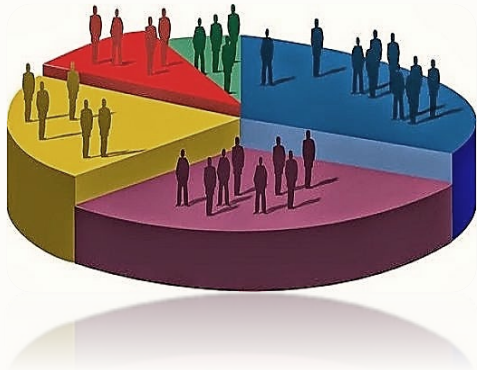
Poll #2:

How many years of experience working with VA data?

- None – I'm brand new to this!
- One year or less
- More than 1, less than 3 years
- At least 3, less than 7 years
- At least 7, less than 10 years
- 10 years or more

Poll #3:

*Have you ever pulled data **yourself** from the Corporate Data Warehouse (CDW)?*



- Yes
- No

Objective

Describe how to pull CDW data for a small study

Applicable to Research or
Operations

Slight difference in table naming

Not covering data from
Oracle/Cerner Millennium

VA's newest electronic health record (EHR), in
use at 6 VA medical centers

Session roadmap

- **Review of important CDW concepts**
- Describe aims of example study
- Identify procedures associated with the study aim
- Determine CDW tables of interest
- Prepare SQL code to pull the study data

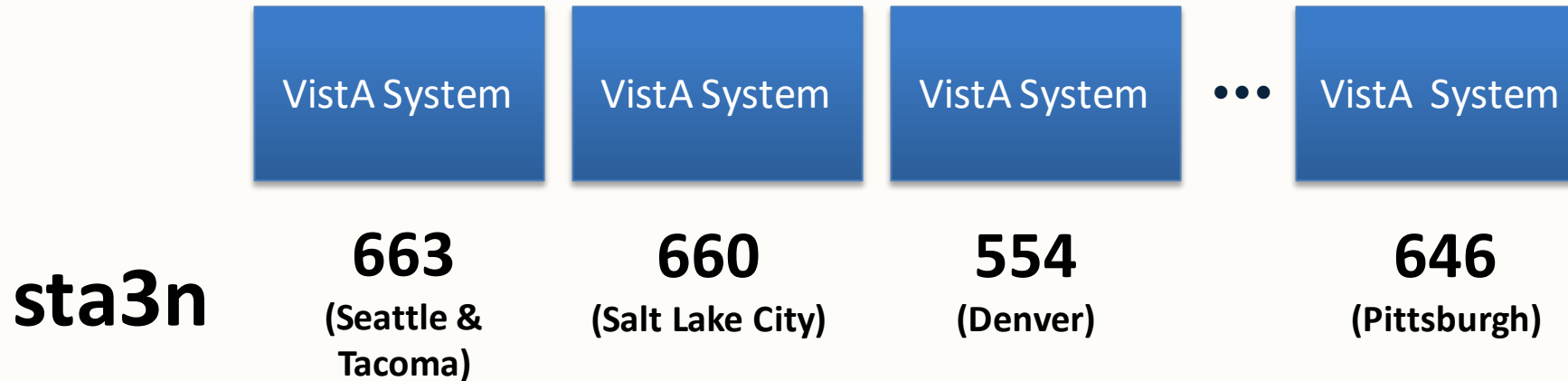


Basics Concepts of the CDW



Veterans Health Administration (VHA) Has 130 VistA Systems

Each VistA system is represented by a 3-digit number (sta3n)



More than one
medical center
can be in a sta3n!

Identifying fields in CDW Tables

Every row in a CDW table has...

- **PatientSID** (patient identifier)
- **Sta3n** (tells you which VistA system the record came from)

VisitSID	PatientSID	sta3n	VisitDateTime	InstitutionSID	PrimaryStop CodeSID
1034566	10111223	554	2/2/2020 14:22	1039581211	98734
1320002	10034555	646	3/3/2003 15:33	1197622000	34567



Patient Identifiers

Patient A



VistA System
1

**PatientSID
10055888**

VistA System
2

**PatientSID
10777666**

...

VistA System
n

**PatientSID
1229999**

*Note: only fictitious patient identifiers used in this presentation

Patient Identifiers

Patient A



VistA System
1

**PatientSID
10055888**

VistA System
2

**PatientSID
10777666**

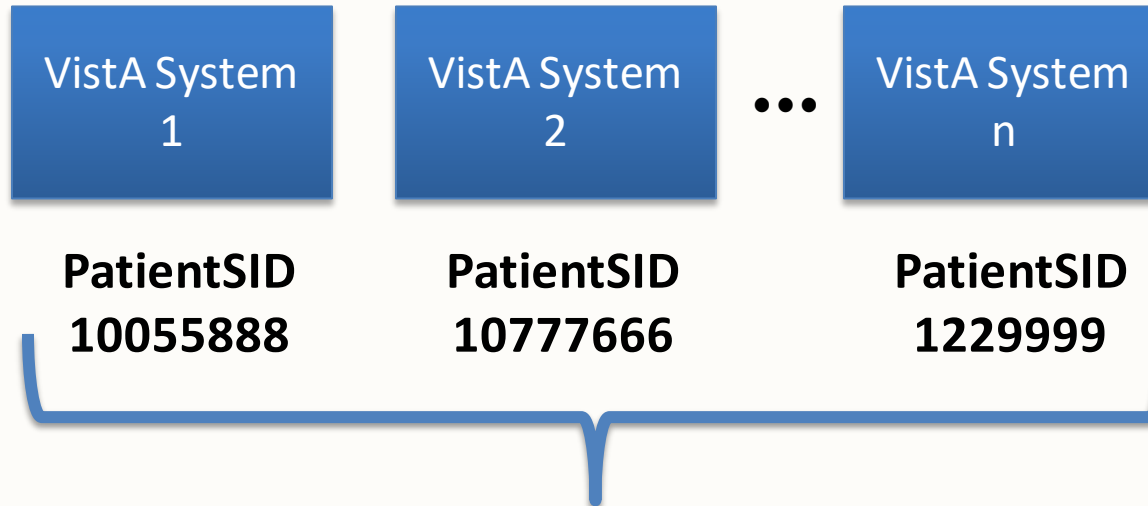
...

VistA System
n

**PatientSID
1229999**

Patient Identifiers

Patient A



Patient A has 1-n PatientSIDs

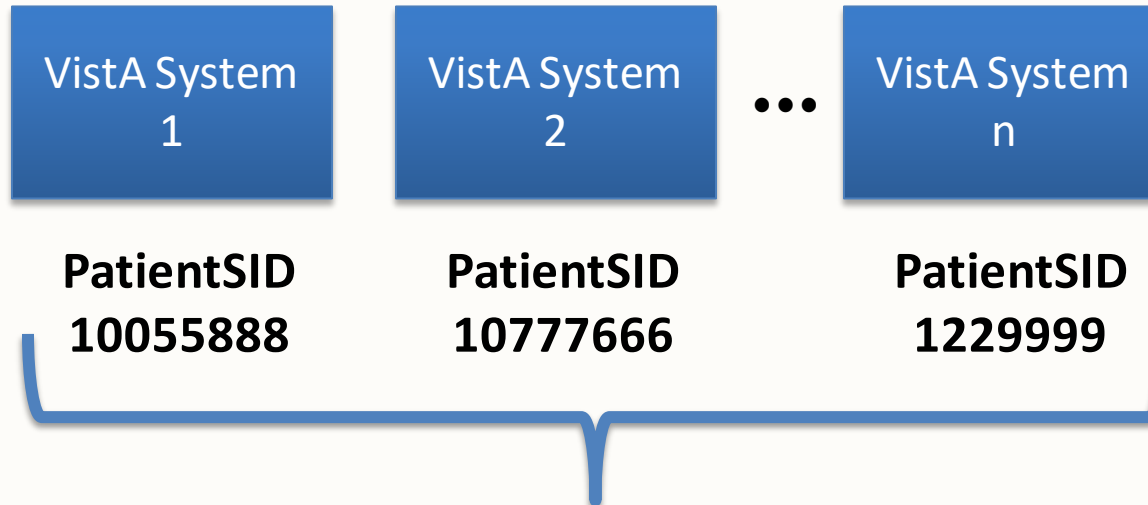
You need *all* of these to completely define all records belonging to Patient A

Patient Identifiers

Patient A



- PatientSSN
- PatientICN



Patient A has 1-n PatientSIDs

You need *all* of these to completely define all records belonging to Patient A

Patient Identifier Pitfalls: SSN



- Diabetes
- Hysterectomy



Patient A

- Hypertension
- Mitral Valve Stenosis



Patient B

PatientSSN

SSN 123-45-6789

Patient Identifier Pitfalls: SSN



- Diabetes
- Hysterectomy



Patient A PatientSSN 123-45-6789

- Hypertension
- Mitral Valve Stenosis



Patient B (*real* PatientSSN is 123-54-6789)

Patient Identifier Pitfalls: ICN



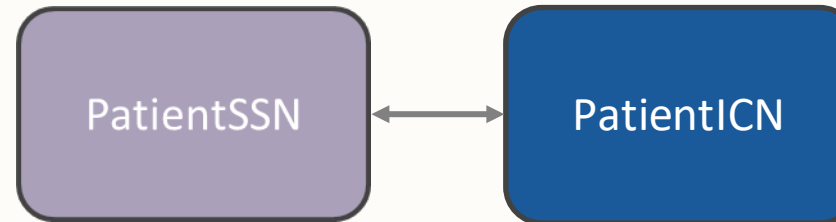
Name	DOB	Gender	Site	ICN
Louise Huston	1/1/1950	F	Denver	12121342
Louise Houston	1/1/1950	F	St. Louis	10072492
Louise Houston Green	1/1/1950	F	Hartford	11000637

One person, 3 different ICNs

Patient Identifier Pitfalls: SSN vs. ICN



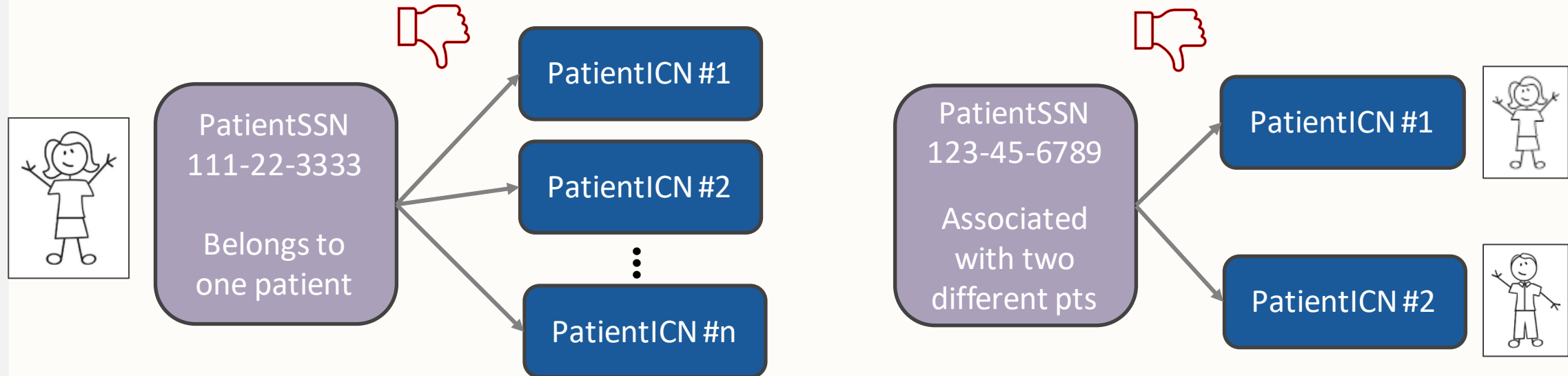
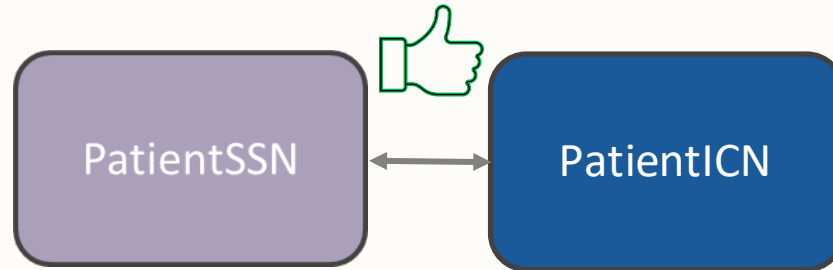
**IDEALLY:
1 – to – 1
correspondence**



Patient Identifier Pitfalls: SSN vs. ICN



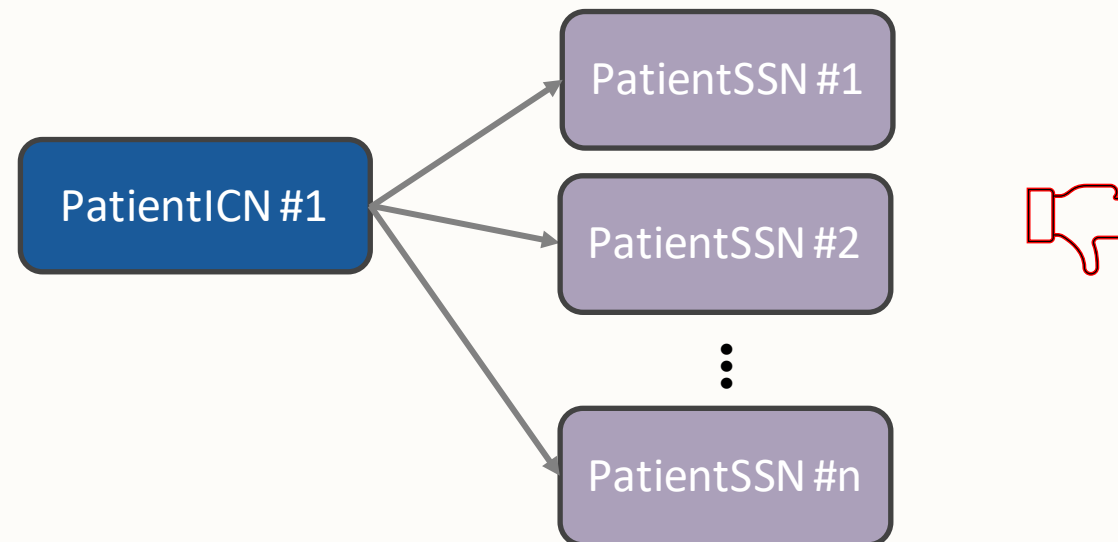
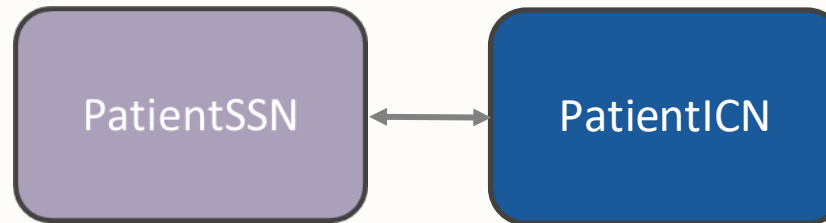
**IDEALLY:
1 – to – 1
correspondence**



Patient Identifier Pitfalls: SSN vs. ICN



**IDEALLY:
1 – to – 1
correspondence**



Patient Identifiers: SSN or ICN?

There are no set rules, but this is how I personally decide how to choose between using PatientSSN or PatientICN:

Use PatientSSN...

...if a cohort of enrolled patients is known.

- Verify if any enrollee has a PatientSSN that seems to be attached to more than one person.
- If so, exclude PatientSID(s) associated with the erroneous person from the finder file.
 - **This may require chart review.**

Use PatientICN...

...if my cohort is defined by patients who have a particular diagnosis, or have had certain types of care or admissions.

- Can still exclude any PatientSSN that maps to multiple PatientICNs, and any PatientICN that maps to multiple PatientSSNs.
 - ***Your VINCI analyst must do this for research projects – you cannot do this yourself.***


*Finder file: list of patient identifiers used to pull data from other tables

Session roadmap

- Review of important CDW concepts
- **Describe aims of example study**
- Identify procedures associated with the study aim
- Determine CDW tables of interest
- Prepare SQL code to pull the study data



Example Study Questions



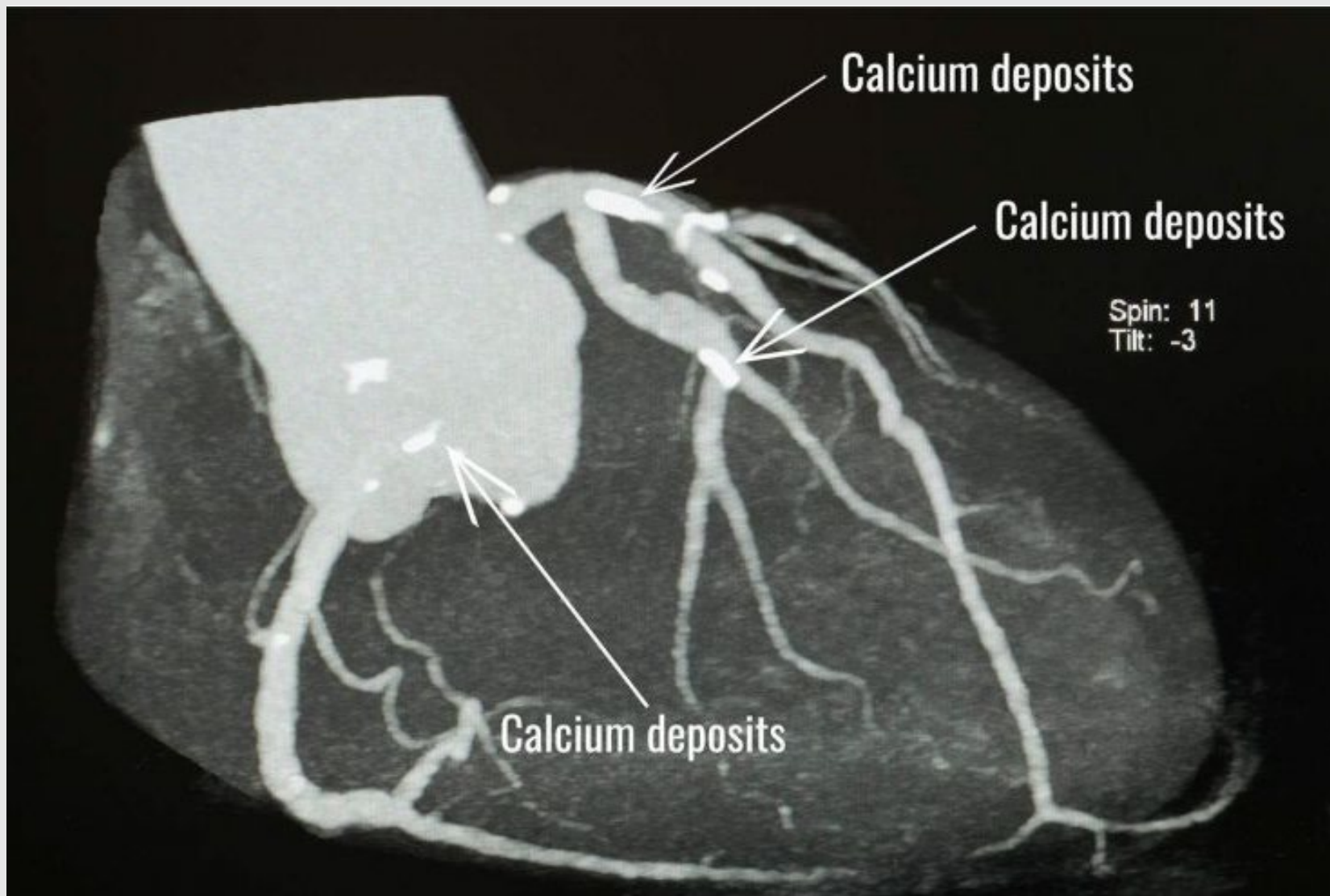
Investigate VHA use of a specific imaging test



Q: Which VA sites perform this test?



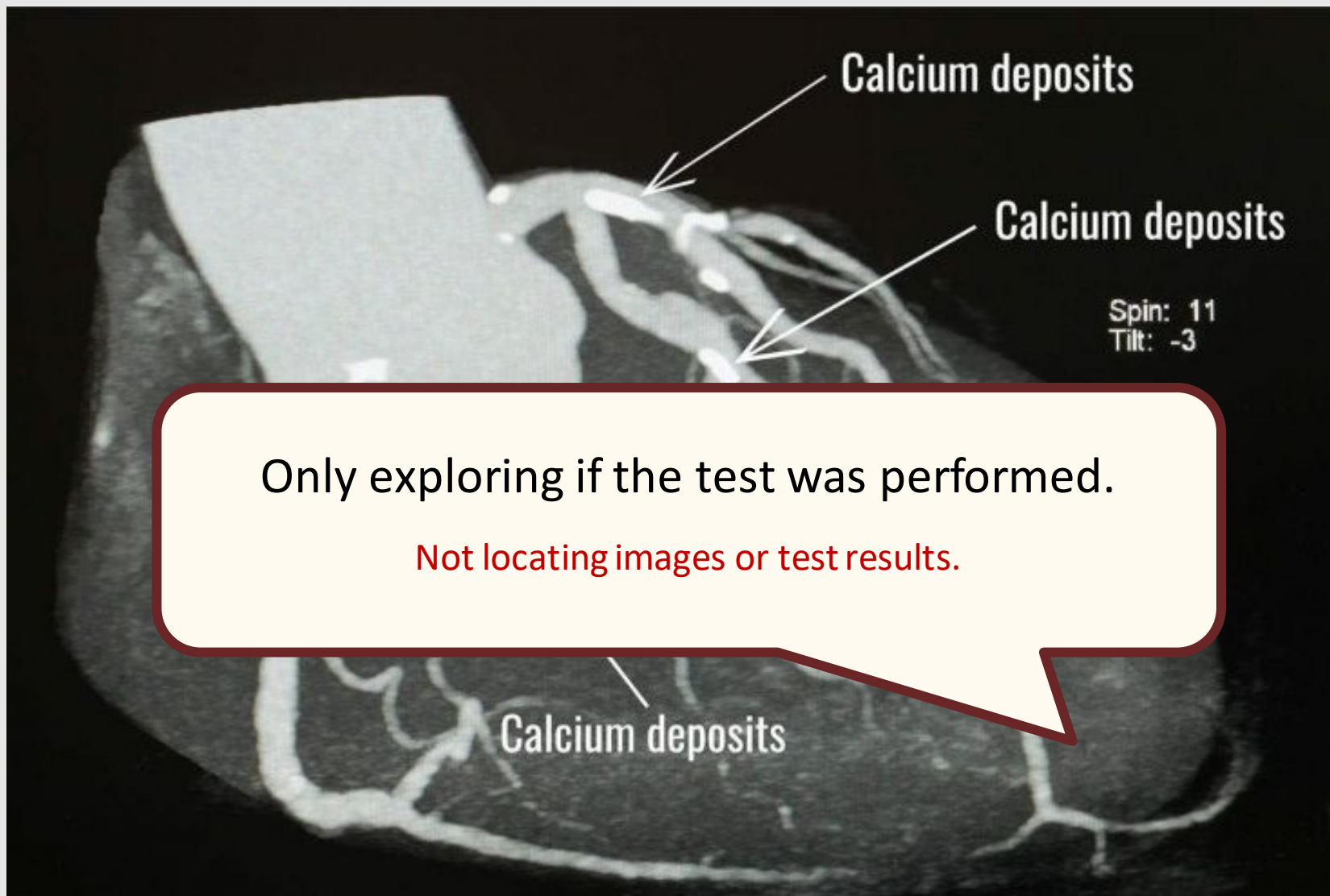
Q: How has use of this test changed over time?



Procedure

Cardiac Computed Tomography (CT) for Calcium Scoring

Imaging technology that is less invasive than a traditional angiogram & provides score for calcium burden



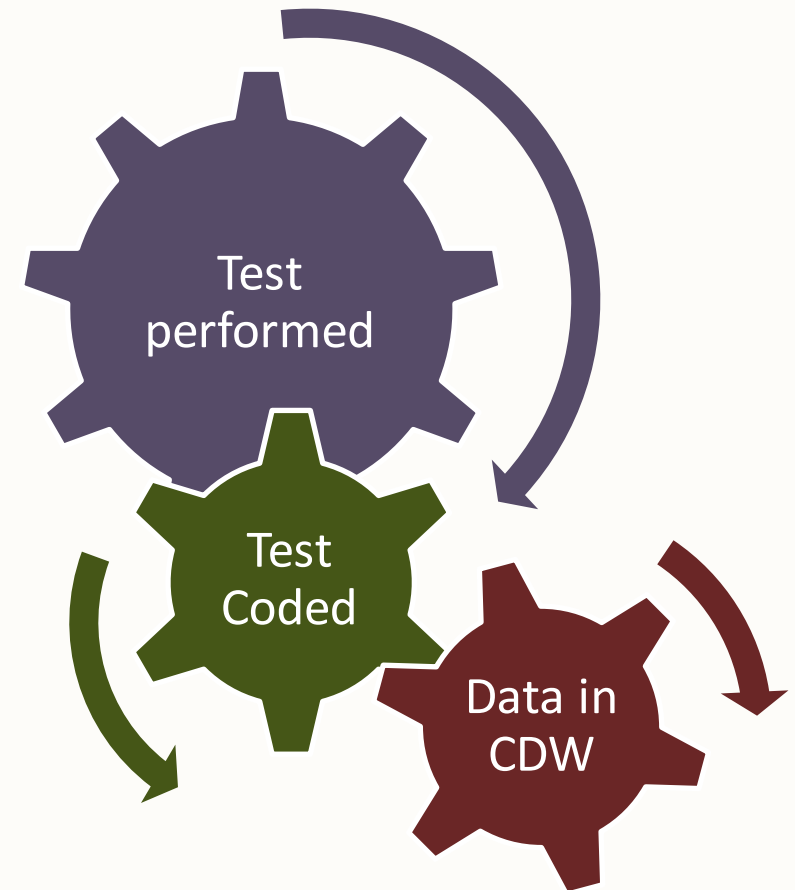
Procedure

Cardiac Computed Tomography (CT) for Calcium Scoring

Imaging technology that is less invasive than a traditional angiogram & provides score for calcium burden

How Do You Measure Use of Cardiac CT for Calcium Scoring?

- Radiology department performs procedure
 - clinician interprets & writes note
- Procedure is assigned a Common Procedural Terminology (CPT) code
 - by clinician or professional coder or is linked to the test ordered
- Data is extracted from EHR to CDW
 - CDW updated nightly



Session roadmap

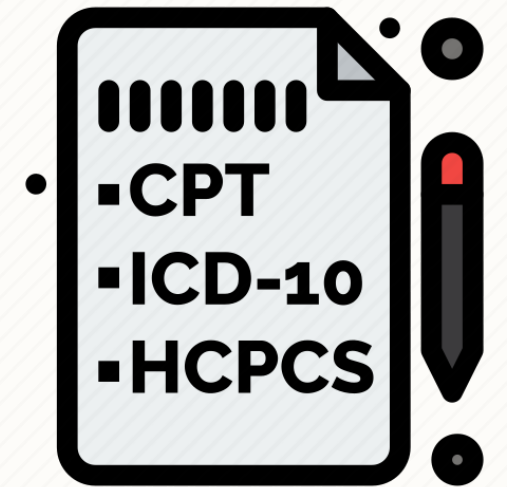
- Review of important CDW concepts
- Describe aims of example study
- **Identify procedures associated with the study aim**
- Determine CDW tables of interest
- Prepare SQL code to pull the study data



Procedure Codes: CPT

CPT Codes

- Developed by the American Medical Association
- Procedure Billing Codes
 - Also used to bill for health care services such as a MD office visit or a PT visit
- Most are 5 numeric digits (e.g., 11712)
- Updated Annually



Other procedure codes: International Classification of Disease (ICD) & Healthcare Common Procedure Codes (HCPCS)



CPT Codes: Cardiac CT for Calcium Scoring

- CPT codes change over time
- What **timeframe** does your study cover?
 - ➔ 1/1/2006 - present

CPT Code	Procedure	Inactive Date
0144T	Coronary Calcium	1/1/2010
0147T	CTA + Coronary Calcium	1/1/2010
0149T	CTA + morphology + Coronary Calcium	1/1/2010
75571	Coronary Calcium	n/a

CTA = CT angiography (visualization of the coronary vessels using CT)



Session roadmap

- Review of important CDW concepts
- Describe aims of example study
- Identify procedures associated with the study aim
- **Determine CDW tables of interest**
- Prepare SQL code to pull the study data



Types of CDW Tables

Fact tables

patient level data

Dimension tables

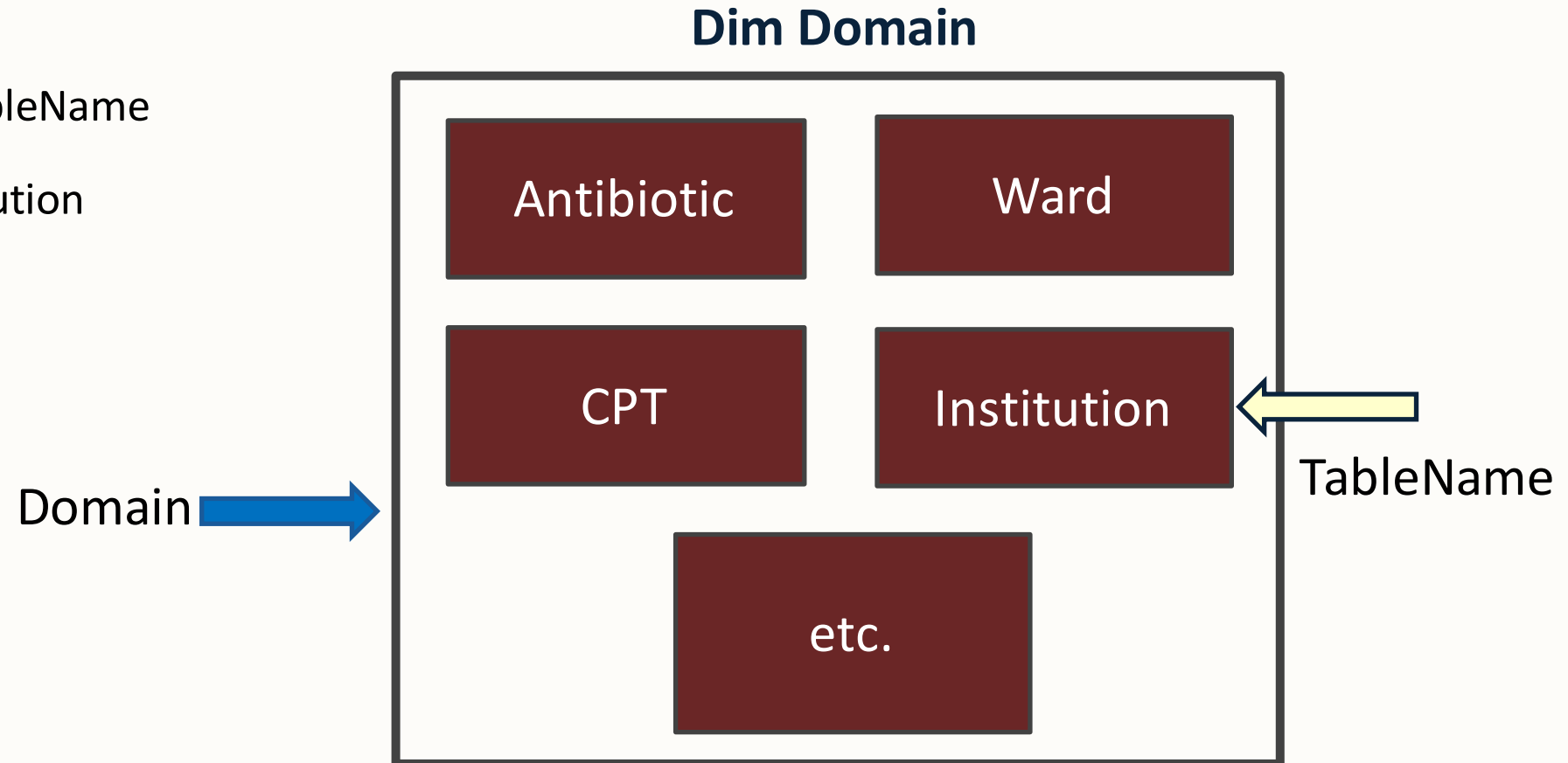
lists (no patient level data*)



Primary Key – field that uniquely identifies each row in a table

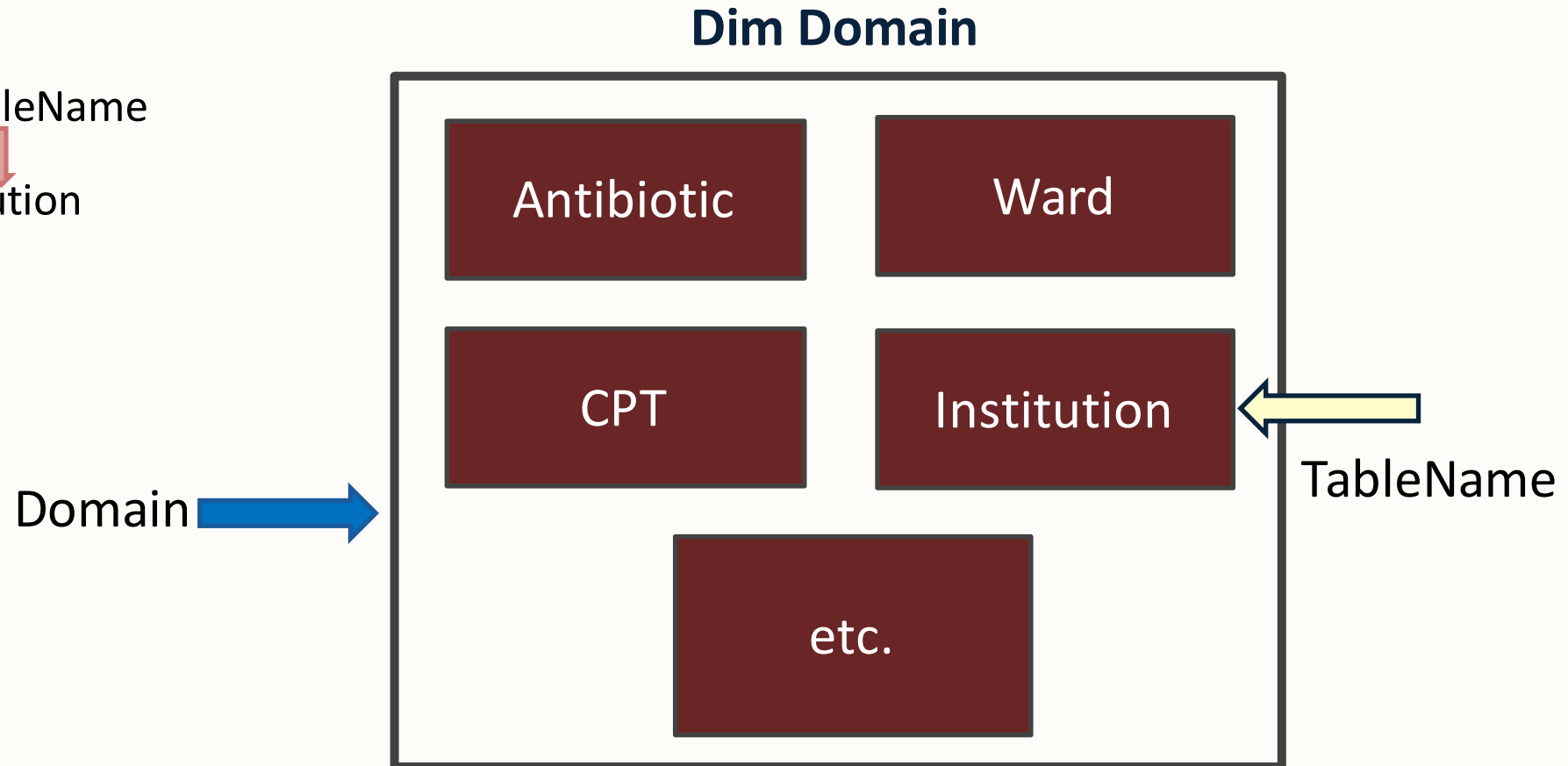
CDW Table Naming Conventions:

- Database.Schema.TableName
- CDWWork.Dim.Institution



CDW Table Naming Conventions:

- Database.Schema.TableName
- CDWWork.Dim.Institution

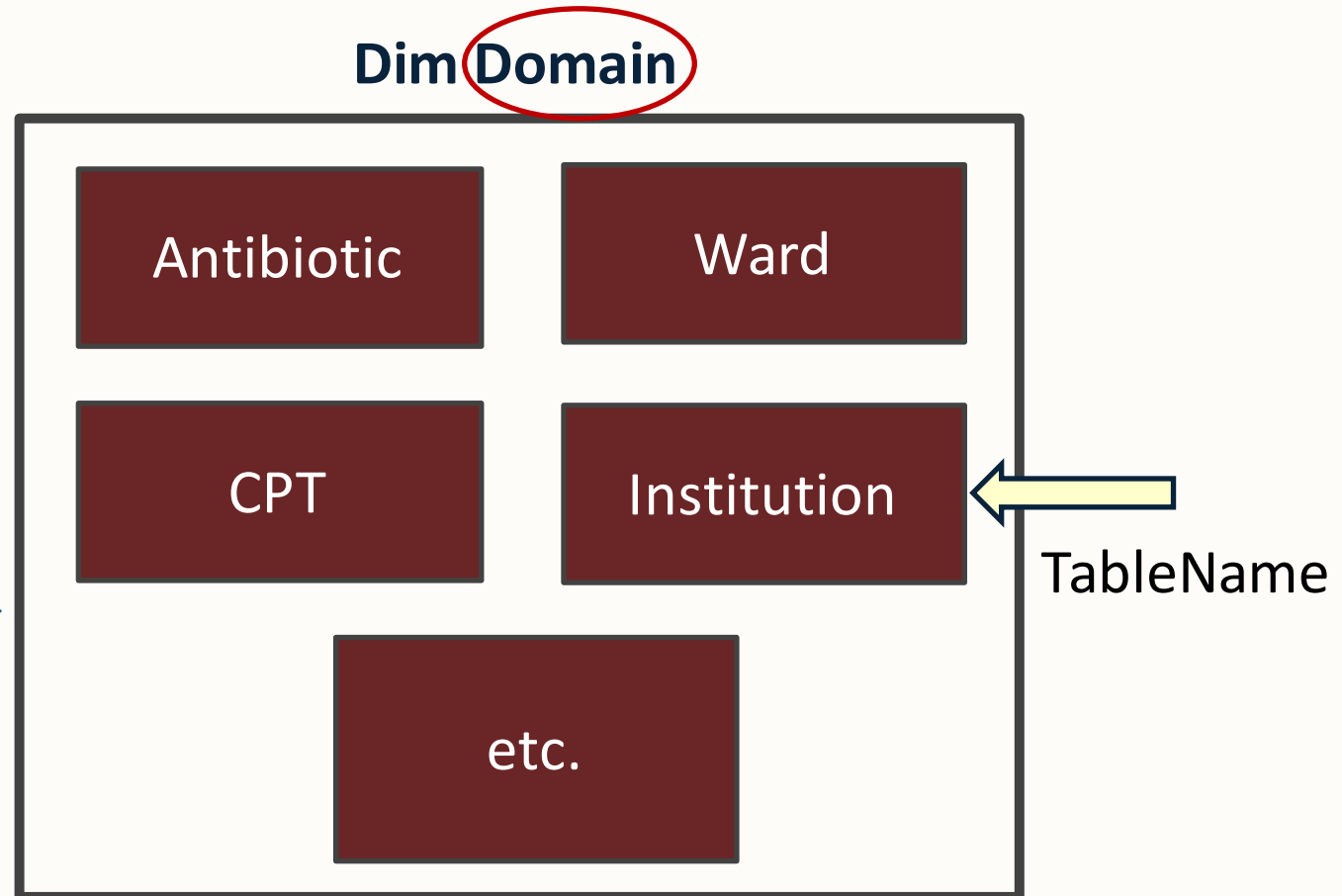


CDW Table Naming Conventions:

- Database.**Schema**TableName
- CDWWork.Dim.Institution

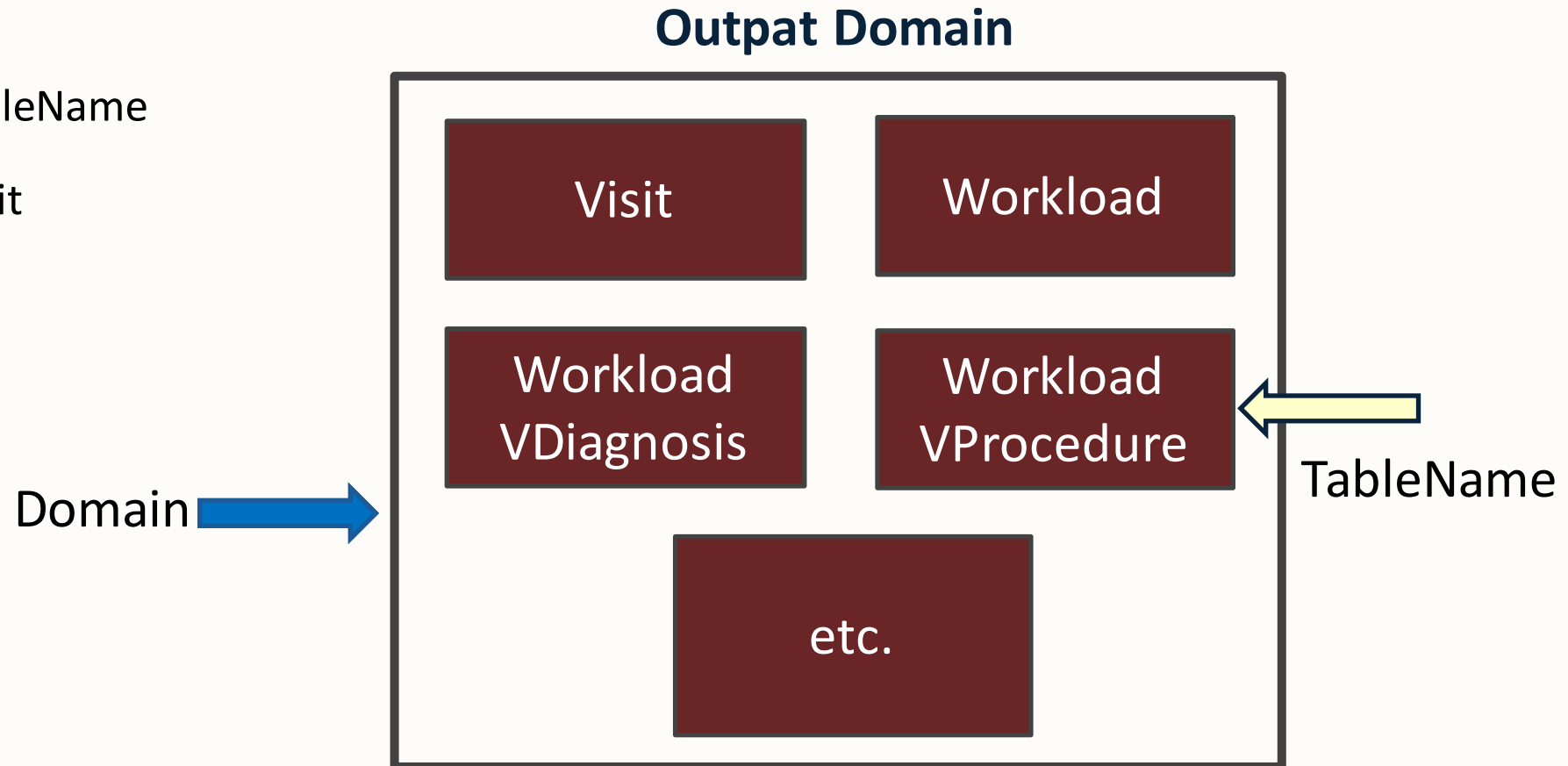
Domain 

Views (read-only copy of the actual tables)



CDW Table Naming Conventions: Operations Project Example

- Database.Schema.TableName
- CDWWork.Outputpat.Visit



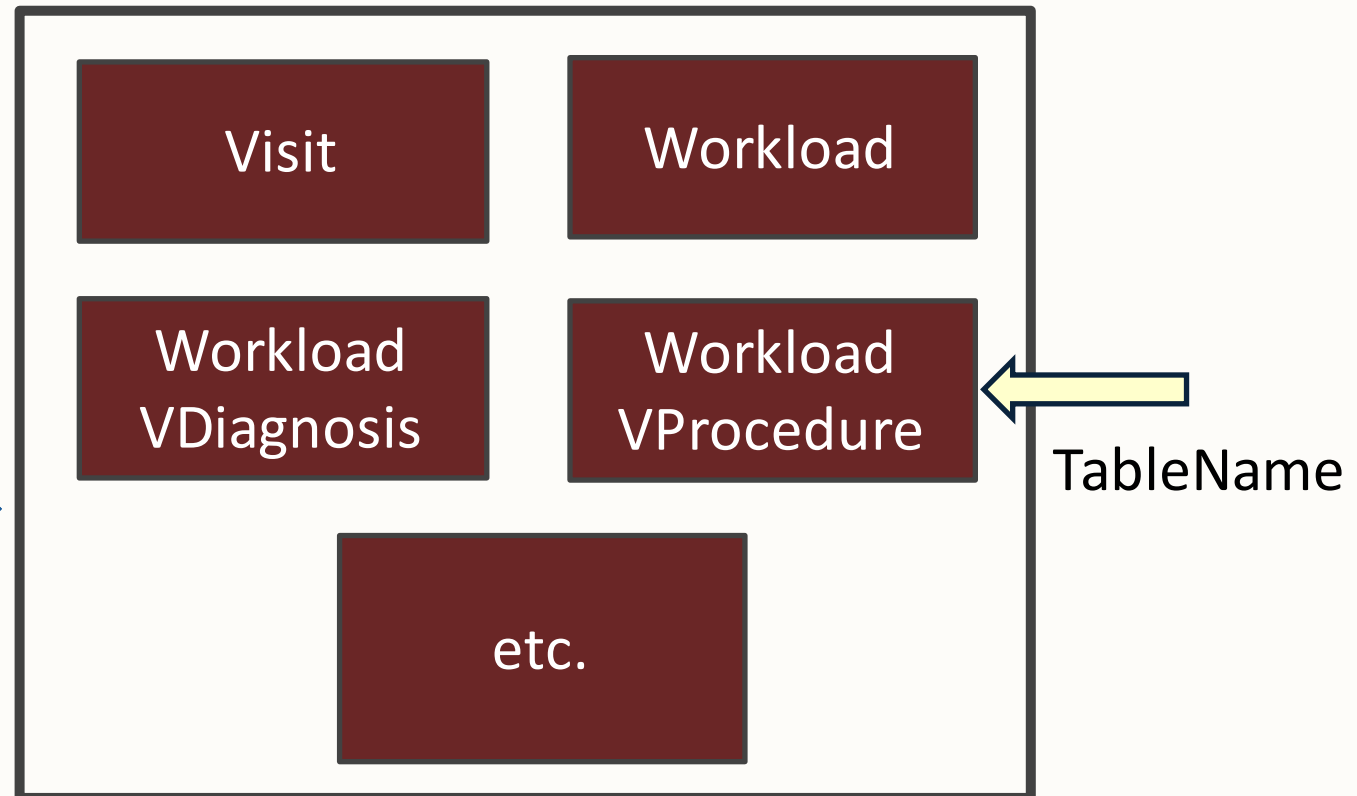
CDW Table Naming Conventions: Operations Project Example

- Database.Schema.TableName
- CDWWork.Outputpat.Visit

Domain 

“Outputpat” is a misnomer – it’s not just outpatient data

Output Domain



CDW Table Naming Conventions: Research Project Example

- Database.Schema.TableName
- ORD_Jones_xxxxx.Src.Output_Visit



Research: CDW Fact tables are in a read-only schema named Src

CDW Table Naming Conventions: Research Project Example

- Database.Schema.TableName
- ORD_Jones_XXXXX.Src.Output_Visit



Study Specific:

ORD_InvestigatorName_XXXXX
is the name of the database
VINCI provided for your study

Research: CDW Fact tables are in a read-only schema named Src



CDW Table Naming Conventions: Operations vs. Research

Operations	Research
CDWork.Outpat.Visit	ORD_Jones_XXXXX.Src.Outpat_Visit
CDWork.Dim.CPT	CDWork.Dim.CPT
	(Dim tables are not in your project database - use the copy in CDWork)
*No filtering of patients in CDW tables	*Tables in your research database only contain data for the patients in your cohort



Dimension Tables: DIM.CPT as Example

Dimension Tables

Primary key



Dim.CPT

CPTSID	sta3n	CPTCode	CPTDescription	ActiveDateTime	InactiveFlag	InactiveDateTime
800156848	358	0144T	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
1400515544	402	0144T	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
1400589273	405	0144T	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
800011726	358	75571	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL
1400121168	402	75571	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL
1400189766	405	75571	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL



Dimension Tables: DIM.CPT as Example

Dimension Tables

Dim.CPT

sta3n	CPTCode	CPTSID	CPTDescription	ActiveDateTime	InactiveFlag	InactiveDateTime
358	0144T	800156848	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
402	0144T	1400515544	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
405	0144T	1400589273	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
358	75571	800011726	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL
402	75571	1400121168	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL
405	75571	1400189766	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL

Dimension Tables: DIM.CPT as Example

Dimension Tables

Dim.CPT

sta3n	CPTCode	CPTSID	CPTDescription	ActiveDateTime	InactiveFlag	InactiveDateTime
358	0144T	800156848	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
402	0144T	1400515544	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
405	0144T	1400589273	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, INCLUDING IMAGE POST PROCESSING AND QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2006-01-01	Y	2010-01-01
358	75571	800011726	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL
402	75571	1400121168	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL
405	75571	1400189766	COMPUTED TOMOGRAPHY, HEART, WITHOUT CONTRAST MATERIAL, WITH QUANTITATIVE EVALUATION OF CORONARY CALCIUM	2010-01-01	NULL	NULL

Fact Tables: Containing CPT Codes

Fact Tables

Primary key



Outpat.WorkloadVProcedure

VProcedureSID	Sta3n	CPTSID	PatientSID	VisitSID	VisitDateTime	VProcedure DateTime	VProcedure DateSID	Quan- tity	CPRSOrder SID	Ordering Provider SID	Encounter Provider SID
80XXX295148	600	800667326	XXXX9999	8001XXXXX908	2006-11-06 08:48	2006-11-06 08:48	20061106	1	800XXX149637	14XX165	13XXX91
8000915XX921	459	800182772	XXXX8888	80XXX7402200	2006-10-11 19:13	2006-10-11 19:14	20061011	1	-1	-1	3XXX39
800XXX894344	662	800199389	XXXX7777	8002054XXX10	2007-01-26 08:05	2007-01-26 08:05	20070126	1	8003683XX723	3293XXX	316XXX3
10XXX3946269	636	1000534649	XXXX6666	1XXX245119715	2007-04-20 17:35	2007-04-20 17:35	20070420	1	-1	-1	28XX744
14002045XXX36	512	1400555589	XXX5555	14001616XXXX1	2008-03-14 10:04	2008-03-14 10:04	20080314	1	-1	-1	XX063
1200XXX78360	652	1200752332	XXXX4444	XXXX258253342	2008-11-17 08:22	2008-11-17 08:22	20081117	1	1200XXX18203	XXXX216	2XXX217
8XXX85096690	691	800668185	XXXX3333	80020XXXX615	2006-05-02 09:55	2006-05-02 09:55	20060502	1	-1	-1	363XX13

*Fields ending in SID (Surrogate ID): usually will be linking it to another table to get more information on the field



Fact Tables: Containing CPT Codes

Other tables that contain CPT procedure codes:

Inpat.InpatientCPTProcedure

Surg.SurgeryPrincipalAssociatedProcedure

Surg.SurgeryProcedureDiagnosisCode

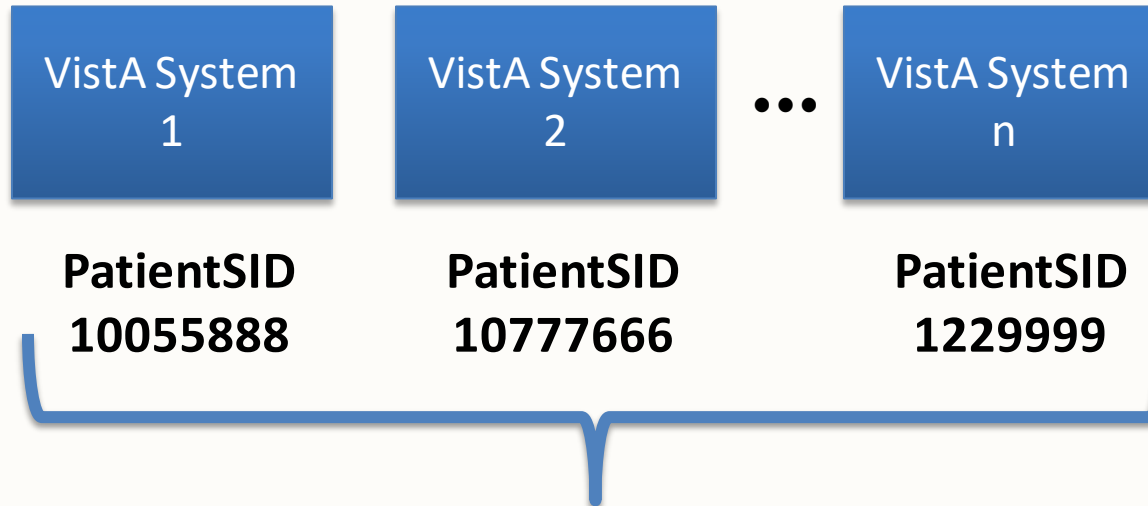
Rad.RadiologyExam

WH.WHProcedure



Patient Identifiers: SPatient Example

Patient A



Patient A has 1-n PatientSIDs

You need *all* of these to completely define all records belonging to Patient A

Fact Tables:

SPatient

Primary key



SPatient.SPpatient

PatientSID	Sta3n	PatientLast Name	PatientFirst Name	CDWPossible TestPatient Flag	Veteran Flag	PatientICN	ScrSSN	PatientSSN	Age	BirthDate Time	Death Date Time	Gender
XXXX8888	460	Doe	John J	N	Y	XXXXXXXX38	XXXXXXX44	XXXXXXX49	73	1950-01-XX	NULL	M
XXXX7777	573	Doe	John J	N	Y	XXXXXXXX38	XXXXXXX44	XXXXXXX49	73	1950-01-XX	NULL	M
XXXX6666	657	Doe	John J	N	Y	XXXXXXXX38	XXXXXXX44	XXXXXXX49	73	1950-01-XX	NULL	M
XXX5555	662	Doe	John J	N	Y	XXXXXXXX38	XXXXXXX44	XXXXXXX49	73	1950-01-XX	NULL	M
XXXX4444	673	Doe	John J	N	Y	XXXXXXXX38	XXXXXXX44	XXXXXXX49	73	1950-01-XX	NULL	M
XXXX3333	688	Doe	John J	N	Y	XXXXXXXX38	XXXXXXX44	XXXXXXX49	73	1950-01-XX	NULL	M



Fact Tables:

SPatient

SPatient.SPatient

PatientSID	Sta3n	PatientLast Name	PatientFirst Name	CDWPossible TestPatient Flag	Veteran Flag	PatientICN	ScrSSN	PatientSSN	Age	BirthDate Time	Death Date Time	Gender
XXXX8888	460	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX7777	573	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX6666	657	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXX5555	662	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX4444	673	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX3333	688	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M

Fact Tables:

SPatient

SPatient.SPatient

PatientSID	Sta3n	PatientLast Name	PatientFirst Name	CDWPossible TestPatient Flag	Veteran Flag	PatientICN	ScrSSN	PatientSSN	Age	BirthDate Time	Death Date Time	Gender
XXXX8888	460	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX7777	573	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX6666	657	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXX5555	662	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX4444	673	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX3333	688	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M



Fact Tables:

SPatient

SPatient.SPatient

PatientSID	Sta3n	PatientLast Name	PatientFirst Name	CDWPossible TestPatient Flag	Veteran Flag	PatientICN	ScrSSN	PatientSSN	Age	BirthDate Time	Death Date Time	Gender
XXXX8888	460	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX7777	573	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX6666	657	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXX5555	662	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX4444	673	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M
XXXX3333	688	Doe	John J	N	Y	XXXXXXXX38	XXXXXXXX44	XXXXXXXX49	73	1950-01-XX	NULL	M

*Joining SPatient with PatientICN or PatientSSN to get a demographic variable will generally give multiple rows



CDW Tables Needed for Example Study

Fact & Dim Tables for Example Study

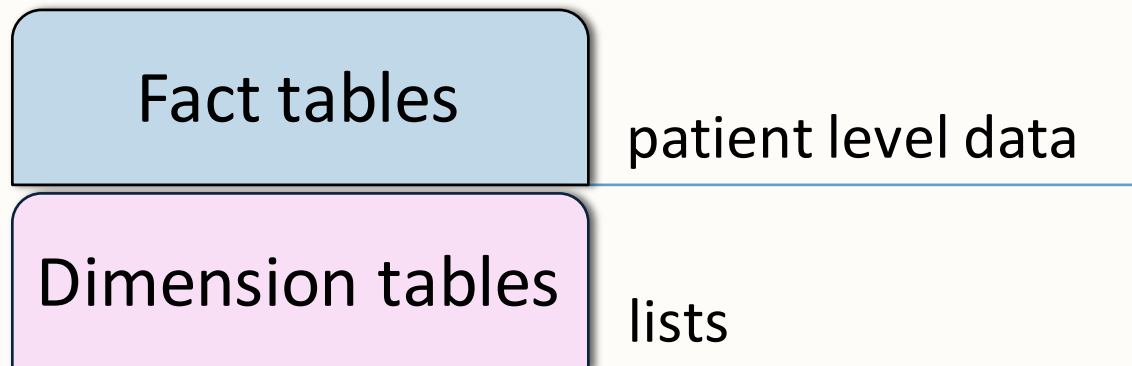



Table Name	Type	Primary Key 
SPatient.SPatient	Fact	PatientSID
Outpat.Workload VProcedure	Fact	VProcedureSID
Outpat.Workload	Fact	VisitSID
Dim.CPT	Dimension	CPTSID
Dim.Date	Dimension	DateSID
Dim.Sta3n	Dimension	Sta3n

Joining Tables: Concept

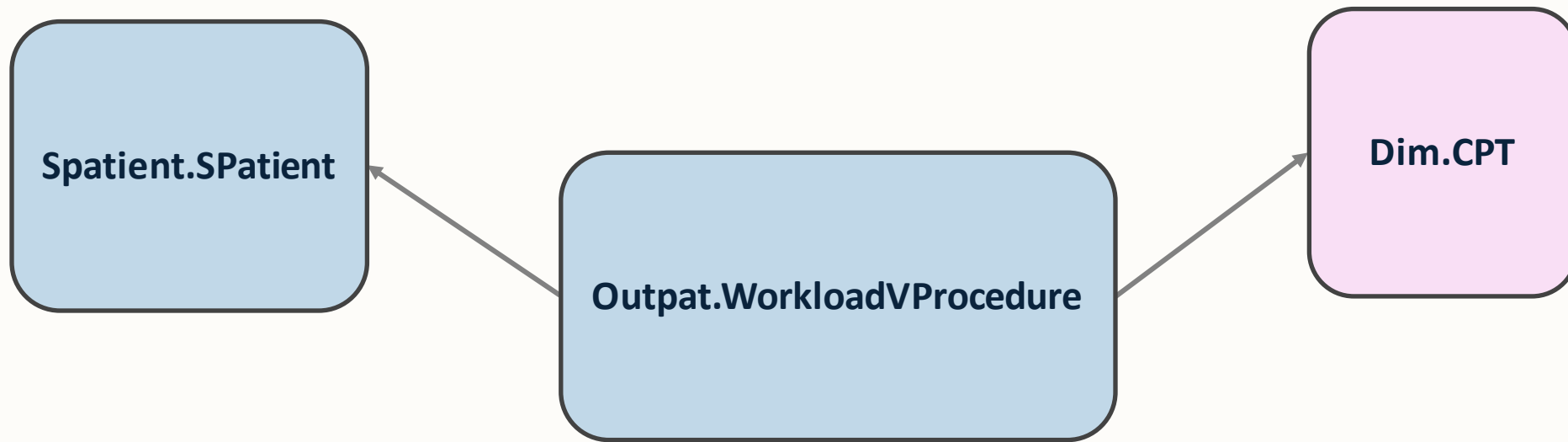
CDW data are stored in a relational SQL database

Similar data is grouped
into tables

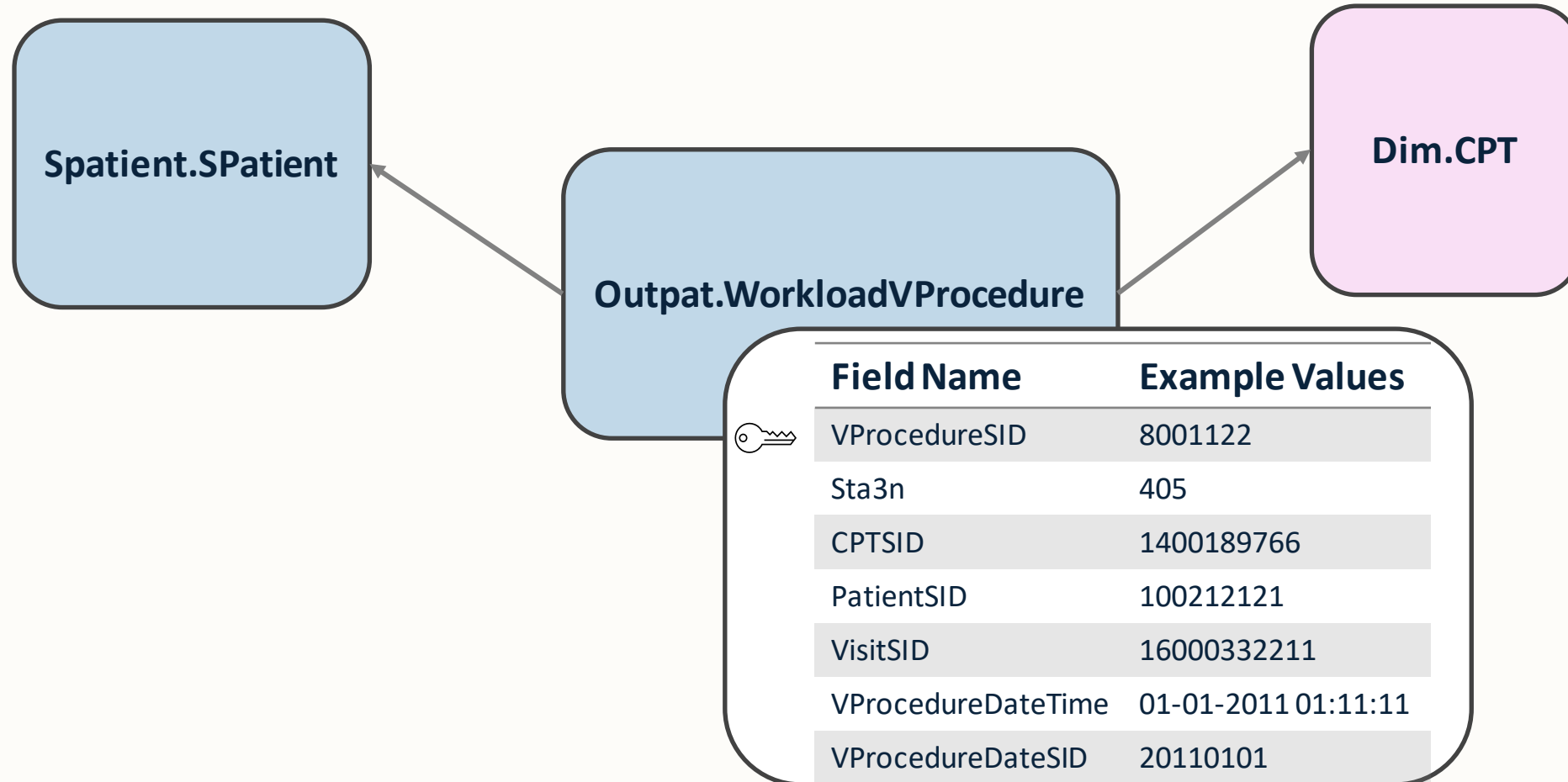
Tables join together to
provide information
for study questions



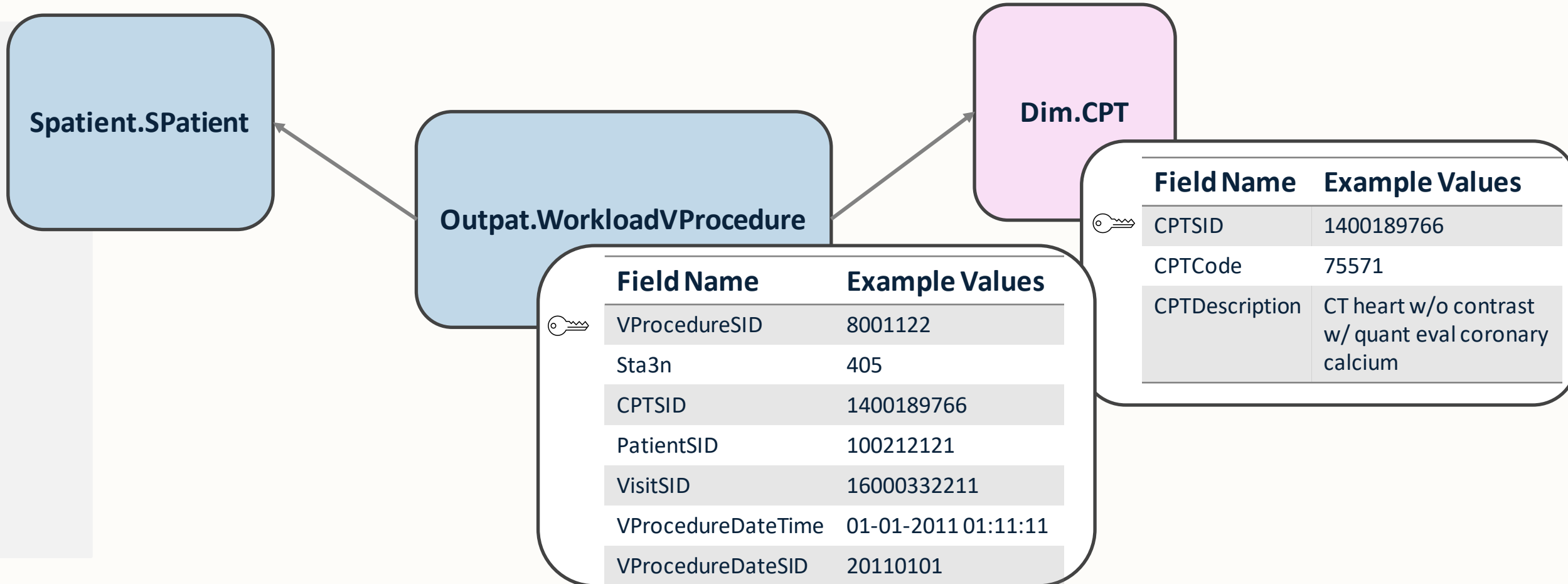
Joining Tables: Example Study



Joining Tables: Example Study



Joining Tables: Example Study



Joining Tables: Example Study

Spatient.SPatient

Outpat.WorkloadVProcedure

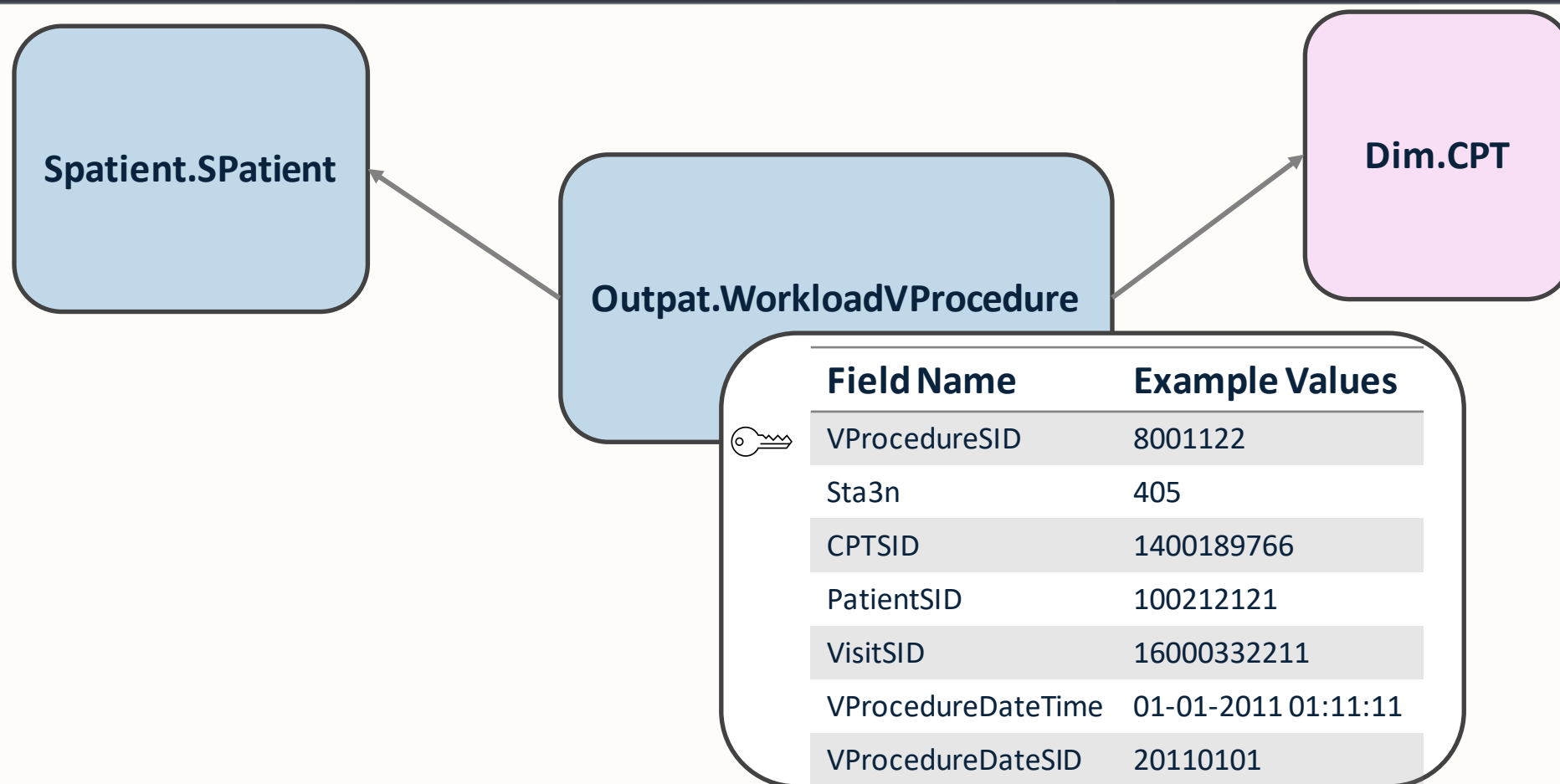
Dim.CPT

Foreign Key (**FK**): a column that refers to a primary key in another table. It specifies that a record in one table refers to a record in another table.

	Field Name	Example Values
	VProcedureSID	8001122
	Sta3n	405
FK	CPTSID	1400189766
	PatientSID	100212121
	VisitSID	16000332211
	VProcedureDateTime	01-01-2011 01:11:11
	VProcedureDateSID	20110101

	Field Name	Example Values
	CPTSID	1400189766
	CPTCode	75571
	CPTDescription	CT heart w/o contrast w/ quant eval coronary calcium

Joining Tables: Example Study



Joining Tables: Example Study

Spatient.SPatient

Field Name Example Values



PatientSID	100212121
Sta3n	554
PatientName	ZZDoe, John J
CDWPossibleTest	Y
PatientFlag	
PatientICN	1001122333
PatientSSN	555447777
BirthDateTime	1850-01-01 00:00:00

Output.WorkloadVProcedure

Field Name Example Values

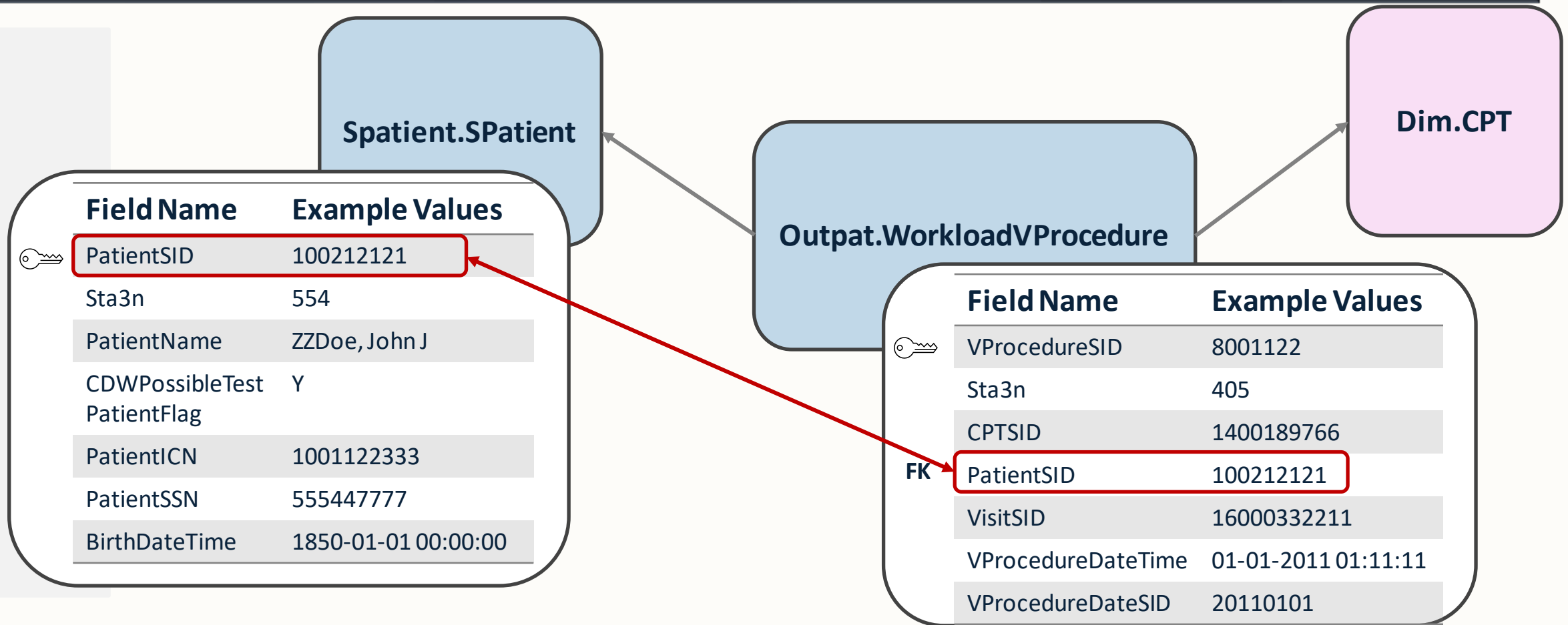


VProcedureSID	8001122
Sta3n	405
CPTSID	1400189766
PatientSID	100212121
VisitSID	16000332211
VProcedureDateTime	01-01-2011 01:11:11
VProcedureDateSID	20110101

Dim.CPT




Joining Tables: Example Study



Result:

Joining 3 Tables Together

PatientICN	Sta3n	PatientSID	VProcedure DateTime	CPTCode
1000671111	516	22xxx35	2009-01-15	0147T
1000671111	573	33xxxx11	2023-08-22	75571



Result:

Joining 3 Tables Together

PatientICN	Sta3n	PatientSID	VProcedure DateTime	CPTCode
1000671111	516	22xxx35	2009-01-15	0147T
1000671111	573	33xxxx11	2023-08-22	75571

One patient (identified by PatientICN) had a *Cardiac CT for Calcium Scoring* test at two different sites...



Result:

Joining 3 Tables Together

PatientICN	Sta3n	PatientSID	VProcedure DateTime	CPTCode
1000671111	516	22xxx35	2009-01-15	0147T
1000671111	573	33xxxx11	2023-08-22	75571

...but different PatientSID at each site.



Result:

Joining 3 Tables Together

PatientICN	Sta3n	PatientSID	VProcedure DateTime	CPTCode
1000671111	516	22xxx35	2009-01-15	0147T
1000671111	573	33xxxx11	2023-08-22	75571

PatientICN groups data for different PatientSIDs
that belong to the same patient



Session roadmap

- Review of important CDW concepts
- Describe aims of example study
- Identify procedures associated with the study aim
- Determine CDW tables of interest
- **Prepare SQL code to pull the study data**



SQL Code

SQL = “Structured Query Language”
We use a version called T-SQL

SQL Code

SQL = “Structured Query Language”
We use a version called T-SQL

If this is a RESEARCH project, you must be logged into a VINCI workspace, which is behind a firewall.

[Learn how to gain access to the VINCI Workspace](#)

SQL Code

Create List of CPTSIDs for
Cardiac CT for Calcium
Scoring



Find patients who have had
the test



Determine # of tests per FY



Determine which sites
perform the test

*Goal: Pull patients
who have had
**Cardiac CT for
Calcium Scoring***



SQL:

Select & From Statements

Create List of CPTIDs for Cardiac CT for Calcium Scoring

The Select & From statements

```
use cdwork
go
```

```
select CPTCode, CPTSID }
from Dim.CPT;          }
```

What we want

Where we get it from

	CPTCode	CPTSID
1	37721	800008494
2	37722	800008495
3	37730	800008496
4	37731	800008497
5	37735	800008498
6	37737	800008499
7	37760	800008500
8	37761	800008501
9	37765	800008502
10	37766	800008503
11	37780	800008504
12	37781	800008505

*CDWWork is the database used for Dim tables and Operations projects

SQL:

Where Clause

The Where clause

```
use cdwork
go
```

```
select CPTCode, CPTSID } What we want
from Dim.CPT           } Where we get it from
Where CPTCode in('0144T', '0147T', '0149T', '75571'); } How much do we want
```

	CPTCode	CPTSID
1	0144T	800156848
2	0144T	1400515544
3	0144T	1400589273
4	0144T	800653035
5	0144T	1000551333
6	0144T	1000579686
7	0144T	800603303
8	0144T	800116370
9	0144T	1400563605
10	0144T	800638856
11	0144T	800642710
12	0144T	1000572500



SQL:

Order By Clause

The Order By clause

```
use cdwork
go
```

```
select CPTCode, CPTSID } What we want
from Dim.CPT           } Where we get it from
Where CPTCode in('0144T', '0147T', '0149T', '75571') } How much do we want
Order By CPTSID;      } How we want to display it
```

	CPTCode	CPTSID
1	75571	800011726
2	75571	800024168
3	75571	800041452
4	75571	800049492
5	75571	800063393
6	75571	800082520
7	75571	800096248
8	0147T	800101684
9	0149T	800108507
10	0147T	800115336
11	0144T	800116370
12	0147T	800127543



SQL:

Creating a Temporary Table

Creating a Temp Table

```
use cdwork  
go
```

```
select CPTCode, CPTSID
```

```
Into #CPTCodes } Where we store it
```

```
from Dim.CPT
```

```
Where CPTCode in('0144T', '0147T', '0149T', '75571')
```

```
Order By CPTSID;
```

'#' indicates #CPTCodes is a Temporary Table

SQL:

Creating a Temporary Table

Creating a Temp Table, con't

```
use cdwork  
go
```

```
Drop table if exists #CPTCodes;
```



Delete the Temp Table if there is one

```
select CPTCode, CPTSID
```

```
Into #CPTCodes
```

```
from Dim.CPT
```



Where we store it

```
Where CPTCode in('0144T', '0147T', '0149T', '75571')
```

```
Order By CPTSID;
```

'#' indicates #CPTCodes is a Temporary Table



SQL:

Create List of CPTSIDs

Create List of CPTSIDs for Cardiac CT for Calcium Scoring

```
use cdwork
go

drop table if exists #CPTCodes;

select CPTCode, CPTSID
into #CPTCodes
from Dim.CPT
where CPTCode in ('0144T', '0147T', '0149T', '75571');
```

1. Specify CDW Database

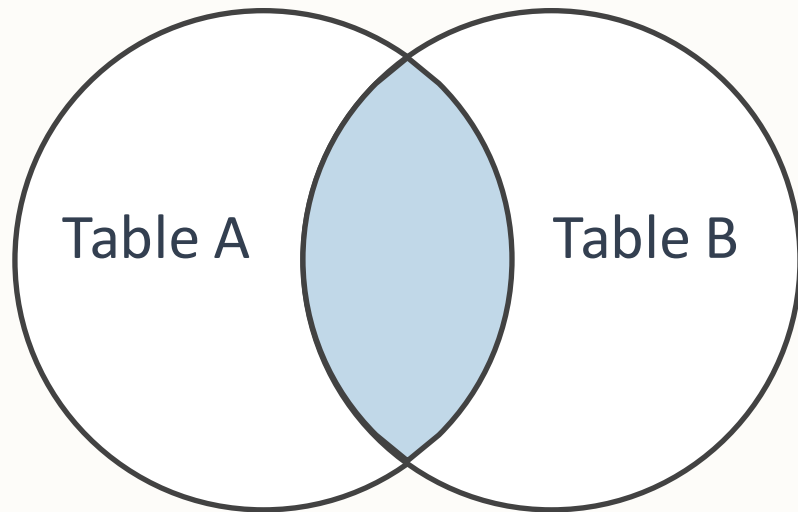
2. Drop Temp Table

3. Create Temp Table containing CPTSIDs for the 4 CPT codes of interest

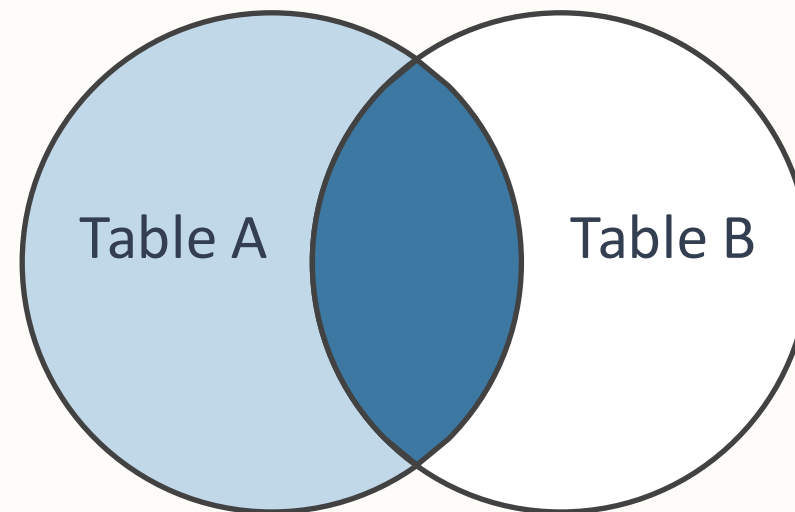
SQL Joins

Most Common Joins in SQL

Inner Join (Join)



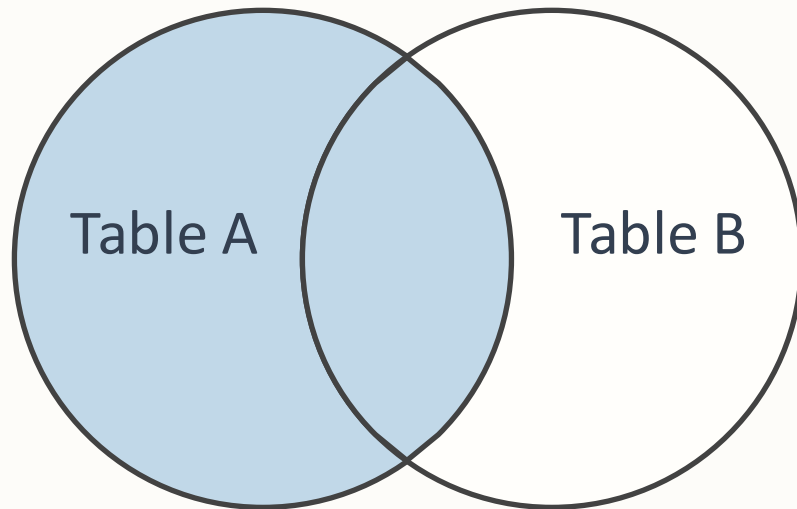
Left Outer Join (Left Join)



SQL Joins

Most Common Joins in SQL

Left Outer Join (Left Join)

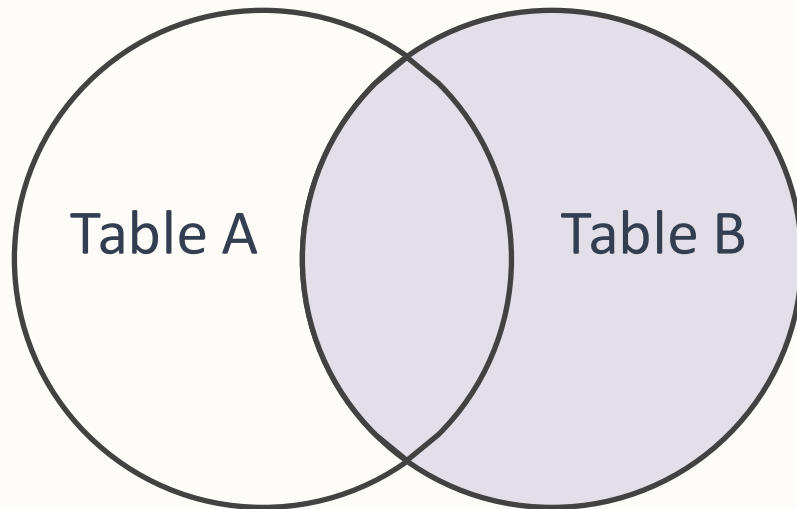


Patients in Cohort (Table A)
Jim
Eric
Colleen
Mark
Vivian
Sam

SQL Joins

Most Common Joins in SQL

Left Outer Join (Left Join)

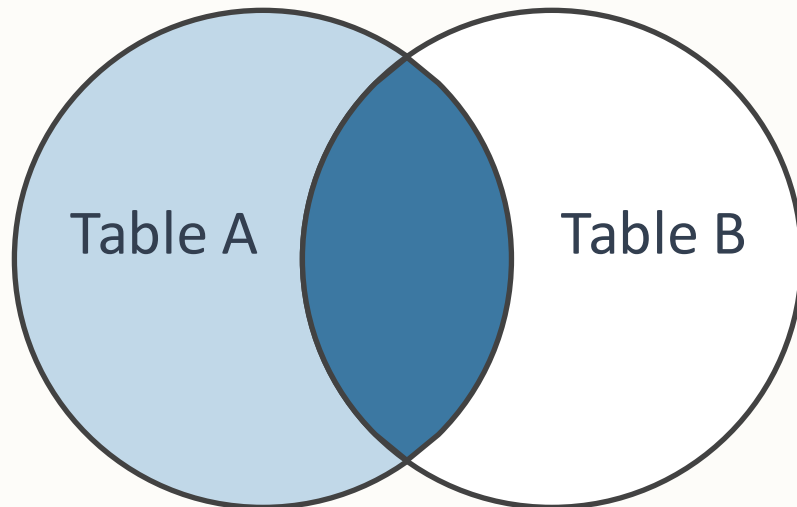


Patients in Cohort (Table A)	Patients Enrolled in Tai Chi (Table B)
Jim	Jim
Eric	
Colleen	Colleen
Mark	Mark
Vivian	
Sam	Sam
	Conrad

SQL Joins

Most Common Joins in SQL

Left Outer Join (Left Join)



Give me data for all patients in Table A, and any matching records for those patients in Table B

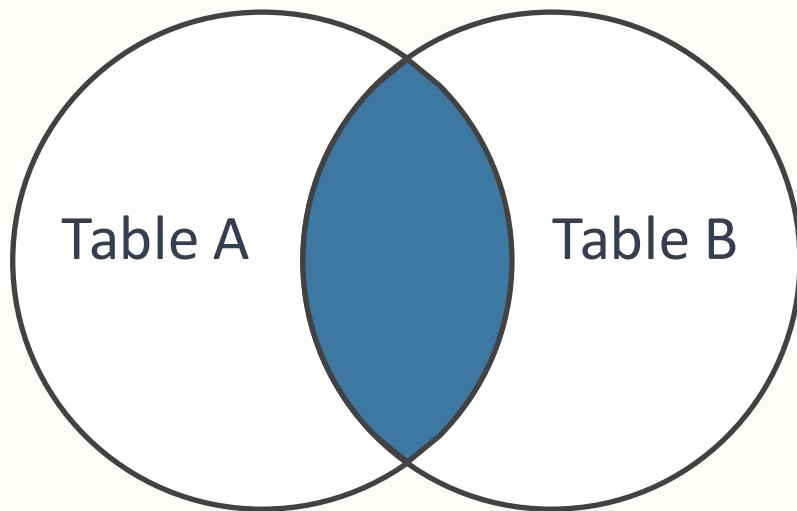
Join Results

Patients in Cohort (Table A)	Patients Enrolled in Tai Chi (Table B)
Jim	Jim
Eric	
Colleen	Colleen
Mark	Mark
Vivian	
Sam	Sam
	Conrad

SQL Joins

Most Common Joins in SQL

Inner Join (Join)

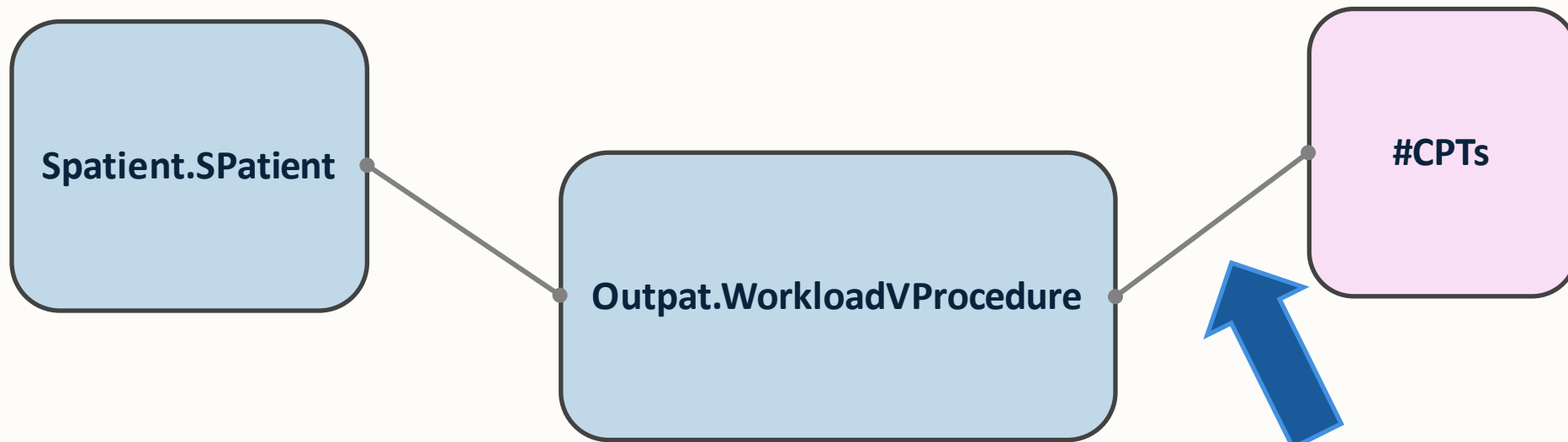


Give me data for patients that have records in *both* Table A and have matching records in Table B

Join Results

Patients in Cohort (Table A)	Patients Enrolled in Tai Chi (Table B)
Jim	Jim
Eric	
Colleen	Colleen
Mark	Mark
Vivian	
Sam	Sam
	Conrad

Joining Tables:



Inner join: we will only get records from WorkloadVProcedure for the procedures listed in #CPTs

SQL:

Pull Procedures

Find patients who have had test based on list

Goal: Pull patients who have had *Cardiac CT for Calcium Scoring*

```
drop table if exists #CoronaryCalciumProcedures;
```

```
select
```

```
  b.PatientICN
```

```
, a.Sta3n
```

```
, a.PatientSID
```

```
, a.vProcedureDateTime
```

```
, a.vProcedureDateSID
```

```
, c.CPTCode
```

```
, b.CDWPossibleTestPatientFlag
```

```
into #CoronaryCalciumProcedures
```

```
from Output.WorkloadVProcedure as a
```

```
join Spatient.Spatient as b
```

```
on a.PatientSID = b.PatientSID
```

```
join #CPTCodes as c
```

```
on a.CPTSID = c.CPTSID
```

Tables with information

(con't next slide)



SQL: Pull Procedures

Find patients who have had test based on list

Goal: Pull patients who have had **Cardiac CT for Calcium Scoring**

```
drop table if exists #CoronaryCalciumProcedures;
```

What we want

```
select
  b.PatientICN
,a.Sta3n
,a.PatientSID
,a.vProcedureDateTime
,a.vProcedureDateSID
,c.CPTCode
,b.CDWPossibleTestPatientFlag
into #CoronaryCalciumProcedures
from Output.WorkloadVProcedure as a
join Spatient.Spatient as b
on a.PatientSID = b.PatientSID
join #CPTCodes as c
on a.CPTSID = c.CPTSID
```

(con't next slide)

SQL:

Pull Procedures

Goal: Pull patients who have had **Cardiac CT for Calcium Scoring**

```

drop table if exists #CoronaryCalciumProcedures;

select
  b.PatientICN
  , a.Sta3n
  , a.PatientSID
  , a.vProcedureDateTime
  , a.vProcedureDateSID
  , c.CPTCode
  , b.CDWPossibleTestPatientFlag
into #CoronaryCalciumProcedures
from Output.WorkloadVProcedure as a
join Spatient.Spatient as b
on a.PatientSID = b.PatientSID
join #CPTCodes as c
on a.CPTSID = c.CPTSID

```

WorkloadVProcedure →

(con't next slide)

SQL:

Pull Procedures

Goal: Pull patients who have had **Cardiac CT for Calcium Scoring**

```

drop table if exists #CoronaryCalciumProcedures;

select
    b.PatientICN
    ,a.Sta3n
    ,a.PatientSID
    ,a.vProcedureDateTime
    ,a.vProcedureDateSID
    ,c.CPTCode
    ,b.CDWPossibleTestPatientFlag
into #CoronaryCalciumProcedures
from Output.WorkloadVProcedure as a
join Spatient.Spatient as b
on a.PatientSID = b.PatientSID
join #CPTCodes as c
on a.CPTSID = c.CPTSID

```

Join to SPatient {

(con't next slide)

SQL:

Pull Procedures

Goal: Pull patients who have had *Cardiac CT for Calcium Scoring*

```
drop table if exists #CoronaryCalciumProcedures;
```

```
select
```

```
  b.PatientICN
```

```
, a.Sta3n
```

```
, a.PatientSID
```

```
, a.vProcedureDateTime
```

```
, a.vProcedureDateSID
```

```
, c.CPTCode
```

```
, b.CDWPossibleTestPatientFlag
```

```
into #CoronaryCalciumProcedures
```

```
from Output.WorkloadVProcedure as a
```

```
join Spatient.Spatient as b
```

```
on a.PatientSID = b.PatientSID
```

Join to #CPTCodes

```
{ join #CPTCodes as c
  on a.CPTSID = c.CPTSID
```

(con't next slide)



SQL:

Pull Procedures

Goal: Pull patients who have had **Cardiac CT for Calcium Scoring**

```
drop table if exists #CoronaryCalciumProcedures;
```

```
select
```

```
  b.PatientICN
```

```
, a.Sta3n
```

```
, a.PatientSID
```

```
, a.vProcedureDateTime
```

```
, a.vProcedureDateSID
```

```
, c.CPTCode
```

```
, b.CDWPossibleTestPatientFlag
```

```
into #CoronaryCalciumProcedures
```

```
from Output.WorkloadVProcedure as a
```

```
join Spatient.Spatient as b
```

```
on a.PatientSID = b.PatientSID
```

```
join #CPTCodes as c
```

```
on a.CPTSID = c.CPTSID
```

Where we save it

(con't next slide)

SQL:

Pull Procedures

Where Clause limits the records that will be returned

How much do
we want

```
where a.VisitDateTime >= cast('1/1/2006' as datetime2(0))--partition key
and b.PatientICN is not null
and b.PatientICN not like '%missing%'
and b.PatientICN not like '%unknown%'
and isnull(b.CDWPossibleTestPatientFlag, 'N') <> 'Y';--ALWAYS remove test patients!
```


SQL: Pull Procedures

```
where a.VisitDateTime >= cast('1/1/2006' as datetime2(0))
```

Table
Partition
Key



Table data are partitioned and stored in separate locations – in this table partitions are separated by the VisitDateTime – *akin to knowing what file drawer to look in for visits after 1/1/2006*



SQL:

Pull Procedures

Casting literal ('1/1/2006') to same data type as the partition key is required for partition elimination; matching data types in any comparison speeds up your query too

```
where a.VisitDateTime >= cast('1/1/2006' as datetime2(0))
```

Table
Partition
Key



Table data are partitioned and stored in separate locations – in this table partitions are separated by the VisitDateTime – *akin to knowing what file drawer to look in for the data*



SQL:

Partition Key

How can I locate the Partition Key for a given table?

- VIREC Fact Books
- Meta Queries & Tutorials

```
where a.VisitDateTime >= cast('1/1/2006' as datetime2(0))
```

Table
Partition
Key



VINCI CDW Metadata Tutorials (March 2024): [VINCI Training Videos and Presentations](#)

SQL:

Pull Procedures

Query Results:

PatientICN	Sta3n	PatientSID	VProcedure DateTime	VProcedure DateSID	CPTCode	CDW PossibleTest PatientFlag
1000671111	516	22XXX35	2009-01-15 15:34	20090115	0147T	N
1000671111	573	33XXXX11	2023-08-22 13:22	20230822	75571	N
1000932222	556	444XXX9	2006-07-07 11:24	20060707	0144T	N
1006233333	459	55XXX103	2008-01-03 14:10	20080103	0149T	N

SPatient

WorkloadVProcedure

#CPTCodes

SPatient



Summary

Covered some important basics of working with CDW data

Demonstrated how to code for a procedure

Discussed the tables we needed to conduct the study

Reviewed SQL programming basics and learned how to pull the study data

Code for simple summaries of the data are in the bonus slides

VINCI SQL Boot Camp (Quarterly)

[Course Description and Sign-up Form](#)

Purpose: To support VINCI's goal of driving VA research by training research analysts on VA data basics. For new analysts, the training will rapidly accelerate their learning curve to productively contribute to research projects sooner and avoid crippling beginner data mistakes, especially the ones that unnecessarily burden shared data resources.

Delivery: This 4-week remote training will include:

- Access to a training database similar to a normal ORD research database.
- Lectures.
- Exercises (with scoring keys).
- Practice building 2-4 cohorts using CDW data.
- Practice building an analytic or "flat" file.
- Code reviews.
- Office hours.
- Documents and references for the trainee to keep.



SQL Resources & Continuing Support

[CDW Beginner SQL Office Hours \(sharepoint.com\)](#) Biweekly Wed. 1pm ET

CDW SQL Office Hours – Tuesdays: Weekly 2pm ET [CDW Training \(sharepoint.com\)](#)
Fridays: Weekly 10am ET

[VINCI Training & Office Hour \(va.gov\)](#) – Wednesdays: Weekly 3pm ET

VINCI Help Desk – VINCI@va.gov

[PBM Clinical Informatics - Analytics and Data training](#) (Pharmacy data focus) - link to SQL office hours under the PBM Communication menu. Thursdays: Weekly 11am ET

VINCI SQL Cyberseminars

Managing Research Data in SQL Server

After VINCI creates your Research Database and provisions your source data, you are free to use those resources in service of your research. But how can you best make use of those resources? This training will cover data management techniques and tips for:

- Designing/creating tables, views and schemas.
- Using indices and compression.
- Avoiding permission issues.
- Monitoring resource usage.
- Troubleshooting common issues.



Presented by Andrew Holbrook, VINCI Data Services

1

[Managing Research Data in VINCI \(va.gov\)](https://va.gov)

SQL Query Optimization for Researchers

- As research analysts, we often need costly SQL queries.
 - Wide time ranges
 - Nationwide studies
 - Complicated cohort criteria
- In this presentation, we will talk about how to safely and efficiently approach heavy data needs, and we will troubleshoot some illustrative example queries.
- This is an intermediate level presentation.

“True optimization is the revolutionary contribution of modern research to decision processes.” – George Dantzig



Presented by Andrew Holbrook, VINCI Data Services

1

[SQL Query Optimization for Researchers \(va.gov\)](https://va.gov)



THANK YOU!
Questions?



CONTACT INFORMATION

Anne E. Hines, PhD

Sr. Technical Analyst, VIREC

anne.hines@va.gov



 DATABASE & METHODS CYBERSEMINAR SERIES



Next session:

Monday, July 1st at 1 pm Eastern

Research Applications for JLV (Joint Longitudinal Viewer)

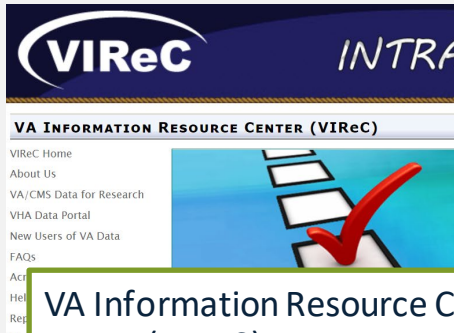


Database & Methods
BONUS SLIDES

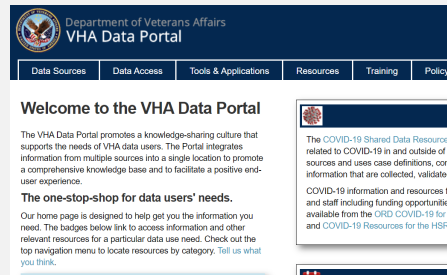


Resources for VA Data Users

Select image to visit page



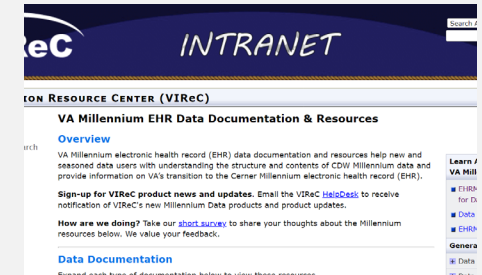
VA Information Resource Center (VIREC) (VA Intranet)



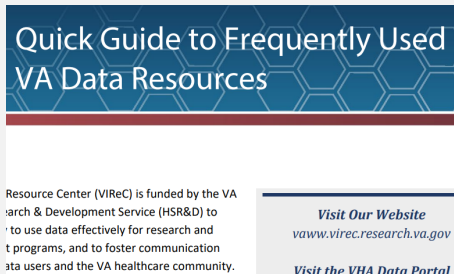
VHA Data Portal (VA Intranet)



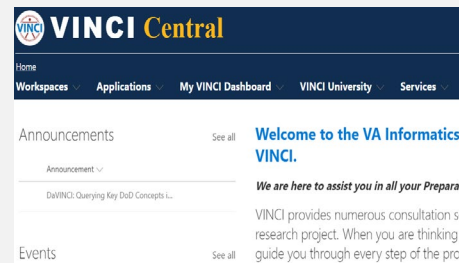
VIREC Cyberseminars



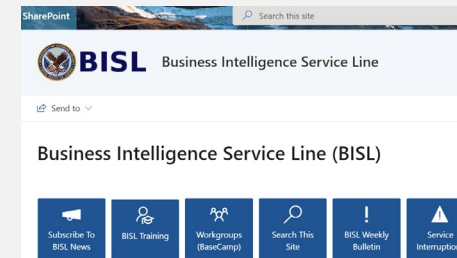
VA Millennium EHR Data Documentation (VA Intranet)



Quick Guide: Resources for Using VA Data (VA Intranet)



VA Informatics and Computing Infrastructure (VINCI) (VA Intranet)



BISL/CDW (VA Intranet)



Health Economics Resource Center (HERC) (VA Intranet)



Questions about using VA Data?

HSRData Listserv

- Community knowledge sharing
- ~1,800 VA data users
- Researchers, operations, data stewards, managers
- Subscribe by visiting vaww.virec.research.va.gov/Support/HSRData-L.htm (VA Intranet)

VIReC HelpDesk

- Individualized support
- Request Form: varedcap.rcp.vaec.va.gov/redcap/surveys/?s=KXMEN77LXK (VA Intranet)

SQL:

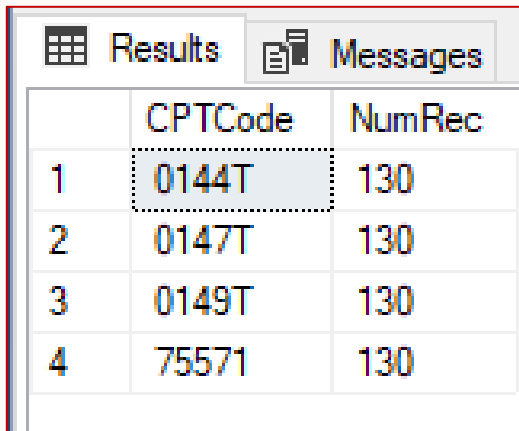
Create List of CPTSIDs

Goal: Count # of CPTSIDs for *Cardiac CT for Calcium Scoring*

```
Select CPTCode, count(CPTCode) as NumRec  
from #CPTCodes  
group by CPTCode  
order by CPTCode;
```

4. Confirm you have 130 entries for each code (one for each VistA system)

Results:



	CPTCode	NumRec
1	0144T	130
2	0147T	130
3	0149T	130
4	75571	130

SQL:

Tests Administered by FY

Goal: Determine number of **Cardiac CT for Calcium Scoring** tests performed each Fiscal Year across VA

```
select b.FiscalYear, count(b.FiscalYear) as NumRec
from #CoronaryCalciumProcedures as a
left join Dim.Date as b
on a.VProcedureDateSID = b.DateSID
group by b.FiscalYear
order by b.FiscalYear;
```

Use Group By when you need Summary Statistics

Example Summary Statistics:

- Count
- Sum, Avg
- Max, Min

SQL:

Tests Administered by FY

*Query Results: Number of **Cardiac CT for Calcium Scoring** tests performed each Fiscal Year across VA*

	FiscalYear	NumRec
1	2006	17
2	2007	141
3	2008	376
4	2009	1017
5	2010	1304
6	2011	1240
7	2012	730
8	2013	1720
9	2014	1884
10	2015	1827
11	2016	2109
12	2017	2356
13	2018	3479
14	2019	4329
15	2020	3429
16	2021	4701
17	2022	6002
18	2023	7918

Change due to?

Likely due to COVID



SQL:

Which Sites Perform this Test?

Goal: Determine which Sta3ns perform *Cardiac CT for Calcium Scoring*

```
select b.Sta3nName
       , count(a.Sta3n) as NumRec
       , min(year(a.VProcedureDateTime)) as MinDate
       , max(year(a.VProcedureDateTime)) as MaxDate
from #CoronaryCalciumProcedures as a
left join Dim.Sta3n as b
on a.Sta3n = b.Sta3n
group by a.Sta3n, b.Sta3nName
order by count(a.Sta3n) desc;
```

Summary Statistics

Use Group By when you need Summary Statistics

- Must also include any additional variable you need in the select list (Sta3nName)
- Verify its addition doesn't change your counts

SQL:

Which Sites Perform this Test?

*Results: Sta3ns that perform **Cardiac CT for Calcium Scoring***

	Sta3nName	NumRec	MinDate	MaxDate
1	(573) N. Florida/S. Georgia HCS (Gainesville FL)	17758	2009	2023
2	(618) Minneapolis, MN (CACHE 5.0)	2201	2007	2023
3	(549) North Texas HCS (Dallas TX)	1597	2008	2023
4	(621) Mountain Home, TN	1519	2008	2023
5	(501) New Mexico HCS (Albuquerque NM)	1229	2008	2023
6	(644) Phoenix, AZ	1191	2009	2023
7	(671) South Texas HCS (San Antonio TX)	1119	2008	2023
8	(652) Richmond, VA	975	2006	2023
9	(506) Ann Arbor, MI	821	2012	2023
10	(534) Charleston, SC	802	2013	2023
11	(526) Bronx, NY	759	2008	2023

	Sta3nName	NumRec	MinDate	MaxDate
74	(756) El Paso, TX	14	2015	2017
75	(667) Shreveport, LA	11	2018	2023
76	(402) Togus ME	3	2022	2022
77	(673) Tampa, FL	3	2023	2023
78	(578) Hines, IL	3	2012	2021
79	(674) Central Texas HCS (Temple TX)	3	2023	2023
80	(612) Northern California HCS (Martinez CA)	2	2015	2018
81	(626) Tennessee Valley HCS (Nashville TN)	2	2009	2009
82	(646) Pittsburgh HCS (Pittsburgh PA)	2	2011	2011
83	(503) Altoona, PA	2	2023	2023
84	(687) Walla Walla, WA	1	2022	2022

SQL:

Tests Administered by FY

Goal: Determine number of **Cardiac CT for Calcium Scoring** tests performed each Fiscal Year, by Sta3n: FY2019-FY2022

```
select a.Sta3n, b.FiscalYear, count(*) as NumRec
from #CoronaryCalciumProcedures as a
left join Dim.Date as b
on a.VProcedureDateSID = b.DateSID
where b.FiscalYear in (2019, 2020, 2021, 2022)
group by a.Sta3n, b.FiscalYear
order by a.Sta3n, b.FiscalYear;
```

We want to group by two separate variables

SQL:

Tests Administered by FY

Goal: Determine number of **Cardiac CT for Calcium Scoring** tests performed each Fiscal Year, by Sta3n: FY2019-FY2022

```
select a.Sta3n, b.FiscalYear, count(*) as NumRec
from #CoronaryCalciumProcedures as a
left join Dim.Date as b
on a.VProcedureDateSID = b.DateSID
where b.FiscalYear in (2019, 2020, 2021, 2022)
group by a.Sta3n, b.FiscalYear
order by a.Sta3n, b.FiscalYear;
```

“in” operator limits to 4 FY of interest

SQL:

Tests Administered by FY

Results: Number of **Cardiac CT for Calcium Scoring** tests performed each Fiscal Year, by Sta3n: FY2019-FY2022

	Results	Messages	
	Sta3n	FiscalYear	NumRec
17	502	2019	7
18	502	2020	4
19	502	2021	69
20	502	2022	84
21	506	2019	105
22	506	2020	74
23	506	2021	123
24	506	2022	140
25	508	2019	54
26	508	2020	43
27	508	2021	19
28	508	2022	27
29	509	2020	2
30	509	2021	6
31	509	2022	56



SQL:

Tests Administered by FY

Results: Number of **Cardiac CT for Calcium Scoring tests** performed each Fiscal Year, by Sta3n: FY2019-FY2022

	Sta3n	FiscalYear	NumRec
17	502	2019	7
18	502	2020	4
19	502	2021	69
20	502	2022	84
21	506	2019	105
22	506	2020	74
23	506	2021	123
24	506	2022	140
25	508	2019	54
26	508	2020	43
27	508	2021	19
28	508	2022	27
29	509	2020	2
30	509	2021	6
31	509	2022	56

Where Clause: "in" operator limits to four FY of interest



SQL:

Tests Administered by FY

Results: Number of **Cardiac CT for Calcium Scoring** tests performed each Fiscal Year, by Sta3n: FY2019-FY2022

	Results	Messages	
	Sta3n	FiscalYear	NumRec
17	502	2019	7
18	502	2020	4
19	502	2021	69
20	502	2022	84
21	506	2019	105
22	506	2020	74
23	506	2021	123
24	506	2022	140
25	508	2019	54
26	508	2020	43
27	508	2021	19
28	508	2022	27
29	509	2020	2
30	509	2021	6
31	509	2022	56



SQL:

Inpatient Status

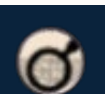
Goal: Determine distribution of *Cardiac CT for Calcium Scoring* tests across inpatients and outpatients

```
drop table if exists
#CoronaryCalciumProceduresWithVisitSID;

select  b.PatientICN
, a.Sta3n
, a.PatientSID
, a.VisitSID ←
, a.vProcedureDateTime
, a.vProcedureDateSID
, c.CPTCode
, b.CDWPossibleTestPatientFlag
into #CoronaryCalciumProceduresWithVisitSID
from Outpat.WorkloadVProcedure as a
join Spatient.Spatient as b
on a.PatientSID = b.PatientSID
join #CPTCodes as c
on a.CPTSID = c.CPTSID

where a.VisitDateTime >= cast('10/1/1999' as datetime2(0))
and b.PatientICN is not null
and b.PatientICN not like '%missing%'
and b.PatientICN not like '%unknown%'
and isnull(b.CDWPossibleTestPatientFlag, 'N') <> 'Y';
```

Need to add the **foreign key VisitSID** to join to Outpat.Workload table



SQL:

Inpatient Status

Goal: Determine distribution of **Cardiac CTs for Calcium Scoring** tests across inpatients and outpatients

- Join to parent table Output.Workload

```
drop table if exists #ProcedureInpatientStatus;

select a.PatientICN, b.PatientStatusInOut
into #ProcedureInpatientStatus
from #CoronaryCalciumProceduresWithVisitSID as a
join Output.Workload as b
on a.VisitSID = b.VisitSID
where b.VisitDateTime >= cast('1/1/2006' as datetime2(0));
```

SQL:

Inpatient Status

Results: Distribution of **Cardiac CT for Calcium Scoring** tests across inpatients and outpatients

```
select PatientStatusInOut
       , count(PatientStatusInOut) as NumRec
from #ProcedureInpatientStatus
group by PatientStatusInOut
order by PatientStatusInOut;
```

	PatientStatusInOut	NumRec
1	0	38789
2	1	5790

Ideas to Expand the Study



Incorporate non-VA data and its effect on VA use over time and sites

Add cost data

