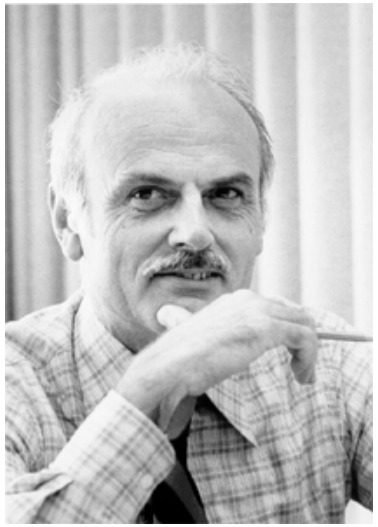
DATABASE FOR HEALTHCARE: OVERVIEW AND EXAMPLES

Kei-Hoi Cheung, PhD

Biomedical Informatics and Data Science (BIDS)

Yale School of Medicine, New Haven, CT

Database Pioneers



E.F. Codd

Peter Chen

Jim Gray

What is a relational (SQL) database?

- The relational data model is a way to model data in the form of relations or tables
- It uses a collection of tables (relations) to represent both data and the relationships among those data
- Each table has multiple columns (attributes), and each column has a unique name
- A relational database is the implementation of the relational model as a software system called the Relational Database Management System (RDBMS)

  (e.g., Oracle, MS SQLServer, MySQL, etc)
- It comes with a standard Structured Query Language (SQL)

Constraints in Relational Model

- **Domain constraint**: a column can only take values that lie inside the domain range (e.g., Age>0)
- **Key constraint**: a table should have a least one column (or multiple columns) that define a row uniquely. Such a column (or a set of columns) is called a (primary) key. No two rows can have the same key. Also, a key cannot have NULL values (e.g., Patient_ID is the key of the Patients table)
- **Referential integrity**: one column of a table can only take values from another column (foreign key) of the same or a different table (e.g. A patient's race can only take values which are present in the name of the Race table)

Normalization

- It is the process of organizing data in relational database to minimize data redundancy and inconsistent dependency
- There are different forms of normalization (normal forms)
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)
- Functional dependency
  - Given two columns A and B in a table, A->B (A determines B) means that A uniquely identifies B (e.g., Patient_ID determines Patient_Name)

First normal form (1NF)

- **1NF is used to remove repeated groups from a table to guarantee atomicity**

Patient_ID	Patient_Name	Phone_Number
1	John Smith	1234567890
2	Mary Doe	4567890123, 2345678910

Patient_ID	Patient_Name	Phone_Number
1	John Smith	1234567890
2	Mary Doe	4567890123
2	Mary Doe	2345678910

Second normal form (2NF)

- **2NF is used to reduce data redundancy by eliminating partial dependencies**

Patient_ID	Provider_ID	Patient_Name	Provider_Name
1	D1	John Smith	Susan Lee
2	D2	Mary Doe	Peter Pan
3	D1	Kay Stone	Susan Lee

Patient_ID	Patient_Name
1	John Smith
2	Mary Doe
3	Kay Stone

Provider_ID	Provider_Name
D1	Susan Lee
D2	Peter Pan

Third normal form (3NF)

- **3NF is used to reduce data redundancy by removing transitive dependencies**

Patient_ID	Patient_Name	Zip_Code	City
1	John Smith	06511	New Haven
2	Mary Doe	06611	Trumbull

Patient_ID	Patient_Name	Zip_Code
1	John Smith	06511
2	Mary Doe	06611

Zip_Code	City
06511	New Haven
06611	Trumbull

Structured Query Language (SQL)

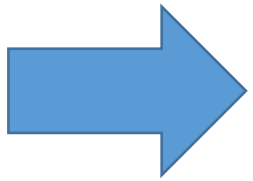
- It is an ANSI and ISO standard language for accessing and manipulating relational databases
- It supports the following database operations:
  - Create a new database (CREATE DATABASE)
  - Create a new table (CREATE TABLE)
  - Insert records in a database (INSERT)
  - Update record in a database (UPDATE)
  - Delete records from a database (DELETE)
  - Data retrieval (SELECT)

SELECT Statement (JOIN)

```
SELECT A.*, B.Report_Text
FROM Patient_DB.Patient AS A
INNER JOIN Patient_DB.LabTest. AS B
ON A.ID = B.Patient_ID
```

ID	Name	Address	Age	Sex
1	John Doe	XYZ	40	M
2	Jane Smith	ABC	34	F
3	Mary Queen	PQSRT	46	F
4	Mike Lee	DWQER	60	M

Patient_ID	ID	Report_Text
1	1	……
1	2	…….



ID	Name	Address	Age	Sex	Report_Text
1	John Doe	XYZ	40	M	…..
2	Jane Smith	ABC	34	F	…..
3	Mary Queen	PQSRT	46	F	…..
4	Mike Lee	DWQER	60	M	…..

Pivot and De-pivot

SQL PIVOT diagram

You can use PIVOT to rotate rows in a table by turning row values into multiple columns. The following diagram illustrates what PIVOT can do where we take 4 rows of data and turn this into 1 row with 4 columns. As you can see, the PIVOT process converts rows into columns by pivoting the table.

Col1	Col2
1	3445
2	2334
3	5555
4	5555

Col2	1	2	3	4
Col2	3445	2334	5555	5555

SQL UNPIVOT diagram

On the other hand, unpivot does the opposite and rotates multiple columns into multiple rows.

Col1	1	2	3	4
Col2	3445	2334	5555	5555

Col1	Col2
1	3445
2	2334
3	5555
4	5555

ExamData

Name	Subject	Marks
Amit	Operating Systems	90
Amit	DSA	80
Amit	DBMS	70
Amit	Computer Networks	85
Manisha	Operating Systems	70
Manisha	DSA	60
Manisha	DBMS	70
Manisha	Computer Networks	55
Ramesh	Operating Systems	80
Ramesh	DSA	90
Ramesh	DBMS	95
Ramesh	Computer Networks	88

```
SELECT * FROM (
  SELECT
  [id],
    [Name],
    [Subject],
    [Marks]
  FROM ExamData
) ExamResults
PIVOT (
  SUM([Marks])
  FOR [Subject]
  IN (
    [Operating Systems],
    [DSA],
    [DBMS],
    [Computer Networks]
  )
) AS PivotTable
```

Name	Operating Systems	DSA	DBMS	Computer Networks
Amit	90	80	70	85
Manisha	70	60	70	55
Ramesh	80	90	95	88

Student

StudentID	Math	Science	English
1	70	80	90
2	90	55	60
3	80	70	90
4	75	65	80

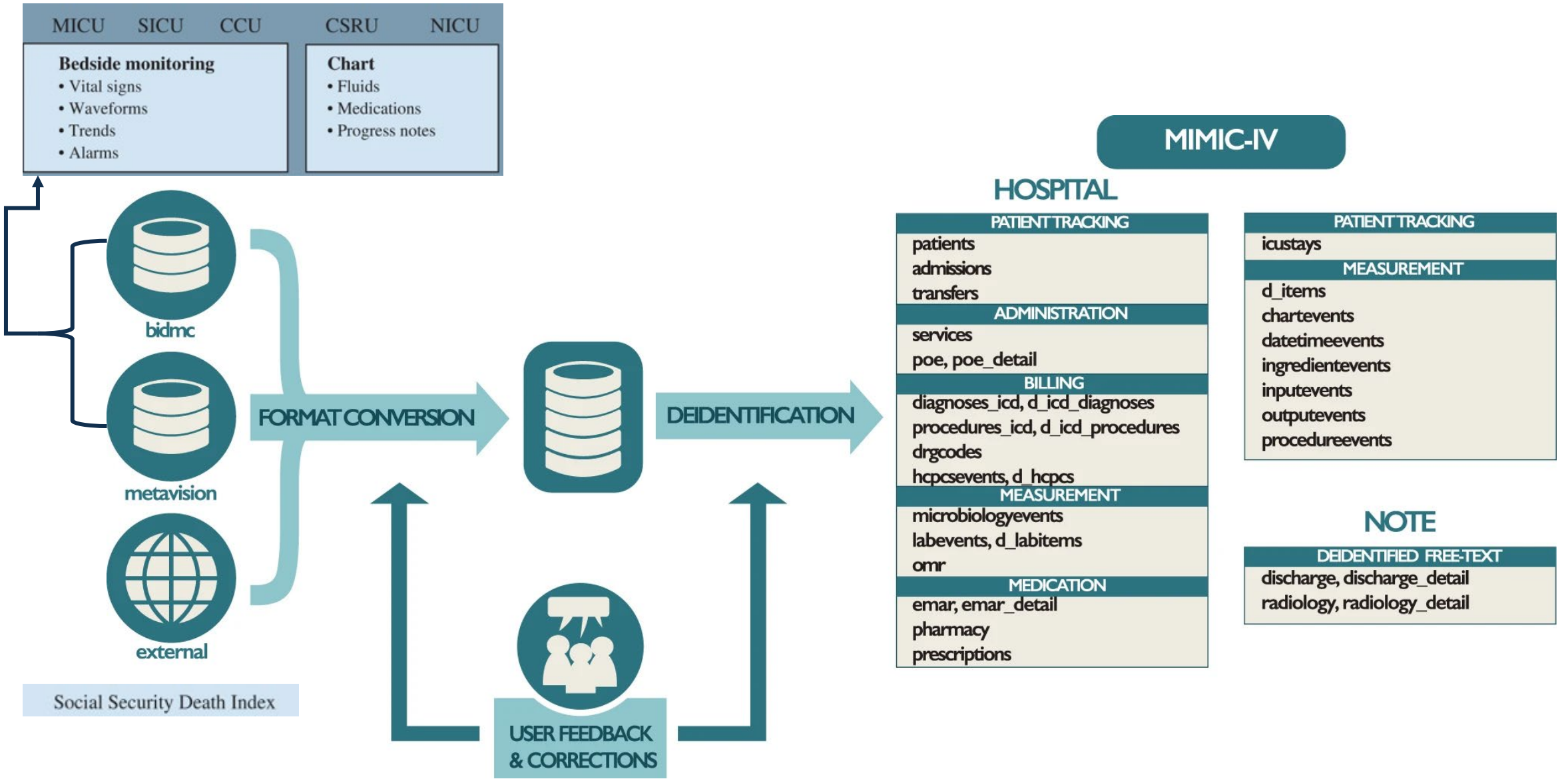
```
SELECT StudentID, [SubjectNames], Marks
FROM (
    SELECT StudentID, Math, Science, English
    FROM Student
) AS s
UNPIVOT
(
    Marks FOR [SubjectNames] IN (Math,
Science, English)
) AS unpvt;
```

StudentID	SubjectNames	Marks
1	Math	70
1	Science	80
1	English	90
2	Math	90
2	Science	55
2	English	60
3	Math	80
3	Science	70
3	English	90
4	Math	75
4	Science	65
4	English	80

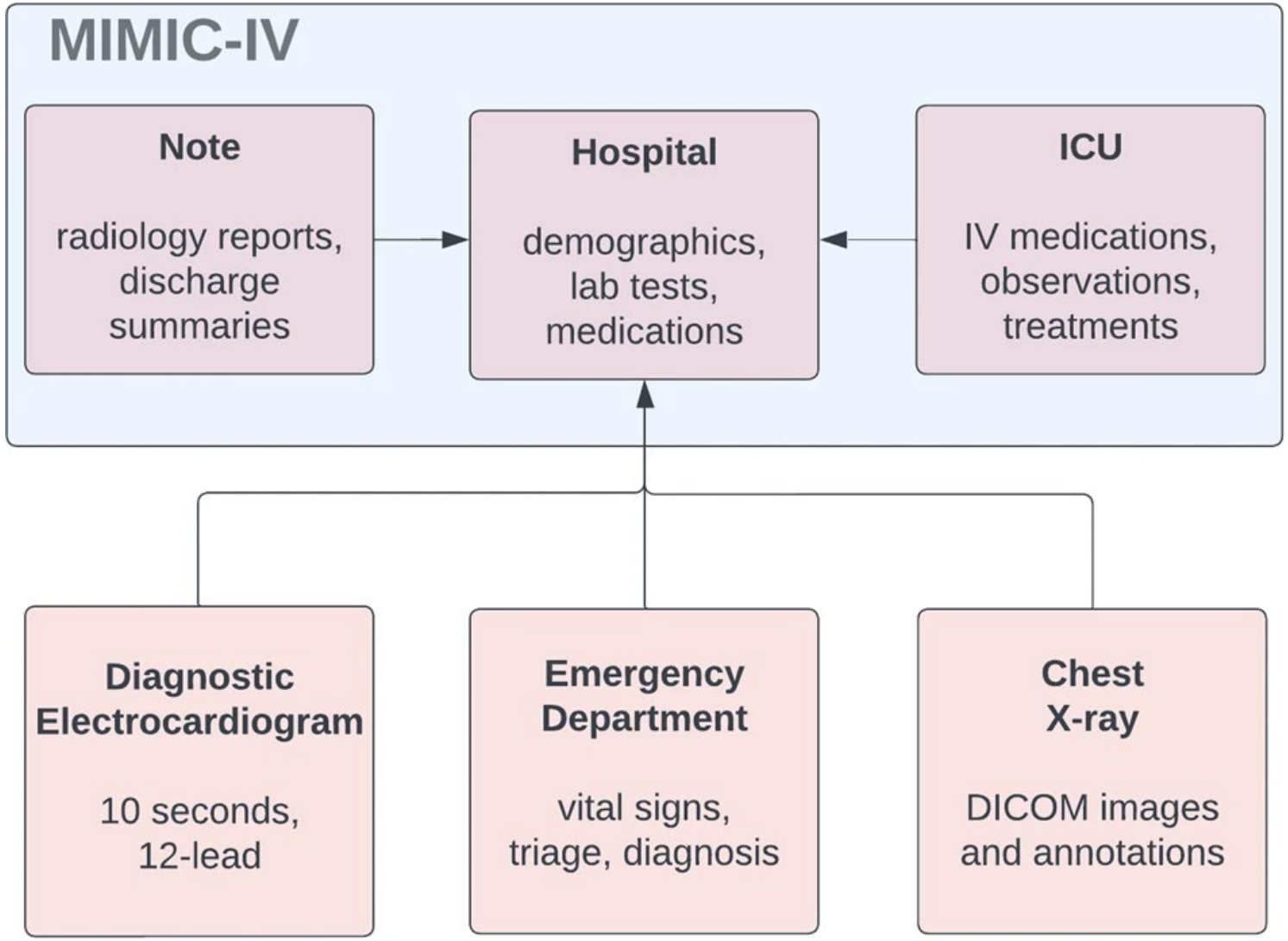
Data Warehouse and Data Mart

- A data warehouse is a central repository that integrates large amounts of data from various databases (across an entire organization)
- It allows comprehensive analysis
- A data mart is a smaller subset of data warehouse focused on a specific area of interest for targeted user needs
- A data warehouse/data mart is read/retrieval only for online analytical processing (OLAP)
- MIMIC (Medical Information Mart for Intensive Care) and CDW (Corporate Data Warehouse)

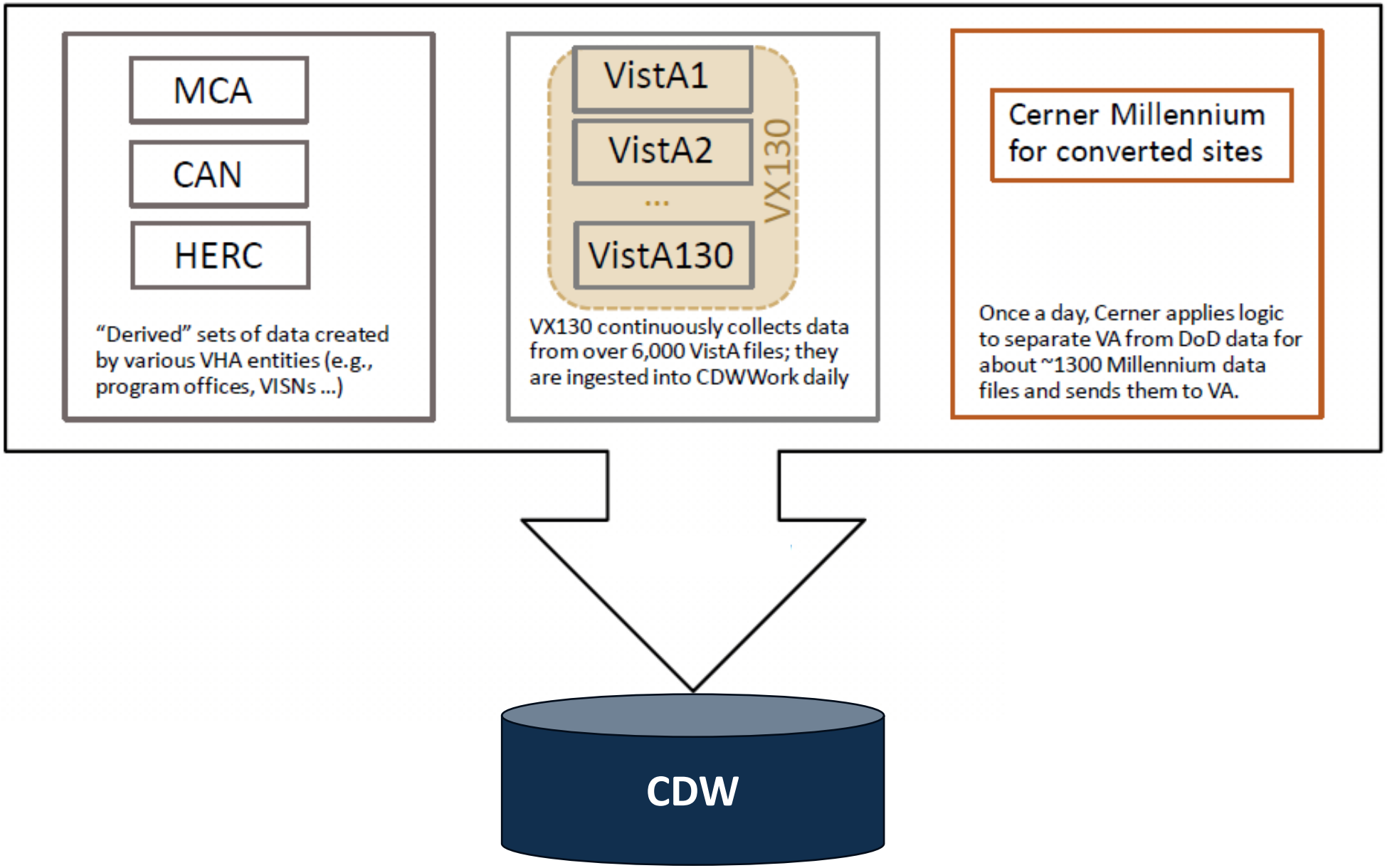
Overview of the development process of MIMIC



Modular structure



Corporate Data Warehouse (CDW) at VA

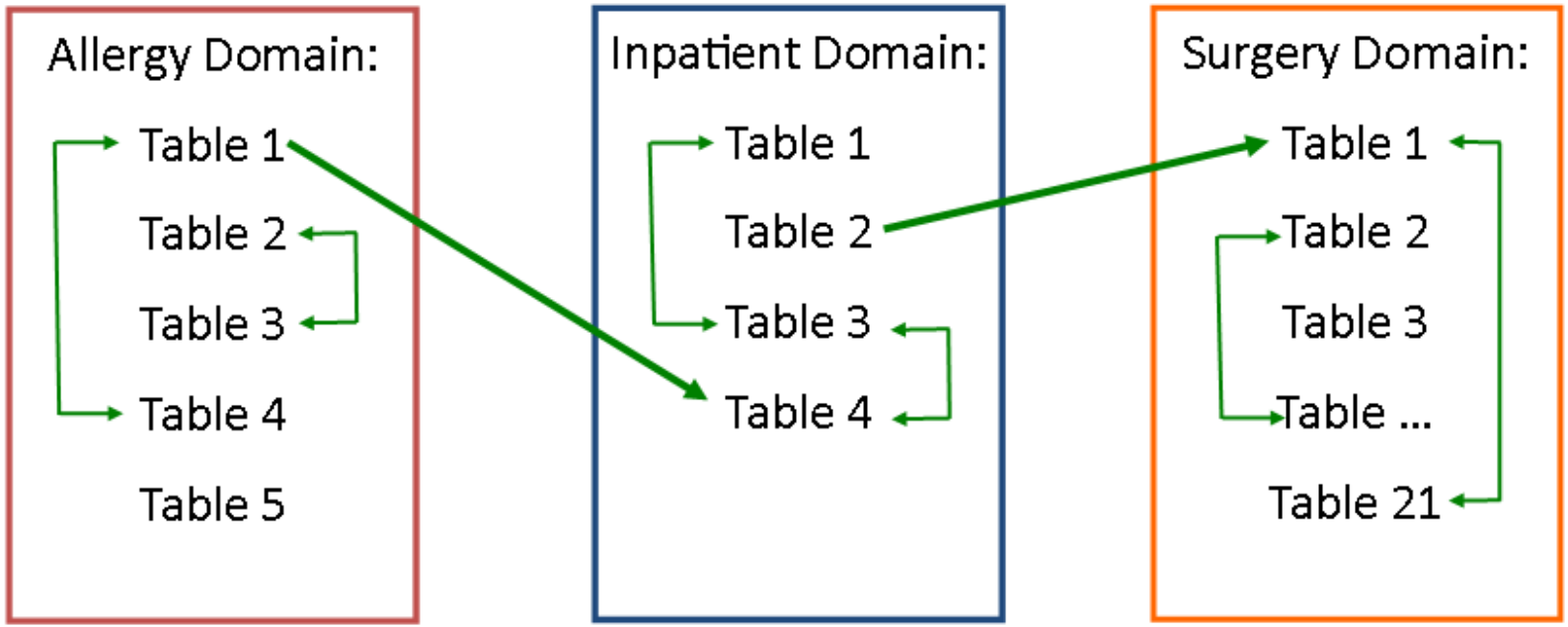


Corporate Data Warehouse (CDW)

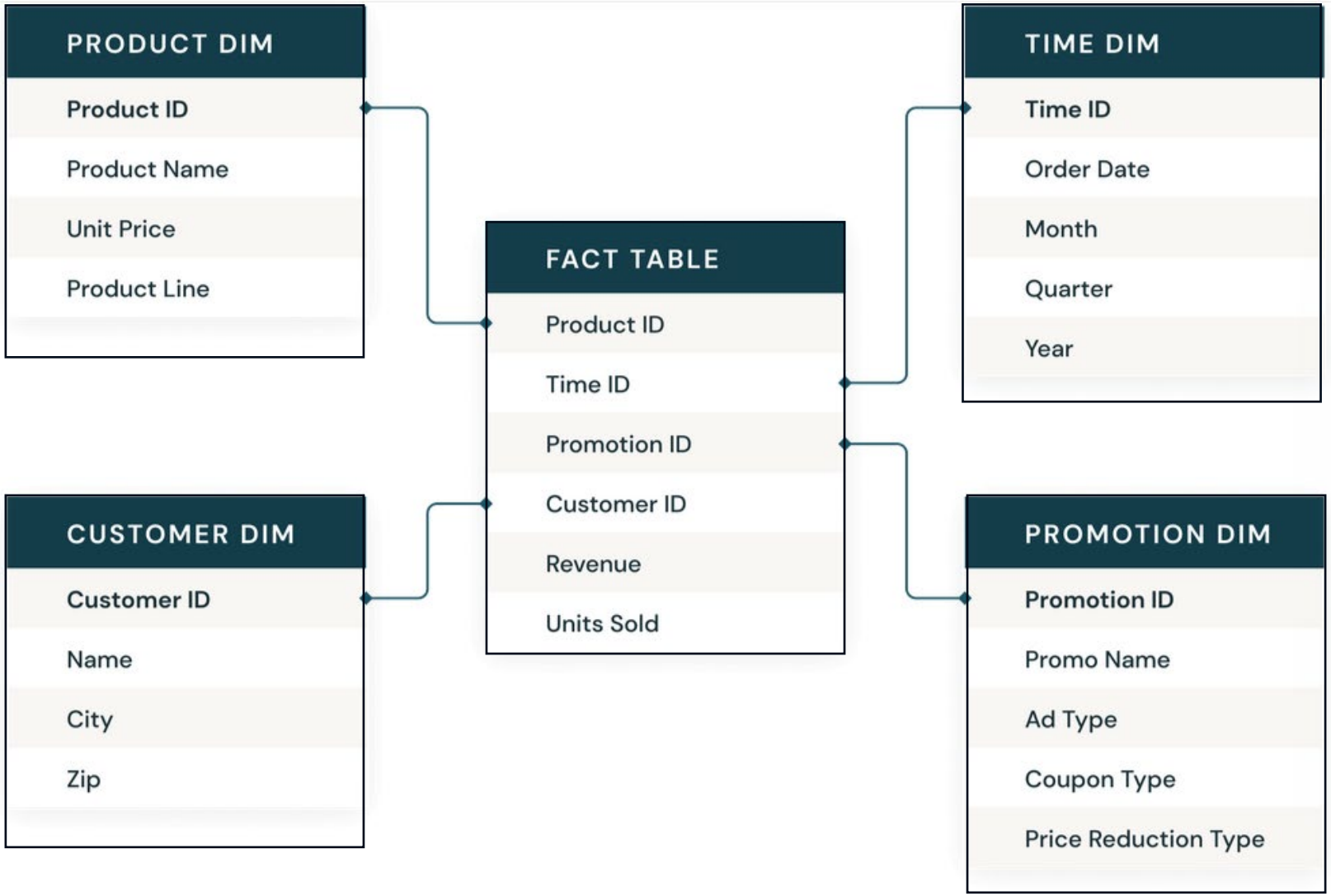
CDW organizes data into data domains

with multiple tables in each domain

and builds linking keys to connect tables and domains:



Multi-Dimensional Data Model (Star Schema)



Fact and Dim Tables in CDW

PatientTable (Fact)

PatientKey (PK)	LastName	FirstName	Address	CityState	Zipcode
1	Jones	Marianna	123 Oak St	Bee, AR	70788
2	Frank	Josie	11 Pine Ave	Flip, OK	30032
3	Plank	Bill	230 5th St	Miner, TX	11201

DiagnosisTable (Dim)

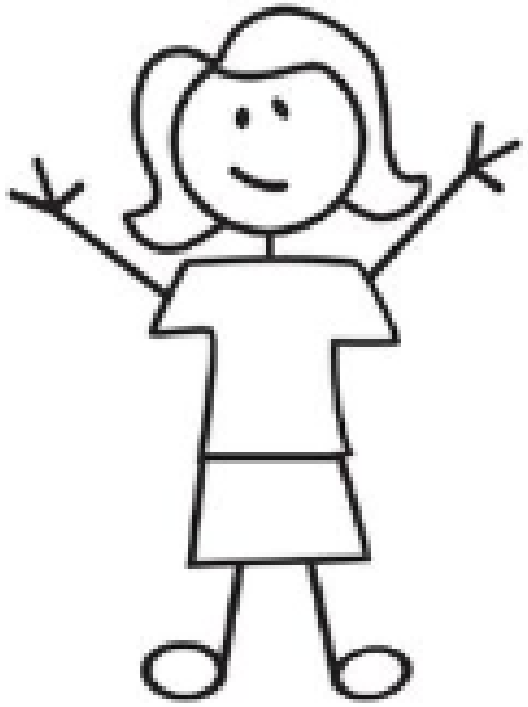
DxKey (PK)	ICD9
1	110.6
2	202.05
3	280.8
4	377.75

VisitTable (Fact)

VisitKey (PK)	DxKey (FK)	PatientKey (FK)	Sta3n	Date
1	1	1	578	1/1/2014
2	4	2	358	2/2/2014
3	2	3	402	3/3/2014

Patient Identifiers in CDW

Patient A



- PatientSSN
- PatientICN

VistA System 1 | VistA System 2 | ... | VistA System n

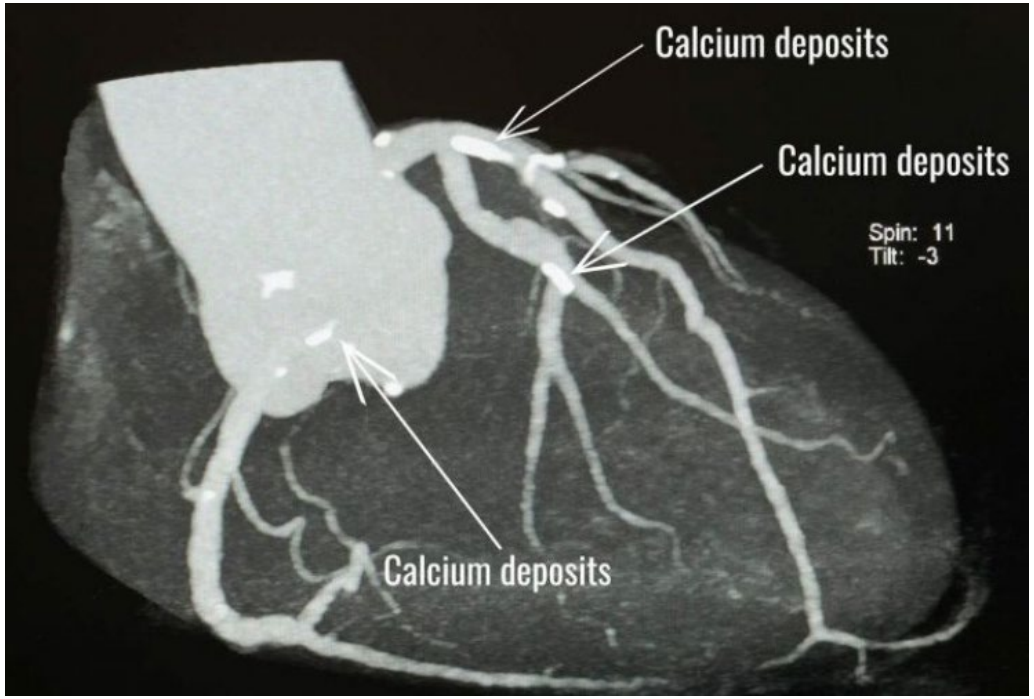
PatientSID 10055888 | PatientSID 10777666 | PatientSID 1229999

Patient A has 1-n PatientSIDs

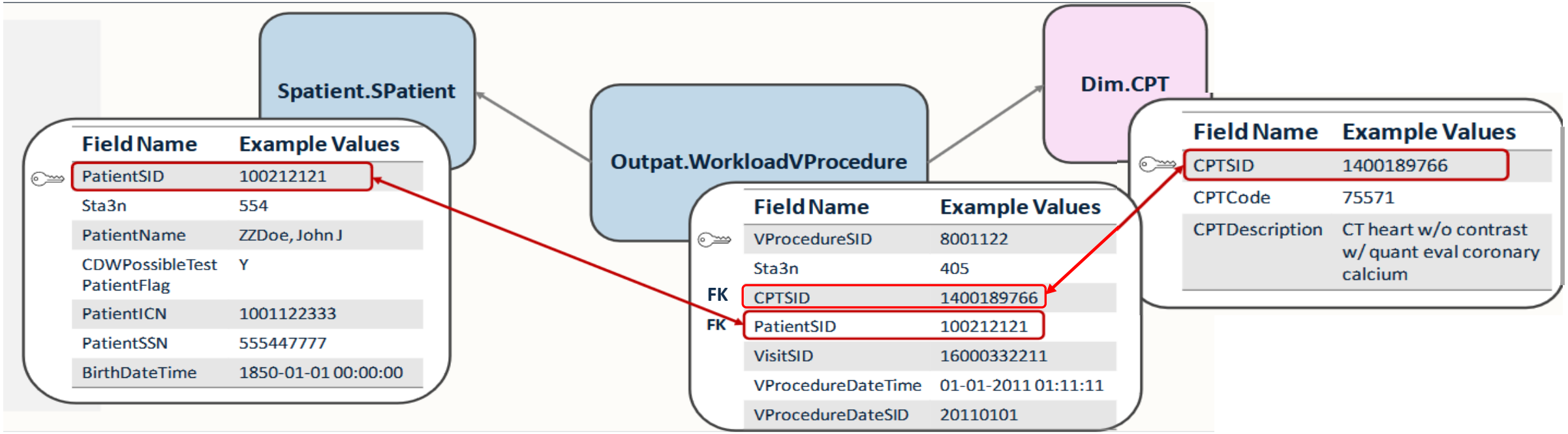
You need *all* of these to completely define all records belonging to Patient A

Example Use of CDW: Investigate VHA use of Cardiac CT Scan for Calcium Scoring

- A cardiac CT scan for calcium scoring is a noninvasive test that uses X-rays to measure calcium deposits in the heart's coronary arteries
- The investigation involves the following:
  - Identify which VA sites perform this test
  - How has use of this test changed over time



Tables Involved



CPT code (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

```
use cdwwork
go
drop table if exists #CPTCodes;
select CPTCode, CPTSID
into #CPTCodes
from Dim.CPT
where CPTCode in ('0144T','0147T','0149T','75571');
```

Make sure to include procedures valid for the period of interest

SQL Code to Join the 3 Tables

```
drop table if exists #CoronaryCalciumProcedures;
select
 b.PatientICN
,a.Sta3n
,a.PatientSID
,a.vProcedureDateTime
,a.vProcedureDateSID
,c.CPTCode
,b.CDWPossibleTestPatientFlag
into #CoronaryCalciumProcedures
from Outpat.WorkloadVProcedure as a
left join Spatient.Spatient as b
on a.PatientSID = b.PatientSID
join #CPTCodes as c
on a.CPTSID = c.CPTSID
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!
```

Make sure Patients are real

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

Further Querying

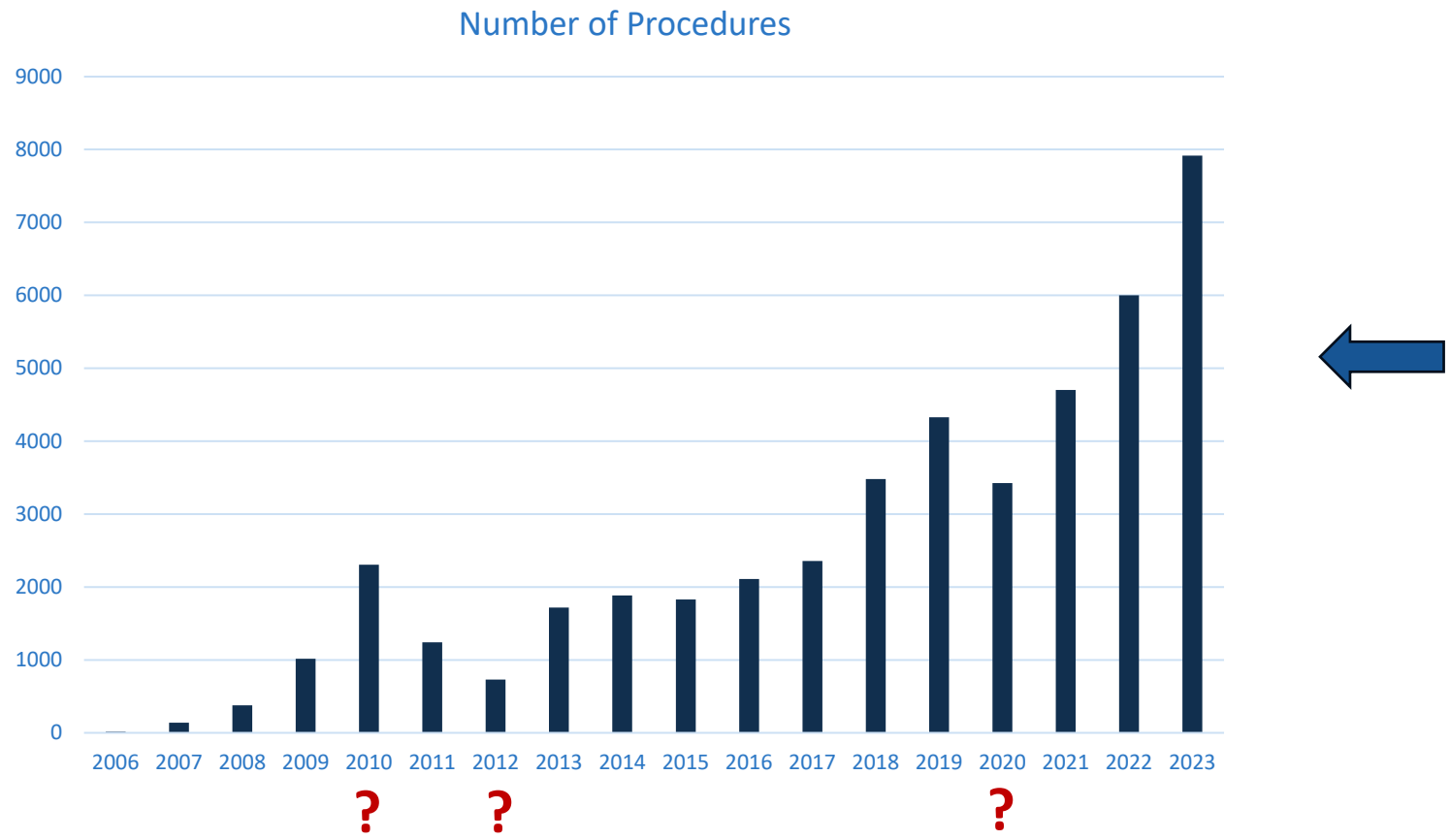
Determine number of Cardiac CT for Calcium Scoring tests performed each Fiscal Year across sites

```
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

Query Results

Number of Cardiac CT for Calcium Scoring tests performed each Fiscal Year across sites



SQL table

Results | Messages

	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

Suggested readings

- MIMIC-IV, a freely accessible electronic health record dataset https://pubmed.ncbi.nlm.nih.gov/36596836/
- Prediction of Intensive Care Unit Length of Stay in the MIMIC-IV Dataset https://www.mdpi.com/2076-3417/13/12/6930
- VHA Corporate Data Warehouse height and weight data: opportunities and challenges for health services research https://pubmed.ncbi.nlm.nih.gov/21141302/
- Building Protein-Protein Interaction Graph Database Using Neo4j https://pubmed.ncbi.nlm.nih.gov/37450167/

THANKS!