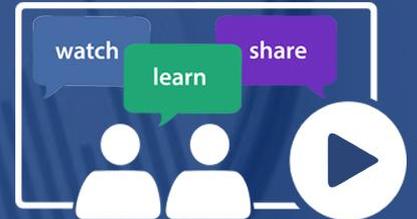


# Effectively Explore Clinical Research Data in Highly Regulated Environments Using 'Data Marts'

June 22, 2016, 1 PM ET / 10 AM PT



WEBiNAR  
SERIES



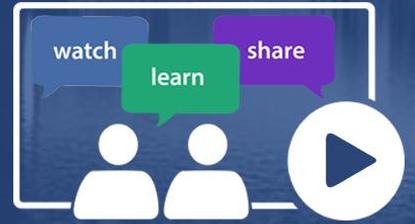
# Multiply the Value of Your Data by Bridging the Clinic and Research Divide

May 11, 2016

Access to the video and slides:  
<http://www.prometheusresearch.com/webinar-bridging-the-clinic-and-research-divide/>



WEBINAR  
SERIES





# PROMETHEUS | RESEARCH

**Mission:** Empower teams to securely harness complex, sensitive data in regulated environments.

**Vision:** We are **inspired** by the many opportunities to improve research productivity through better **sharing** and **reuse** of data

# Roadmap

1. The **Problem** of Plenty
2. Brief **deep dive** into “organizing data”
3. **Conceptual analysis** of the problem, and a solution
4. Pros and cons of **data marts**
5. **Practical** examples
6. **Query interfaces** and their trade-offs

# Part 1: The **Problem** of Plenty

# Previously . . .

We've spent most of our time in the previous webinars talking about how to centralize your research data across data types, studies, and sites

1. Data and Assets from Multiple Sources are Centralized



Data

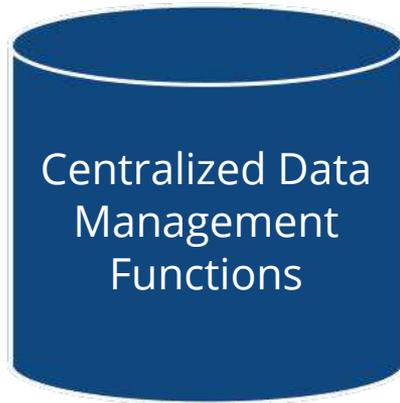


Participants



Biospecimens

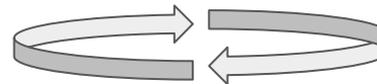
2. Data is optimized for sharing and reuse in a central repository via a central process



3. Data is available for reuse for many purposes



Data is available for many uses



Results can be integrated back into the central system

**Centralizing** and **organizing**  
clinical research data creates  
many opportunities.

# If you've centralized all your research data . . .

- All your data across **many studies** and **many data types** is in one place
- All your research assets (files, biospecimens) are **properly inventoried**
- All your research **operational activities** (visits, events, data entry) are tracked

But it creates **new challenges**:

Tension between optimal ways of organizing data for different uses

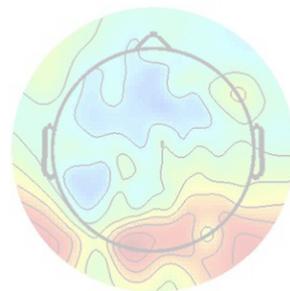
# Querying: Cohort Selection **Example**

## Find:

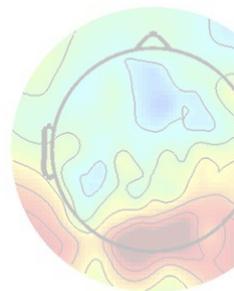
- All females
- W/ an above-cut-off score on depression inventory
- A brain MRI in the last year
- And a deletion at 16p11.2

## Return:

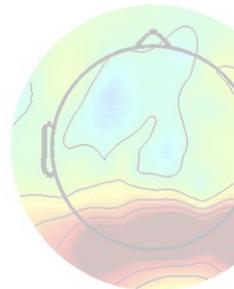
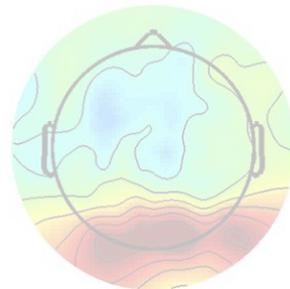
- Basic demographics
- Depression inventory raw scores
- Radiology report on latest brain MRI



2



3.9



# Part 2: Brief **Deep Dive** into “Organizing Data”

# Basic Terminology: **Relational** Databases

**Table: Employees**

Employee_ID	First_Name	Last_Name	Department_ID
10245	Jane	Doe	100
21455	Larry	Barry	100
67982	Tom	Jones	200
12454	Edward	Camel	200

Primary Key:  
Employee\_ID

Attribute/Column:  
First\_Name

Foreign Key:  
Department\_ID

**Table: Departments**

Department_ID	Department
100	Biology
200	Math

Primary Key:  
Department\_ID

# Basic Terminology: **Queries**

## Example of a “**Query**”:

*Who is employed by the Math department?*



## Query Results:

Employee_ID	First_Name	Last_Name	Department_ID	Department
67982	Tom	Jones	200	Math
12454	Edward	Camel	200	Math

## Tables

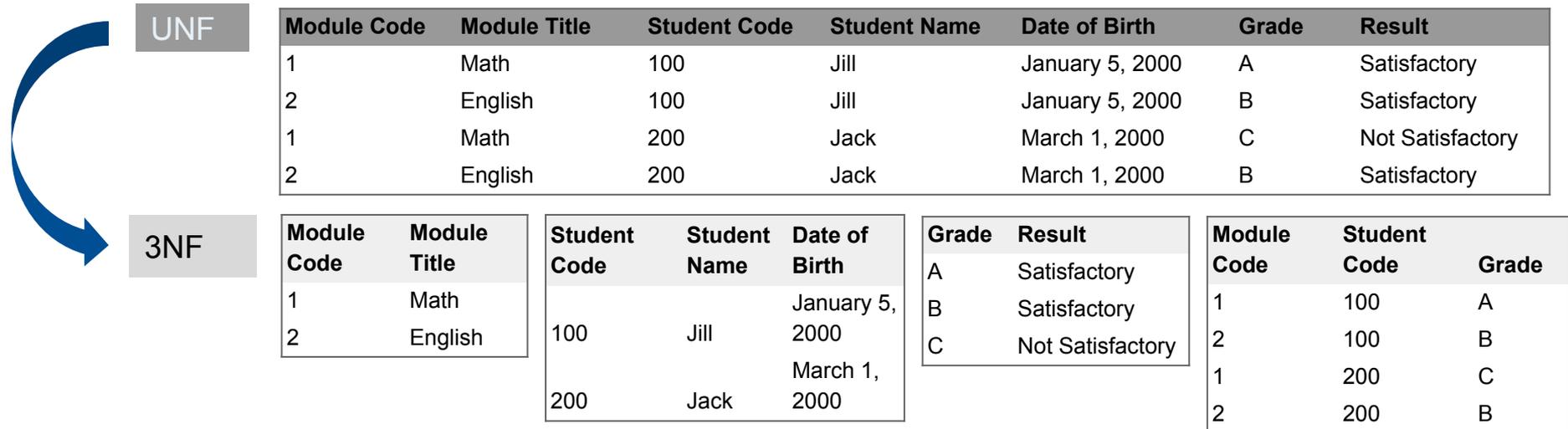
Employees	Departments
Employee_ID	Department_ID
First_Name	Department
Last_Name	
Department_ID	

“Join”

# Basic Terminology: Data **Normalization**

How is data usually organized in relational databases?

**Third Normal Form (3NF)**: describes a state of organization where most redundancy has been removed from the data.



# Basic **Terminology**: Databases, Warehouses, Marts



Part 3: **Conceptual** Analysis  
of the Problem,  
and a Solution

# Why are **transactional databases** hard to search?

- There are **many tables** and **many columns**
- The tables may be organized in a complex way to **optimize for proper storage** of data
- Potentially, there are **many rows**, many choice lists, value sets, etc.
- Privileges for accessing data may be **complex**



The **more kinds** of data you  
organize in a database . . .

the **harder** it becomes to teach data  
consumers how to **query effectively**.

# Example

Epic (EHR) Clarity (data warehouse) has **tens of thousands of tables**

Only an expert in the data model can perform complex searches

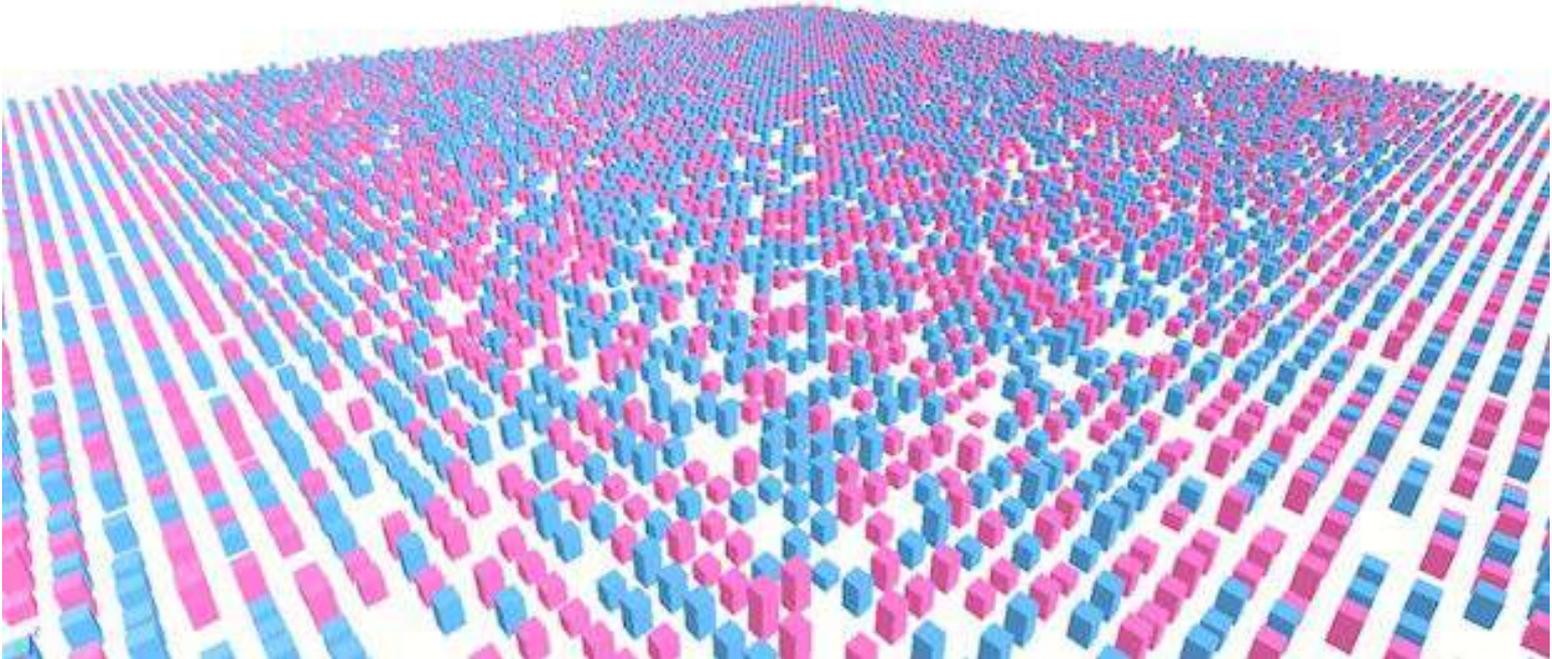


Image Source: <http://lumacode.com/articles/big-data-vr-challenge.html>

There is a **tension** between  
optimal ways of **organizing data**  
**for different uses**



# Data Collection & Information Sharing

# Study Coordination & Tracking To- Dos





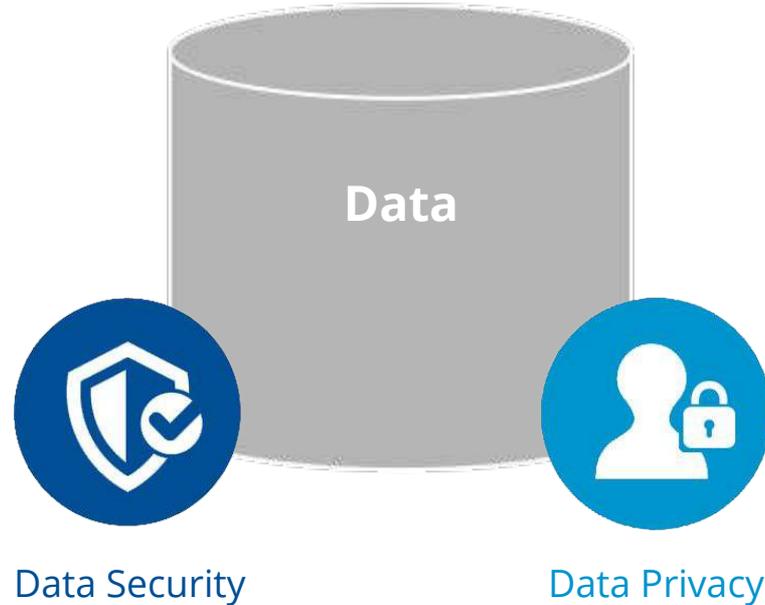
# Data Exploration & Data Analysis

# Data Sharing in a **Regulated** Environment

System security

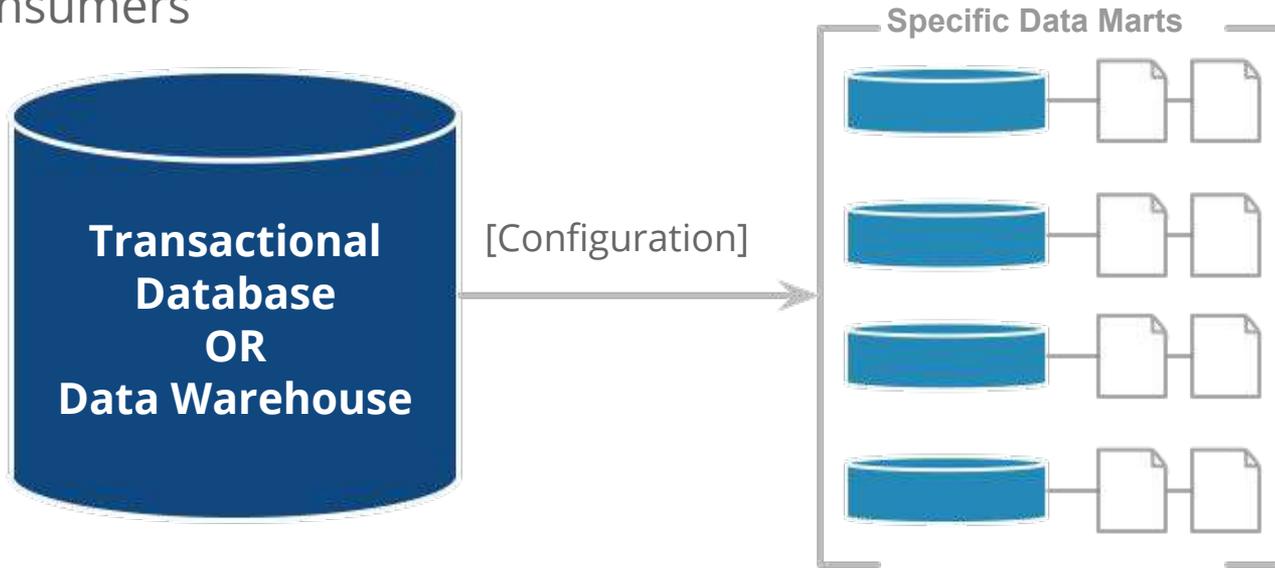
Access privileges

- Granular
- Comprehensive
- Adapted to local practices
- Easy to manage



# Conceptual Solution

Provide a mechanism to selectively reorganize data for different uses by data consumers



**Data managers** can serve as data model experts for the central system

**Data consumers** only need to understand the (simpler) organization of their specific data mart

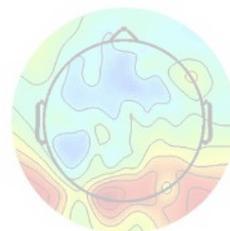
# Back To Our Cohort Selection Example

Find:

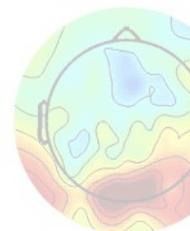
- All females [**Individual.sex='F'**]
- With an above-cut-off score on a depression inventory [**Dep\_Inv1.v21>75**]
- A brain MRI in the last year [**exists(Imaging\_Asset)?type='MRI'**]
- And a deletion at 16p11.2 [**exists(Genomic\_Variant?type='16p11.2')**]

Return:

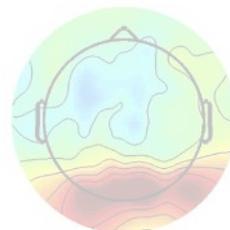
- Basic demographics [**one or many?**]
- Depression inventory raw scores [**one or many?**]
- Radiology report on brain MRI [**one or many?**]



2



3.9



# Part 3: Pros and Cons of **Data Marts**

# Architecture

## PRO

*Reduce **unpredictable load** on transactional system (from ad hoc queries)*



## CON

*Data Mart information isn't instant; need process to **refresh the data***

Refresh

# Security/Privileging

## PRO

*Separate privileges for data marts than for transactional system enable greater flexibility when sharing data (e.g., multi-site networks); better ability to segregate*



## CON

*Need to **manage privileges** of data mart in addition to transactional system privileges*



# Usability

## PRO

*Data can be selectively reorganized to **simplify querying** by a particular class of users for a particular purpose*



## CON

*Data that appears stale may confuse users of the transactional system if inconsistent; need to **manage datamart definitions** as transactional system changes*



Upkeep

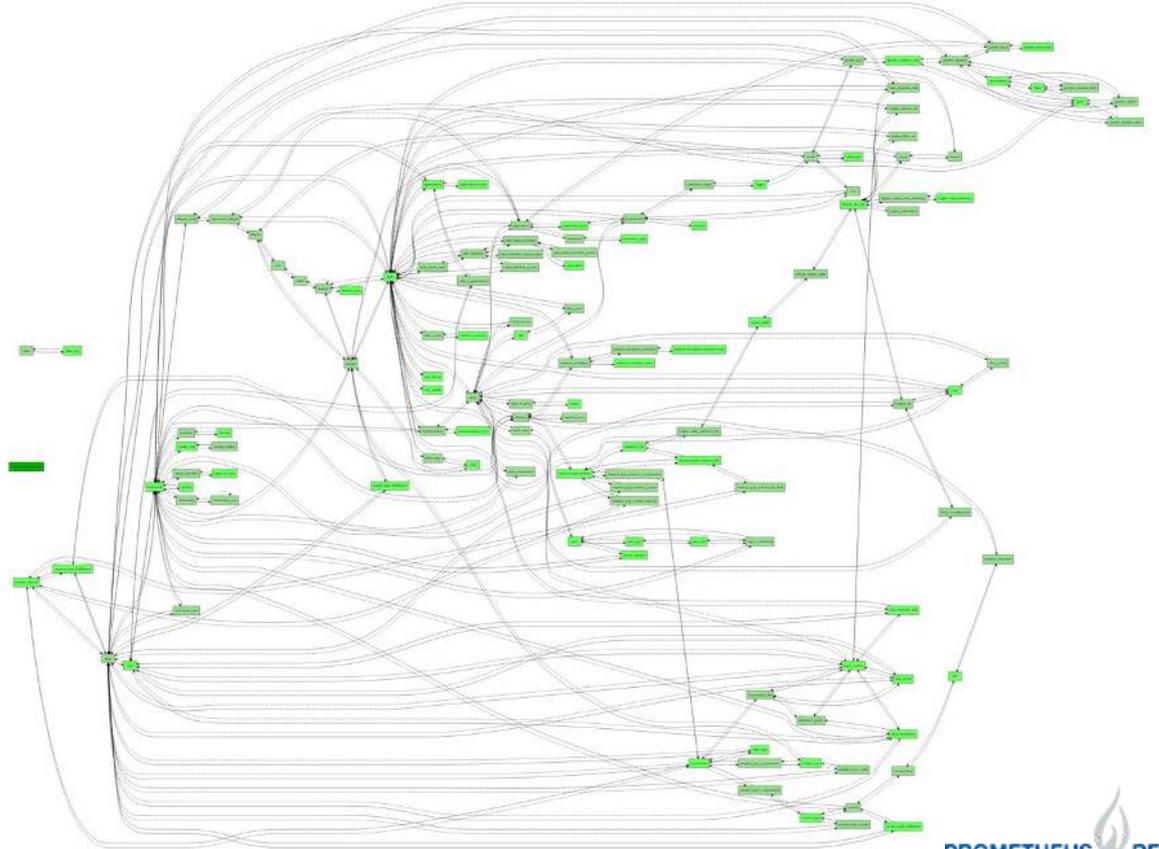
# Part 4: **Practical** Examples

# Example: **Normalized** Clinical Research Data Model

Autogenerated ERD  
from demo system

!! Note the **large number** of tables

!! Note **complex linking** between tables

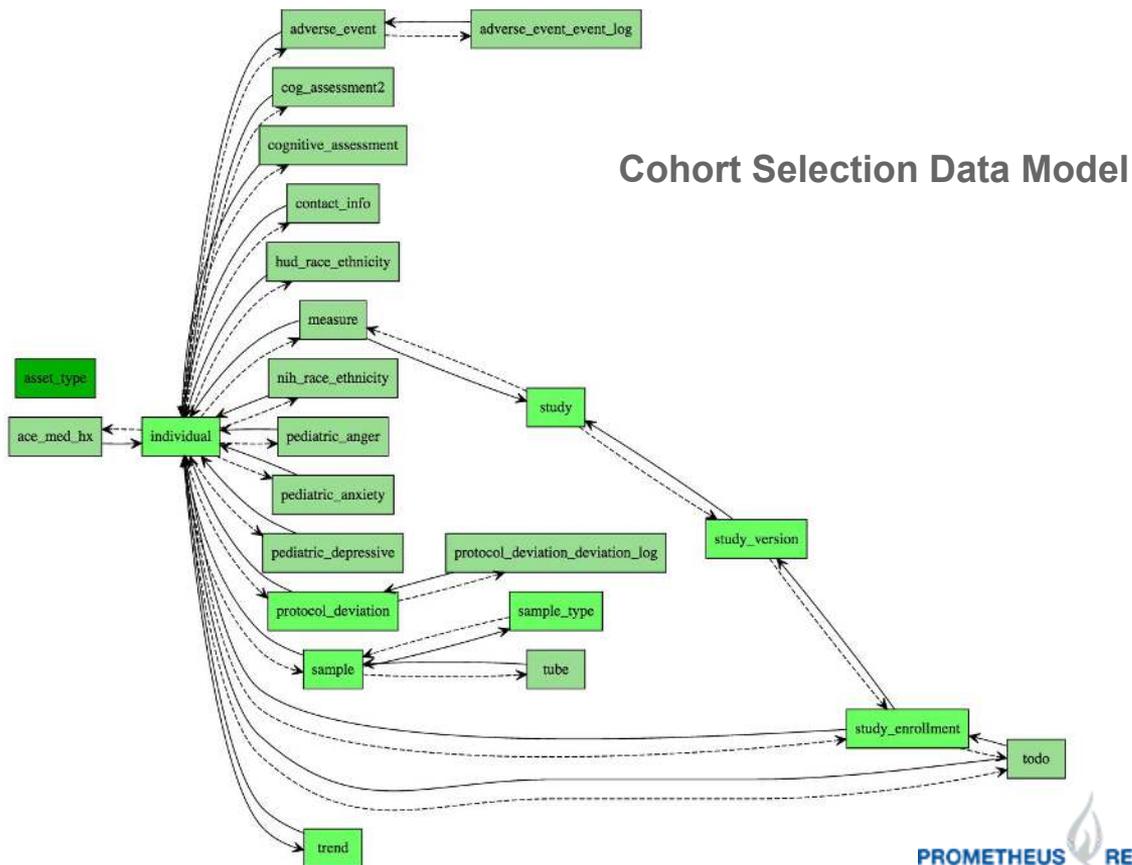


# Data Mart for Cohort Selection

**Need:** When selecting cohorts for recruitment **you care about individuals and their attributes.**

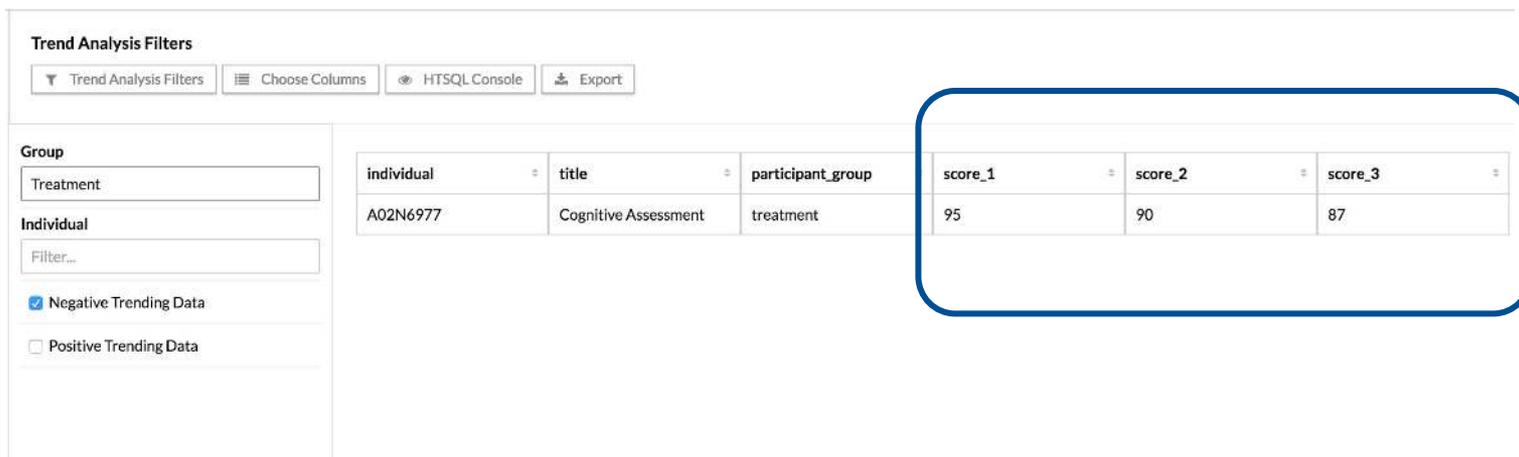
**Challenge:** Having too big a data model can be **distracting.**

Fewer tables, links



# Cohort Selection Data Mart: Search UI

*Question: How did scores in my longitudinal study change over time?*



The screenshot displays a web interface for trend analysis. On the left, there are filter options for 'Group' (set to 'Treatment') and 'Individual' (with a 'Filter...' input). Below these are checkboxes for 'Negative Trending Data' (checked) and 'Positive Trending Data' (unchecked). At the top, there are buttons for 'Trend Analysis Filters', 'Choose Columns', 'HTSQL Console', and 'Export'. The main area shows a table with the following data:

individual	title	participant_group	score_1	score_2	score_3
A02N6977	Cognitive Assessment	treatment	95	90	87

**Take-away:** Data Marts can also utilize **configurable guide interfaces** that make it even easier and faster to get the answers you need from your data.

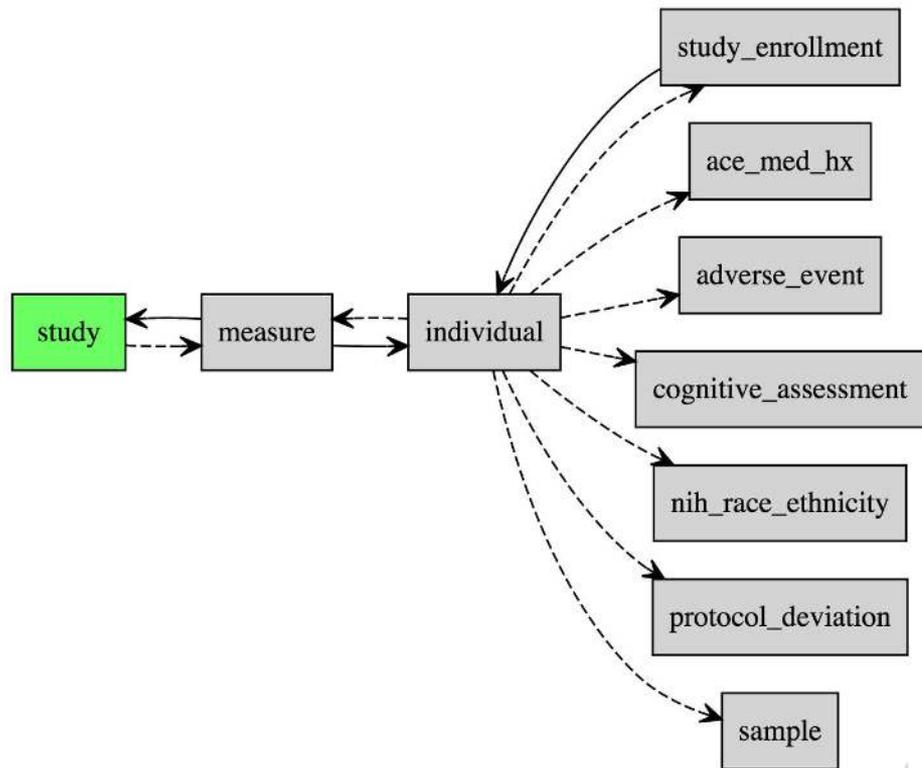
# Study-Specific Data Mart for Multiple Sites

**Common Use Case:** Limiting data to a subset to satisfy a permission requirement or improve usability.

**Example:** research network has 5 sites that need to combine data. Stats core can't see PHI.

**Solution:** Study-specific data mart.

*Data marts allow you to set up data access privileges finely tuned to data sharing needs.*

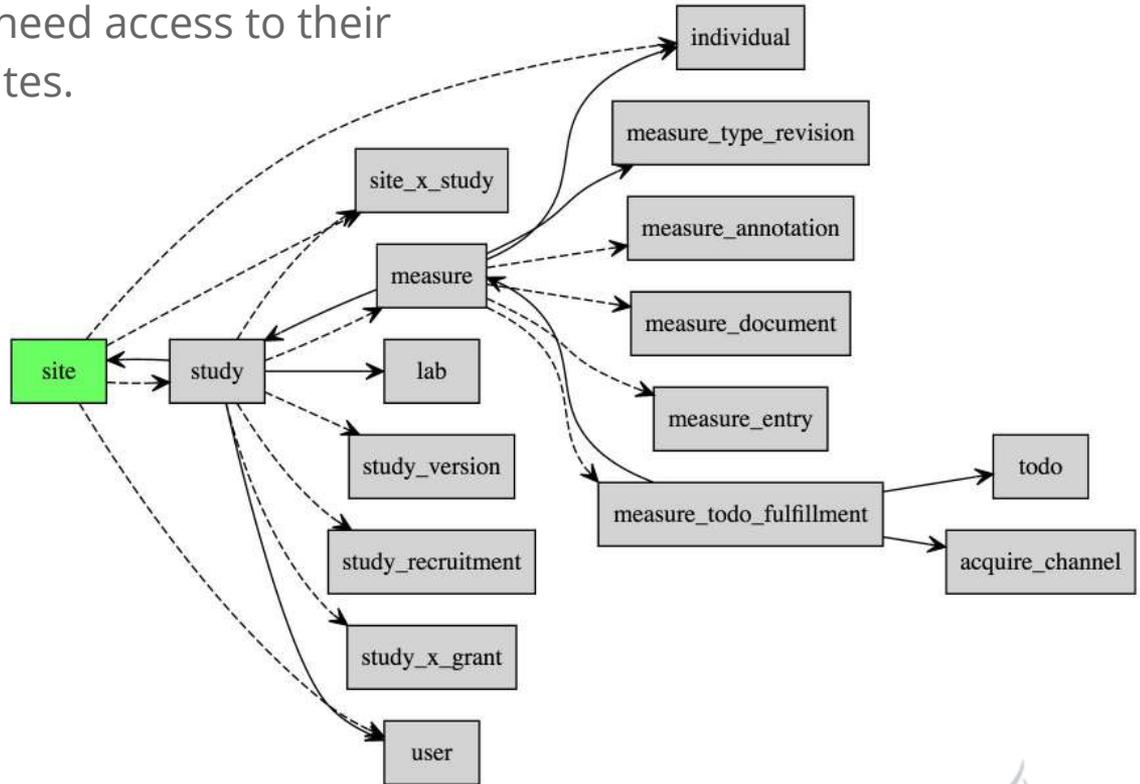


# Data Mart for **Data Quality**

**Need:** Complex privileging. Sites need access to their own data, but not data of other sites.

**Need:** Data Quality teams needs access to data from all the sites, but only to a subset of tables.

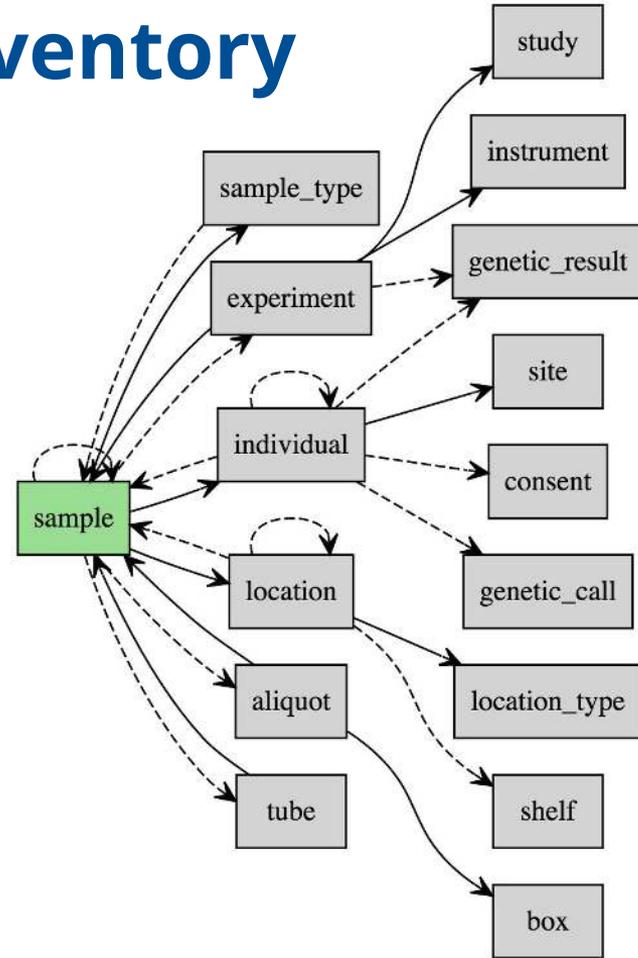
**Solution:** The latter requirement can be met with a **data quality data mart** that includes research operations and measure annotation data.



# Data Mart for **Sample Inventory**

Sample inventory data can be made available to potential consumers in a controlled way, **w/o access to phenotype data or PHI.**

*Note that “Sample” is the “top level” table in this model, making it easier for a data consumer to **search for samples by a wide variety of criteria.***

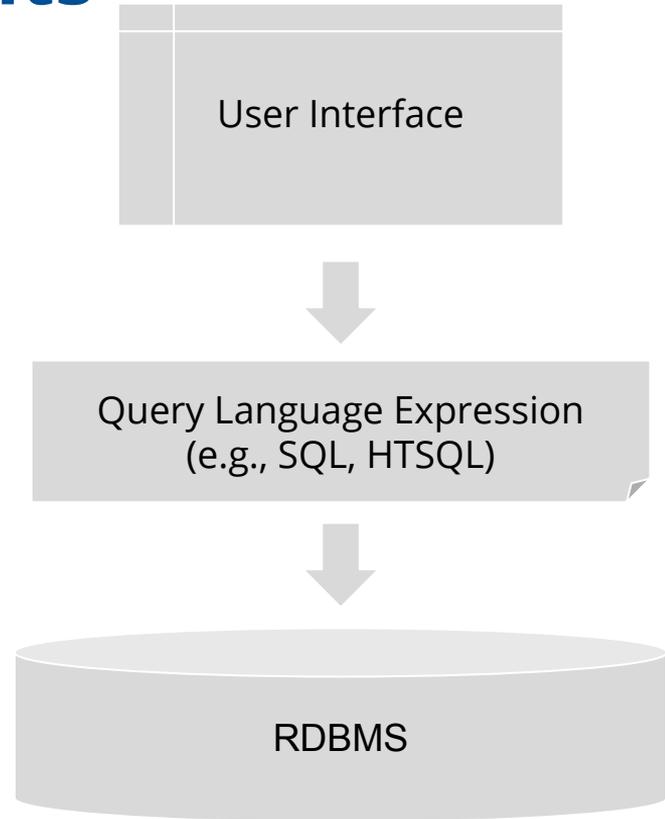


# Part 5: **Query Interfaces** & Their Tradeoffs

# Query Interface **Constraints**

The user interface must emit a query expression in a language that the **Relational Database Management System (RDBMS)** can process.

Thus, any Graphical User Interface (GUI) for building queries must ultimately **generate a query expression**.



Query UIs tend to be either  
**expressive** or **easy to learn**,  
but not both

# Expressiveness vs Learnability in Query Builders



Google Search

I'm Feeling Lucky

Hide complexity, but lose expressiveness



<b>Find results</b>	with <b>all of the words</b>	<input type="text"/>	10 results <input type="text"/>
	with the <b>exact phrase</b>	<input type="text"/>	<input type="button" value="Google Search"/>
	with <b>at least one</b> of the words	<input type="text"/>	
	<b>without</b> the words	<input type="text"/>	
<b>Language</b>	Return pages written in	<input type="text" value="any language"/>	
<b>Region</b>	Search pages located in:	<input type="text" value="any region"/>	
<b>File Format</b>	<input type="text" value="Only"/> return results of the file format	<input type="text" value="any format"/>	
<b>Date</b>	Return web pages first seen in the	<input type="text" value="anytime"/>	
<b>Numeric Range</b>	Return web pages containing numbers between	<input type="text"/> and <input type="text"/>	
<b>Occurrences</b>	Return results where my terms occur	<input type="text" value="anywhere in the page"/>	
<b>Domain</b>	<input type="text" value="Only"/> return results from the site or domain	<input type="text"/>	e.g. google.com, .org <a href="#">More info</a>
<b>Usage Rights</b>	Return results that are	<input type="text" value="not filtered by license"/>	<a href="#">More info</a>
<b>SafeSearch</b>	<input checked="" type="radio"/> No filtering <input type="radio"/> Filter using <a href="#">SafeSearch</a>		

## Page-Specific Search

<b>Similar</b>	Find pages similar to the page	<input type="text"/>	<input type="button" value="Search"/>
		e.g. www.google.com/help.html	
<b>Links</b>	Find pages that link to the page	<input type="text"/>	<input type="button" value="Search"/>

More complex, more expressive

# Expressiveness vs Learnability in Query Builders

Reset

Search

```
SELECT reference, dtpropid, surveyid, release_date, start_date, date_obs, dtpi, ra, dec, telescope, instrument, filter, exposure, obstype, obsmode, proctype, prodtype, seeing, depth, dtacqnam, reference AS archive_file, filesize, md5sum FROM voi.siap ORDER BY date_obs ASC LIMIT 50000
```

## Examples of valid SQL queries

- This selects the first 100 rows from the voi.siap table within a box bounded by ra (10.352, 11.017) and dec (41.019, 41.519). Note that position constraints must be in the same units (here, degrees). Tip: LIMIT 100 or some other small number is useful for testing without getting a very large data set back.

```
SELECT * FROM voi.siap WHERE ((ra >= 10.352 AND ra <= 11.017) AND (dec >= 41.019 AND dec <= 41.519)) LIMIT 100
```

- This selects the first 100 rows from the voi.siap table for a specified NOAO proposal ID (here, noao) using a wildcard match. A wildcard may be specified by constructing a condition using the operator "LIKE" or "ILIKE" (the later is case insensitive) followed by a string which may contain one or more "%" characters which are the wildcards.

```
SELECT * FROM voi.siap WHERE dtpropid ILIKE '%noao%' LIMIT 100
```

- This selects the first 50000 rows from the voi.siap table at a specified proposal ID (here, 2016A) using a wildcard match. Tip: This kind of condition can be useful to find proposals where you are a co-investigator.

```
SELECT * FROM voi.siap WHERE dtpropid ILIKE '%2016A%' LIMIT 50000
```

- This selects the first 50000 rows from the voi.siap table within a range of dates (2015-08-17 and 2015-10-17) at the start of an observing night. Tip: You can specify which columns you want in the result set.

```
SELECT reference, release_date, start_date, filesize, dtpropid, md5sum, date_obs, ra, dec, telescope, instrument FROM voi.siap WHERE start_date BETWEEN '2015-08-17' AND '2015-10-17' LIMIT 50000
```

- This selects the first 50000 rows from the voi.siap table for a specified telescopes (kp4m, kp09m and ct4m) and instruments (mosaic and mosaic\_2). Note that this query is using a wildcard match for some instruments.

```
SELECT * FROM voi.siap WHERE (telescope = 'kp4m' AND instrument LIKE 'mosaic%') OR (telescope = 'kp09m' AND instrument LIKE 'mosaic%') OR (telescope = 'ct4m' AND instrument = 'mosaic_2') LIMIT 50000
```

- This selects the first 50000 rows from the voi.siap table with release date less than some date (here, 2016-06-17). Tip: For getting public files, you have to select a date less than today.

```
SELECT * FROM voi.siap WHERE release_date < '2016-06-17' LIMIT 50000
```

- This selects the first 50000 rows from the voi.siap table at a specified user (here, dummyuser). Tip: If you are adding a noao\_id condition, then make sure that the noao\_id column is present in the SELECT clause.

```
SELECT reference, release_date, start_date, filesize, dtpropid, md5sum, noao_id FROM voi.siap WHERE noao_id = 'dummyuser' LIMIT 50000
```

Expressive, but hard to learn

# Expressive, looks easier to learn, but . . .

The screenshot shows a PostgreSQL query builder interface. The main window is titled "Query - pgbench on postgres@localhost:5432". The interface includes a menu bar (File, Edit, Query, Favorites, Macros, View, Help), a toolbar, and a main workspace divided into several panes.

The main workspace is divided into three panes:

- SQL Editor / Graphical Query Builder:** This pane shows a graphical query plan. It consists of three tables: **tellers**, **accounts**, and **branches**. The **tellers** table is connected to the **accounts** table, which is in turn connected to the **branches** table. The **tellers** table has columns **tid**, **bid**, **tbalance**, and **filler**. The **accounts** table has columns **aid**, **bid**, **abalance**, and **filler**. The **branches** table has columns **bid**, **bbalance**, and **filler**.
- Columns:** This pane shows a table with columns **Relation**, **Column**, and **Alias**. It lists the columns selected in the query:

	Relation	Column	Alias
1	tellers	tbalance	Teller Balance
2	tellers	bid	
3	tellers	tid	
4	tellers	filler	
5	accounts	abalance	
6	accounts	bid	

The **Output pane** is currently empty, with tabs for **Data Output**, **Explain**, **Messages**, and **History**.

The status bar at the bottom shows "ready", "Unix", and "Ln 1 Col 1 Ch 1".

# Solution Space

Hard to Learn

Query Language  
Syntax

Graphical  
Query Builder



Low  
Expressiveness

High  
Expressiveness

Query Form



Canned Query



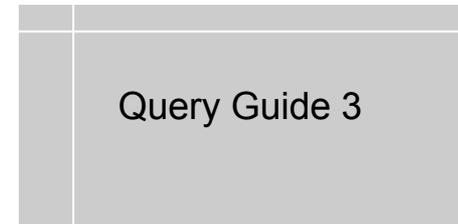
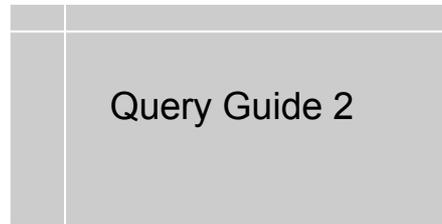
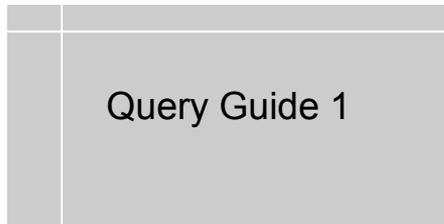
Configurable  
Query Guides

The screenshot shows a 'Configurable Query Guide' interface. It has a search bar at the top with '10 results' and a 'Search' button. Below the search bar, there's a table with columns 'Table', 'Columns', and 'Filter'. The table contains several rows of data, including 'Table', 'Join', 'Filter', and 'Aggregate'.

Easy to Learn

# Better **learnability** through specialization of expertise

(3) A **data consumer** only needs to **learn** the query guides.



(2) A **power user**, with understanding of the (simplified) data mart data model, **configures** the query guides.

(1) A **data manager**, who understands the overall data model, **configures** the data mart.



# Example of a Query Guide

**RexStudy**  
This is a demonstration instance and does not contain PHI.

Home Individuals Data Entry Studies **Query** Manage

Demo RexMart > Choose a Type of Mart to Explore > Choose a Mart > Medical History > Medical History Filters > Choose Columns > **Medical History Filters**

**Medical History Filters**

Choose Columns HTSQL Console Export

**Medical History Filters**

Choose Columns HTSQL Console Export

**Gender**

male  
 female

**Age (in days)**

365 Max

Developmental Delay

Has Autism

Hearing Loss

Verbal

**School Performance**

mixed  
 above-grade-level  
 at-grade-level  
 below-grade-level

individual	instrument_ve	date_of_evalu	age_at_evalua	sex	r2_developme
A18V6861	ace-med-hx.1	2011-06-01	827	male	true
A18V6861	ace-med-hx.1	2012-06-01	839	male	true
A18V6861	ace-med-hx.1	2013-06-01	851	male	true
A40G5231	ace-med-hx.1	2011-06-01	935	male	false
A40G5231	ace-med-hx.1	2012-06-01	947	male	false
A40G5231	ace-med-hx.1	2013-06-01	959	male	false

Additional Resource: <http://tinyurl.com/RexMart-ConfigureDataMarts>

# Technology Take-Aways: **3 Things to Look For**

*If your clinical research data is hard to to query, consider thinking in terms of specialized analytic data marts*

Three technical capabilities you should look for in your informatics systems:

1. Configurable **data marts**: provide ability for research staff or data managers to define and create data marts quickly and at low cost
2. Configurable **query interfaces** (guides): allow power users or data managers to simplify searches for other users
3. Configurable **access privileges** for data marts: allow data managers to meet complex data sharing needs in a regulated environment

# Summary

- **Tension** between optimal ways of **organizing data for different uses** complicates data use; organizing data in marts manages that tension and makes **querying** easier for data consumers
- Regulated env. imposes high burden of **granular access privileging**; Marts that limit access to specific users or groups **improve security** and simplify data sharing.
- **Configurable query guide** interfaces further improve learnability
- Marts insulate central databases from ad hoc querying load, improving overall **system stability and performance**

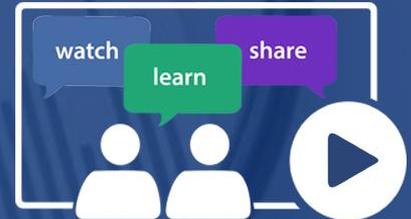
# Leveraging RIOS: an NIH-Funded Standard for REDCap, Qualtrics, and RexDB Form Exchange

July 13, 2016, 1 PM ET / 10 AM PT

Register: <http://www.prometheusresearch.com/webinar-RIOS/>



WEBiNAR  
SERIES



# Thank you

Leon Rozenblit, JD, PhD

+1 203 672 5810

LeonR@prometheusresearch.com

Frank Farach, PhD

+1 203 672 5885

FrankF@prometheusresearch.com



**PROMETHEUS | RESEARCH**

*Integrating data for extraordinary outcomes*

[prometheusresearch.com](http://prometheusresearch.com)

# Q&A

# Resources & Next Steps

## Attend our next webinar

- [Reserve your spot](#) for our next webinar: *“Leveraging RIOS, an NIH-funded universal standard to facilitate the exchange of REDCap, Qualtrics, and RexDB-based form configuration”*

## Read our Series on Good Data Management Practices

- Good Data Management Practices for Data Analysis
  - [Part 1: Data Cleaning & Data Transformation](#)
  - [Part 2: Tidy Data](#)
  - [Part 3: Relational Databases](#)

## Contact us with questions about clinical research informatics

- Frank Farach ([FrankF@PrometheusResearch.com](mailto:FrankF@PrometheusResearch.com))
- Julie Hawthorne ([Julie@PrometheusResearch.com](mailto:Julie@PrometheusResearch.com))

# Thank you

Leon Rozenblit, JD, PhD

+1 203 672 5810

LeonR@prometheusresearch.com

Frank Farach, PhD

+1 203 672 5885

FrankF@prometheusresearch.com



**PROMETHEUS | RESEARCH**

*Integrating data for extraordinary outcomes*

[prometheusresearch.com](http://prometheusresearch.com)