CSI TriSano™ 2.0 Documentation

AVR Quick Start Guide

Copyright © 2009 • Collaborative Software Initiative, Inc.

TriSano is a trademark of Collaborative Software Initiative, Inc.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval sys- tem, or translated into any language in any form by any means without the express written per- mission of Collaborative Software Initiative, Inc.

This material is presented for informational purposes only and is subject to change without notice. Collaborative Software Initiative, Inc. assumes no responsibility for errors that may appear in these instructions.

Contents

1. AVR Quick Start Guide Preface 3
2. Introduction to AVR in CSI TriSano™ 4
3. Introduction to the Pentaho™ User Console 8
4. Pentaho™ Ad Hoc Query Tool 11
    4.1 Ad Hoc Query Tool Completed Report 12
    4.2 Accessing the Ad Hoc Query Tool 13
    4.3 Creating a New Report 16
5. Pentaho™ Report Designer Tool 55
    5.1 Start the Report Designer 56
    5.2 Start the Report Design Wizard 58
    5.3 Data Source and Query 60
    5.4 Query Designer 61
    5.5 Using the Report Designer 90
    5.6 Pentaho™ Report Designer Tool Summary 110
6. Pentaho™ Analysis - OLAP Tool 111
    6.1 Starting an OLAP Session 112
    6.2 Showing and Hiding Measures and Dimensions 114
    6.3 Adjusting Rows and Columns 118
    6.4 Filtering 120
    6.5 Multi-Level Dimension Hierarchies 123
    6.6 Miscellaneous Toolbar Options 124
7. CSI TriSano™ Data Warehouse Data Dictionary 126

AVR Quick Start Guide Preface

How to use this Guide

Purpose of this Guide

The *CSI TriSano™ AVR Quick Start Guide* provides information on how to use the analysis, visualization and reporting (AVR) features in CSI TriSano™. This guide helps you to:

- Understand the components of CSI TriSano™ AVR, including
  - Pentaho™ Ad Hoc Query Tool
  - Pentaho™ Report Designer
  - Pentaho™ Analysis Services (OLAP)
  - The structure of the data warehouse

Audience

This guide is intended for those responsible for creating and maintaining reports, charts and analysis from the CSI TriSano™ data warehouse using the Pentaho™ set of tools. It assumes that you have an understanding of CSI TriSano™. For more information on the features of CSI TriSano™ refer to the CSI TriSano™ Administrator Guide, CSI TriSano™ User Guide, CSI TriSano™ Form Builder Guide, as well as the online help.

Assumptions

None

Note

All the infomation shown in the guide is completely fictictious.

Formatting Conventions

This guide uses the following conventions:

- **Bold** text indicates user action.
  - For example: Type **0**, then press **Enter** for each of the remaining fields.
- *Italic* text indicates a new word or a word that requires some emphasis.
  - For example: Before analyzing, *always* prepare fresh matrix.

Next Topic:

Introduction to AVR in CSI TriSano™

Copyright © 2009 Collaborative Software Initiative, Inc.

Introduction to AVR in CSI TriSano™

CSI TriSano™ Data Warehouse for Analysis, Visualization and Reporting (AVR)

Overview of CSI TriSano™

CSI TriSano™ is a citizen-centric surveillance and outbreak management system for infectious disease, environmental hazards, and bioterrorism attacks. It allows local, state and federal health advocates to track, control and ultimately prevent illness and death. It is the next generation of surveillance and outbreak management software.

CSI TriSano™ provides extensive surveillance capabilities including:

- Case management, including capture of demographic, clinical, laboratory, epidemiological and other information, such as encounters and tasksTasks
- Contact management
- Environmentals (place exposure)
- Electronic Lab Reporting
- Address lookup, geocoding and mapping
- Form Builder for disease-specific information
- Separate Data Warehouse database
- Tools for for Analysis and Visualization Reporting

The CSI TriSano™ Data Warehouse

The CSI TriSano™ Data Warehouse provides ad hoc query, reporting and online analytical processing (OLAP) capabilities against information extracted from the CSI TriSano™ transaction system. Data is extracted at user-defined intervals, transformed for ease of access, and loaded into a separate CSI TriSano™ data warehouse database. After loading, one or more business models for the data warehouse are defined for ease of use. The number of business models and content of each is controlled through the definition of AVR Groups. See the [CSI TriSano™ Administrator Guide|Administrator Guide] for information on AVR Groups.

In addition to supporting CSI TriSano™ core data, the data warehouse also fully supports data captured through disease-specific forms created with the CSI TriSano™ Form Builder and access to this data using all the tools discussed below.

Analysis, Visualization and Reporting (AVR)

CSI TriSano™ includes the following advanced Analysis, Visualization and Reporting features through the embedding of Pentaho™ Enterprise Edition and the creation of Pentaho™ business models for the extracted transaction data. The business models provides a pre-configured metadata overlay so that the Pentaho™ Business Intelligence platform can utilize the TriSano™ Data Warehouse.

Web-based Reporting using the Pentaho™ Ad Hoc query tool

The web-based Ad Hoc query and reporting tool can be used by casual and power users. It allows you to create reports using selection fields in friendly business terms provided by a pre-defined business model. Users create reports with an interactive "drag and drop" interface that supports grouping, summarizing, filtering and formatting of the data. The reports created can be exported in HTML, PDF, Excel and CSV formats. The user can save the reports locally or in the Pentaho™ Solution Repository, later run the saved reports in the background, or schedule them to run in the future. Reports created using the Ad Hoc query tool can be "promoted" to the Report Designer for fine-tuning.

Here is an example of a report created using the Pentaho™ Ad Hoc query tool that displays information, including disease-specific data from a form, for Influenza A, Novel Virus.

Copyright © 2009 Collaborative Software Initiative, Inc.

Influenza A, Novel Virus Report on Hospitalization

Disease name: Influenza A, Novel Virus

Last name	First name	Birth date	Date disease onset	Disease event hospitalized	fever	high temp	icu
Black	Dave	Apr 06, 1987	Aug 03, 2009	Yes	Yes	100.5	Yes
Chapman	Penny	Aug 04, 1980	May 01, 2009	No	Yes	102	-
Craig	Larry	Jul 29, 1948	May 06, 2009	No	Yes	102.5	-
Evans	Betty	Aug 23, 1962	Mar 03, 2009	Unknown	No	-	-
Evans	Paul	Jan 12, 2000	May 01, 2009	Yes	Unknown	-	Yes
Hendricks	Mitchell	Oct 06, 1937	Apr 04, 2009	Yes	Unknown	-	-
Jones	Bradley	Jul 21, 1993	Jul 02, 2009	Yes	Yes	101.5	No
Nelson	Grant	May 16, 1996	Jun 06, 2009	Yes	Yes	105	Yes
Phillips	Danielle	Mar 09, 1953	Jul 05, 2009	No	No	-	-
Washington	Wayne	Dec 21, 1972	May 04, 2009	Yes	Yes	103	Yes
Wilson	Jim	May 05, 1965	Jul 17, 2009	Yes	Yes	101	Yes
Wilson	Kevin	Jul 23, 1970	Jun 02, 2009	Yes	Yes	104	No
Yoder	Jeff	Jul 02, 1955	Jul 27, 2009	No	No	-	-
Young	Ashley	May 03, 1994	May 04, 2009	Yes	-	-	Yes

State Health Department

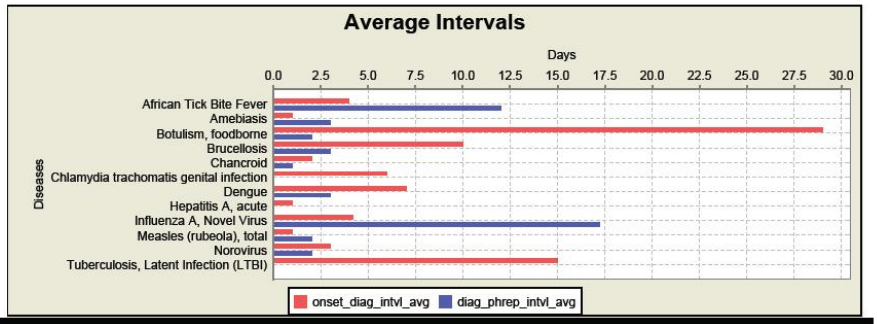
Rich reporting capability to produce reports, charts and graphs using the Pentaho™Report Designer

The Pentaho™ Report Designer is an advanced reporting tool. With it, you can create highly detailed "pixel perfect" reports using data stored in the CSI TriSano™ data warehouse. It features a drag-and-drop graphical interface with full control of data access, layout, grouping, calculations, charting and formatting. An integated, step-by-step reporting wizard guides you through the design process. It also includes report templates to accelerate report creation and provide a consistent look and feel. After creation, the reports can be uploaded to a server, shared by others and run from a web browser.

Here is an example of a report and chart created using the Pentaho™ Report Designer showing average intervals in days between disease onset, disease diagnosis and reporting to public health.

CSI TriSano 2.0 Sample Report — August 09, 2009 @ 07:38

Average Date Intervals by Disease



Disease	Interval: Onset to Diagnose	Interval: Diagnose to PH Reporting
African Tick Bite Fever	4	12
Amebiasis	1	3
Botulism, foodborne	29	2
Brucellosis	10	3
Chancroid	2	1
Chlamydia trachomatis genital infection	6	0
Dengue	7	3
Hepatitis A, acute	1	0
Influenza A, Novel Virus	4.214	17.214
Measles (rubeola), total	1	2
Norovirus	3	2
Tuberculosis, Latent Infection (LTBI)	15	0

Property of State Health Department

Data Cube/OLAP support through the Pentaho™ Analysis Services

The Date Cube/OLAP capability in CSI TriSano™ allows the user to work with large amounts of data in real-time using multi-dimensional analysis. The Pentaho™ Analysis Services makes it easy for public health professionals to explore disease surveillance data by interactively drilling into and cross-tabulating data.

CSI TriSano™ provides two pre-configured OLAP cubes, one for morbidity events and one for contact events. Using the OLAP capabilities provided, you can easily view summarized data across multiple dimensions, drill through to the underlying data, and display your results in reports and graphs. This OLAP capability also

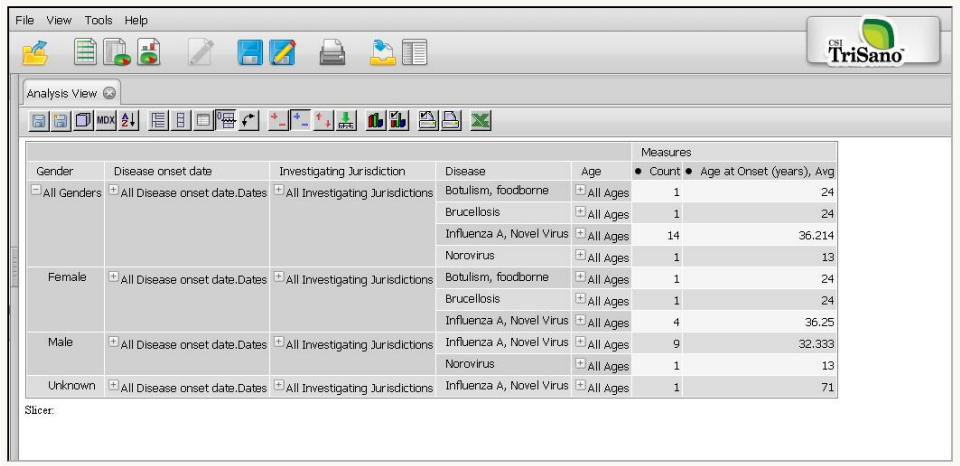
Copyright © 2009 Collaborative Software Initiative, Inc.

integrates with the Report Designer discussed above.

Some of the capabilities provided in the OLAP support include:

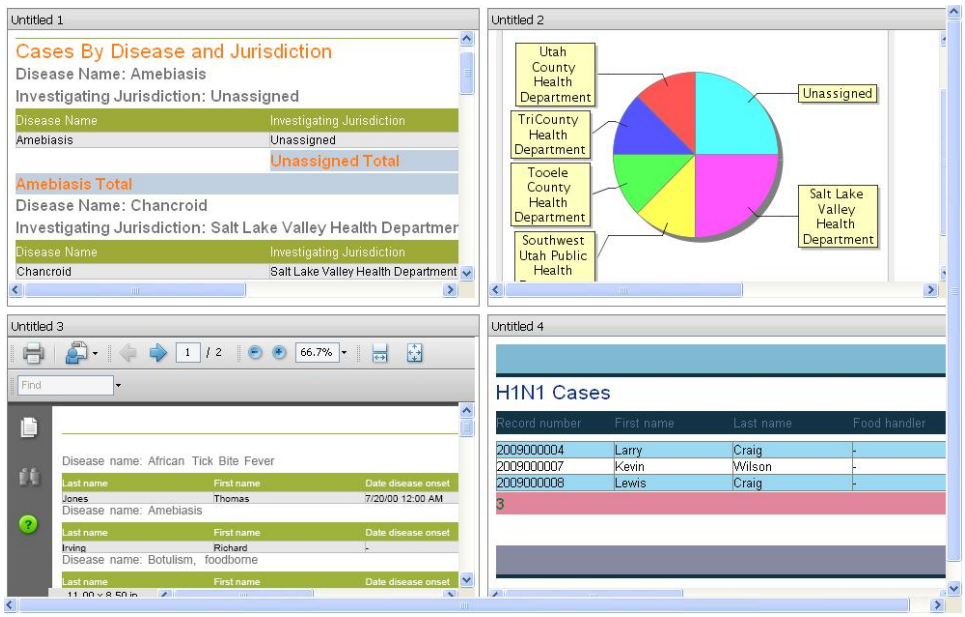
- Showing Measures and Dimensions
- Hiding Measures and Dimensions
- Moving a Measure to a Dimension or a Dimension to a Measure
- Choosing additional measures to display including Population and Rate measures
- Expanding and Collapsing the cube
- Reordering items along the row axis or column axis.
- Filtering the data to be analyzed, for example, by picking a subset of jurisdictions
- Print the results in PDF format
- Chart the results

The illustration below shows an example of using the OLAP capability in Pentaho™ for several diseases showing the count of the diseases and the average age of onset by gender.



Dashboard

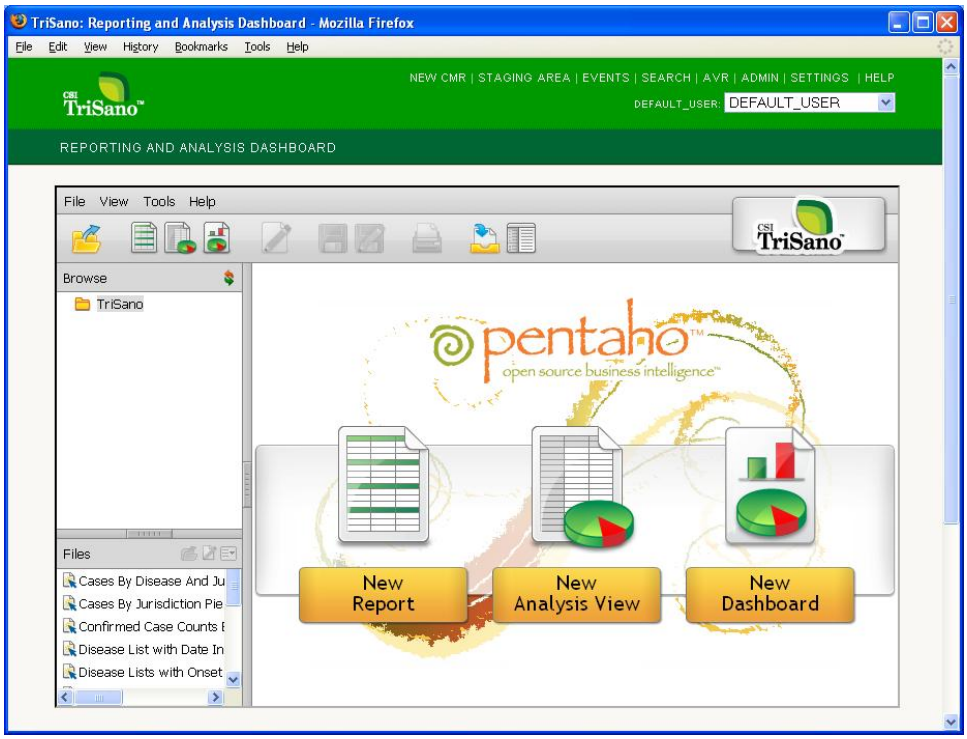
The CSI TriSano™ AVR feature includes the capability of creating a dashboard that can display key reports and charts automatically when you login to the system. With this, a public health official using CSI TriSano™ can quickly see the status of case, disease, jurisdiction or any other information. The dashboard is fully integrated with Pentaho™ Reporting and Pentaho™ Analysis so that users can drill into underlying reports and analysis to understand what factors are contributing to the current situation.



Summary

Copyright © 2009 Collaborative Software Initiative, Inc.

The CSI TriSano™ Advanced AVR capability helps your organization turn data into information to allow you to make the most well-informed decisions possible. It extends the power of customization by making all the disease-specific form data available for analysis using any of the three tools provided in CSI TriSano™.



The next sections of this guide explore the Ad Hoc Query tool, the Report Designer and the OLAP capability.

Next Topic:

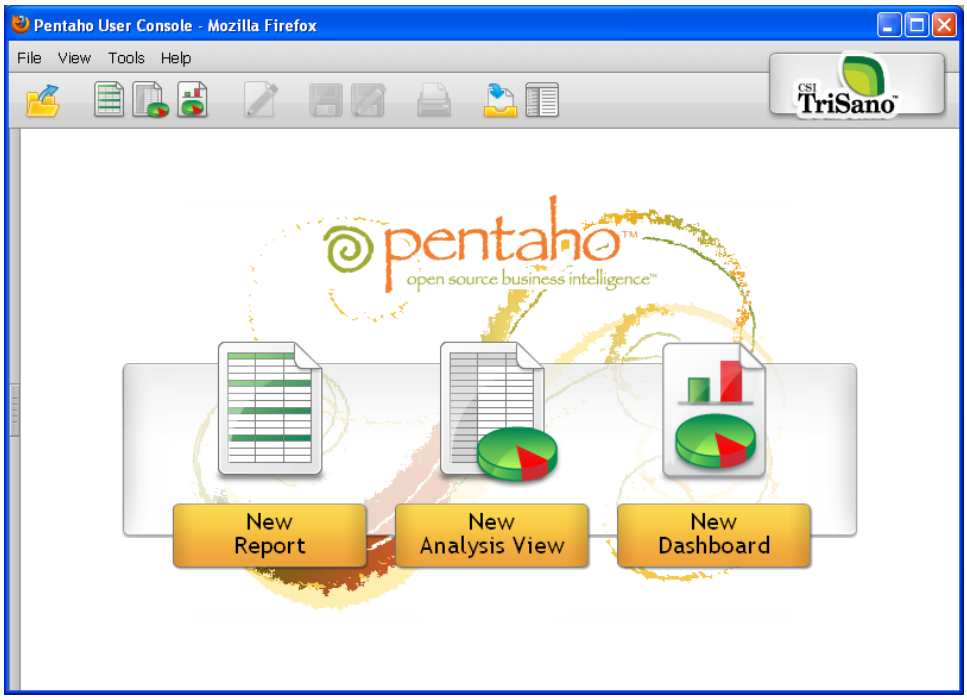
Introduction to the Pentaho™ User Console

Copyright © 2009 Collaborative Software Initiative, Inc.

Introduction to the Pentaho™ User Console

After accessing the AVR capabilities through the **AVR** option in CSI TriSano™. the Pentaho™ User Console will appear.
Both the Ad Hoc Query Tool and the OlAP Analysis Tool, as well as Dashboards are accessed through the Pentaho™ User Console. (The Report Designer is a separate application that runs under Windows, Linux, or OS X.)
The Pentaho™ User Console (shown below) allows you to:

- Create a **New Report** using the Ad Hoc Query Tool
- Create a **New Analysis** using the OLAP Analysis Tool
- Create a **New Dashboard**
- See a list of saved objects (reports, analysis, or dashboards
- Save any of the above to work with later
- Open any saved object to continue creating it
- Run a saved object
- Schedule a saved object to run at a later time or on a scheduled basis



Click on **New Report** or the icon to access the Ad Hoc Query Tool to begin create a new chart or report.

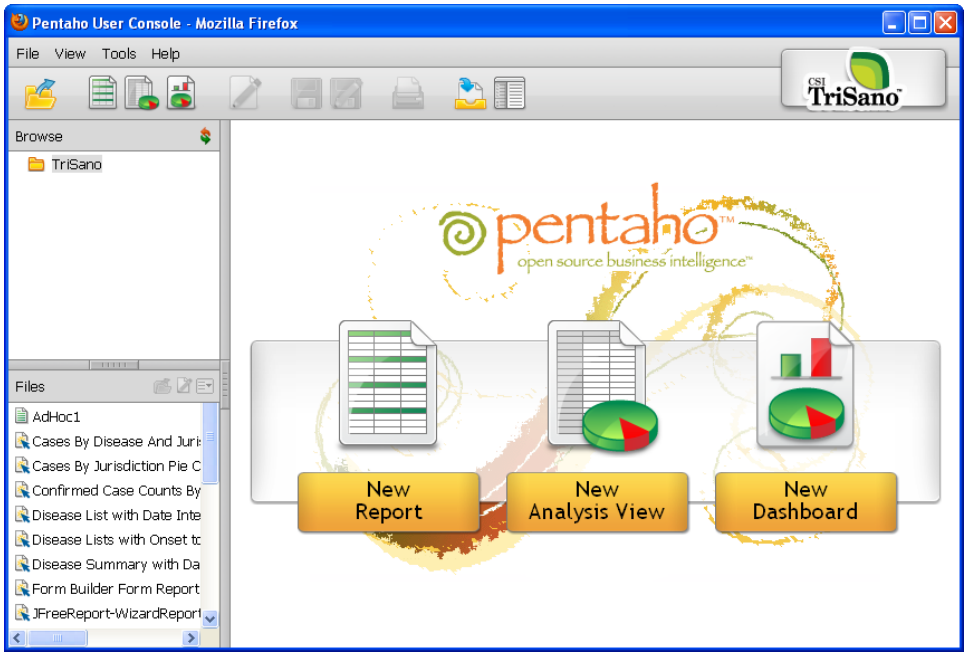
Click on **New Analysis View** or the icon to access the Analysis Services Tool to bein to create a new OLAP analysis.

Click on **New Dashboard** or the icon to create a new user dashboard.

The **Toggle Browser** icon, , can be used to show the folders and files for existing objects.

Click it now, and your screen will look like this:

Copyright © 2009 Collaborative Software Initiative, Inc.



Click on the Folder Name, TriSano in this case, under **Browse** and a list of existing files will display under **Files**.

To run an existing report or analysis, double-click on the name, or right-click on the name, then select run.

To edit an existing report or analysis, right-click on the name, then select edit.

Also, to the right of **Files** are several options:

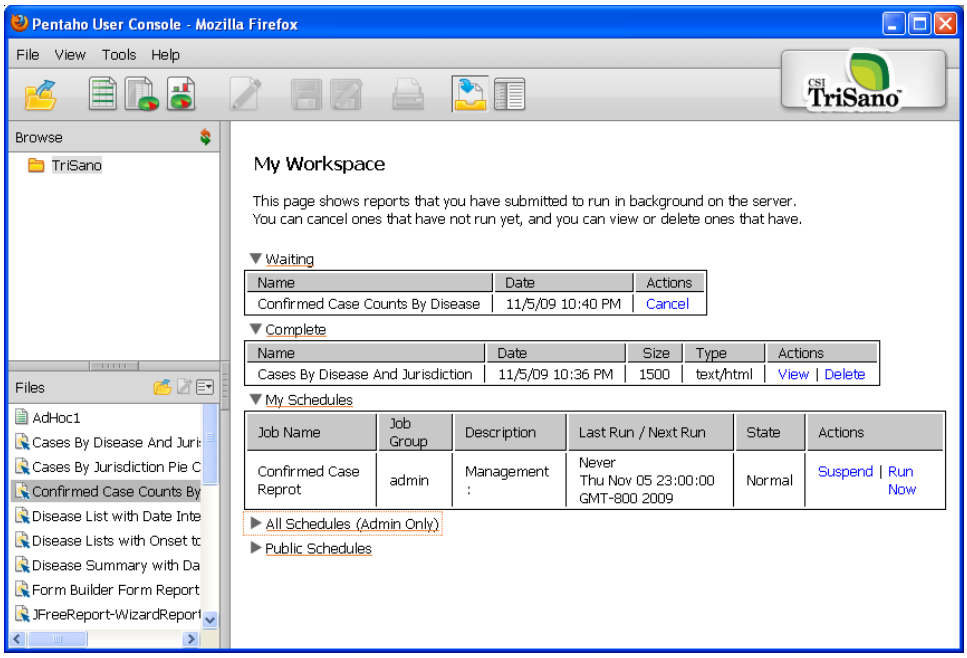
- **Open** - click on this after selecting a file to open the file
- **Edit** - click on this after selecting a file to edit the file
- **Options** - click on this after selecting a file to see additional options for working with the file
    - **Open In New Window** - to open the object in a new window.
    - **Run in Background** - to run the object in the background
    - **Delete** - to delete the object
    - **Share** - to share access to the object with others
    - **Schedule...** - to schedule the object to run at a later time or on a scheduled basis
    - **Properties...** - to see the Properties for the object

One other icon on the top of the screen, the **Workspace** icon, , displays a page with reports that are:

- Waiting to run
- Complete
- My Schedules
- All Schedules (only Admin will see this)
- Public Schedules

Click on the **Workspace** icon and this page will appear. Click it again to return to the User Console

Copyright © 2009 Collaborative Software Initiative, Inc.



That's a quick tour of the Pentaho™ User Console.

Next Topic:

Pentaho™ Ad Hoc Query Tool

Copyright © 2009 Collaborative Software Initiative, Inc.

Pentaho™ Ad Hoc Query Tool

Pentaho Ad Hoc Query Tool

One of the tools that is embedded with CSI TriSano™ is the Pentaho™ Ad Hoc Query tool. This is an easy to use tool that guides you through the creation of a query and report against the CSI TriSano™ core and form builder data. The CSI TriSano™ system provides one or more business models that automatically define and manage the relationship between the various elements of the CSI TriSano™ data warehouse.

The following is an example of using the Pentaho™ Ad Hoc Query Tool to create a query and report from core data in CSI TriSano™ and disease-specific Form Builder data. The form data is from the recent H1N1 form from the CDC. Your system will potentially have a similar form that you could use in place of the one used in this example.

Next Topic:

Ad Hoc Query Tool Completed Report

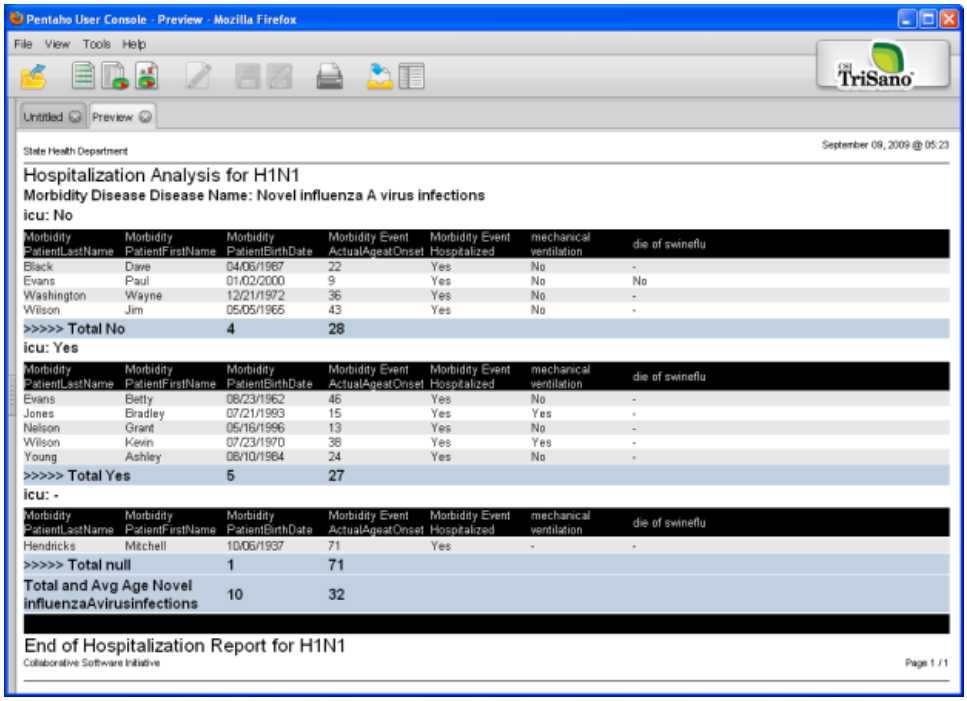
Copyright © 2009 Collaborative Software Initiative, Inc.

Ad Hoc Query Tool Completed Report

This "How To" guide will lead you through the steps to generate the following report (using fictitious information), shown here in HTML format. This report answers the following request, "Create a report that shows patients with H1N1 that have been hospitalized, showing which patients have been placed in the icu, have required mechanical ventilation, show their first and last names, birth dates, age at onset and whether they have died from H1N1. Also calculate the average age of onset along with a count of the number of patients."

Completed Report

Here is the completed report that you will create during this portion of the **AVR Quick Start Guide**.



Pentaho User Console - Preview - Mozilla Firefox

File View Tools Help

Untitled Preview

State Health Department

September 09, 2009 @ 05:23

Hospitalization Analysis for H1N1

Morbidity Disease Disease Name: Novel influenza A virus infections

icu: No

Morbidity PatientLastName	Morbidity PatientFirstName	Morbidity PatientBirthDate	Morbidity Event ActualAgeatOnset	Morbidity Event Hospitalized	mechanical ventilation	die of swineflu
Black	Dave	04/06/1987	22	Yes	No	-
Evans	Paul	01/02/2000	9	Yes	No	No
Washington	Wayne	12/21/1972	36	Yes	No	-
Wilson	Jim	05/05/1965	43	Yes	No	-
>>>>> Total No		**4**	**28**			

icu: Yes

Morbidity PatientLastName	Morbidity PatientFirstName	Morbidity PatientBirthDate	Morbidity Event ActualAgeatOnset	Morbidity Event Hospitalized	mechanical ventilation	die of swineflu
Evans	Betty	08/23/1962	46	Yes	No	-
Jones	Bradley	07/21/1993	15	Yes	Yes	-
Nelson	Grant	05/16/1996	13	Yes	No	-
Wilson	Kevin	07/23/1970	38	Yes	Yes	-
Young	Ashley	08/10/1984	24	Yes	No	-
>>>>> Total Yes		**5**	**27**			

icu: -

Morbidity PatientLastName	Morbidity PatientFirstName	Morbidity PatientBirthDate	Morbidity Event ActualAgeatOnset	Morbidity Event Hospitalized	mechanical ventilation	die of swineflu
Hendricks	Mitchell	10/06/1937	71	Yes	-	-
>>>>> Total null		**1**	**71**			
Total and Avg Age Novel influenzaAvirusinfections		**10**	**32**			

End of Hospitalization Report for H1N1

Collaborative Software Initiative

Page 1 / 1

Next Topic:

Accessing the Ad Hoc Query Tool

Copyright © 2009 Collaborative Software Initiative, Inc.

Accessing the Ad Hoc Query Tool

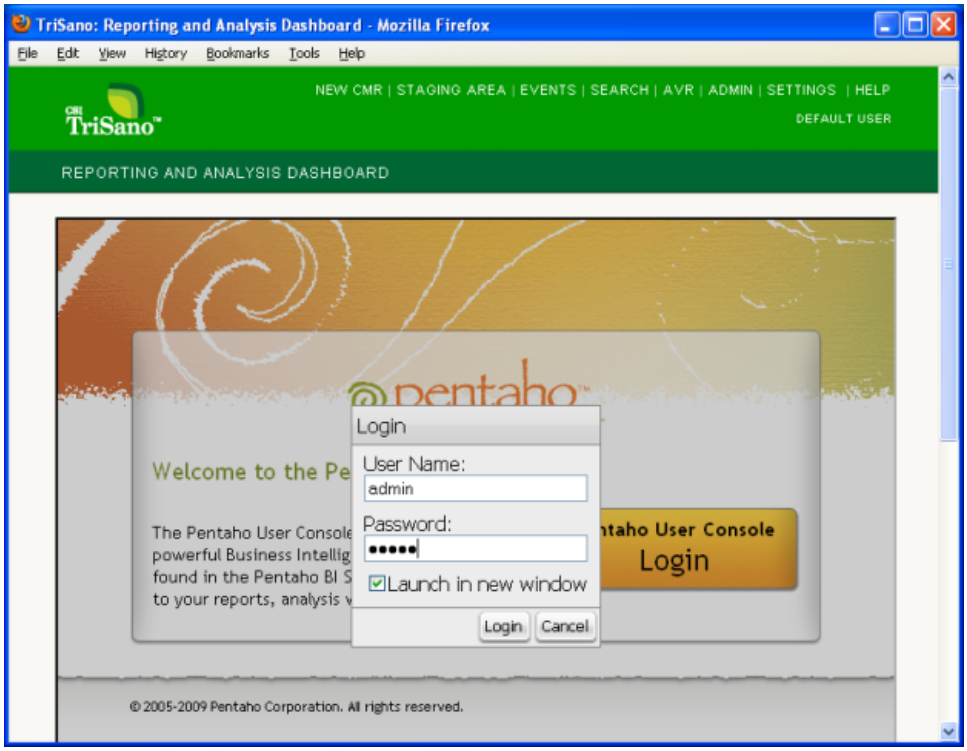
Accessing the Ad Hoc Query Tool

The Pentaho™ Ad Hoc Query tool is accessed by selecting the **AVR** option from any CSI TriSano™ screen. If **AVR** is not shown as an option at the top of the screen, then contact your system administrator to gain access.

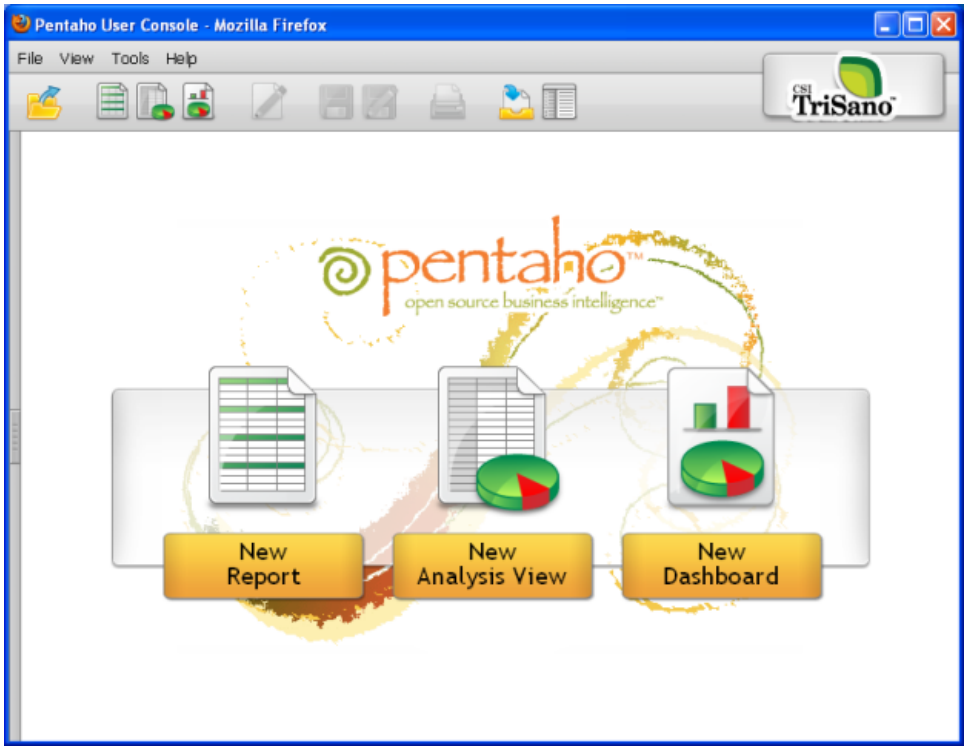
When you click on **AVR**, the Welcome to the Pentaho™ User Console screen is displayed.

Click on **Pentaho User Console Login** button and the **Login** screen will appear.

Copyright © 2009 Collaborative Software Initiative, Inc.



In the **Login** window, enter your **User Name** and **Password**. Your system adminstrator can tell you what to use for User Name and Password. Also, check the **Launch in new window** checkbox. This will give you more room, a larger window, to create the query. The Pentaho™ User Console is displayed.



This screen provides the following options:

Option	Description
New Report	Launches the Ad Hoc Query tool
New Analysis View	Launches the Analysis (OLAP) tool

Copyright © 2009 Collaborative Software Initiative, Inc.

New Dashboard	Allows you to create a new Dashboard for your use

See the section, Introduction to the Pentaho™ User Console, for more information on the capabilities found in the User Console.

Next Topic:

Creating a New Report

Copyright © 2009 Collaborative Software Initiative, Inc.

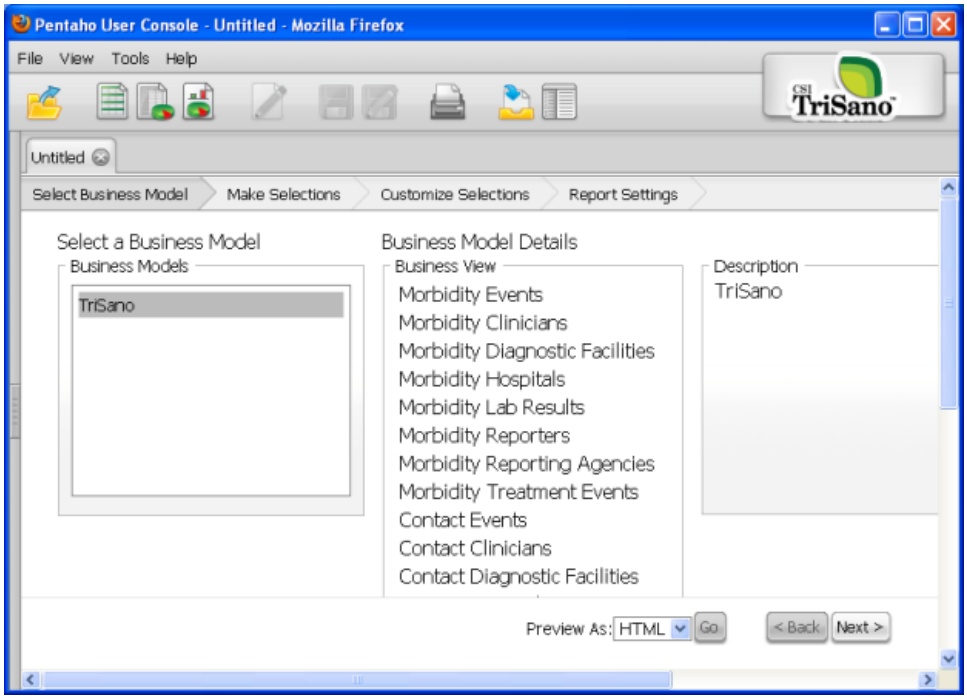
Creating a New Report

Creating a New Report

Click on **New Report** to launch the Ad Hoc Query tool. The **Select Business Model** screen will appear.

The Ad Hoc Query tool consists of a series of four screens, denoted by the four tabs across the screen. They are:

Tab Title	Description
Select Business Model	On this screen, you will select the business model and template to use for the report.
Make Selections	On this screen, you will select the columns to be displayed on your report and any grouping or filtering you would like to use.
Customize Selections	On this screen, you will define formatting for the columns and group headings in your report, define any additional constraints on the query to limit the number of records, or rows, returned, and define any additional sorting required.
Report Settings	On this screen, you will define report and page headers and footers.

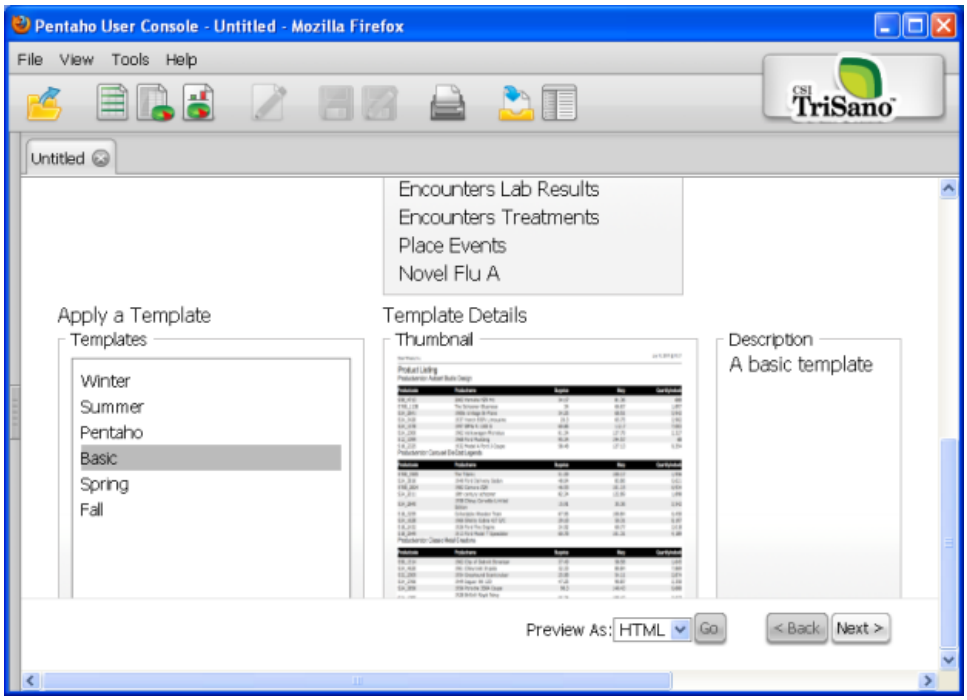


The **Select Business Model** screen lists the business models available and details about each model. In this case, there is only one, **TriSano™**. Since there is only one, it is selected automatically and the details for the business model are shown. Your system may list several business models depending on how the CSI TriSano™ system was implemented, specifically how **AVR Groups** were defined.

One of the options on this screen is to choose a template for the report, but you don't see it here. It is at the bottom of the screen (notice the scroll bar on the right). So scroll to the bottom, and you will see the **Apply a Template** portion of the window.

For our report, choose the **Basic** template. It seems to work well for most reports. Once selected, the thumbnail changes to show you what the report will look like, as shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.



That's it for this screen. To move between the screens of the Ad Hoc Query tool, use the **Back** and **Next** buttons in the lower right hand corner of the screen. If either is not available, it is grayed out. Since you are on the first screen in a series of screens, the **Back** button is grayed out right now. Click on the **Next** button to continue to the **Make Selections** screen.

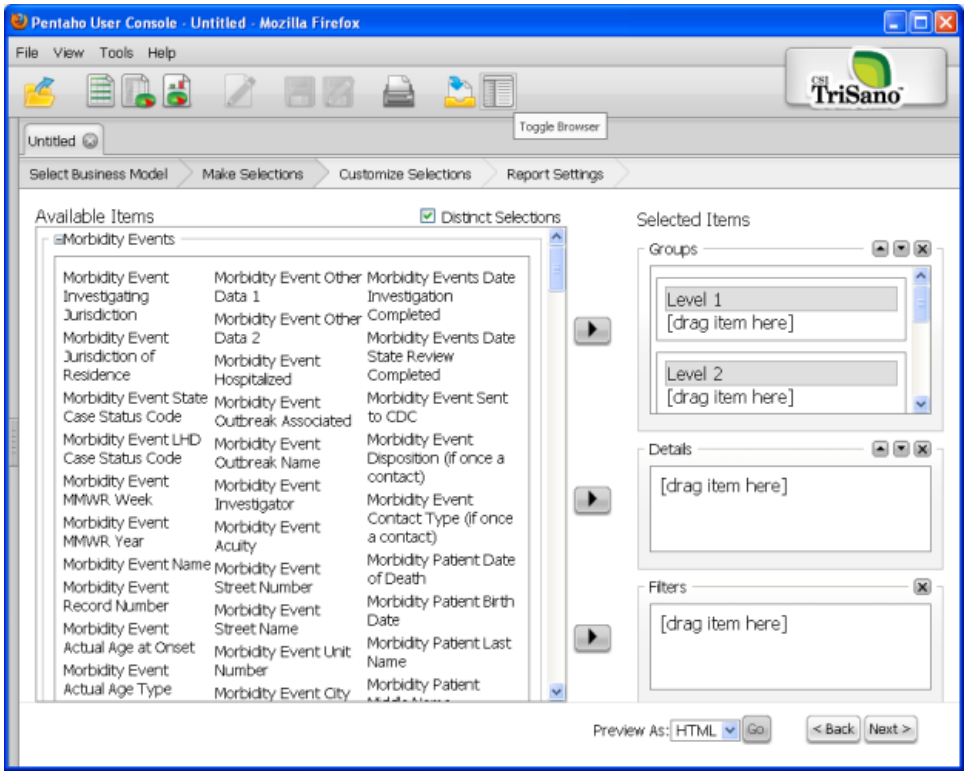
Next Topic:

Make Selections Screen

Copyright © 2009 Collaborative Software Initiative, Inc.

Make Selections Screen

Make Selections Screen



This screen shows details on the information available in the selected business model. On the left is a list of available items in the data warehouse. This information is organized, much like the information in CSI TriSano™, by event type. So the first sets of items, let's call them tables, is related to morbidity events. The first table shown in the list is **Morbidity Events**. It contains the basic CMR data. If you scroll down the Available Items list, you will see:

- Morbidity Clinicians
- Morbidity Diagnostic Facilities
- Morbidity Hospitals
- Morbidity Lab Results
- Morbidity Reporters
- Etc.

After morbidity event tables, there are contact event tables, followed by encounter and place event tables. At the end of the list will be a table for each disease-specific form in the system that is part of the selected business model. This query will be using information from one of these disease-specific tables, the Novel Flu A table. Take some time to scroll through the list before you continue. The query you are going to build includes the following columns of information from the following tables:

Table Name	Column Name
Morbidity Events	Morbidity Disease Disease name
Morbidity Events	Morbidity Patient Last Name
Morbidity Events	Morbidity Patient First name
Morbidity Events	Morbidity Patient Birth date
Morbidity Events	Morbidity Patient Actual Age at Onset
Morbidity Events	Morbidity Event Hospitalized

Copyright © 2009 Collaborative Software Initiative, Inc.

Novel Flu A	icu
Novel Flu A	mechanical ventilation
Novel Flu A	die of swineflu

Let's get started.

Next Topic:

Adding Columns to a Query

Copyright © 2009 Collaborative Software Initiative, Inc.

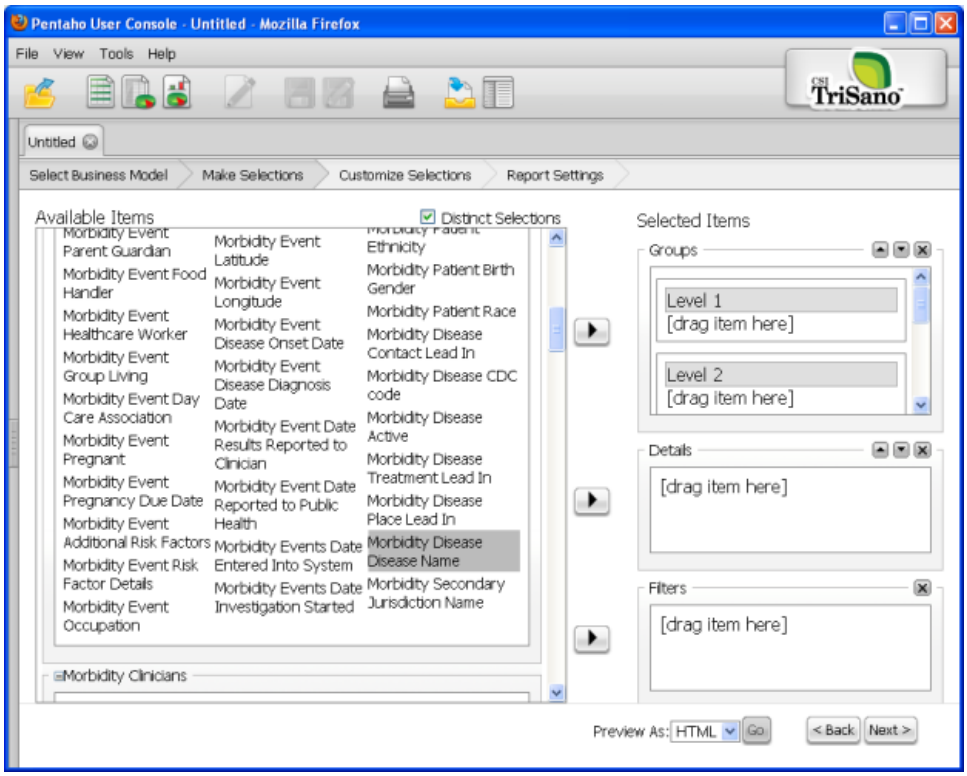
Adding Columns to a Query

Adding Columns to a Query

The first column to be added to the query and report is **Morbidity Disease Disease Name** from the **Morbidity Events** table.

Scroll down the list of **Available Items** until you find the **Morbidity Disease Disease Name** column in the **Morbidity Events** table.

Highlight the **Morbidity Disease Disease Name** column by clicking on it, as shown in the screen below.



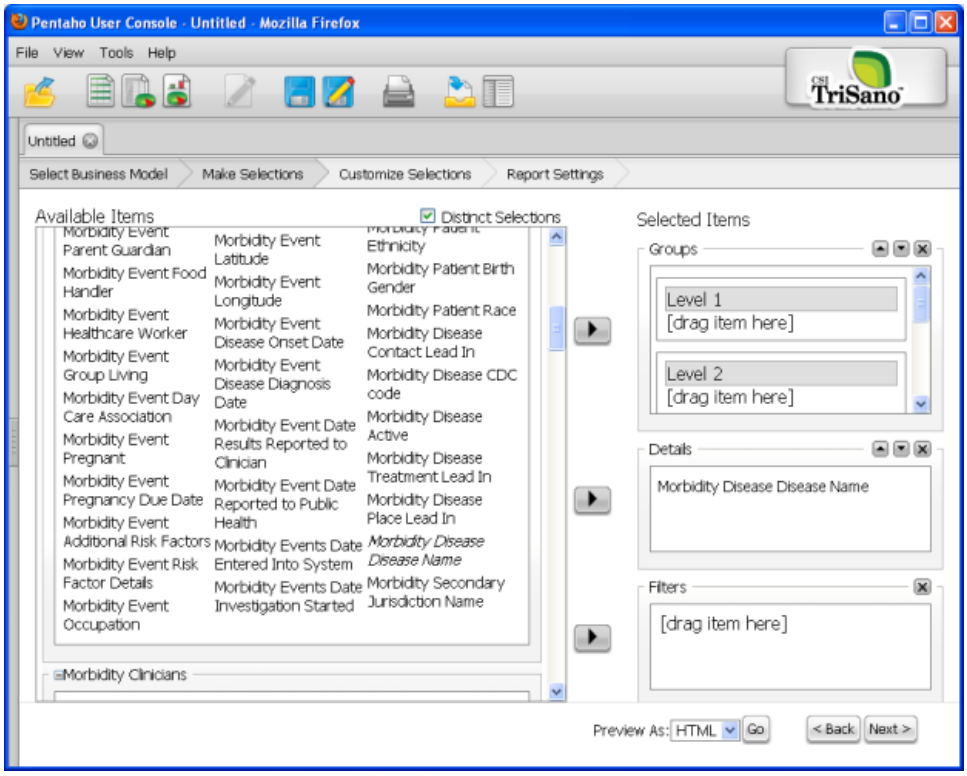
To add this column to the query, click and hold the left mouse button on the **Morbidity Disease Disease Name** column you highlighted before and drag it to the **Details** box on the right. When you are in the right place, you will see a narrow black line appear right above the **[drag item here]** prompt in the **Details** box. Release the left mouse button to add the column to the list of columns in our report.

You may need to try it a couple of times to get the hang of it.

Alternatively, you can highlight the column name and click the dark arrow to the left of the **Details** box. In fact, you can select multiple columns using the **CTRL** key before clicking the arrow.

When finished your screen should look this.

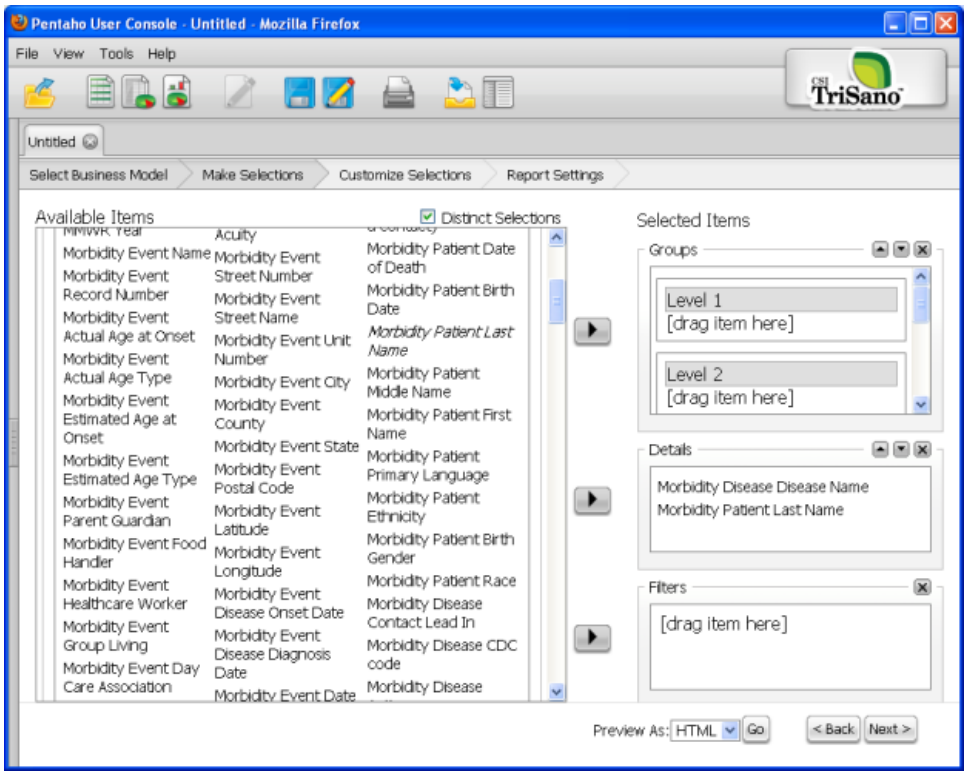
Copyright © 2009 Collaborative Software Initiative, Inc.



Next, we want to add the **Morbidity Patient Last Name** to the report. You may need to scroll the Available Items box up some to find the column. Once you find it, click and hold the mouse and drag the **Last name** column to the **Details** box. You will notice that depending on how you position the cursor, you will see that same narrow black line appear either above or below the **Disease name** column that is already in the **Details** box.

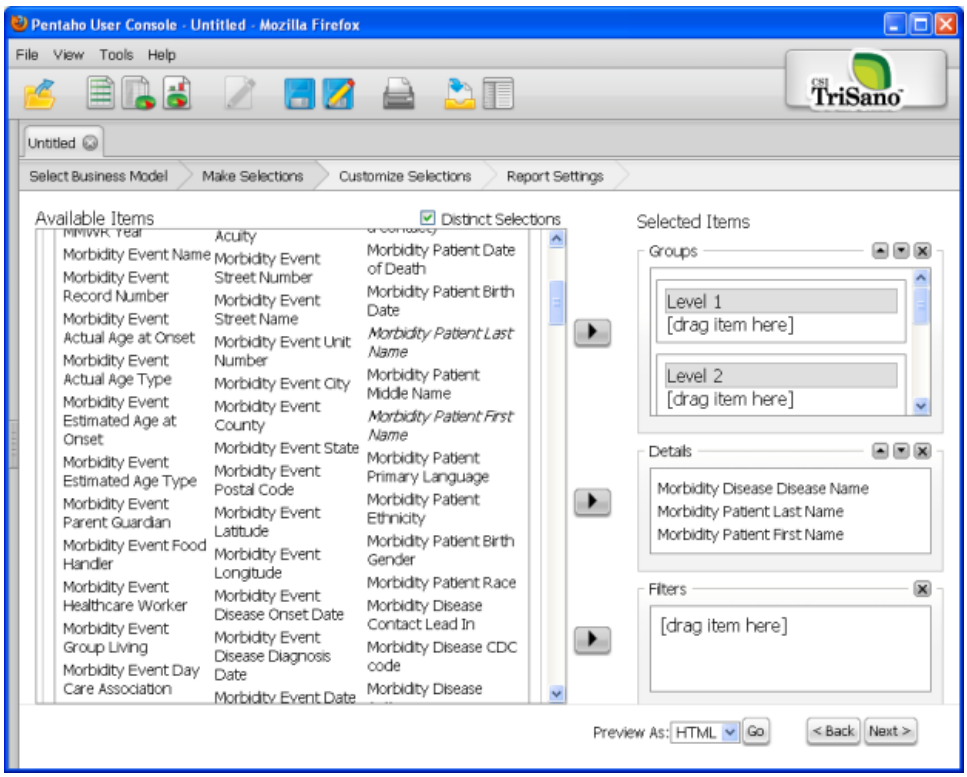
Drop the **Last name** column when the black line is below the **Disease name** by releasing the left mouse button.

Your screen will look like this.



Copyright © 2009 Collaborative Software Initiative, Inc.

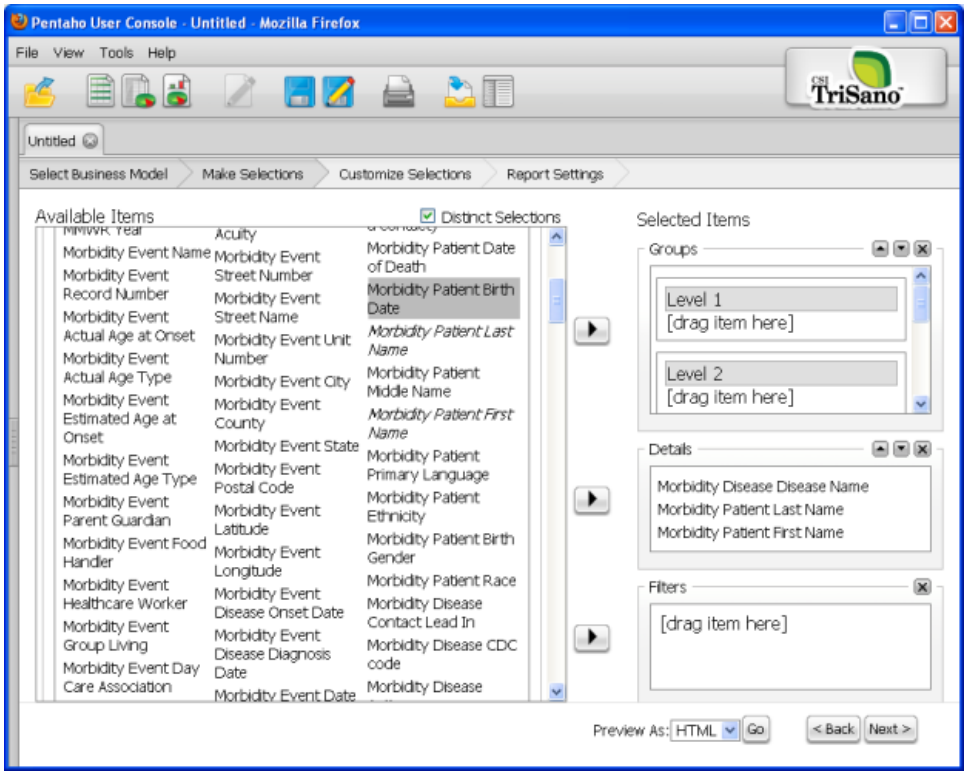
Next select the **Morbidity Events First Name** in the **Morbidity Events** table and drag it to the Details box and place it after the **Morbidity Events Last Name** column. Your screen will look like the one shown below.



Notice that as you drag columns from the list of **Available Items** to the **Details** box that the column name in the **Available Items** list is *italicized*.

Now, depending on the size of your window, the Details box is probably filling up, but you can still drag items to it. (You may want to maximize your window to have as much room as possible.)

To continue, select the **Morbidity Patient Birth Date** column as shown below.



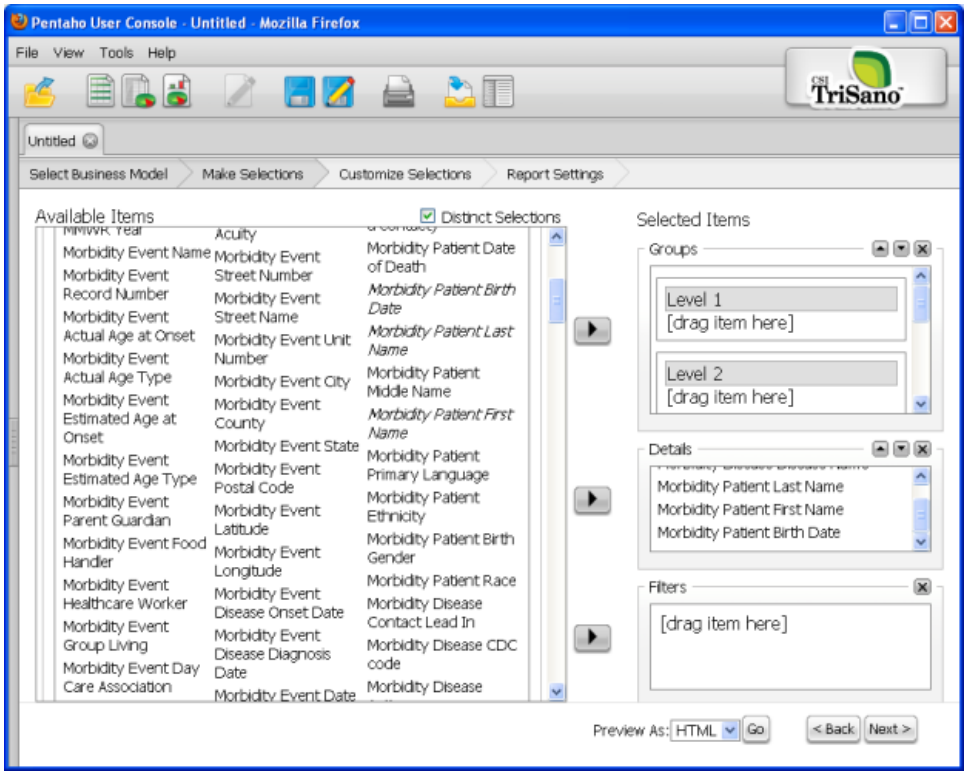
Copyright © 2009 Collaborative Software Initiative, Inc.

Now drag it to just below the **Morbidity Patient First Name** and you will see the black line appear below the **Morbidity Patient First Name** column. Drop the **Morbidity Patient Birth Date** column there.

When you do this, you might not see the **Morbidity Patient Birth Date** column in the **Details** box, as shown below, but it is there in the list under the **Morbidity Patient First Name**.

Use the scroll bar to scroll to the bottom of the list in the **Details** box and see that the **Morbidity Patient Birth Date** column is there.

Your screen should look like the one shown below.



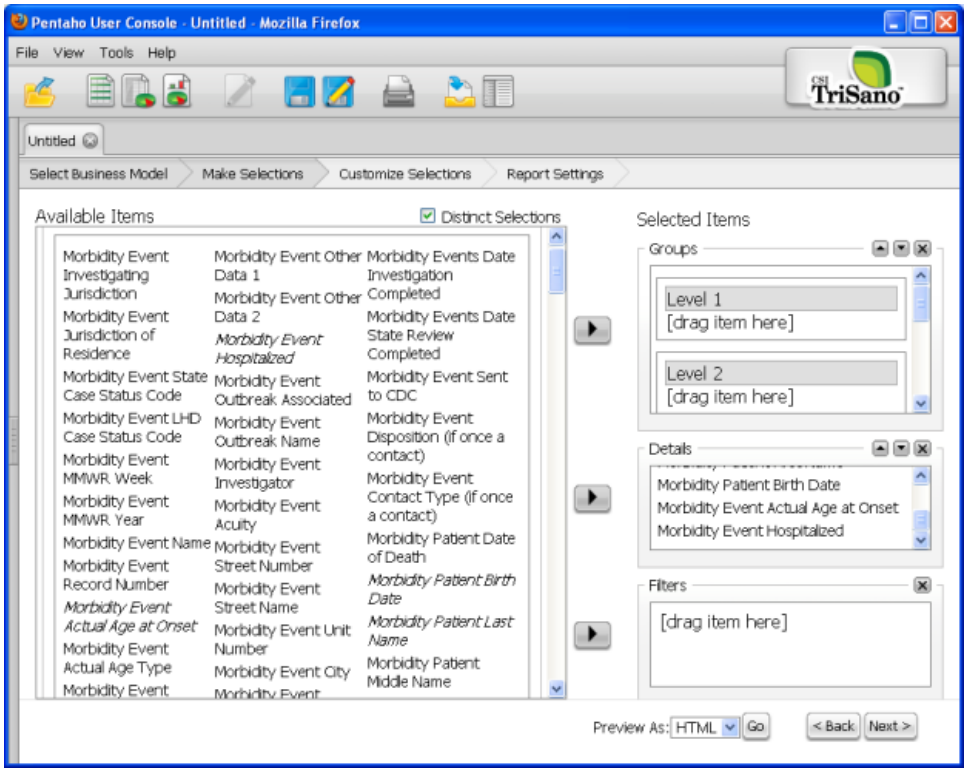
Remember, here is the list of the columns you are adding to the query. You are almost half way there, fine more to go.

Table Name	Column Name
Morbidity Events	Morbidity Disease Disease name
Morbidity Events	Morbidity Patient Last Name
Morbidity Events	Morbidity Patient First name
Morbidity Events	Morbidity Patient Birth date
Morbidity Events	Morbidity Patient Actual Age at Onset
Morbidity Events	Morbidity Event Hospitalized
Novel Flu A	icu
Novel Flu A	mechanical ventilation
Novel Flu A	die of swineflu

Continue to add the **Actual Age at Onset** and the **Morbidity Event Hospitalized** columns from the **Morbidity Events** table to the **Details** box.

Your screen should look like this:

Copyright © 2009 Collaborative Software Initiative, Inc.



You have now added all the columns from the Morbidity Events table. Next, you will add some columns from the form. If you do not have a similar form, see if you can choose a different form for morbidity events, and choose other columns. If you do not have a form to use, just read through this part of the guide.

Next Topic:

Adding Columns from a Form

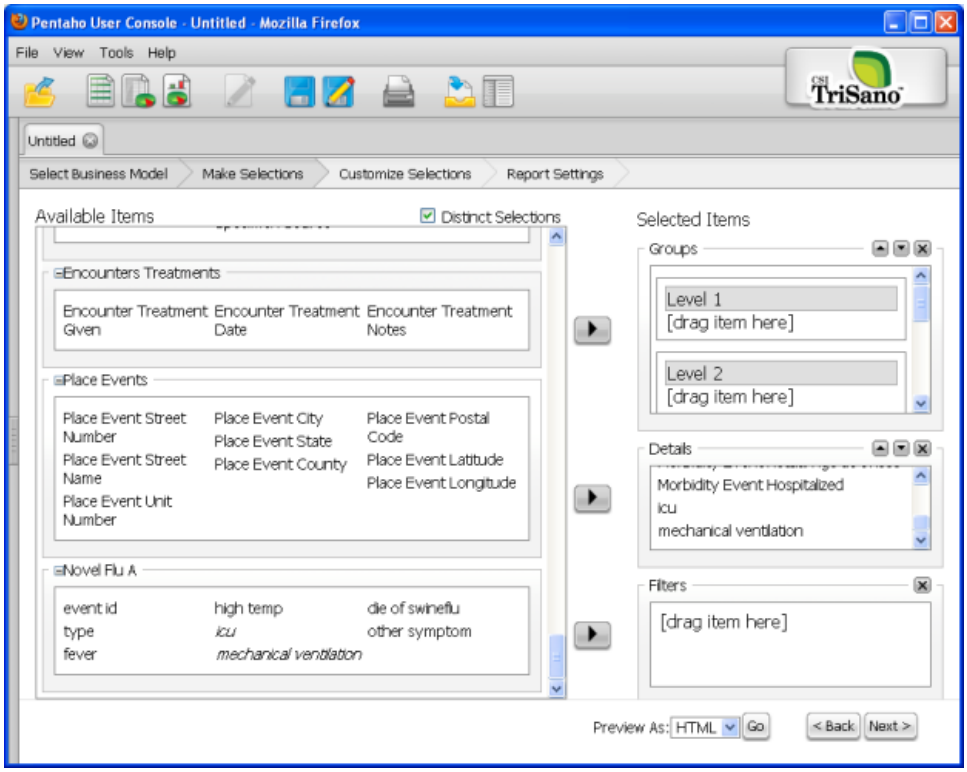
Copyright © 2009 Collaborative Software Initiative, Inc.

Adding Columns from a Form

Adding Columns from a Form

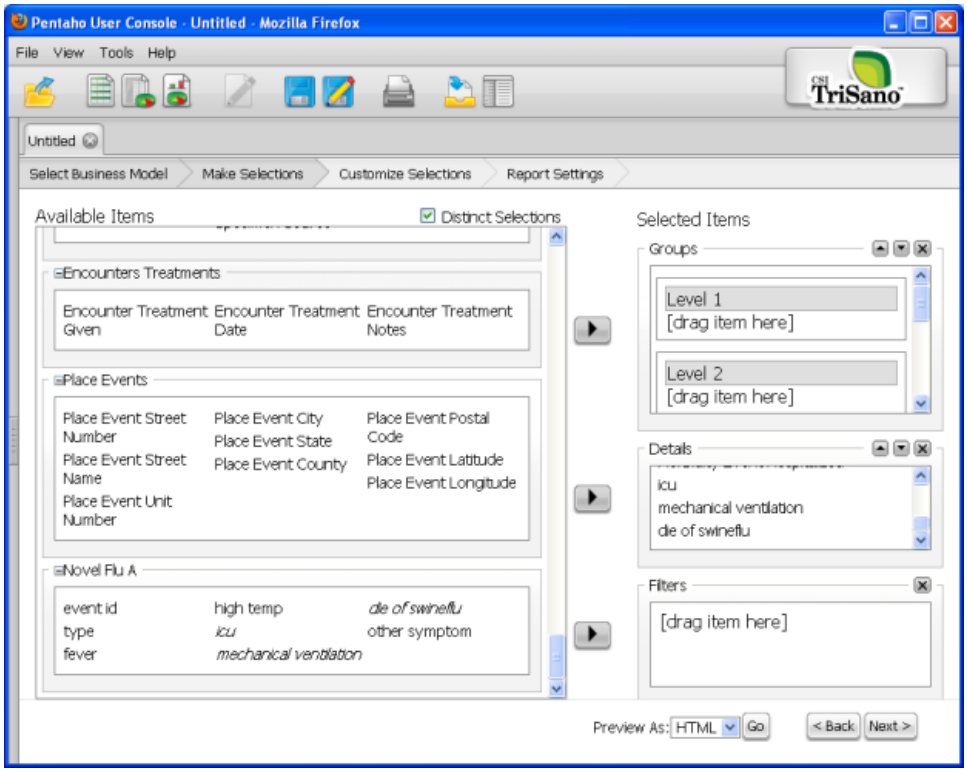
The next two columns to add to the query are from the **Novel Flu A** table that contains data from the **Novel Flu A** form, so scroll to the bottom of the **Available Items** list to find it. (Any tables related to forms will be at the bottom of the list of tables.)

Once you find it, click on the **icu** column and drag it to place it under the **Morbidity Event Hospitalized** column and then click on the **mechanical ventilation** column and drag it to place it after the **icu** column. Your screen should look like the one shown below.



The last column in the query, **die of swineflu**, is also from the Novel Flu A table. Place it after the **mechanical ventilation** column as shown here.

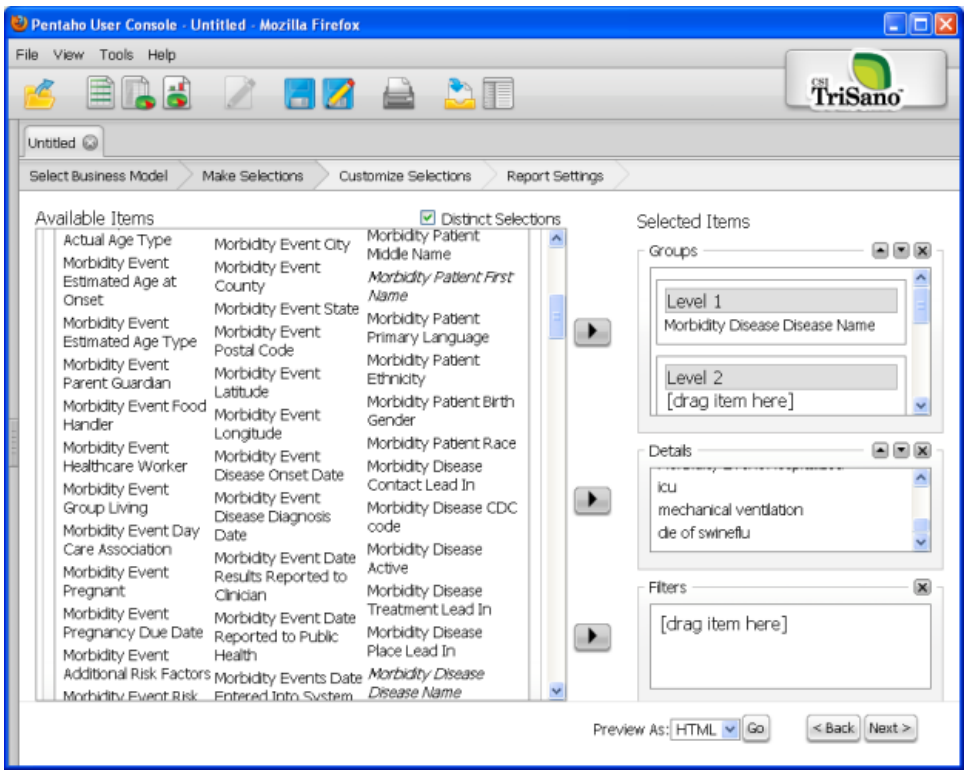
Copyright © 2009 Collaborative Software Initiative, Inc.



Now, you have added all the columns to the query. There is only one more thing to do. You want to have the morbidity events returned grouped by the **Disease name**. To do this, you need to drag the **Morbidity Diseaese Disease name** to the **Level 1** box under the **Groups** heading in the upper right corner of the screen.

Scroll back up the **Morbidity Events** table and select the **Morbidity Disease Disease Name** column and drag it to the Level 1 box.

Your screen will look like the one shown below.



Next Topic:

Copyright © 2009 Collaborative Software Initiative, Inc.

Saving the Query

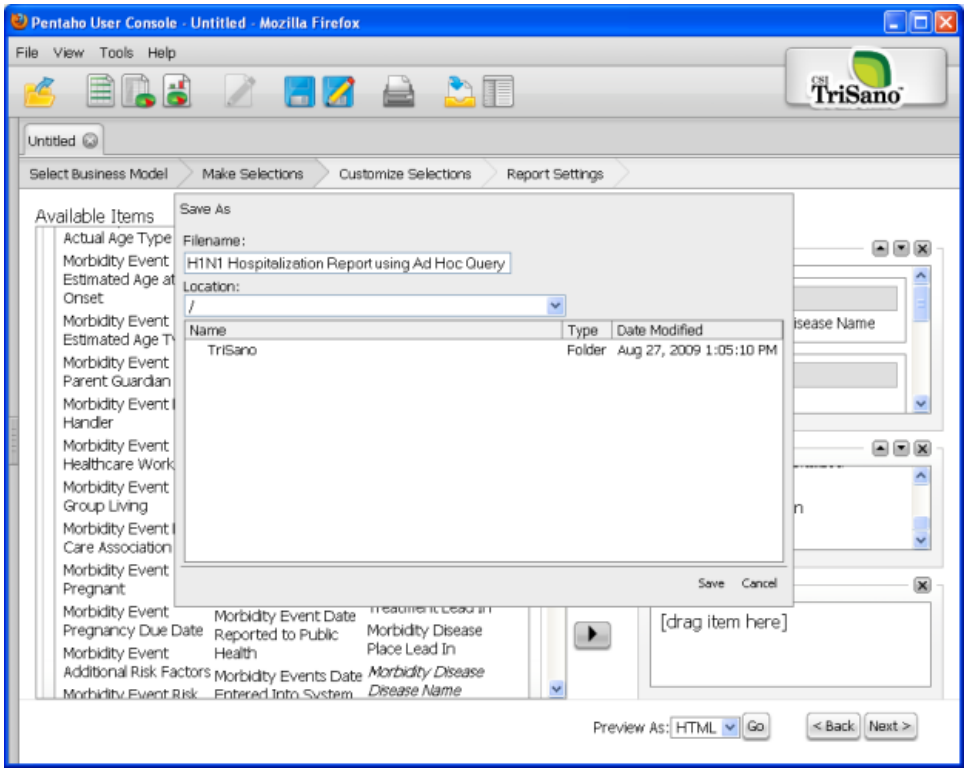
Copyright © 2009 Collaborative Software Initiative, Inc.

Saving the Query

Saving the Query

This might be a good time to **Save** what you have done so far. So click on **File**.

Next, click on **Save As….** , then click on the folder name, **TriSano** in this case, and give your query a name, such as **"H1N1 Hospitalization Report using Ad Hoc Query"** as shown below.



Click on **Save** to save the report.

You should see the name, **"H1N1 Hospitalization Report using Ad Hoc Query"**, in the title bar of the window.

Next Topic:

Previewing the Report

Copyright © 2009 Collaborative Software Initiative, Inc.

Previewing the Report

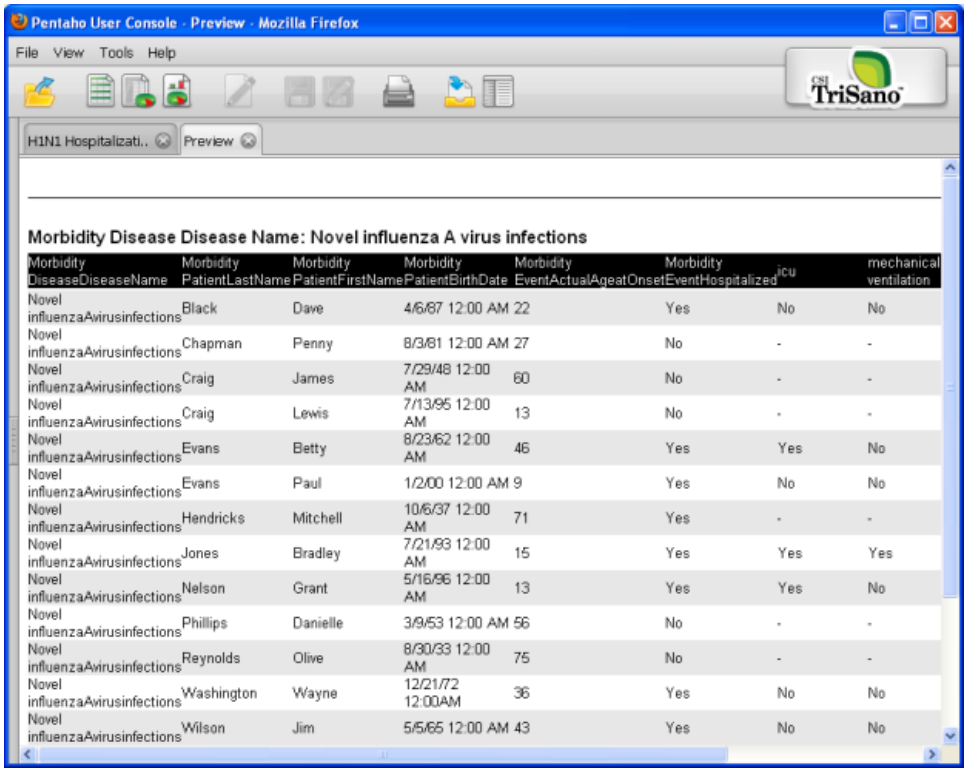
Previewing the Report

Now that you have saved your query and report, this might be a good time to see what the report looks like so far. You can do this at any time.

To preview the report, click on **Go** to see the **HTML** version of the report, or click on the drop down box to the left of **Go** and choose one of the following:

- HTML
- PDF
- Excel
- CSV

For now, click on **Go** to see the HTML preview report.You'll see that the report is shown on a separate tab titled **Preview**.



Morbidity DiseaseDiseaseName	Morbidity PatientLastName	Morbidity PatientFirstName	Morbidity PatientBirthDate	Morbidity EventActualAgeatOnset	Morbidity EventHospitalized	icu	mechanical ventilation
Novel influenzaAvirusinfections	Black	Dave	4/6/67 12:00 AM	22	Yes	No	No
Novel influenzaAvirusinfections	Chapman	Penny	8/3/81 12:00 AM	27	No	-	-
Novel influenzaAvirusinfections	Craig	James	7/29/48 12:00 AM	60	No	-	-
Novel influenzaAvirusinfections	Craig	Lewis	7/13/95 12:00 AM	13	No	-	-
Novel influenzaAvirusinfections	Evans	Betty	8/23/62 12:00 AM	46	Yes	Yes	No
Novel influenzaAvirusinfections	Evans	Paul	1/2/00 12:00 AM	9	Yes	No	No
Novel influenzaAvirusinfections	Hendricks	Mitchell	10/6/37 12:00 AM	71	Yes	-	-
Novel influenzaAvirusinfections	Jones	Bradley	7/21/93 12:00 AM	15	Yes	Yes	Yes
Novel influenzaAvirusinfections	Nelson	Grant	5/16/96 12:00 AM	13	Yes	Yes	No
Novel influenzaAvirusinfections	Phillips	Danielle	3/9/53 12:00 AM	56	No	-	-
Novel influenzaAvirusinfections	Reynolds	Olive	8/30/33 12:00 AM	75	No	-	-
Novel influenzaAvirusinfections	Washington	Wayne	12/21/72 12:00AM	36	Yes	No	No
Novel influenzaAvirusinfections	Wilson	Jim	5/5/65 12:00 AM	43	Yes	No	No

Next Topic:

Removing Columns from the Report

Copyright © 2009 Collaborative Software Initiative, Inc.

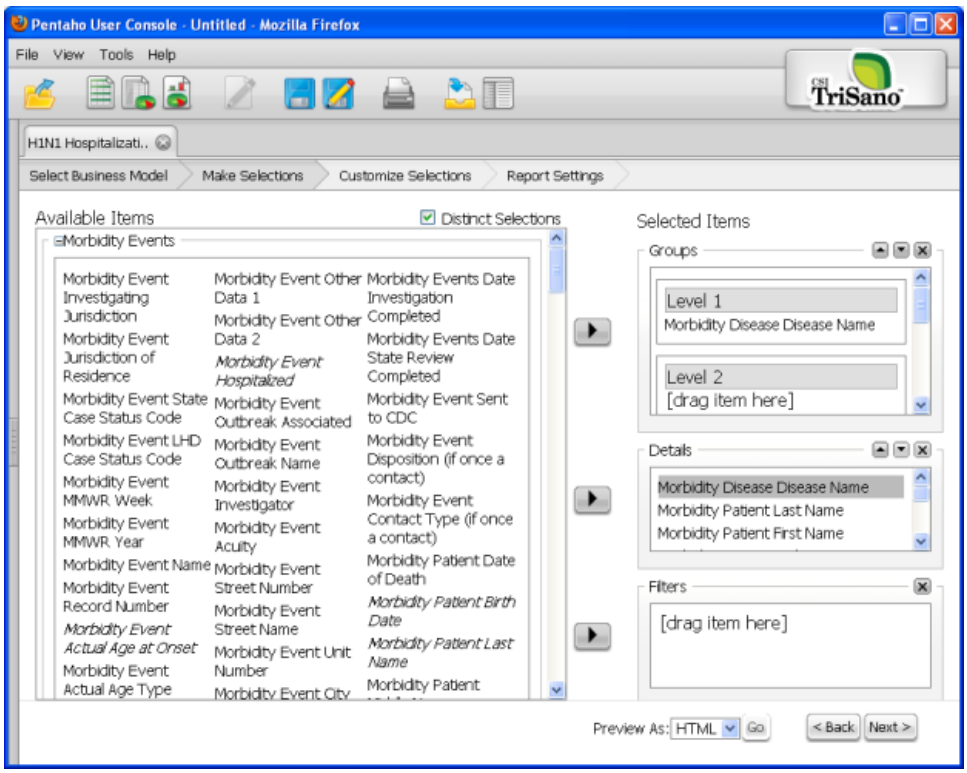
Removing Columns from the Report

Removing Columns from the Report

Notice that the report has a Level 1 grouping with the name of the disease, "**Novel influenza A virus infections**" at the top of the report and the same information as a column in the report. This is duplicate information. You typically don't need the **Morbidity Disease Disease Name** column in the report if you are grouping by that same column.

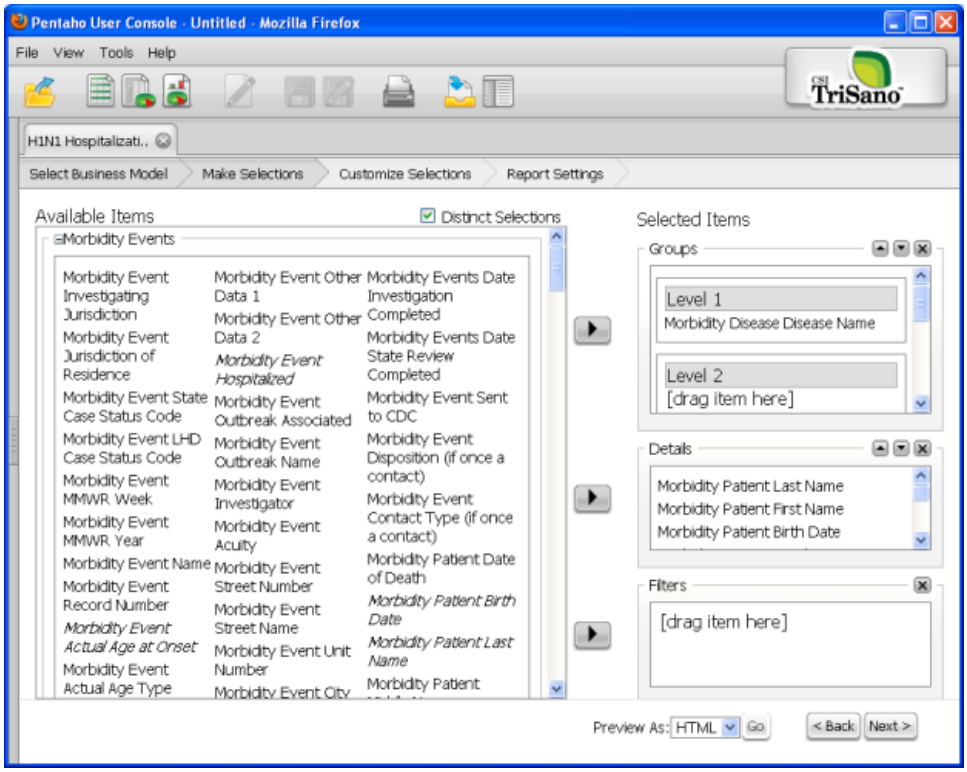
Let's get rid of the column **Morbidity Disease Disease Name** in the report, first click on the **x** next to **Preview** to close the report preview and return to the ad hoc query.

Next, select the **Morbidity Disease Disease name** column in the **Details** box as shown below.

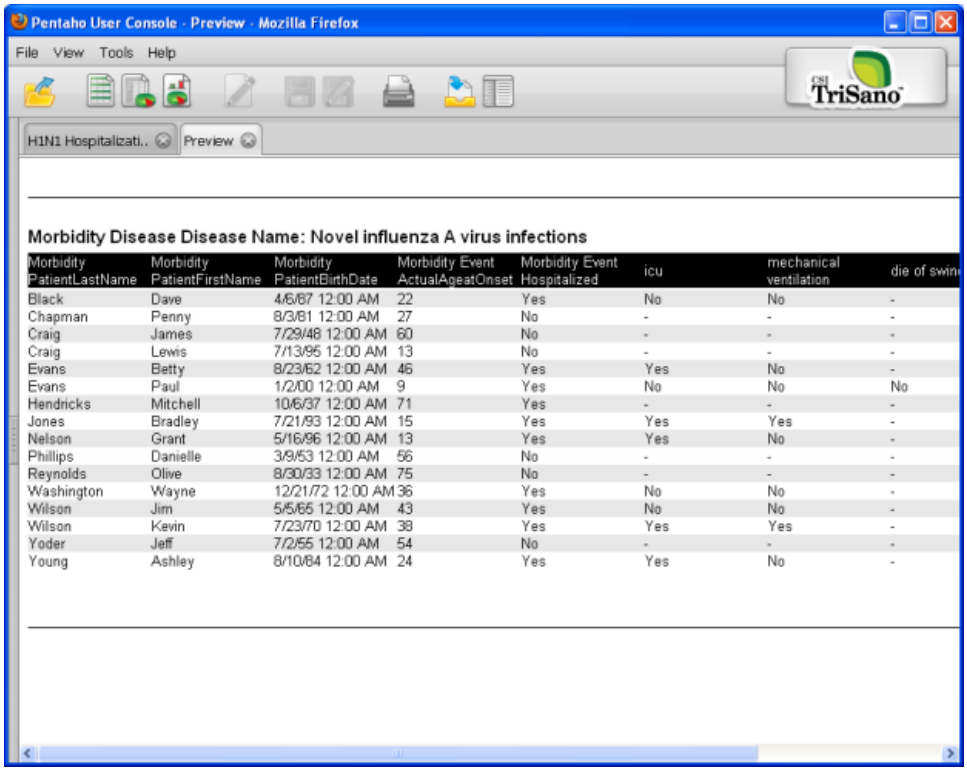


To delete the **Morbidity Disease Disease name** from the **Details** box, just click on the **X** to the right of **Details**.
Your screen should look like the one shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.



Notice that the **Morbidity Event Disease Name** is no longer in the **Details** box. Now **Preview** the report by clicking on **Go**.

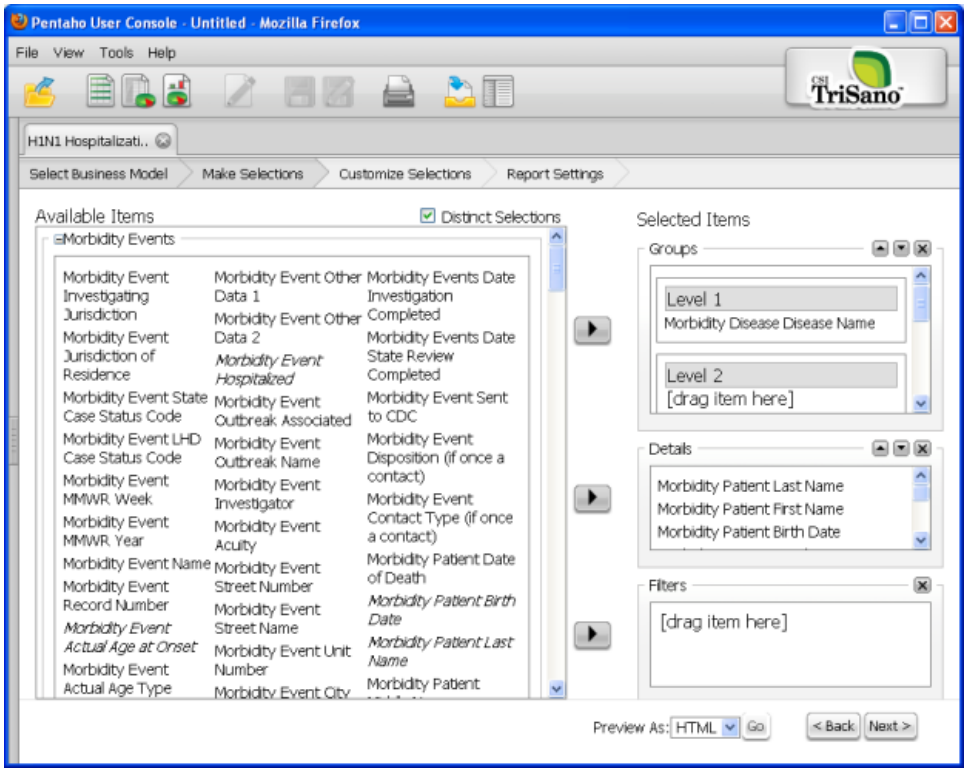


Morbidity PatientLastName	Morbidity PatientFirstName	Morbidity PatientBirthDate	Morbidity Event ActualAgeatOnset	Morbidity Event Hospitalized	icu	mechanical ventilation	die of swin
Black	Dave	4/6/87 12:00 AM	22	Yes	No	No	-
Chapman	Penny	8/3/81 12:00 AM	27	No	-	-	-
Craig	James	7/29/48 12:00 AM	60	No	-	-	-
Craig	Lewis	7/13/95 12:00 AM	13	No	-	-	-
Evans	Betty	8/23/62 12:00 AM	46	Yes	Yes	No	-
Evans	Paul	1/2/00 12:00 AM	9	Yes	No	No	No
Hendricks	Mitchell	10/6/37 12:00 AM	71	Yes	-	-	-
Jones	Bradley	7/21/93 12:00 AM	15	Yes	Yes	Yes	-
Nelson	Grant	5/16/96 12:00 AM	13	Yes	Yes	No	-
Phillips	Danielle	3/9/53 12:00 AM	56	No	-	-	-
Reynolds	Olive	8/30/33 12:00 AM	75	No	-	-	-
Washington	Wayne	12/21/72 12:00 AM	36	Yes	No	No	-
Wilson	Jim	5/5/65 12:00 AM	43	Yes	No	No	-
Wilson	Kevin	7/23/70 12:00 AM	38	Yes	Yes	Yes	-
Yoder	Jeff	7/2/55 12:00 AM	54	No	-	-	-
Young	Ashley	8/10/84 12:00 AM	24	Yes	Yes	No	-

Notice that the **Morbidity Disease Disease Name** is no longer a column in the report, but is still part of the heading.

When finished viewing the report, close that tab and return to the query.

Copyright © 2009 Collaborative Software Initiative, Inc.



You've done all you can do for now on this step, so click on **Next** to continue to the **Customize Selections** screen.

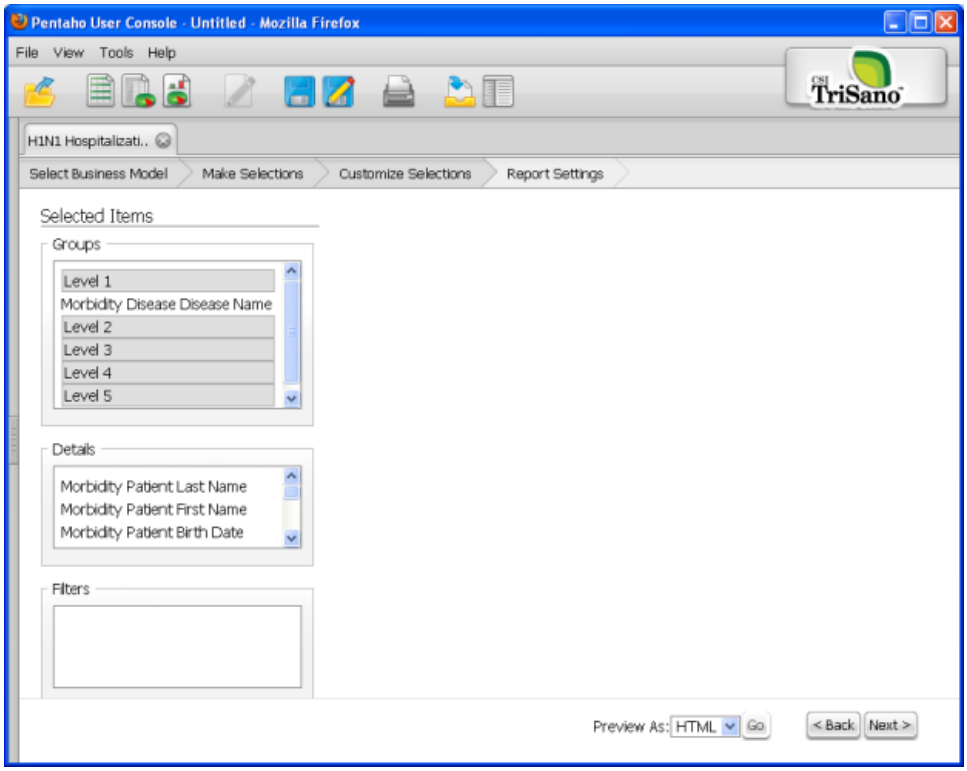
Next Topic:

Customizing the Report Selections

Copyright © 2009 Collaborative Software Initiative, Inc.

Customizing the Report Selections

Customizing the Report Selections



This screen looks pretty plain and simple when it first appears, but in fact it has a lot of capability.

The next few sections of this guide will show you how to:

- Format a column in the report
- Add a constraint for a column
- Define the sort order
- Calculate the average for a column
- Change the formatting for a column
- Changing the text for the Summary line

Next Topic:

Customizing a Column in the Report

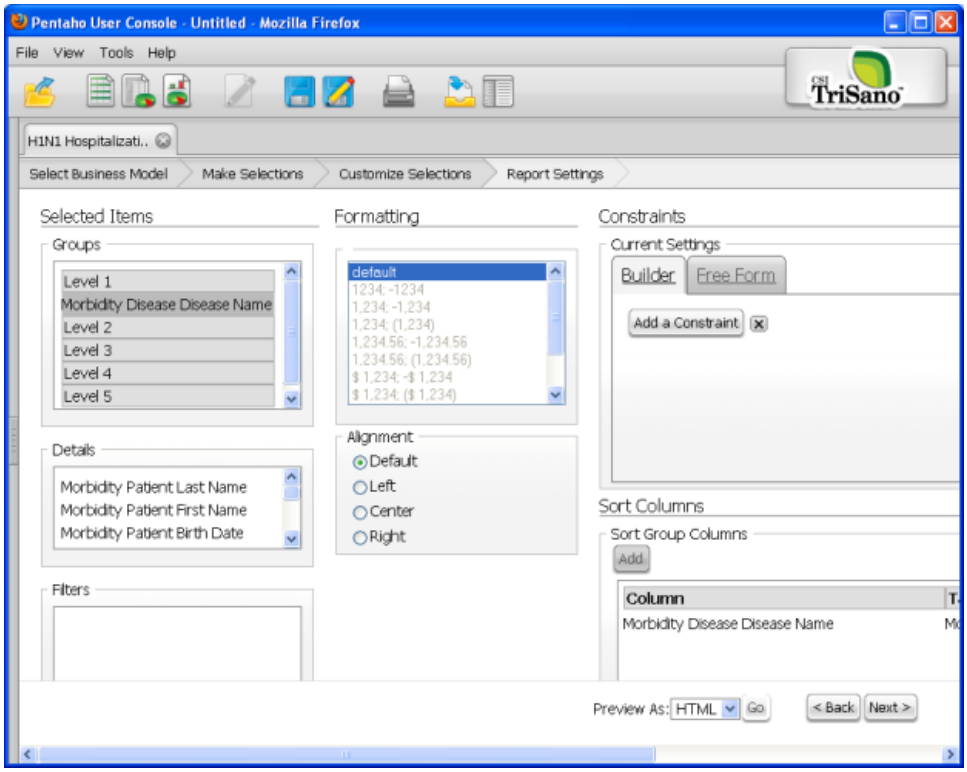
Copyright © 2009 Collaborative Software Initiative, Inc.

Customizing a Column in the Report

Customizing a Column in the Report

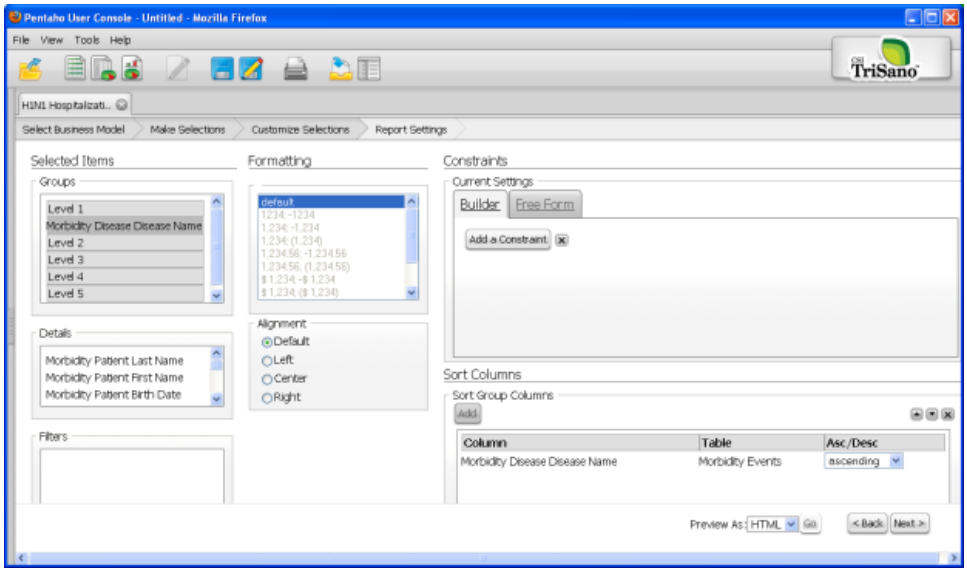
You enable the different, hidden, options on this screen by clicking on the various column names or Level numbers on the left side of the screen.

To start, click on the **Morbidity Disease Disease Name** column under the **Level 1 Group**. You will see additional formatting, constraint and sort options appear.



Notice that **Formatting** and **Constraints** settings appear on the right side of the screen and that the **Morbidity Disease Disease Name** had been automatically added as a sort column.

In this case, the screen is not wide enough to see everything, so I've resized it below.



Copyright © 2009 Collaborative Software Initiative, Inc.

Next Topic:

Adding Constraints to a Column

Copyright © 2009 Collaborative Software Initiative, Inc.

Adding Constraints to a Column

Adding Constraints to a Column

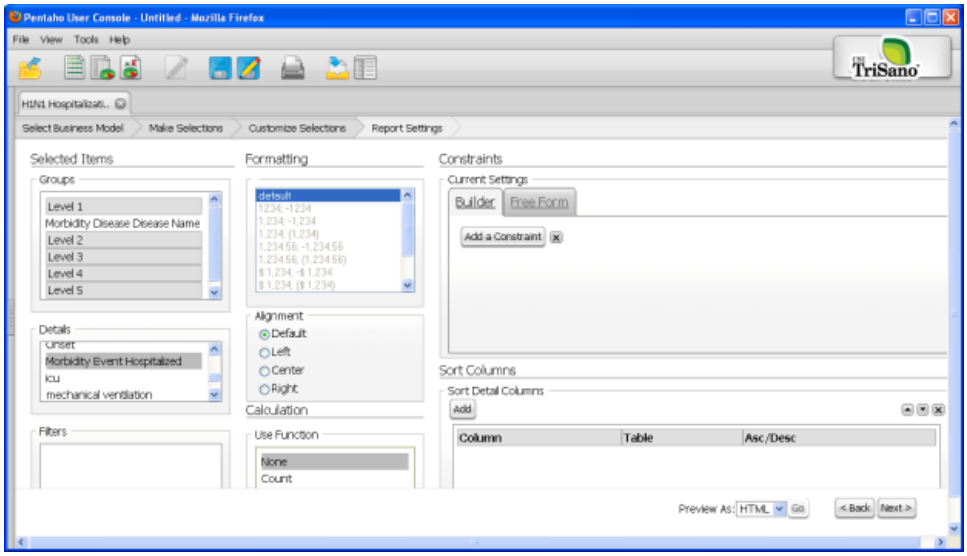
On this screen, you can add constraints for the column, as well as align the heading for this Level 1 Group column.

We are looking only for Morbidity events for the disease "**Novel influenza A virus infections**" that have been hospitalized. Since this query returns information from the CSI TriSano™ core data fields in the Morbidity Events table and the Novel Flu A Form data fields, it automatically only shows information for morbidity events with the disease " **Novel influenza A virus infections** ".

So the only constraint we need to add is for the hospitalized column.

Scroll down the list of columns in the Details box and click on the **Morbidity Event Hospitalized** column.

Your screen should look like this.

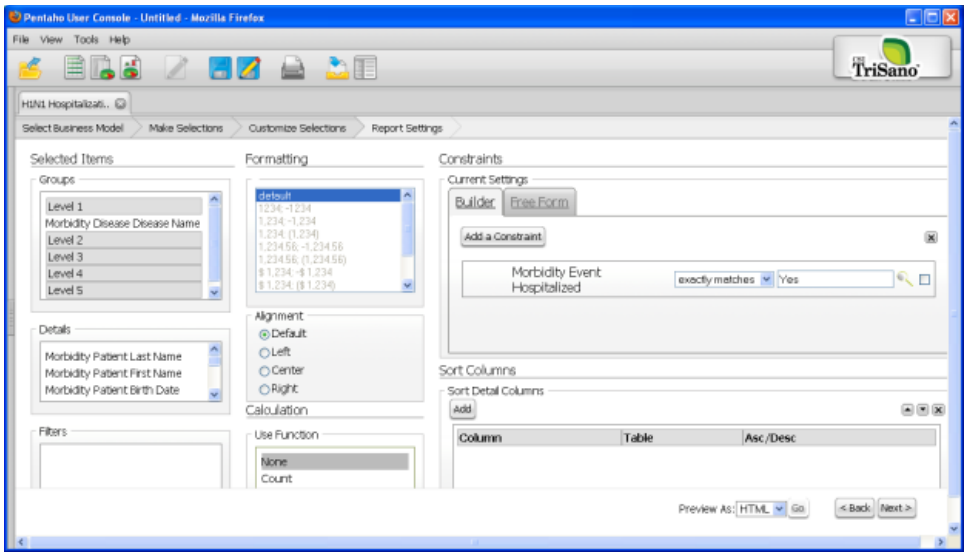


Let's add a constraint by clicking on the **Add a Constraint** button on the right side of the screen. The Ad Hoc Query tool automatically places the column name selected in the constraint.
You can open the drop down box and see that you can have constraints that:

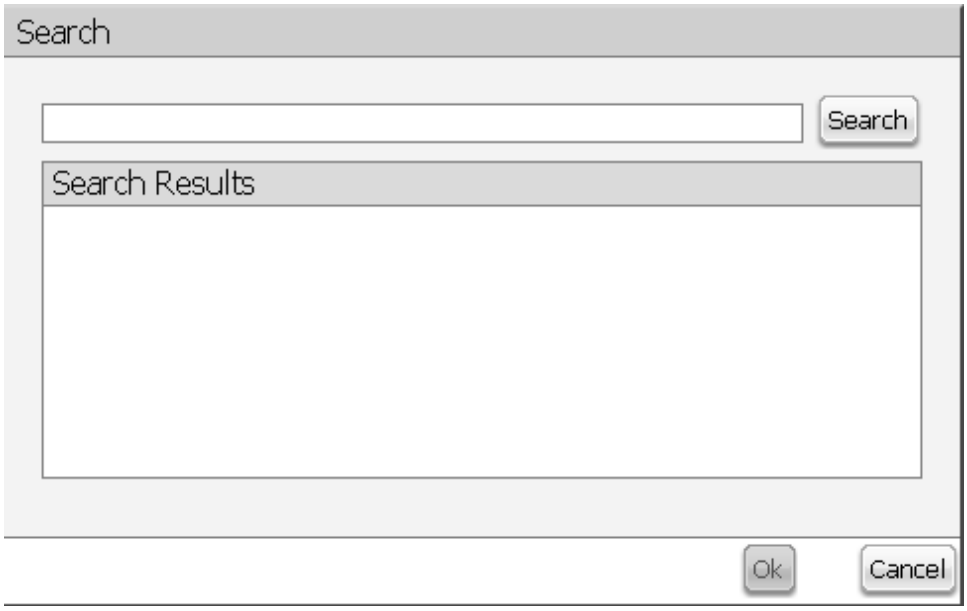
- Exactly matches
- Contains
- Ends with
- Does not contain a specific value
- Is Null (does not contain any value)
- Is Not Null (contains a value).

To find all the events for patients that have been hospitalized, key in Yes in the entry field to the right of "exactly matches", as shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.



You can also open a **Search** box to assist you in creating constraints. To open the **Search** box, click on the magnifying glass to the right of the constraint entry field.

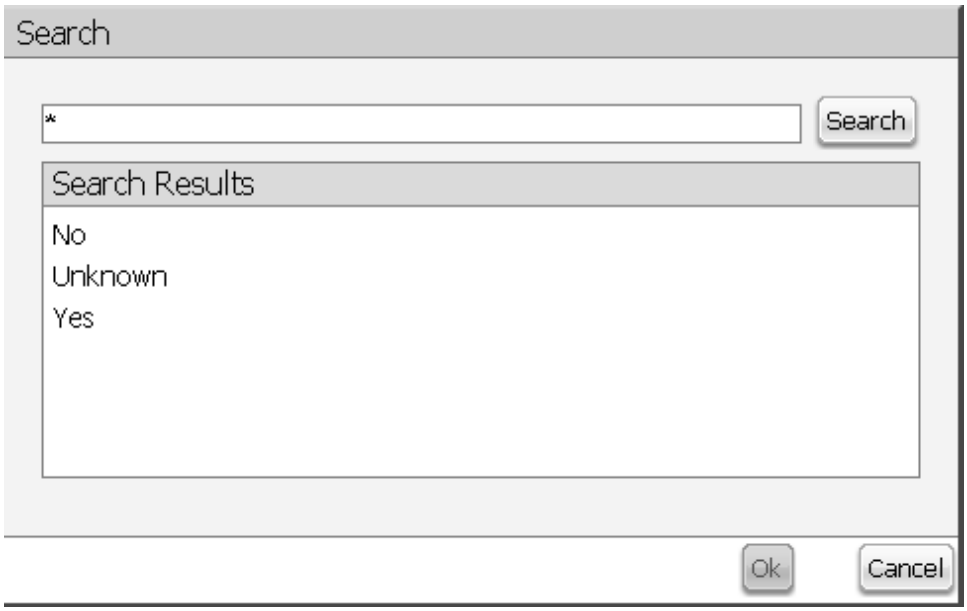


Here are some examples on how to search

- Search is case sensitive.
- Enter an ***** to see all choices
- Enter **A*** to see all choices that start with **A**
- Enter ***A*** to see all choices that have a capital **A** somewhere in the choice
- Entering ***M*** sees all choices with a capital A but would not show choice with a small **m**

After entering your search criteria, click on search. For example, if you enter * for the Morbidity Event Hospitalized constraint and click on Search, the following will appear.

Copyright © 2009 Collaborative Software Initiative, Inc.



To select a value, click on it and then click on **OK**. Your selectiion will be added to the constraint you are defining.

Next Topic:

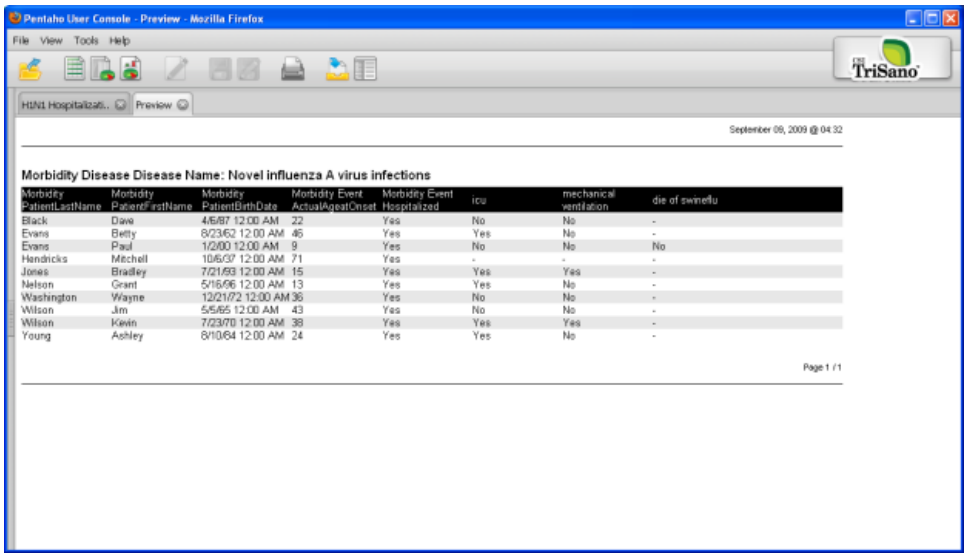
Preview the Report

Copyright © 2009 Collaborative Software Initiative, Inc.

Preview the Report

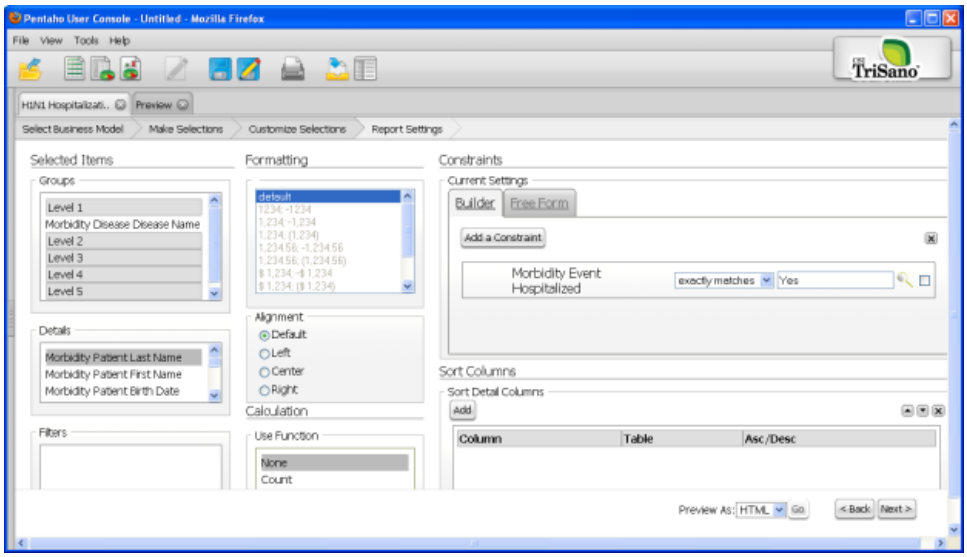
Preview the Report

Click on **Go** to preview the report. You should only see the events for patients that have been hospitalized.



Return back to the query by clicking on the tab that contains the name of the saved query and report.

Next, click on **Morbidity Patient Last Name**, and your screen will look like the one shown below.



Next Topic:

Adding a Column to the Sort Order

Copyright © 2009 Collaborative Software Initiative, Inc.

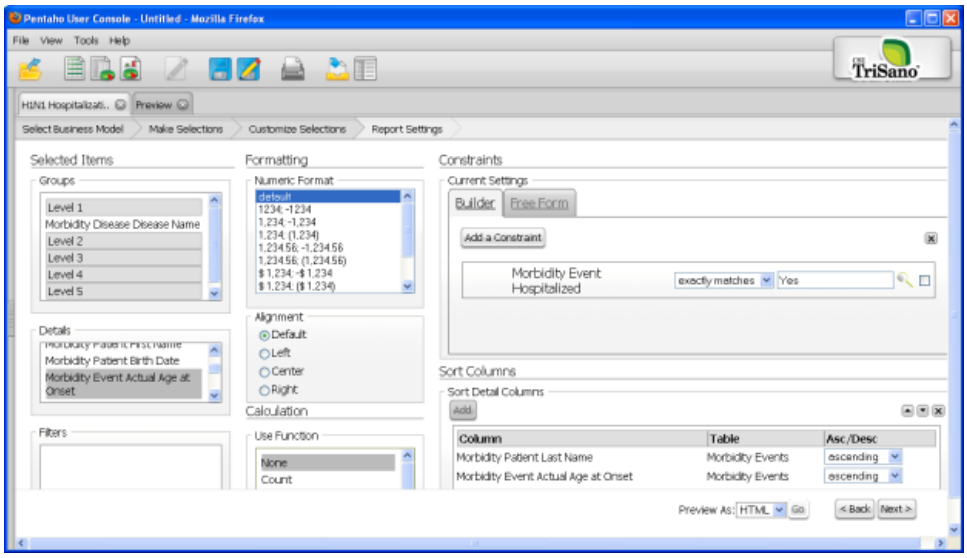
Adding a Column to the Sort Order

Adding a Column to the Sort Order

We would like the **Morbidity Patient Last Name** to be used as one of the sort columns, so click on **Add** under the **Sort Detail Columns**.

Next, do the same for **Morbidity Event Actual Age at Onset** by clicking on the column name then click on **Add**.

Your screen should look like this.



Next Topic:

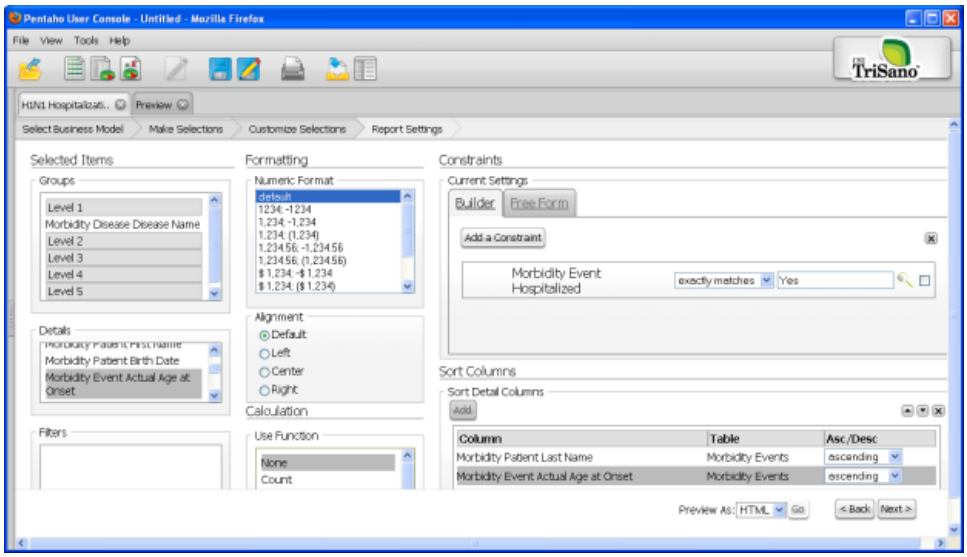
Reordering the Sort Columns

Copyright © 2009 Collaborative Software Initiative, Inc.

Reordering the Sort Columns

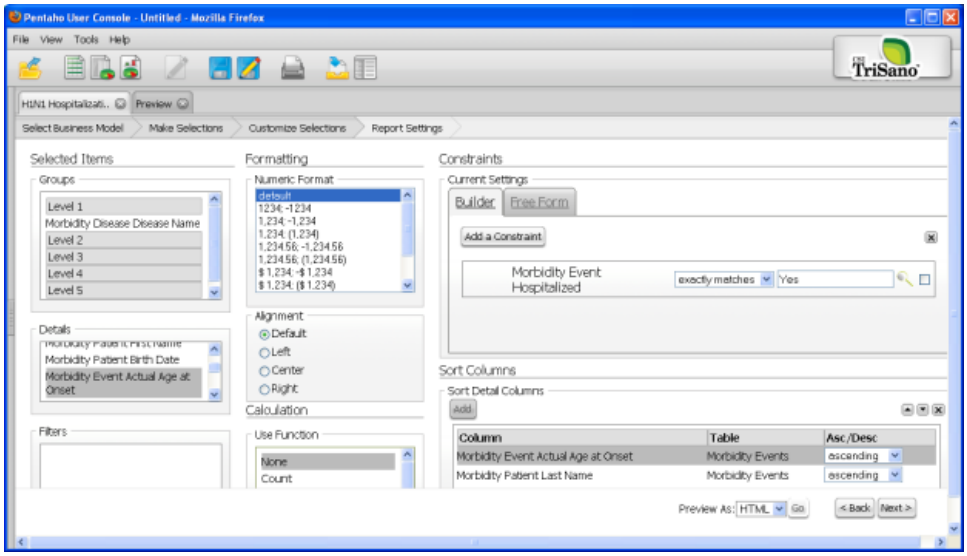
Reordering the Sort Columns

Now we have the two columns as Sort Columns but we really want the **Morbidity Event Actual Age at Onset** to be first in the list, a higher level sort column. To move it to the top of the list, first highlight it as shown below.



Then click on the "up arrow", **^**, in the upper right hand corner of the **Sort Detail columns** box. When you do that, the **Morbidity Event Actual Age at Onset** will move above the **Morbidity Event Last Name** column.

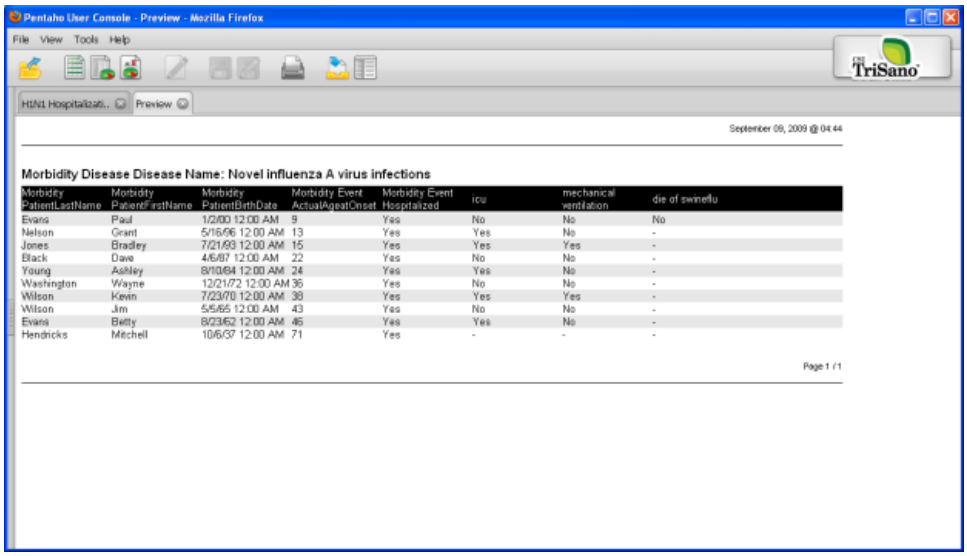
Your screen should look like this.



Preview the Report

Let's see what the report looks like now. Click on **Go** to see it.

Copyright © 2009 Collaborative Software Initiative, Inc.



Morbidity PatientLastName	Morbidity PatientFirstName	Morbidity PatientBirthDate	Morbidity Event ActualAgeatOnset	Morbidity Event Hospitalized	icu	mechanical ventilation	die of swineflu
Evans	Paul	1/2/00 12:00 AM	9	Yes	No	No	No
Nelson	Grant	5/16/96 12:00 AM	13	Yes	Yes	No	-
Jones	Bradley	7/21/93 12:00 AM	15	Yes	Yes	Yes	-
Black	Dave	4/6/87 12:00 AM	22	Yes	No	No	-
Young	Ashley	8/10/84 12:00 AM	24	Yes	Yes	No	-
Washington	Wayne	12/21/72 12:00 AM	36	Yes	No	No	-
Wilson	Kevin	7/23/70 12:00 AM	38	Yes	Yes	Yes	-
Wilson	Jim	5/5/65 12:00 AM	43	Yes	No	No	-
Evans	Betty	8/23/62 12:00 AM	46	Yes	Yes	No	-
Hendricks	Mitchell	10/6/37 12:00 AM	71	Yes	-	-	-

Notice that the report is now sorted by age, and within age, if there are events with the same age, by last name, just what we want…. at least for now.

Close the report and return to the query.

Next Topic:

Calculating the Average for a Column

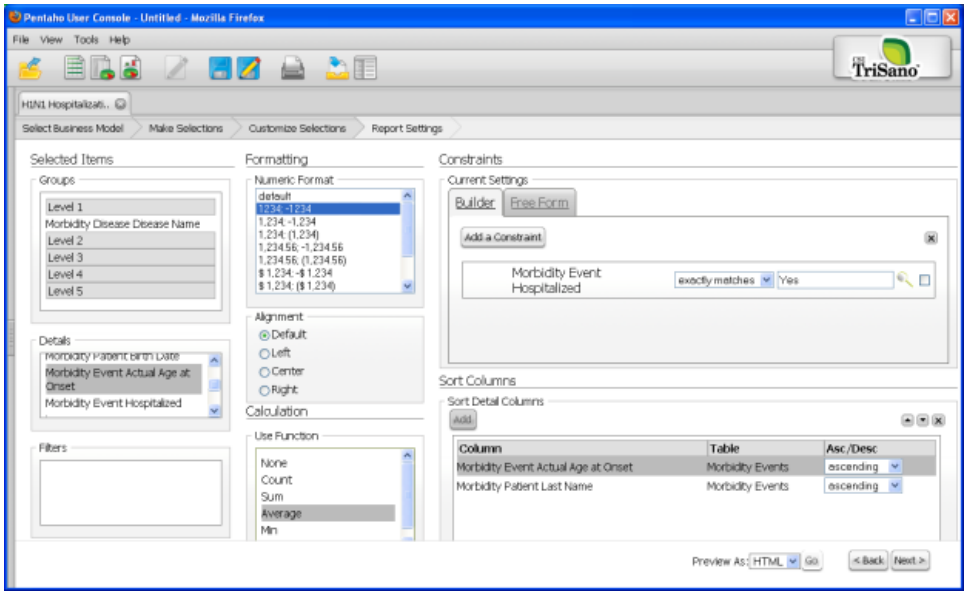
Copyright © 2009 Collaborative Software Initiative, Inc.

Calculating the Average for a Column

Calculating the Average for a Column

Let's do some additional formatting. Make sure the **Morbidity Event Actual Aage at Onset** column on the left is selected.

To have the report calculate the Average age, click on **Average** in the **Use Function** box, and on **1234,-1234** in the **Numeric format** box as shown below.



Remember, you can preview the report at any time by clicking on **Go**.

Next Topic:

Reformatting a Column

Copyright © 2009 Collaborative Software Initiative, Inc.

Reformatting a Column

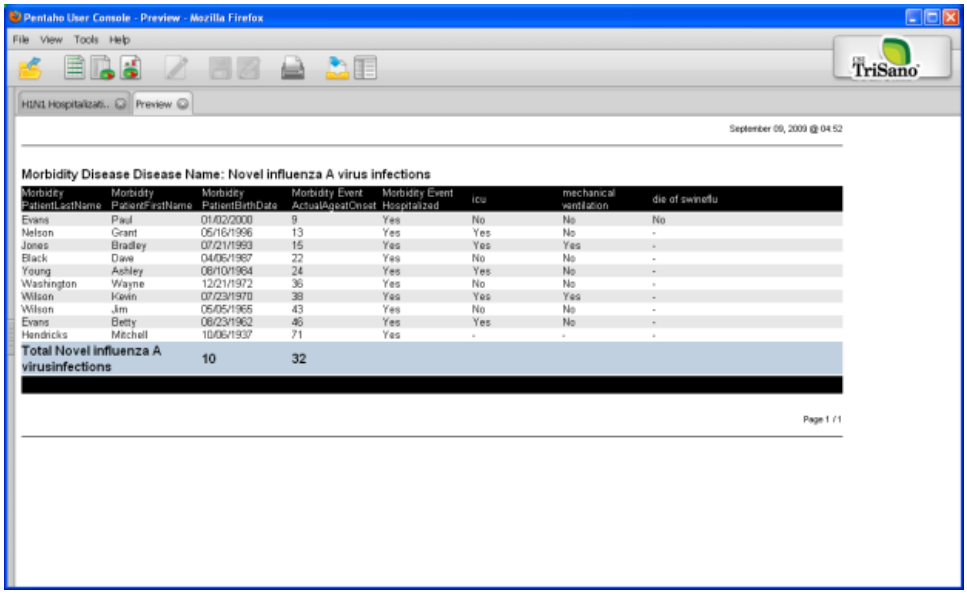
Reformatting a Column

You may have noticed that the **Morbidity Patient Birth Date** column display not only the date but also a time of **12:00 AM**. We want to get rid of that.

To do that, click on **Morbidity Patient Birth Date** as shown below. We want to change the **Morbidity Patient Birth Date** so that the format is **MM/DD/YYYY**, and we want to use this column to **COUNT** the number of cases of "**Novel influenza A virus infections** " that we have that have been hospitalized.

To do this, click on **02/21/2007** in the **Date Format** box, and on **Count** in the **Use Function** box as shown below, and then on **Go** to preview the report.

Your report should look like this.



Notice the "**10**" under the **Birth Date**, which is the number of patients that have been hospitalized, and the "**32** " is the **average age of onset**.

Close the report and return to the query.

Next Topic:

Changing the Summary Line Text

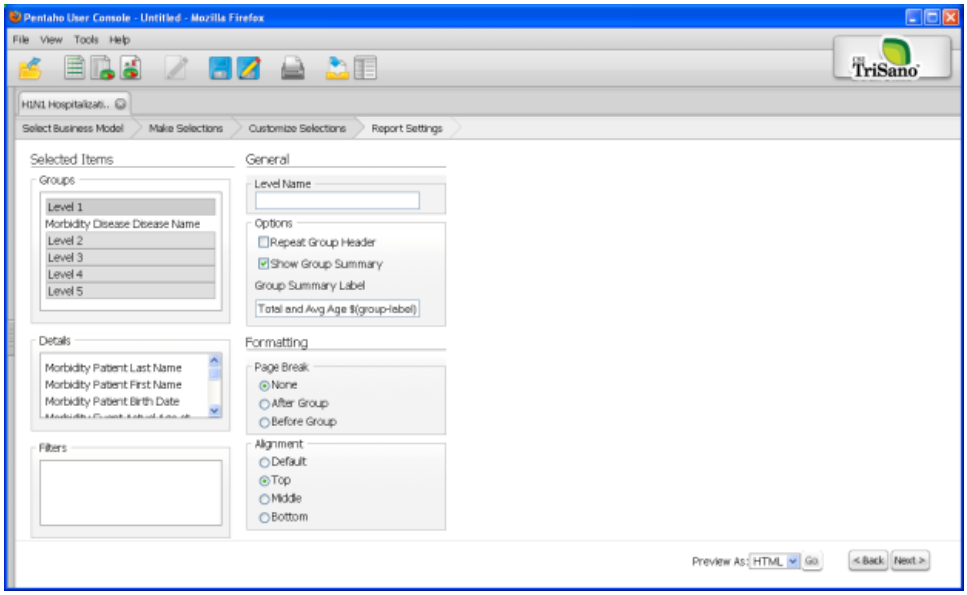
Copyright © 2009 Collaborative Software Initiative, Inc.

Changing the Summary Line Text

Changing the Summary Line Text

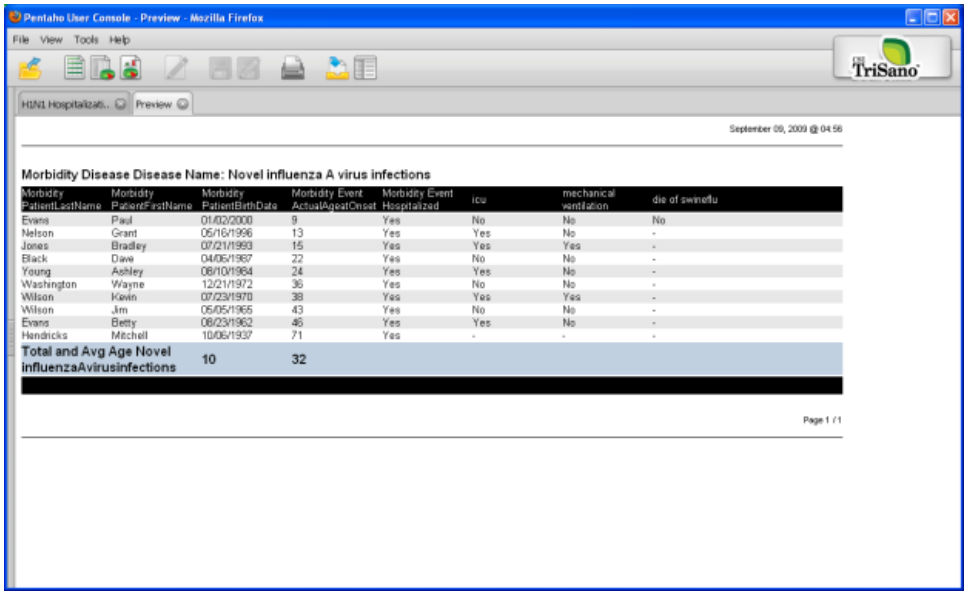
You can also change the text that shows up on the summary line. Right now, it says **Total** followed by the disease name, "**Novel influenza A virus infections** ".

To do this, click on **Level 1** in the **Groups** box and change the **Group Summary Label** from **Total $(group-label)** to **Total and Avg Age $(group-label)** as shown below.



Preview the Report

Let's see the report again. Notice the changed summary line at the bottom of the report.



Next Topic:

Returning to the Make Selections Screen

Copyright © 2009 Collaborative Software Initiative, Inc.

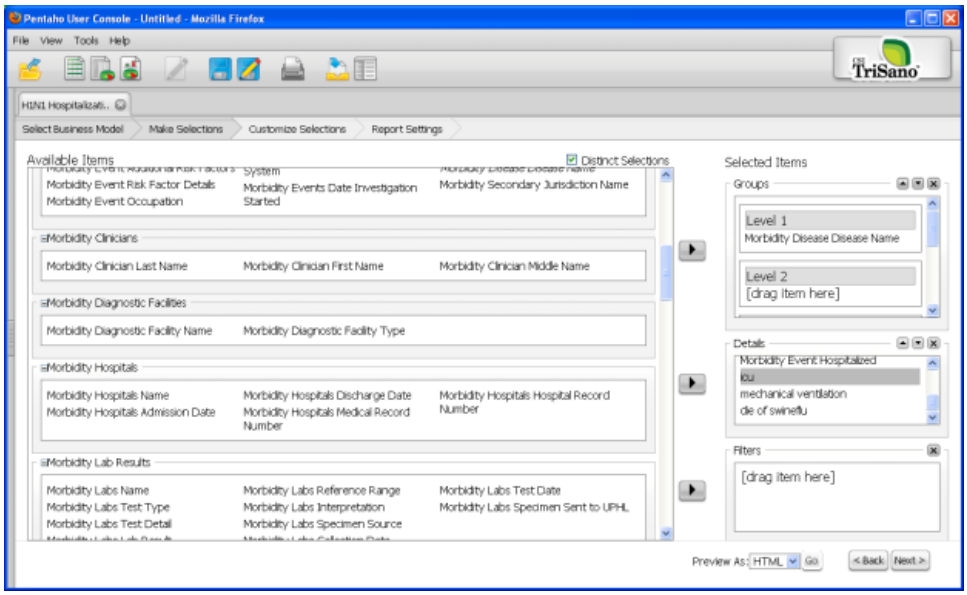
Returning to the Make Selections Screen

When you look at this report, you might notice that the **Yes/No/Unknown** values for **icu** are not grouped together. This report would be more useful if we grouped those together. That's easy to do.

Begin by returning to the query by closing the report.

Click on **Back** to go back to the previous step in building the report.

You can accomplish what we want to do by specifying a second **Group** level, **Level 2**, as the column name **icu**. To do this, click on the **icu** in the **Details** box as shown below.



Next Topic:

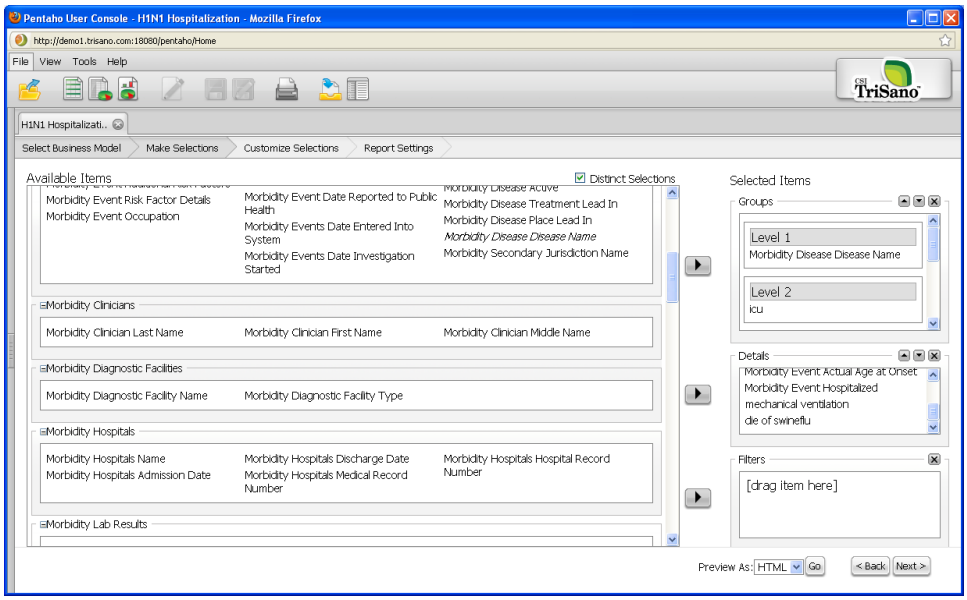
Adding a Second Column for Grouping

Copyright © 2009 Collaborative Software Initiative, Inc.

Adding a Second Column for Grouping

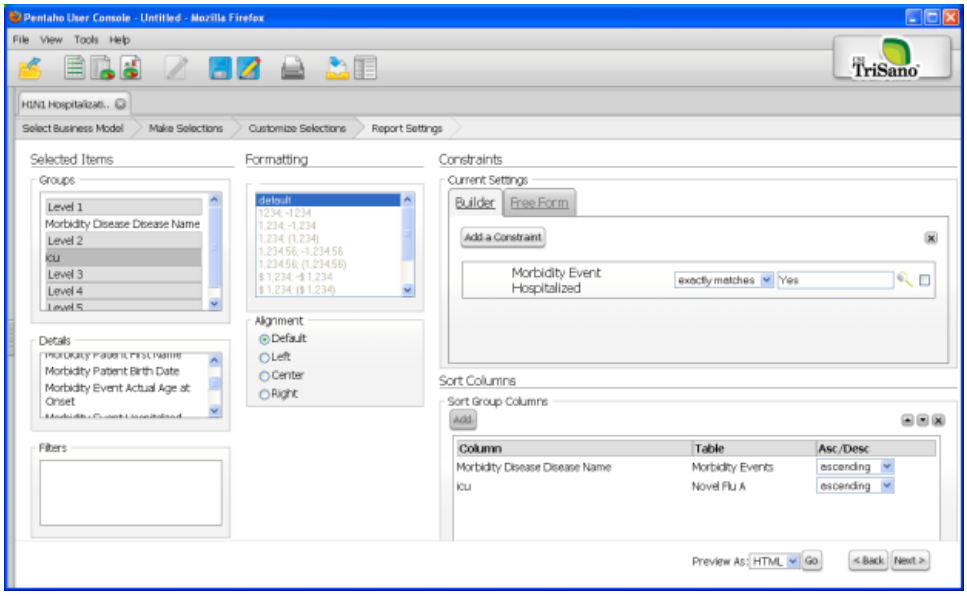
Adding a Second Column for Grouping

Now drag that column name to the **Level 2** entry field. Your screen should look like the one shown below. Notice that the **icu** column is no longer part of the detail for each row of the report.



Now, go on to the next step by clicking on **Next**, and click on the **icu** column and you'll see that the column has been added to the sort list automatically.

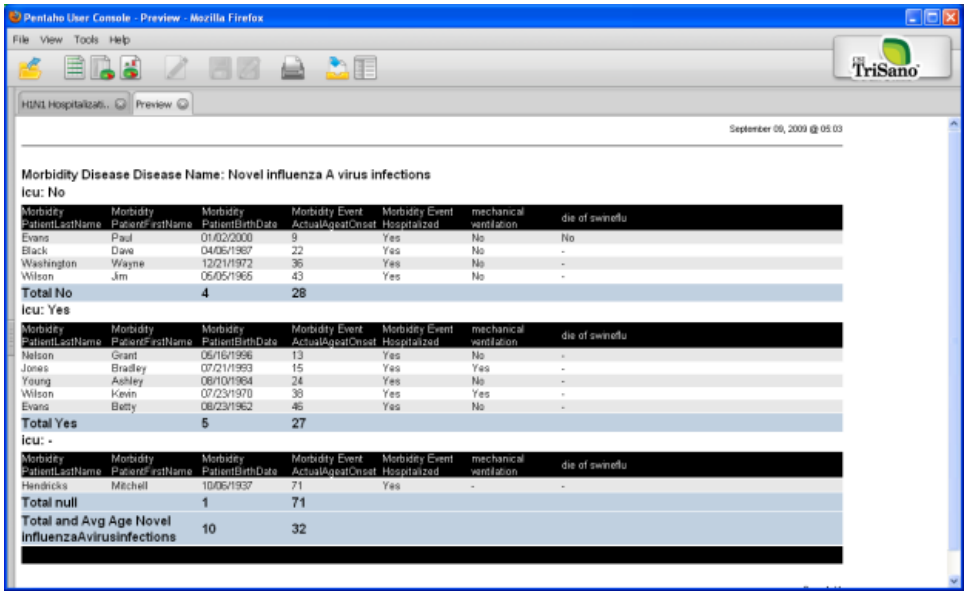
Your screen should look like this.



Preview the Report

Let's see the report again.

Copyright © 2009 Collaborative Software Initiative, Inc.



Notice that the rows are now grouped by the value in the **icu** column, first **No**, then **Yes**, then – (meaning a value was not entered).

Return to the report.

Next Topic:

Changing the Text for the Second Group

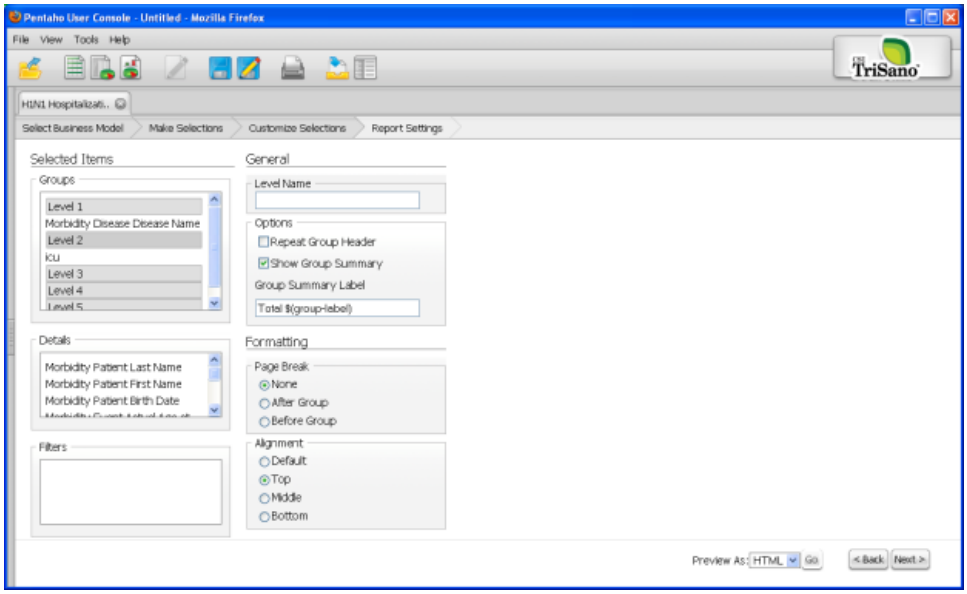
Copyright © 2009 Collaborative Software Initiative, Inc.

Changing the Text for the Second Group

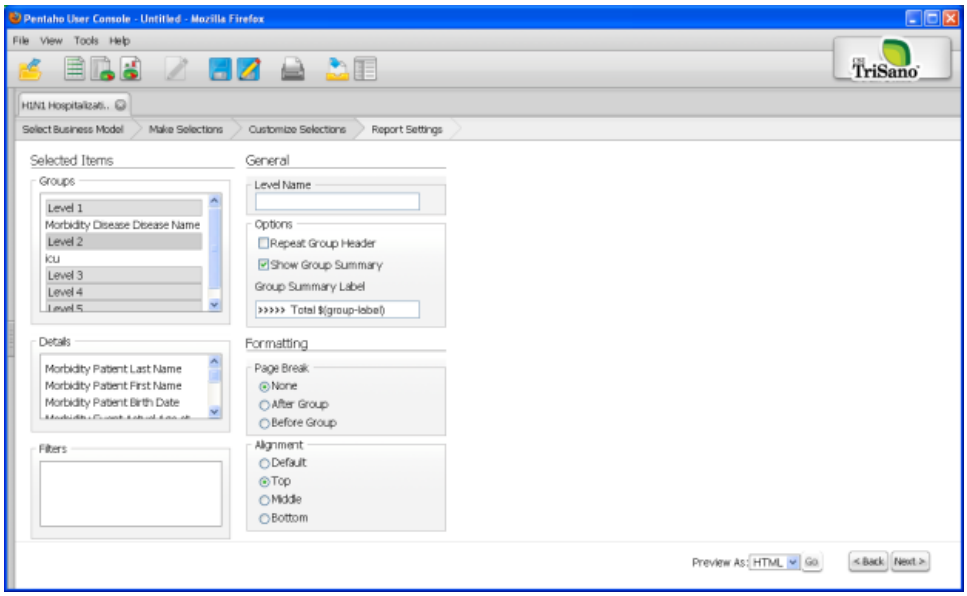
Changing the Text for the Second Group

Let's change the text for the **icu** group level, level 2, to add some ">>>"'s to the beginning just so it shows up better.

Click on **Level 2** in the **Groups** box.



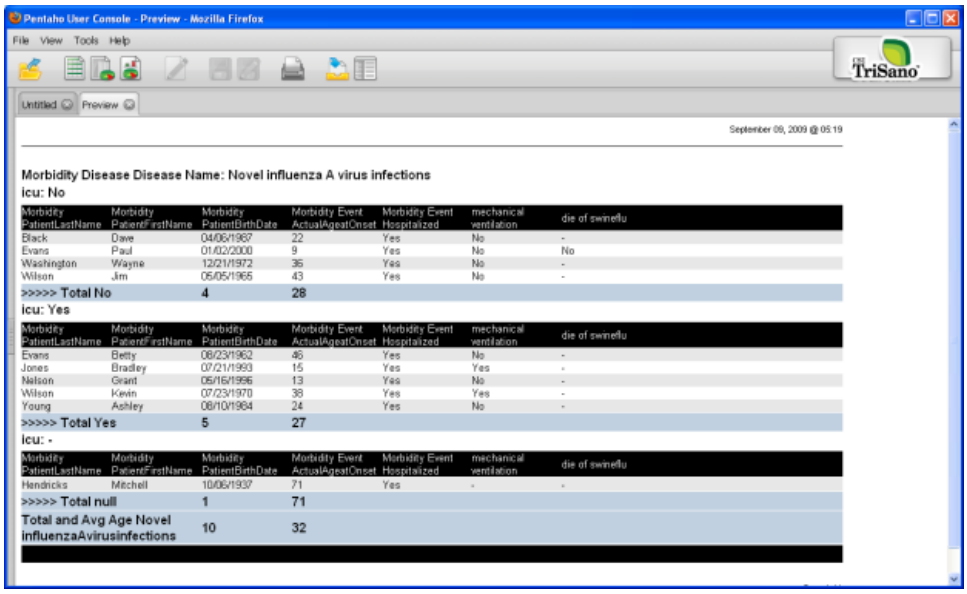
Now add some "**>>>"**'s to the **Group Summary Label** as shown below.



Preview the Report

And view the report.

Copyright © 2009 Collaborative Software Initiative, Inc.



Next Topic:

Adding Report Settings

Copyright © 2009 Collaborative Software Initiative, Inc.

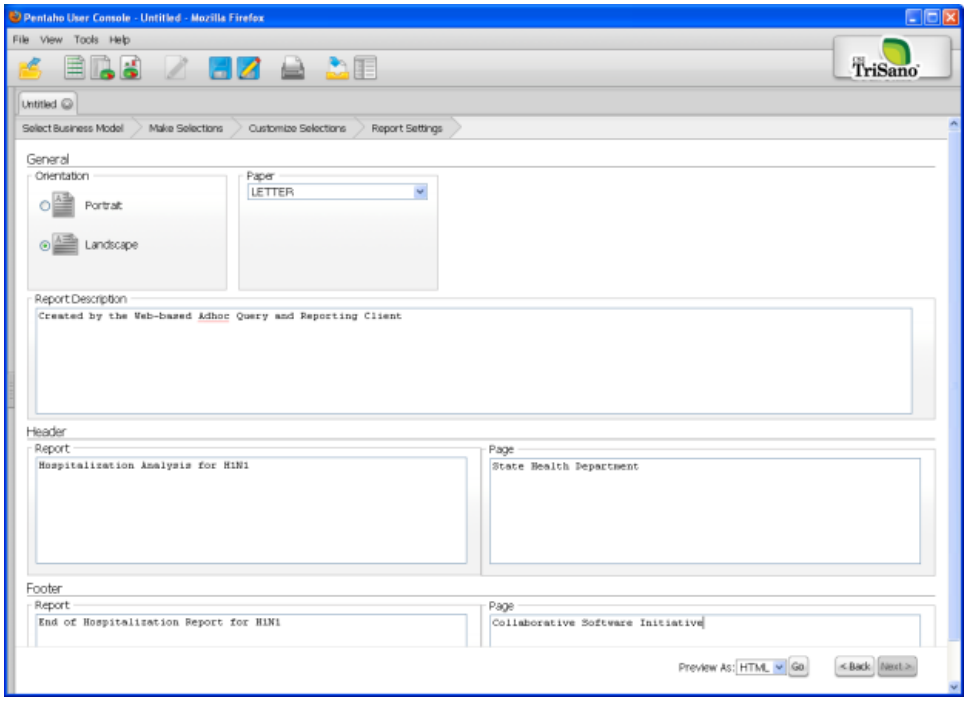
Adding Report Settings

Adding Report Settings

That looks pretty good.

Let's return to the report and move to the next and last step, adding headers and footers.

Close the report and return to the query, then click on **Next** to go to the final step.



Here, I've already keyed in:

Hospitalization Analysis for H1N1 for the Report Header

State Health Department for the Page Header

End of Hospitalization Report for H1N1 for the Report Footer

and **Collaboration Software Initiative** for the Page Footer

Next Topic:

Viewing the Report in HTML Format

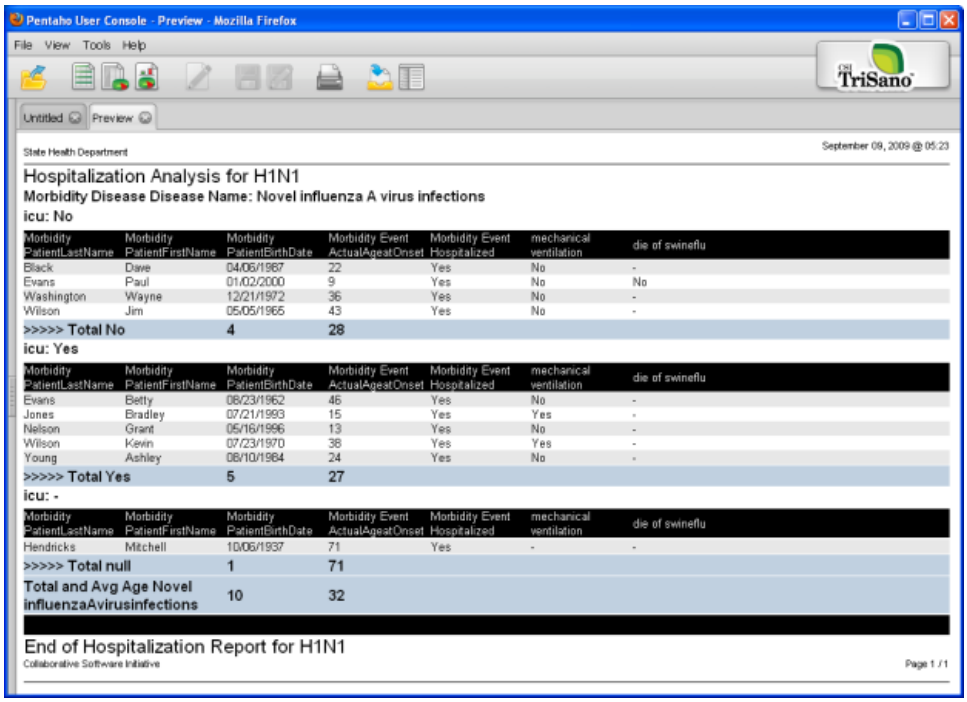
Copyright © 2009 Collaborative Software Initiative, Inc.

Viewing the Report in HTML Format

Viewing the Report in HTML Format

Now view the report for the last time.

Here it is in HTML format:



Next Topic:

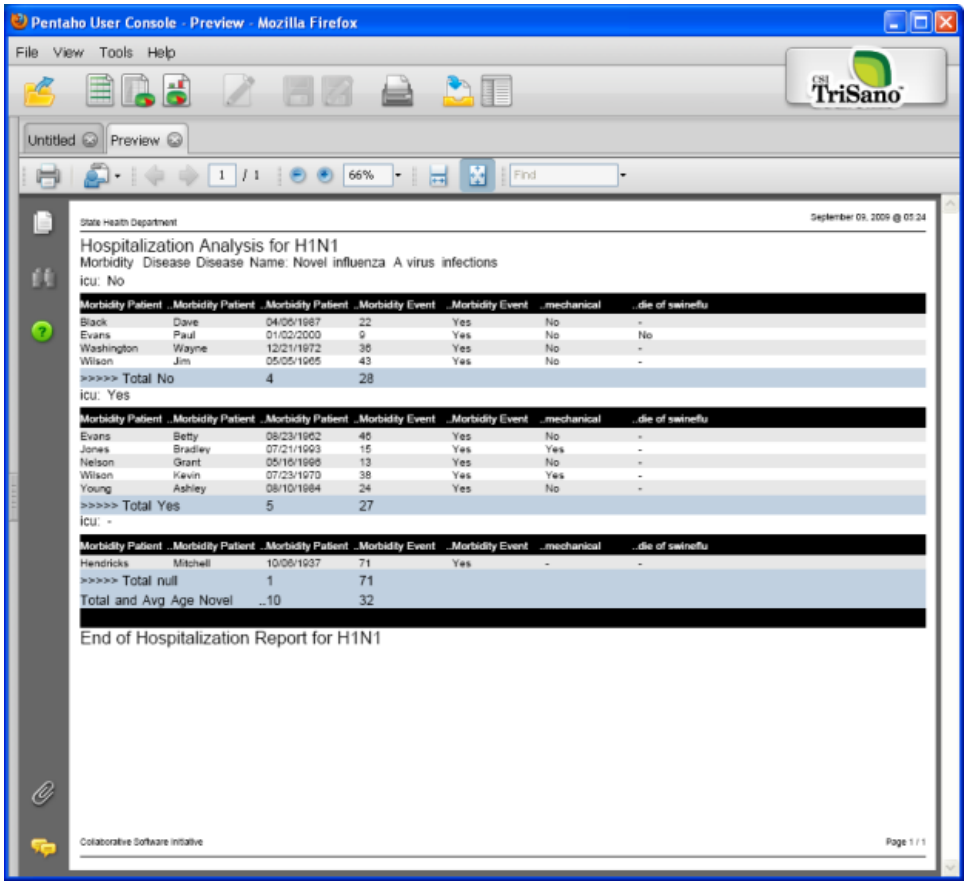
Viewing the Report in PDF Format

Copyright © 2009 Collaborative Software Initiative, Inc.

Viewing the Report in PDF Format

Viewing the Report in PDF Format

and here it is in PDF format.



Next Topic:

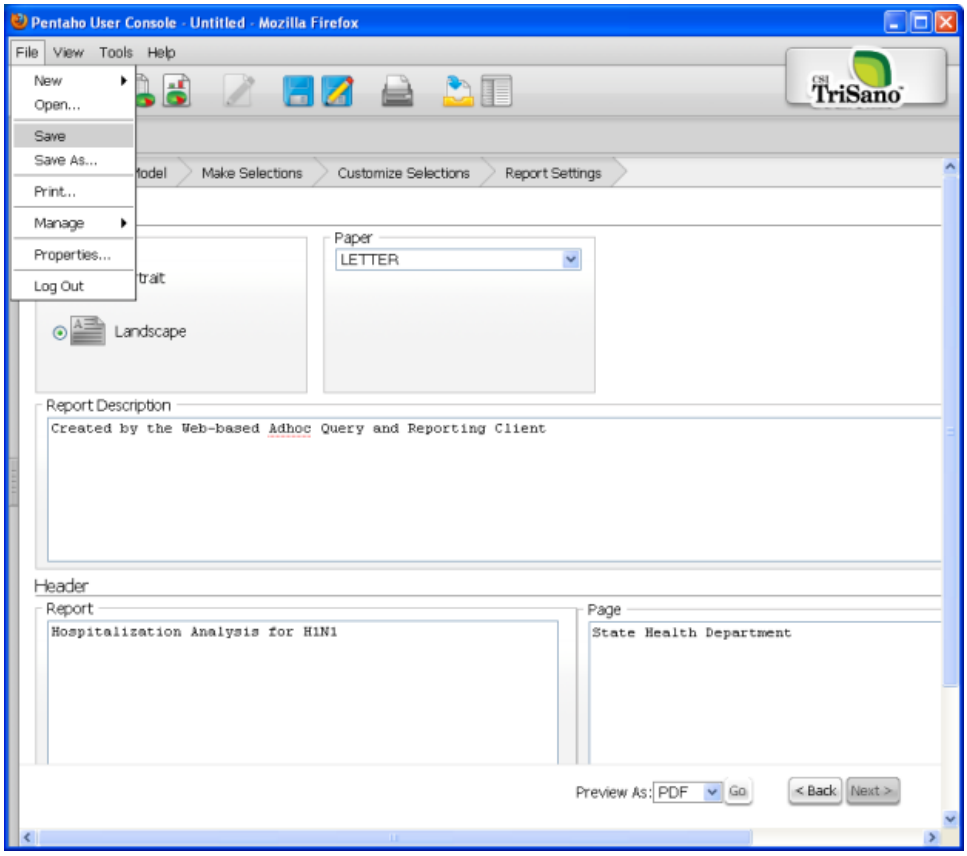
Save the Report

Copyright © 2009 Collaborative Software Initiative, Inc.

Save the Report

Save the Report

Return to the report and save it again.



That's it.

You've seen many of the features of the Pentaho™ Ad Hoc Query tool and report writer.

You should now be able to explore and learn more on your own.

Good luck.

Next Topic:

Pentaho™ Report Designer Tool

Copyright © 2009 Collaborative Software Initiative, Inc.

Pentaho™ Report Designer Tool

The following is an example of using the Pentaho™ Report Designer to create a query and report for core and disease-specific Form Builder data. The example uses a table containing answers from an H1N1 form. Your system will potentially have a similar form.

Create sample report

Here is the description of the report we will be building in this section.

Create a report showing patients last name, first name, age in years, hospitalized?, in ICU?, on Ventilation?, Died?, Died of Swineflu?

The following shows how to use the Pentaho™ Report Designer included with CSI TriSano™ in a Windows environment to create this report.

You may need to install the Pentaho™ Report Designer on your Windows system and establish connectivity to a CSI TriSano™ data warehouse before you begin this series of steps. Those instructions are included elsewhere.

This report was developed using a sample database with fictitious data.

Next Topic:

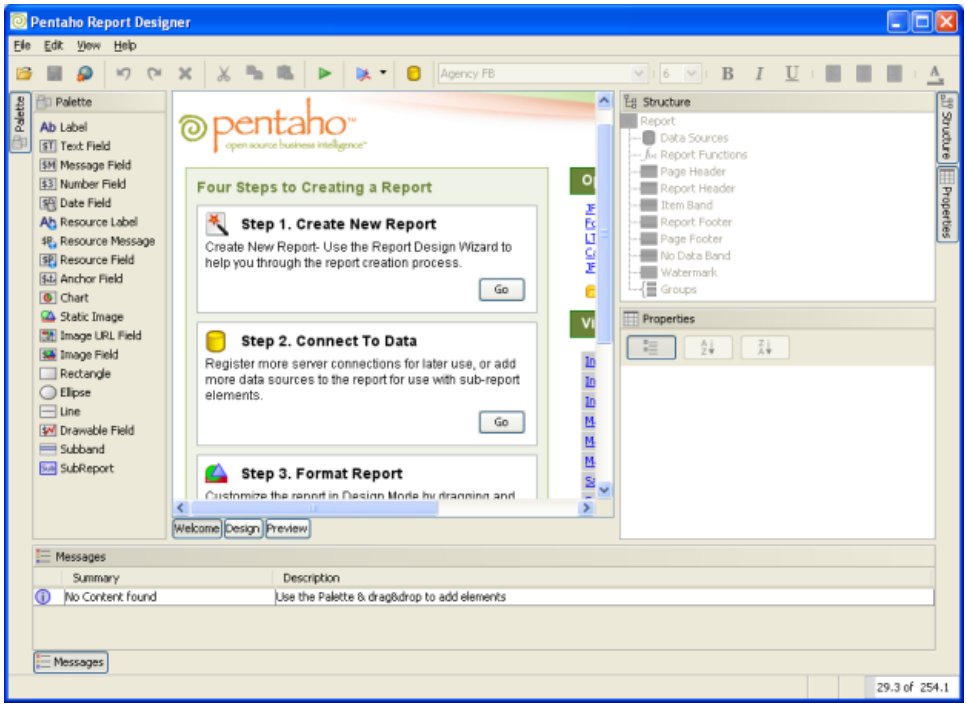
Start the Report Designer

Copyright © 2009 Collaborative Software Initiative, Inc.

Start the Report Designer

Start the Report Designer

To create this report, first start the **Report Designer**. The initial **Report Designer** screen will appear.



The Report Designer Interface contains several sections:

Section	Description
Toolbar	Contains icons for file, data, publishing and text formatting operations
Palette	Contains design tools for use in creating reports
Structure	Shows the data source and report elements
Properties	Shows the properties, or configurable options, of the selected report elements
Tabbed Views	These three views are shown in the middle panel of the Report Designer screen
• **Welcome**	Provides direct access to tthe four-step Report Design Wizard
• **Design**	Allows you to design and modify the different elements of the report
• **Preview**	Lets you preview the finished repot
Messages	Displays messages that occur during the operation of Report Designer

//

Let's use the Report Design Wizard to begin the process of building our report.
//

Copyright © 2009 Collaborative Software Initiative, Inc.

Next Topic:

Start the Report Design Wizard

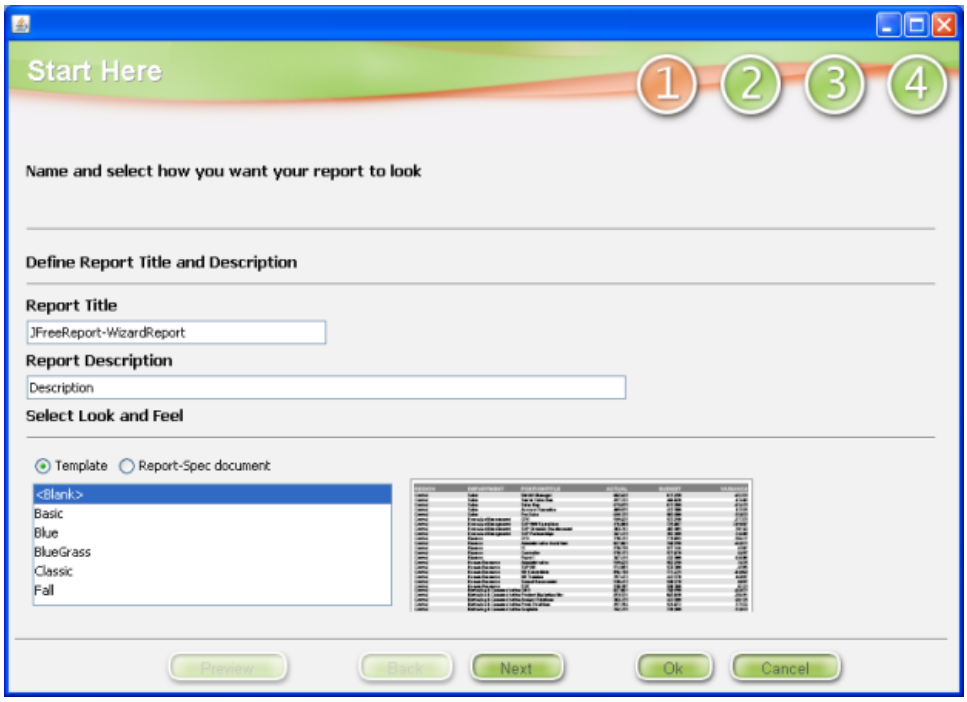
Copyright © 2009 Collaborative Software Initiative, Inc.

Start the Report Design Wizard

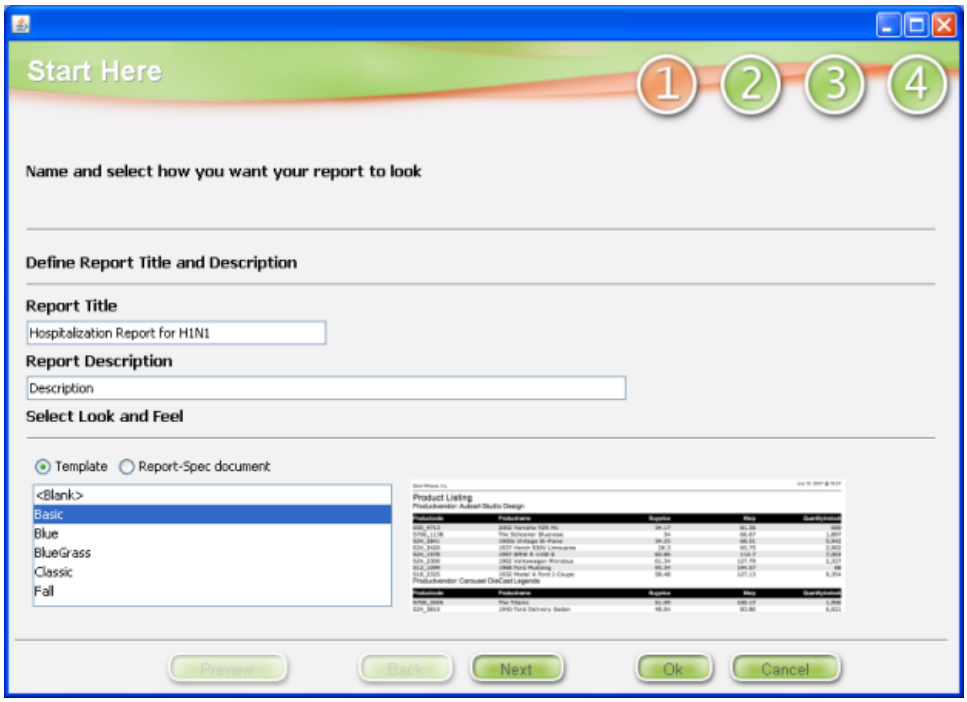
Start the Report Design Wizard

Start the **Report Design Wizard** by clicking on the **Go** button in **Step 1. Create New Report.**

The **Report Wizard** opens as shown below.



On this screen, you can enter a **Report Title** and **Description** and pick the **Template** you would like to use for the report. Enter in **Hospitalization for H1N1** for the report title, and choose **Basic** for the report template, as shown below.



Copyright © 2009 Collaborative Software Initiative, Inc.

Now, press **Next** to go to Step 2.

Next Topic:

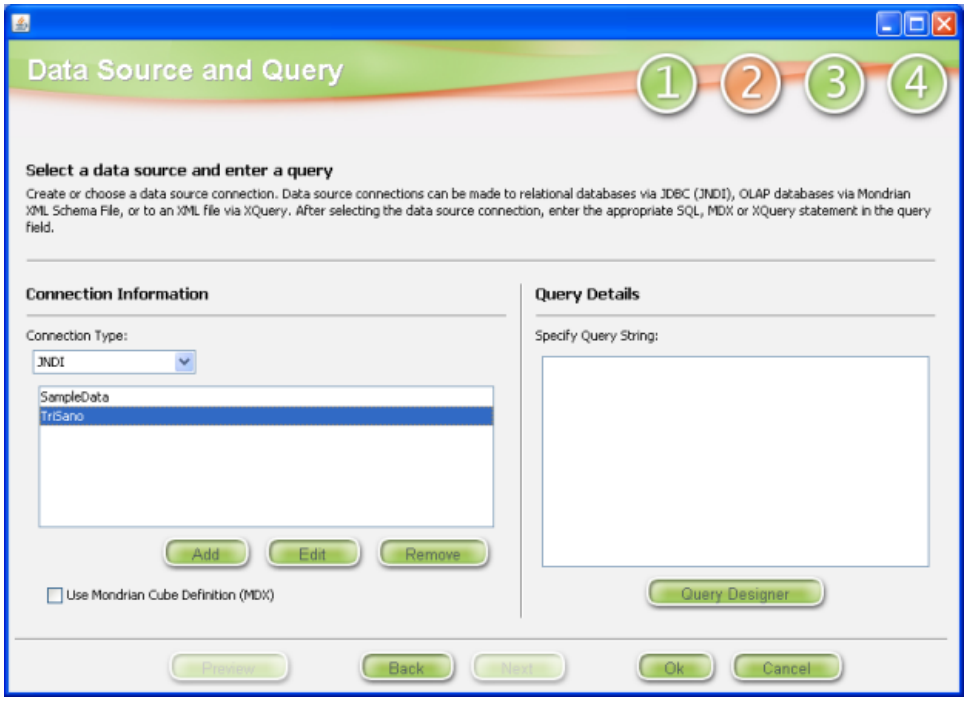
Data Source and Query

Copyright © 2009 Collaborative Software Initiative, Inc.

Data Source and Query

Data Source and Query

The second step is to define the source of our data.



Here you can create or choose the connection to be used to access the CSI TriSano™ data warehouse. To **Add** or **Edit** a connection, use the buttons under the **Connection Information** section. For our demo database, I used the following for CSI TriSano™:

JNDI Name:	TriSano
Driver:	org.postgresql.Driver
Connect string:	jdbc:postgresql://localhost:5432/trisano_warehouse

Your information for these will probably be different.

The next step is to define the query that will be executed against the CSI TriSano™ data warehouse database. You can either key in the query string in the **Specify Query String** box under **Query Details**, or click on **Query Designer** to use a graphical tool to easily build your query. We will use the **Query Designer** to create our query.

Next Topic:

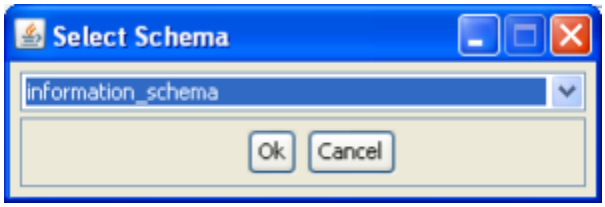
Query Designer

Copyright © 2009 Collaborative Software Initiative, Inc.

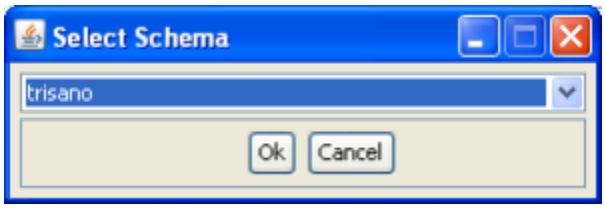
Query Designer

Query Designer

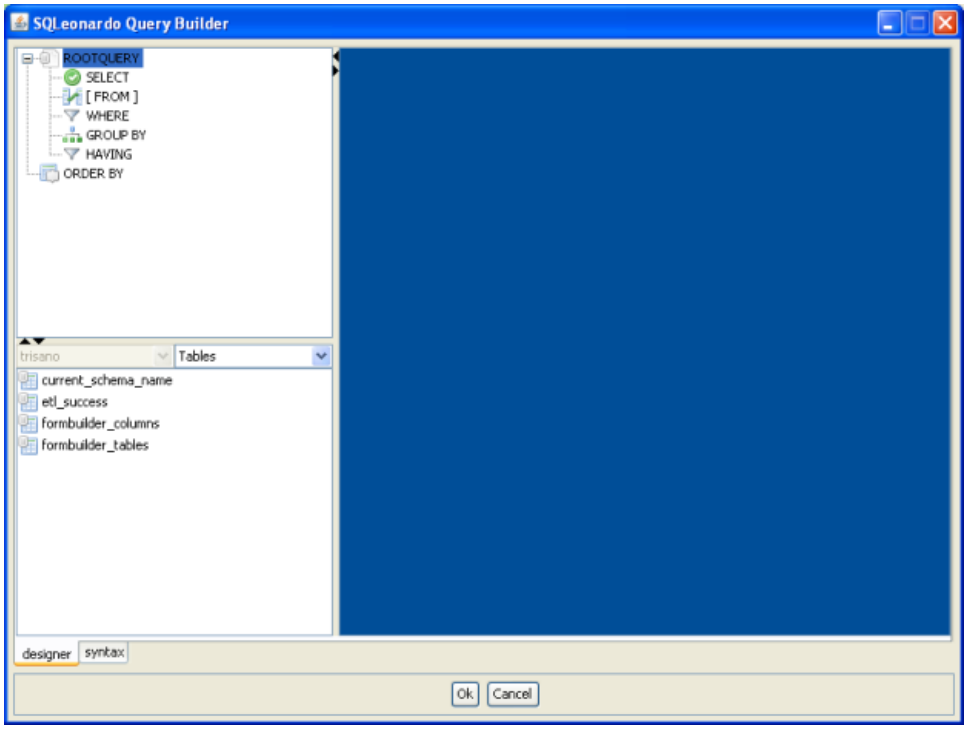
Click on **Query Designer** to begin to create the database query to be used for the report. The **Select Schema** box will appear.



Here you will select the database schema, which contains the tables and views in the data warehouse, to be used for the query. Ours is called **trisano**, so click on the down arrow and select **trisano**, as shown below.



After you click on **OK**, the Query Builder screen will display.



There is a lot you can do from this window. As you can see, it is broken up into three sections:

- In the top left corner is a hierarchical view of the SQL (Structured Query Language) query. As you create the query, information will be added to this section. It is in this section where you define the sort order of the data being retrieved, and other aspects of the SQL statement.
- In the bottom left corner is a view of the **trisano** schema. Right now it is showing a listing of the tables in the schema.
- On the right side of the window is a graphical query editor. As you select tables or views to be added to the query, they will be displayed here. It is here where you will select the columns to be added to

Copyright © 2009 Collaborative Software Initiative, Inc.

the query, and define how multiple tables will be related, or joined, together.

Next Topic:

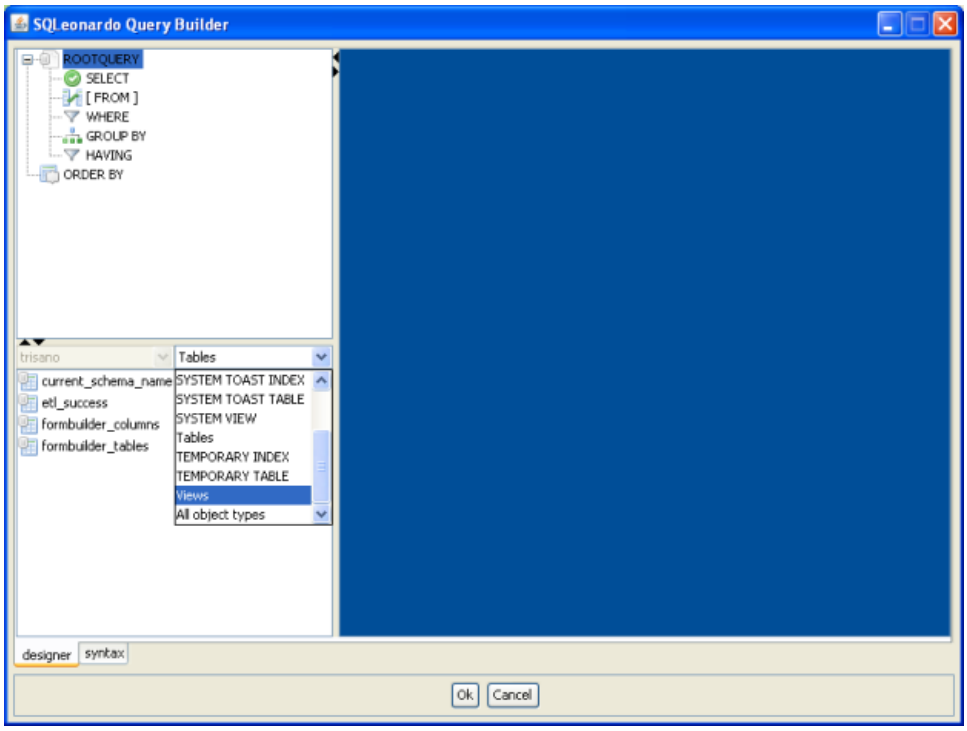
Select Views for Query

Copyright © 2009 Collaborative Software Initiative, Inc.

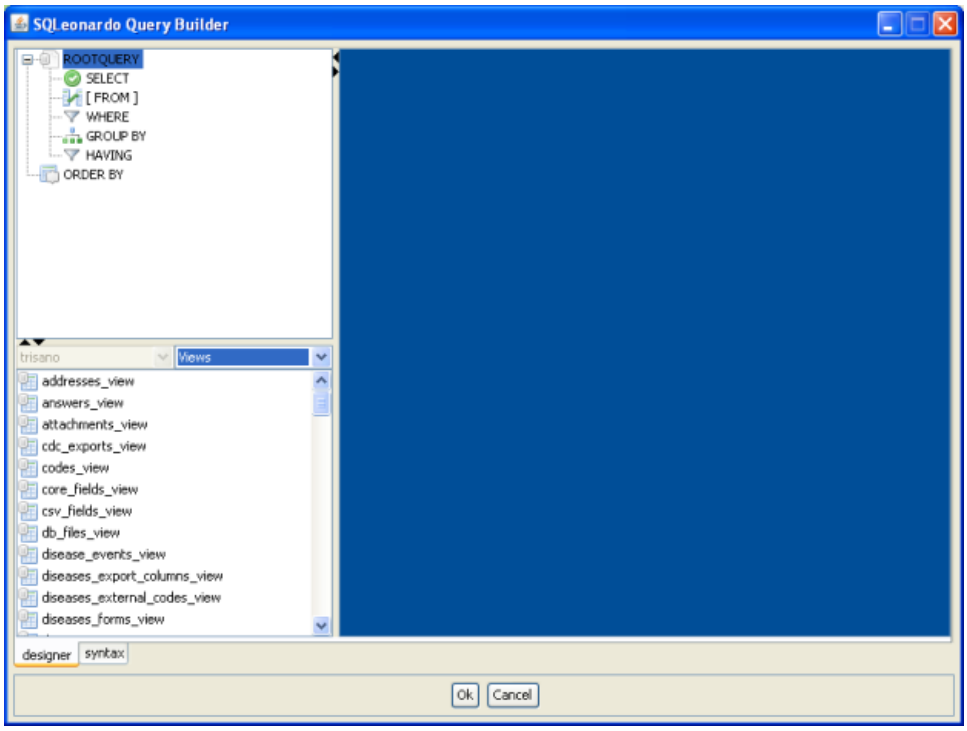
Select Views for Query

Select Views for Query

The information that we want to use has been defined in the **trisano** schema as a series of views. To access the views, click on the down arrow to the right of **Tables** in the schema section (lower left hand corner) of the window. You'll see additional choices, one of which is **Views** as shown below.



Click on **Views**, and a listing of available views will be displayed in the schema section.

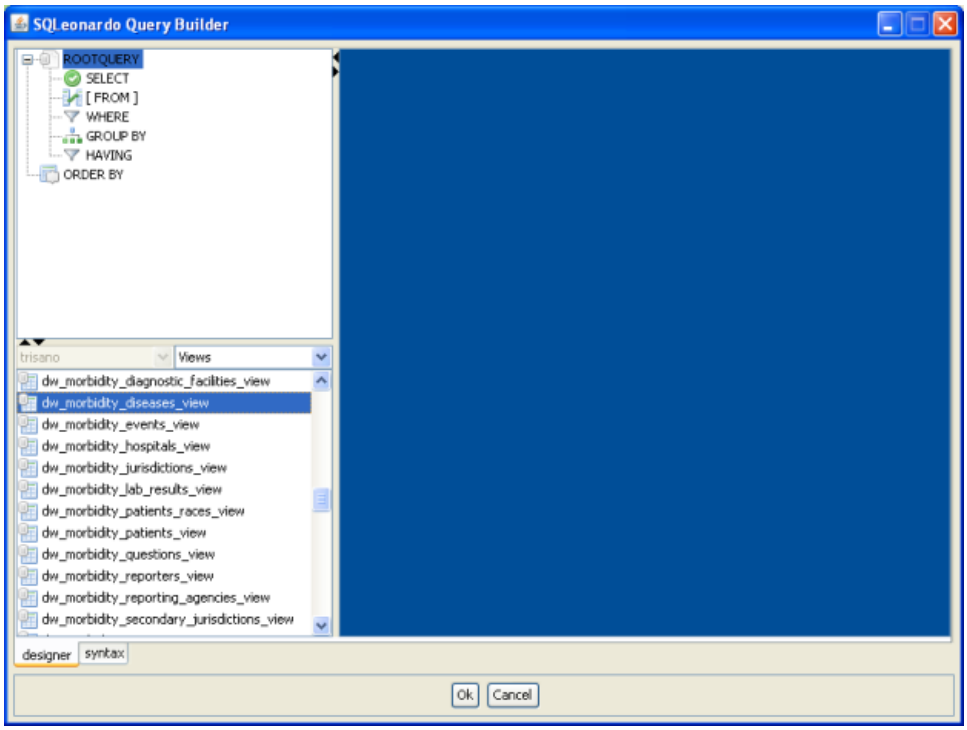


Copyright © 2009 Collaborative Software Initiative, Inc.

You are going to build a query that uses information about morbidity events, patients and diseases, along with a form used for H1N1 cases.

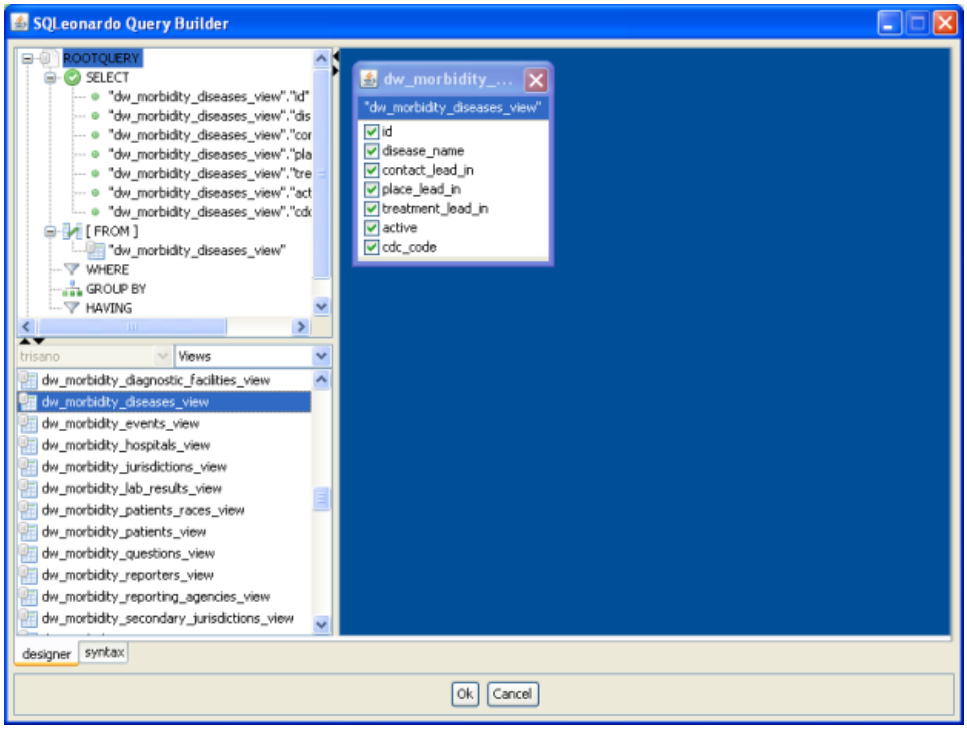
Scroll down through the list of views (about half way) until you come to the views that start with **dw_morbidity....**

The first one you will be using is **dw_morbidity_diseases_view** as shown below.



This view, **dw_morbidity_diseases_view**, is one we need for our query. To add it to the graphical query window on the right side of the screen, double-click on the view name in the lower left hand section of the window.

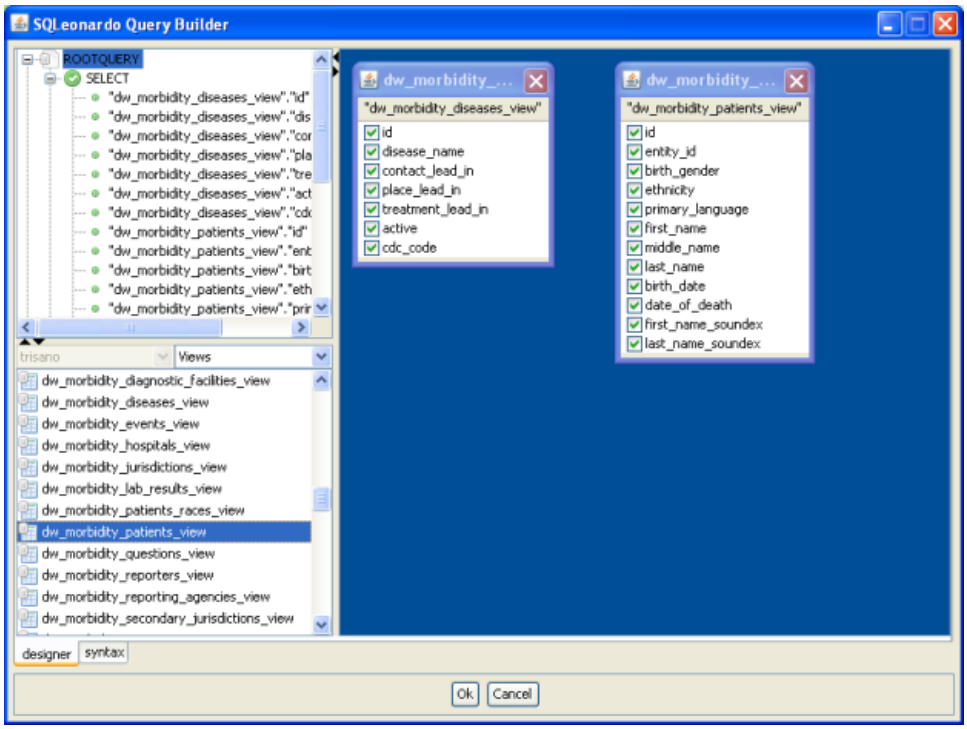
Your screen should look like the one shown below with the **dw_morbidity_diseases_view** displayed on the right.



Copyright © 2009 Collaborative Software Initiative, Inc.

Notice that the name of the view is shown right under the title bar of the small window, and that the column names for all the columns in the views are displayed under that. The default is to select all the columns in the view to be displayed as part of the query. You should also notice that all the columns have been added to the **SELECT** statement in the upper left hand section of the screen. You will change the selected columns a little later, but for now select another view to be added to the query. Find the view **dw_morbidity_patients_view** in the lower left hand section, and double click it to add it to the query on the right.

Your screen should look like the one shown below.

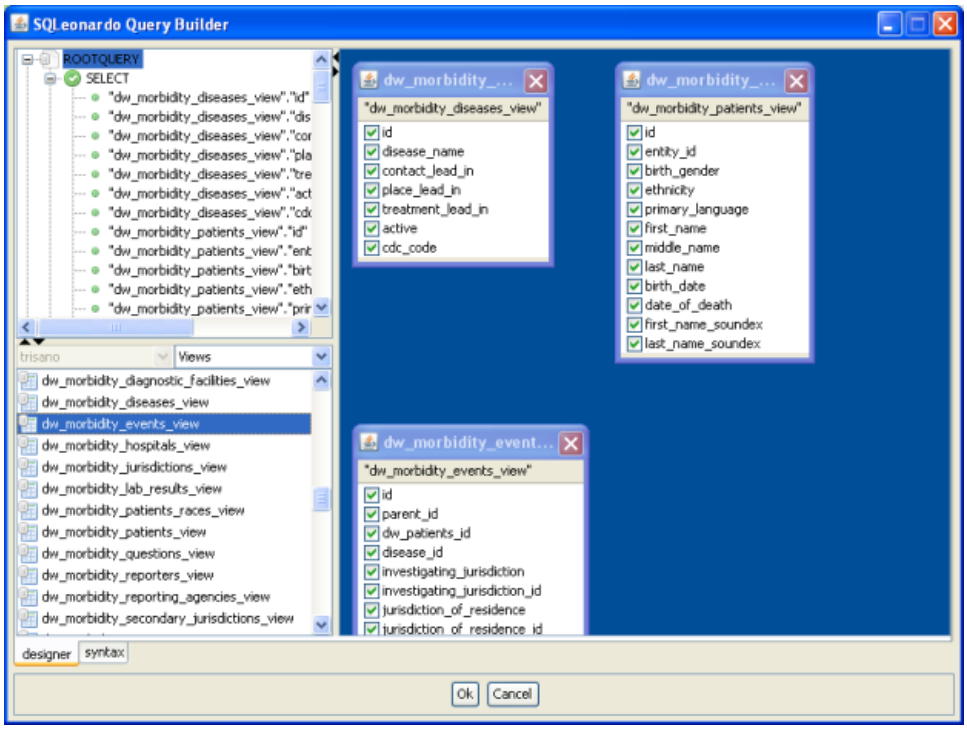


The third view to add is the **dw_morbidity_events_view**. Find it and double click to add to the window. Your window should look like the one shown below.

Note:

Once in a while, a table or view that you double-click on does not immediately display in the right hand section. It may be that you need to scroll down to see it, or it may just not show up. I have found that if you click on the **syntax** tab, next to **designer**, then click **designer**, perhaps a couple of times, the missing table should show up.

Copyright © 2009 Collaborative Software Initiative, Inc.



Your window should now show the three views in the window on the right. These three views all contain what is known as CSI TriSano™ core data. This data is consistent across all installations of CSI TriSano™ for all diseases. For our report, we are going to include some disease-specific data that is part of an H1N1 disease form that has been defined in our CSI TriSano™ demo system. (You may not be able to perform this part, or you can just choose a similar or different form to use.)

Note: You can drag the tables around in the right hand window by clicking and holding down the left mouse button on the title bar of the view and dragging it where you want it to be.

Next Topic:

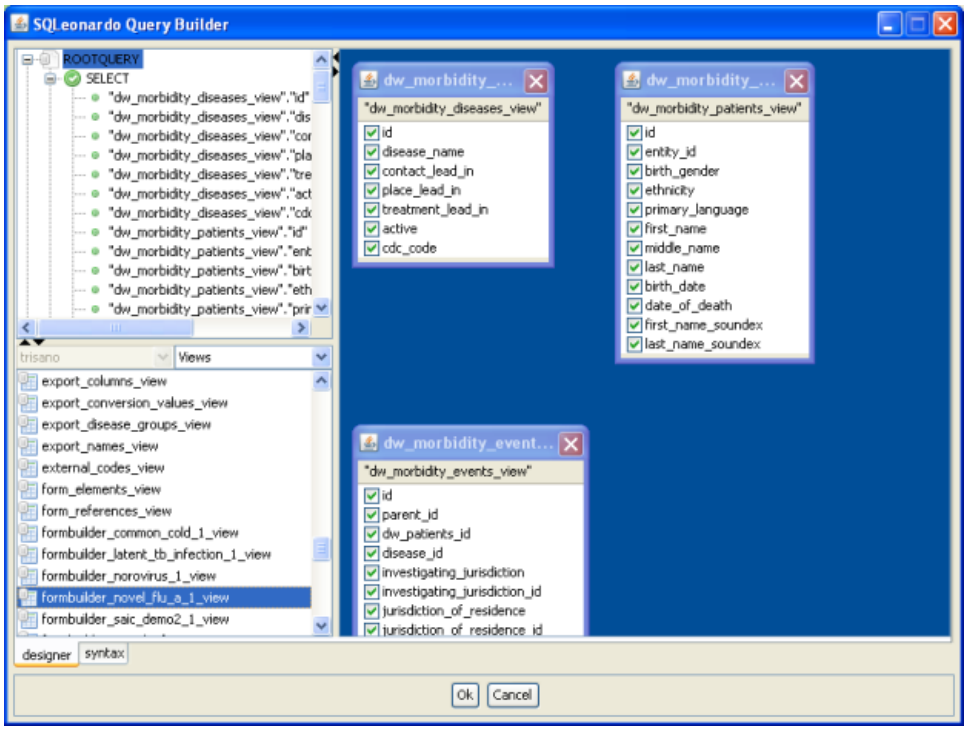
Adding Form Data to the Query

Copyright © 2009 Collaborative Software Initiative, Inc.

Adding Form Data to the Query

Adding Form Data to the Query

The information in this form was requested by the CDC to be collected as part of the H1N1 outbreak in spring 2009. The form is named **novel_flu_a_1**. Scroll down the list of views to find the views that start with **formbuilder...** These are views created when information from the CSI TriSano™ transaction system is loaded into the CSI TriSano™ data warehouse database. Each form in the transaction system becomes a view that contains a column of information for each question/answer in the form. The short name of the question becomes the column heading and the answers become the column values.

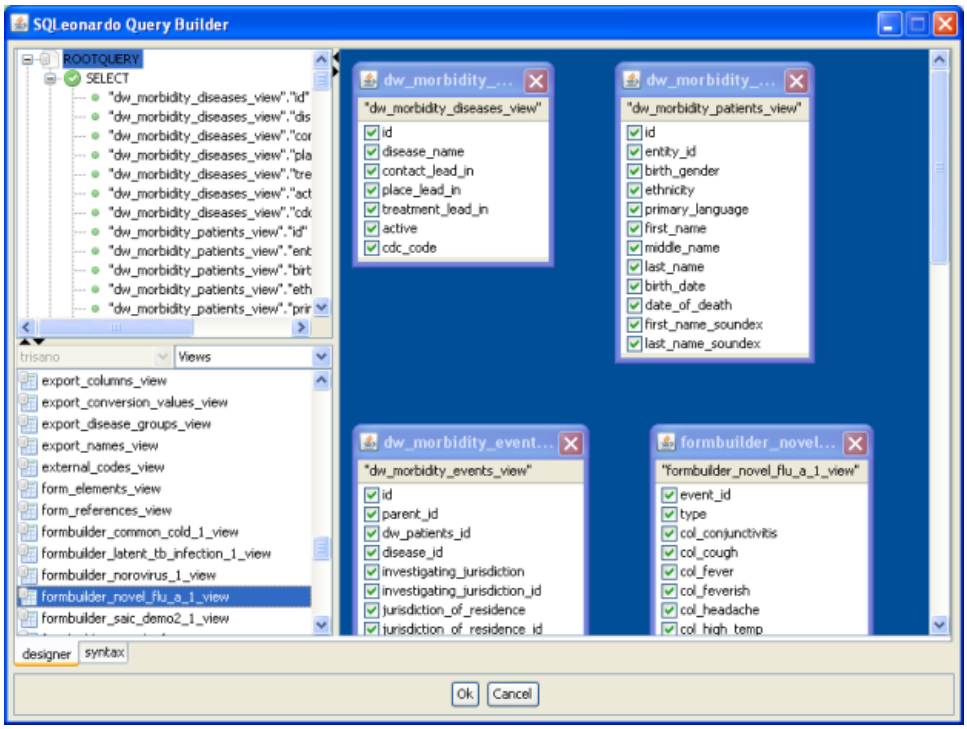


Once you find the view **formbuilder_novel_flu_a_1_view**, double click on it to add it to the query window.

Note:

Once in a while, a table or view that you double-click on does not display in the right hand section. It may be that you need to scroll down to see it, or it may just not show up. I have found that if you click on the **syntax** tab, next to **designer**, then click **designer**, the missing table should show up.

Copyright © 2009 Collaborative Software Initiative, Inc.



You now have all the views you will need for this query, and at this point all the columns have been selected to be displayed. (You can scroll the upper left hand window to see the long, long list!), and you have not defined how these views will be joined together. If you ran the query now, you would get every row in every view joined to every row in every other table, not normally what you want to see.

Next Topic:

Removing Columns from the Query

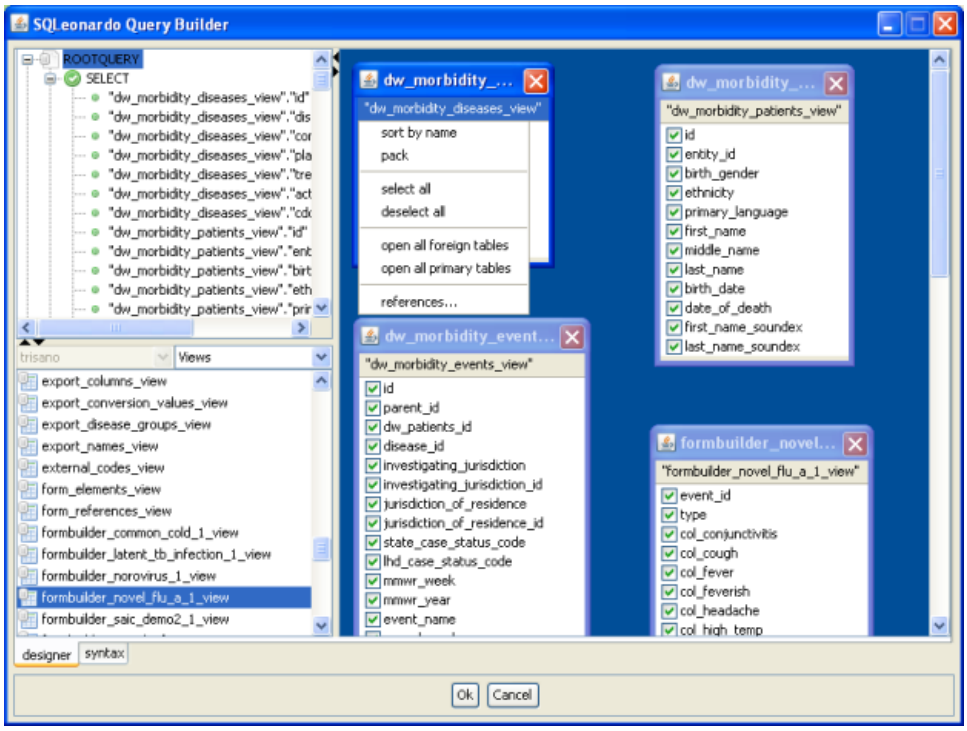
Copyright © 2009 Collaborative Software Initiative, Inc.

Removing Columns from the Query

Removing Columns from the Query

Let's first see how to choose just the columns you want. To the left of each column name is a checkbox with a green checkmark in it. This means that the column has been selected to be displayed. You could just go through every view and every column name and deselect all the columns you don't want. This would leave you with the columns you want to have, but a quicker way usually is to deselect all the columns and then go back and select the columns you want. This also allows you to pick the columns in the order you want them displayed, although you can drag/drop columns in the listing in the upper left hand window by clicking on a column name and holding the left mouse button and dragging the column to where in the SELECT list you would like it to be.

To deselect columns in a table or view, just click on the view name (either mouse button will work) under the title bar of the view. You will see a list of options, one of which is **deselect all** as shown below.



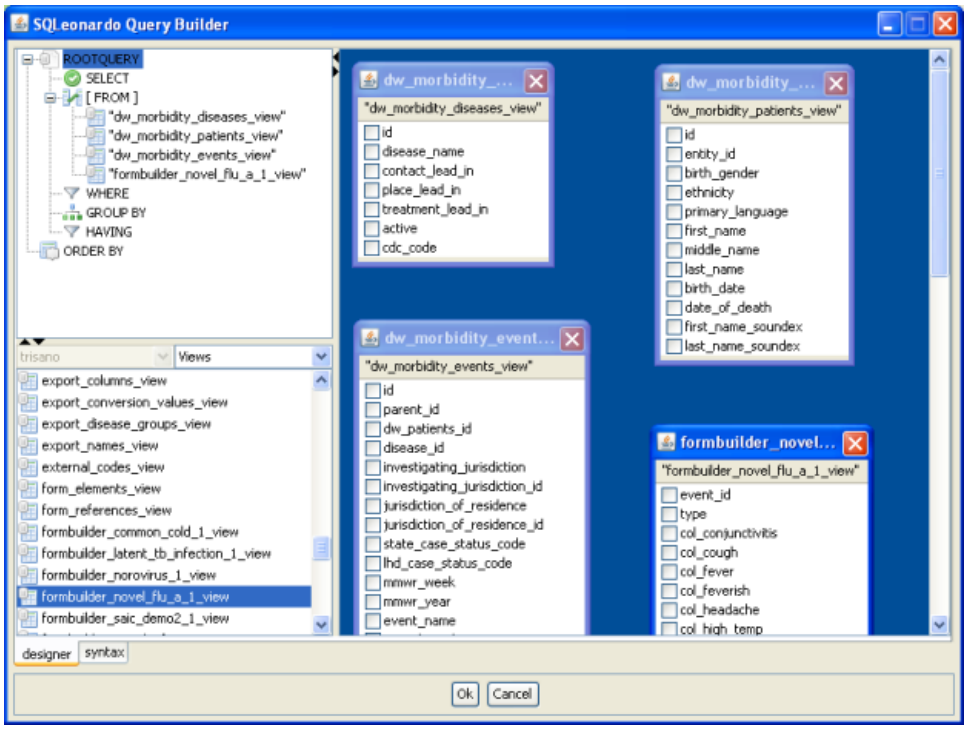
Click on **deselect** all for each of the four views in your query and your screen should look like the one shown below.

You will notice that all the little green checkmarks are gone and that there are no column names listed under **SELECT** in the upper left hand corner section of the screen.

Note:

You can move the views on the right around by clicking and dragging the view window to where you want it to go, and sometimes you can resize the entire window to give more room for tables/views. Also, you can re-proportion the screen amond the three parts by clicking and dragging the borders between the parts.

Copyright © 2009 Collaborative Software Initiative, Inc.



You'll add some columns back to the query is a little bit, after you join the views together.

Next Topic:

Joining Views for the Query

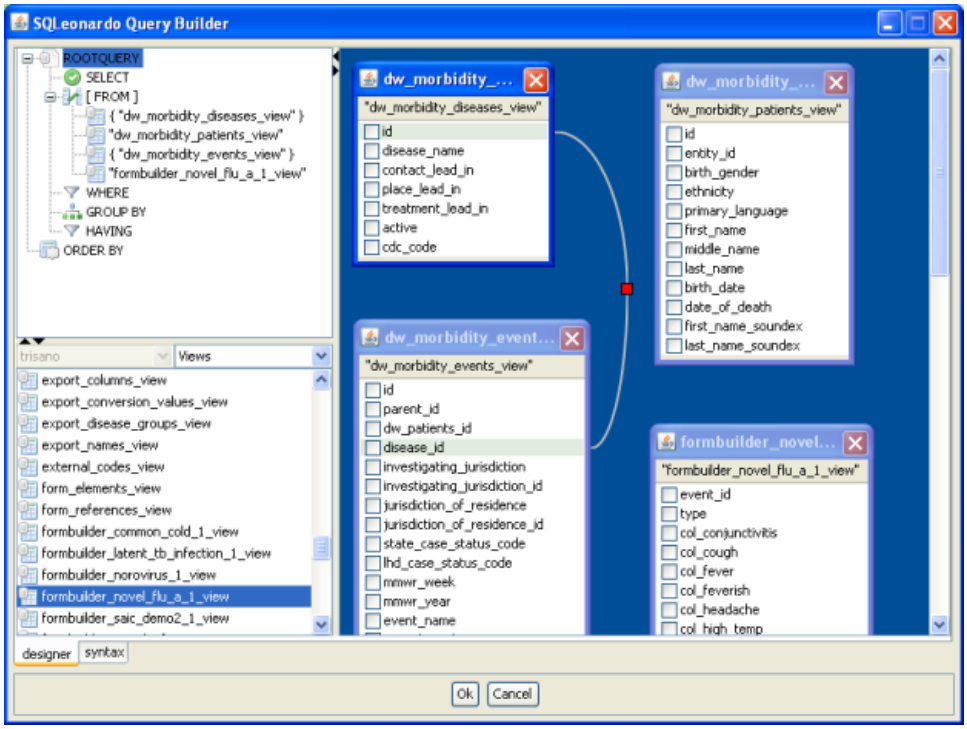
Copyright © 2009 Collaborative Software Initiative, Inc.

Joining Views for the Query

Joining Views for the Query

Next, though, it's time to create the joins between the views. The first join you are going to define is between the **dw_morbidity_diseases_view id** column and the **dw_morbidity_events_view disease_id** column. To create the join, click and hold with the left mouse button on one of the two columns and drag it to the other column and release. Your mouse pointer will change to a plus sign when you do this operation.

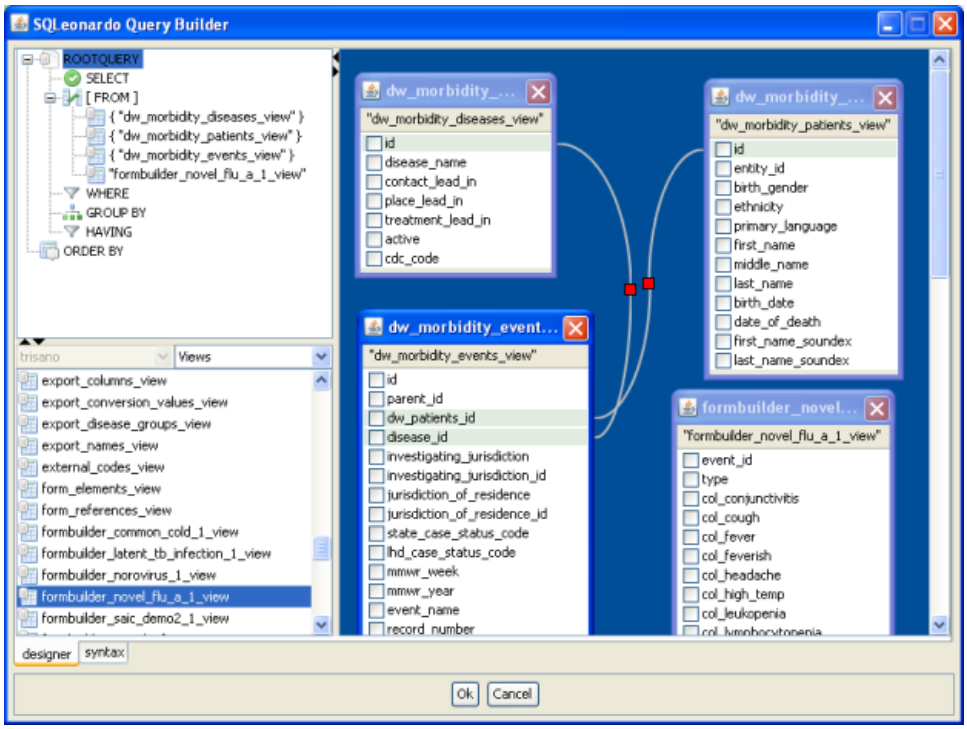
After dragging (and it may take you several tries to get it right), your screen should look like the one below. If you get it wrong and the wrong columns are joined, just right mouse click on the little red box in the middle of the join connnection, and click **remove**.



Next, connect the **dw_morbidity_patients_view id** column to the **dw_morbidity_events_view dw_patients_id** column in the same way.

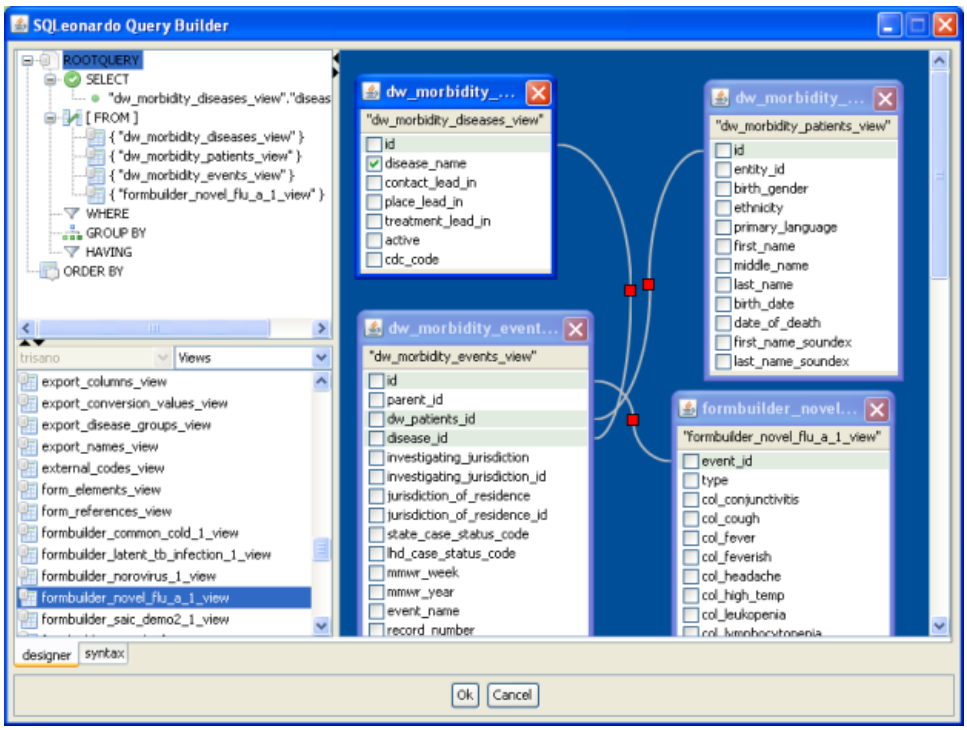
Your screen should look like this.

Copyright © 2009 Collaborative Software Initiative, Inc.



One more join and you'll be done!

There is one more view left out, the **formbuilder_novel_a_1_flu view**. Join the **event_id** column in the **formbuilder_novel_a_1_flu view** view to the **id** column in the **dw_morbidity_events_view**, as shown below.



That's all you need to do to join these view together. In most cases, tables and views are joined together by a single column but sometimes it can be two or more.

Next Topic:

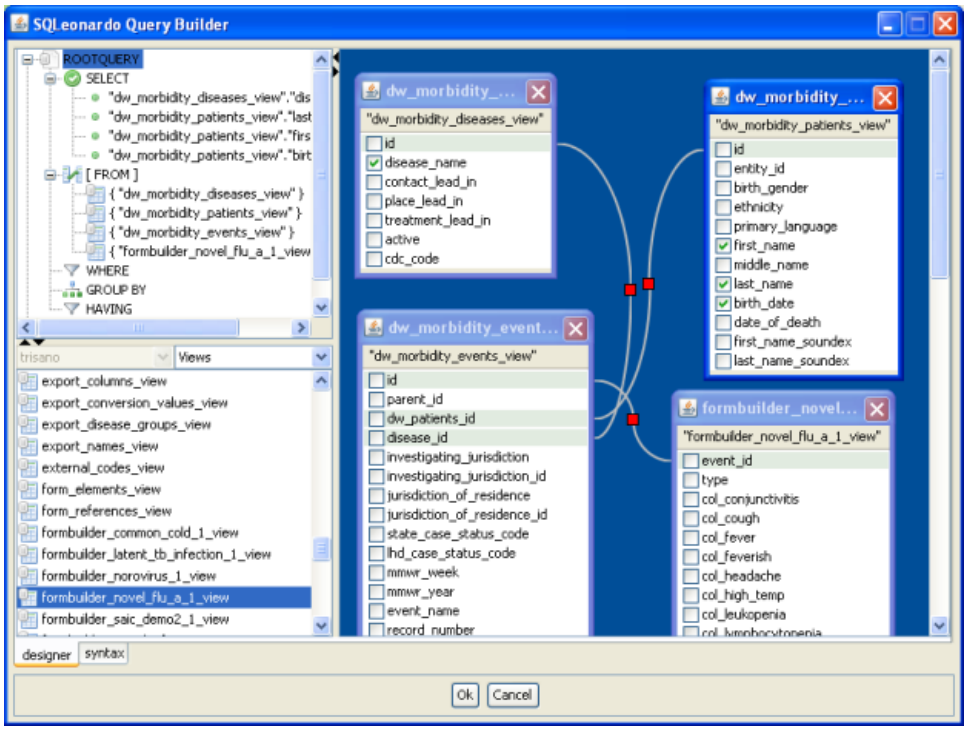
Selecting Columns for the Query

Copyright © 2009 Collaborative Software Initiative, Inc.

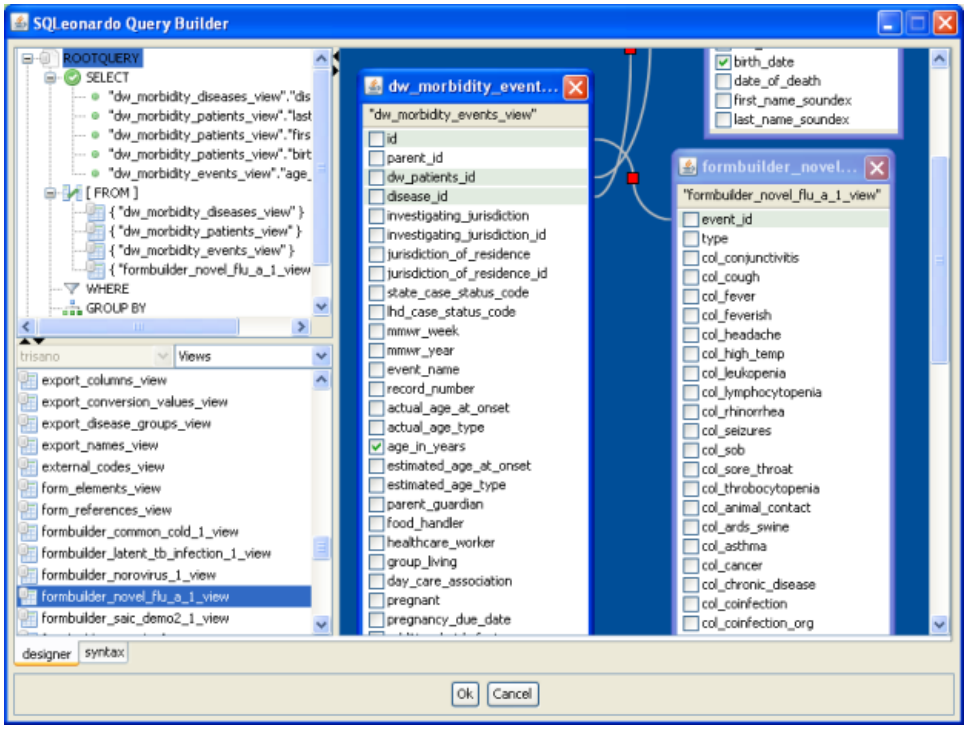
Selecting Columns for the Query

Selecting Columns for the Query

Now, on to selecting columns for our query. To select a column, just click in the checkbox in front of the column name. The columns will appear in the upper left hand section in the same order that you select the columns. In the screen below, I checked **disease_name** from the **dw_morbidity_diseases_view**, then **last_name**, **first_name** and **birth_date** from the **dw_morbidity_patients_view**.

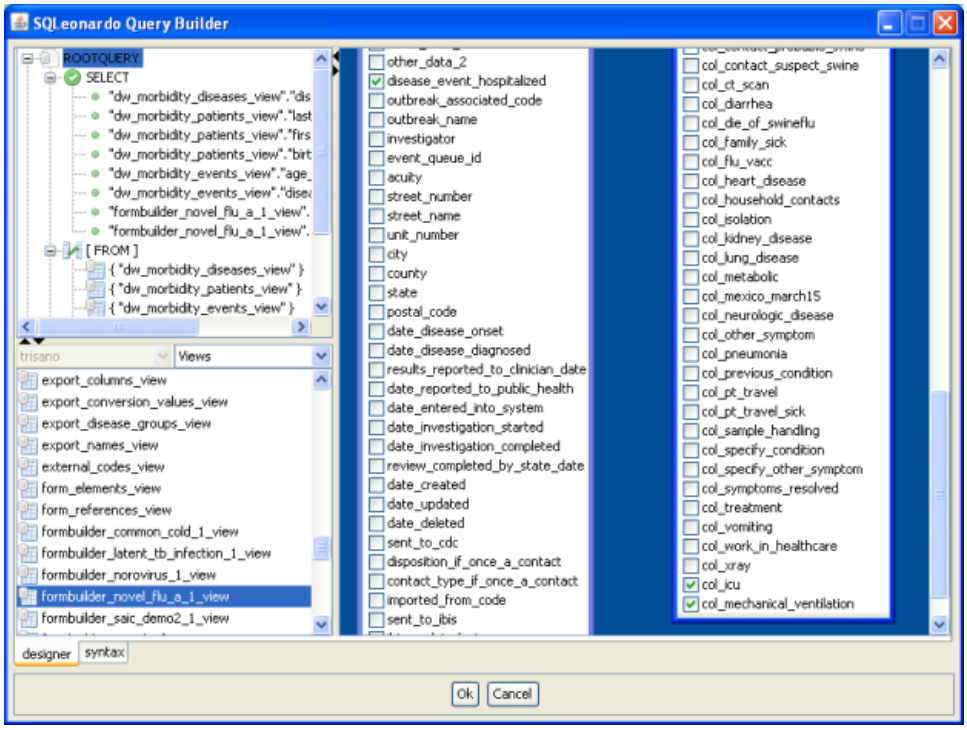


Next, I scrolled down and selected **age_in_years** in the **dw_morbidity_events_view**.



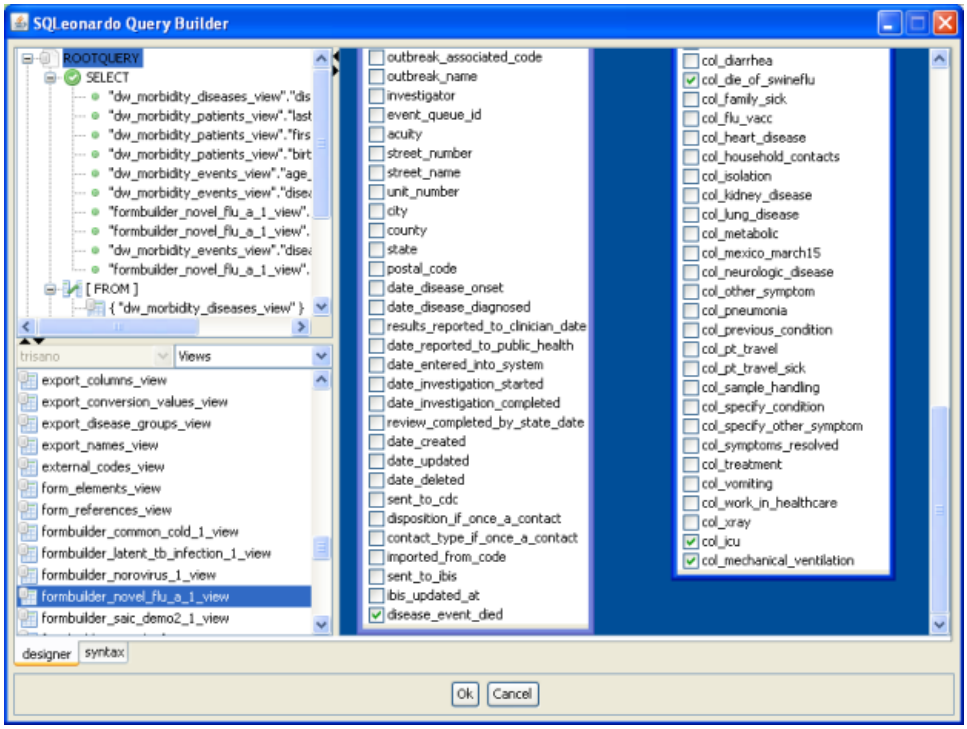
Copyright © 2009 Collaborative Software Initiative, Inc.

By scrolling to near the bottom of the list, as shown below, select **disease_event_hospitalized** from the **dw_morbidity_events_view**, and **col_icu** and **col_mechanical_ventilation** from the **formbuilder_novel_flu_a_1_view**.



Only two more columns to select. We want to select the column, **disease_event_died** in the **dw_morbidity_events_view** and the column **col_die_of_swineflu** from the **formbuilder_novel_flu_a_1_view**. Scroll to the very bottom of the graphical query builder section and you will see the first of these columns. Select it, and then select the **col_die_of_swine_flu** from the **formbuilder_novel_flu_a_1_view**.

After selecting these, your screen should look like the one shown below.



Copyright © 2009 Collaborative Software Initiative, Inc.

You are now finished adding columns to the query. Remember that you can reorder the columns in the SELECT list by dragging and dropping them.

Next Topic:

Viewing the Syntax of the Query

Copyright © 2009 Collaborative Software Initiative, Inc.

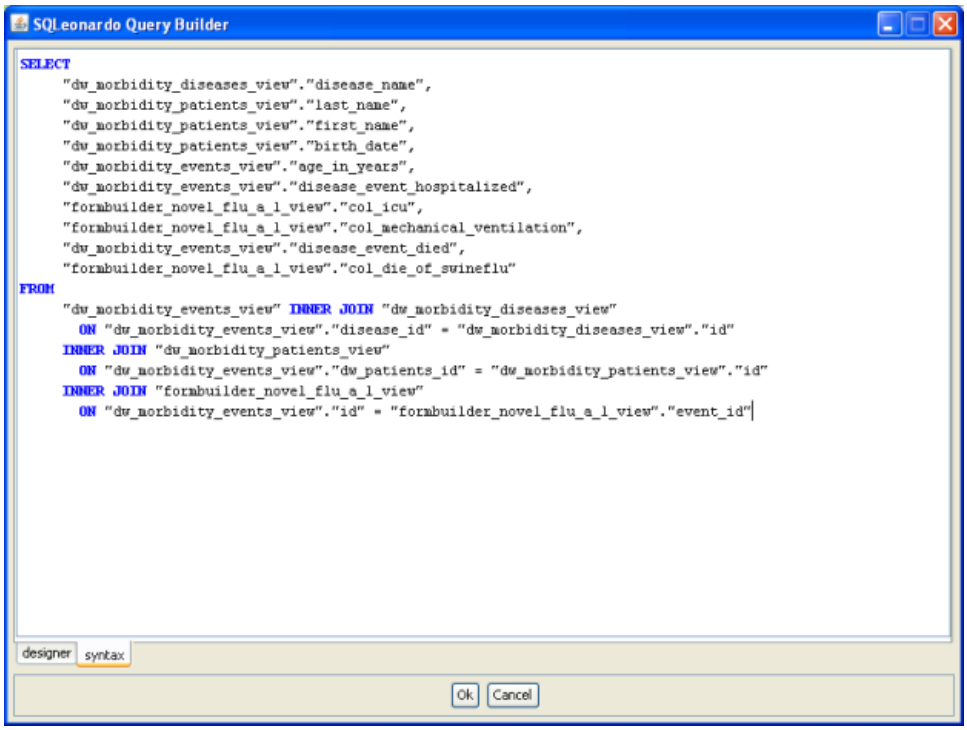
Viewing the Syntax of the Query

Viewing the Syntax of the Query

At any time, you can see the syntax of the query by clicking on the **syntax** tab in the bottom left hand corner of the screen. Your screen should look like the one shown below.

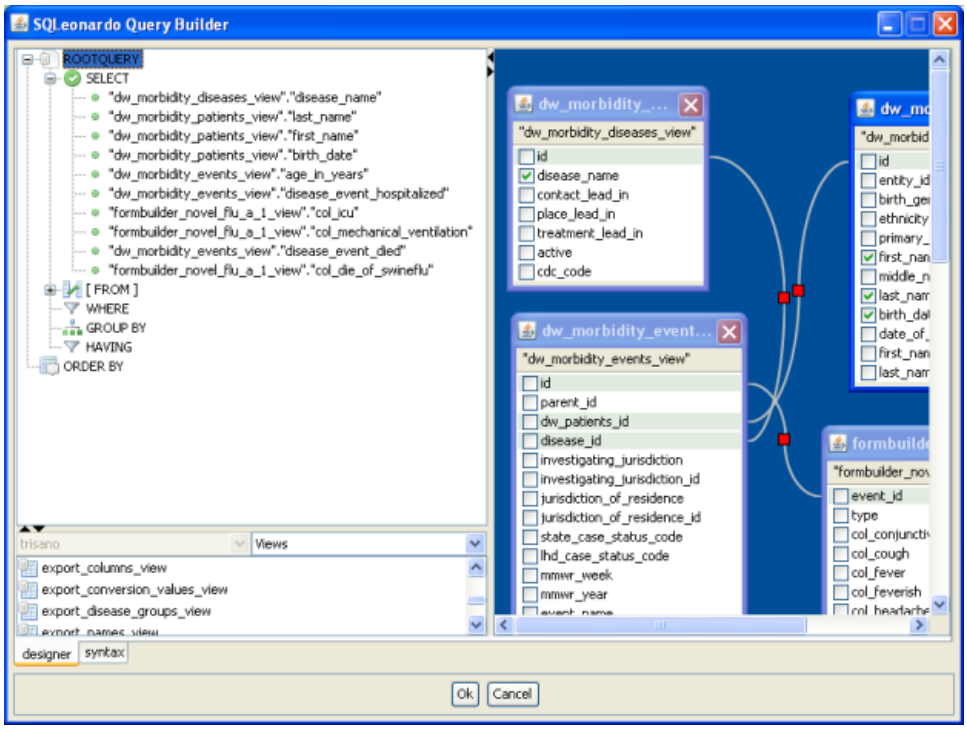
Note:

I've slightly modified the text so that it all appears in the window.



To return to the query designer, click on **designer** in the bottom left hand corner of the screen. As mentioned earlier, you can drag the borders that are between the different sections of the Query Builder to resize the sections. I've done that below to make the **SELECT** statement in the upper left hand corner completely visible and easier to work with.

Copyright © 2009 Collaborative Software Initiative, Inc.



Next Topic:

Ordering the Result of the Query

Copyright © 2009 Collaborative Software Initiative, Inc.

Ordering the Result of the Query

Ordering the Result of the Query

We would like the results of our query to be returned in a specific order. To do this, just drag the columns you would like the results to be sorted by from the **SELECT** list to the **ORDER BY** clause. As shown below, drag the following columns to the **ORDER BY** clause.

Note:

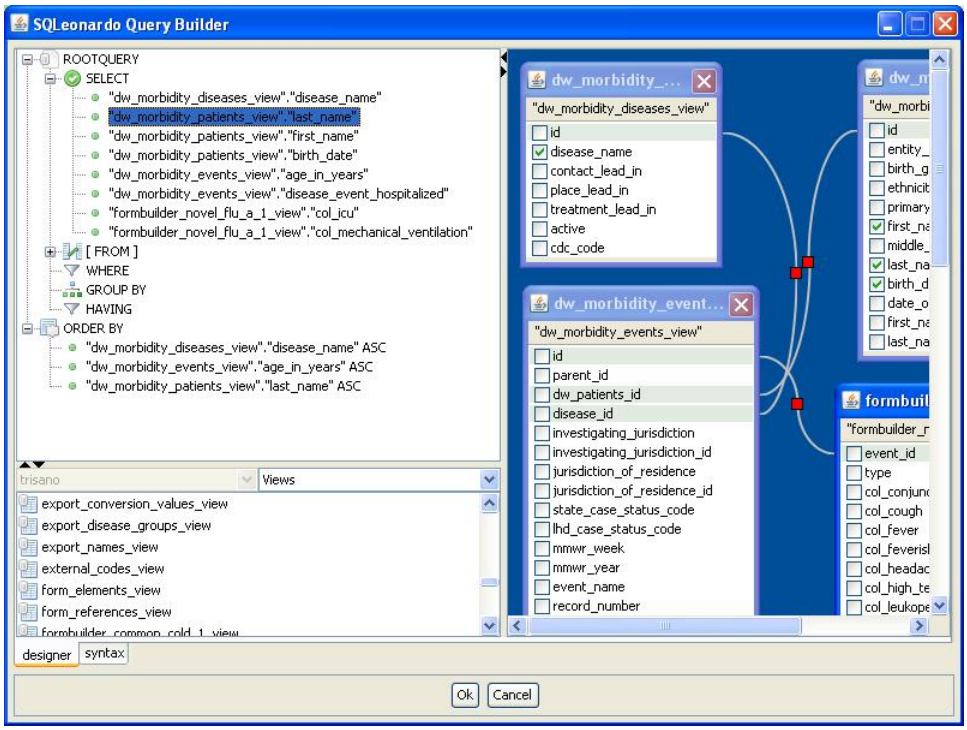
After you drag the first one to right below the **ORDER BY**, drag the others to this exact same place, just below the **ORDER BY** clause, and they will be placed **AT THE BOTTOM** of the **ORDER BY** list. If you try to drag the second and third column name to the very bottom of the **ORDER BY** list, it will not work.

As with the column list, you can reorder the **ORDER BY** clause by dragging the columns to the place you want them.

In this case, drag the following columns in the this order:

- First, **disease_name**
- Second, **age_in_years**
- Third, **last_name**

Your screen should look like this.



Next Topic:

Leaving the Query Builder

Copyright © 2009 Collaborative Software Initiative, Inc.

Leaving the Query Builder

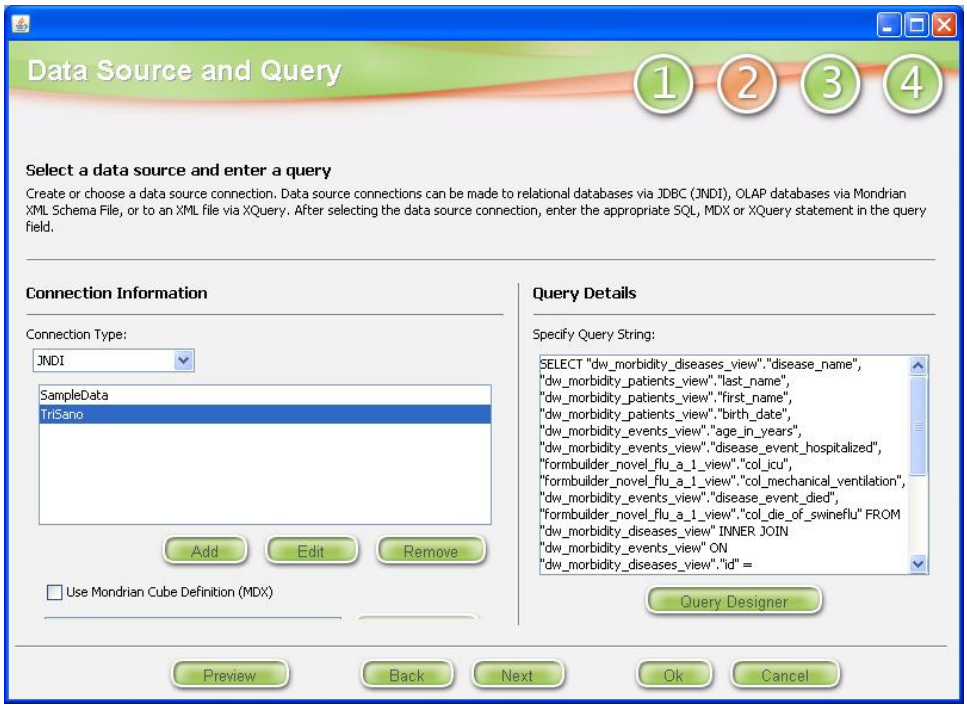
Leaving the Query Builder

It's time to leave the **Query Builder** now, and go back to the **Report Wizard**. To do this, click on **OK**.

Note:

Sometimes, you will have to click on **OK** twice to get back to the **Report Wizard**.

Your screen should look like the one shown below. The query you just created is shown in the **Query Details** section. In fact, you can write the SQL from scratch in this window if you wish.



You can return to the Query Designer by pressing the **Query Designer** button to update the query, but you can also edit the query directly in the query window.

One update I have made in the window shown below is to change the format of the birth_date column. This column is returned from the database in the form of a TIMESTAMP. For example, the date of August 11, 2009 would by default be returned as '08/11/2009 12:00 AM'. To remove the '12:00 AM' from the birth_date column, change:

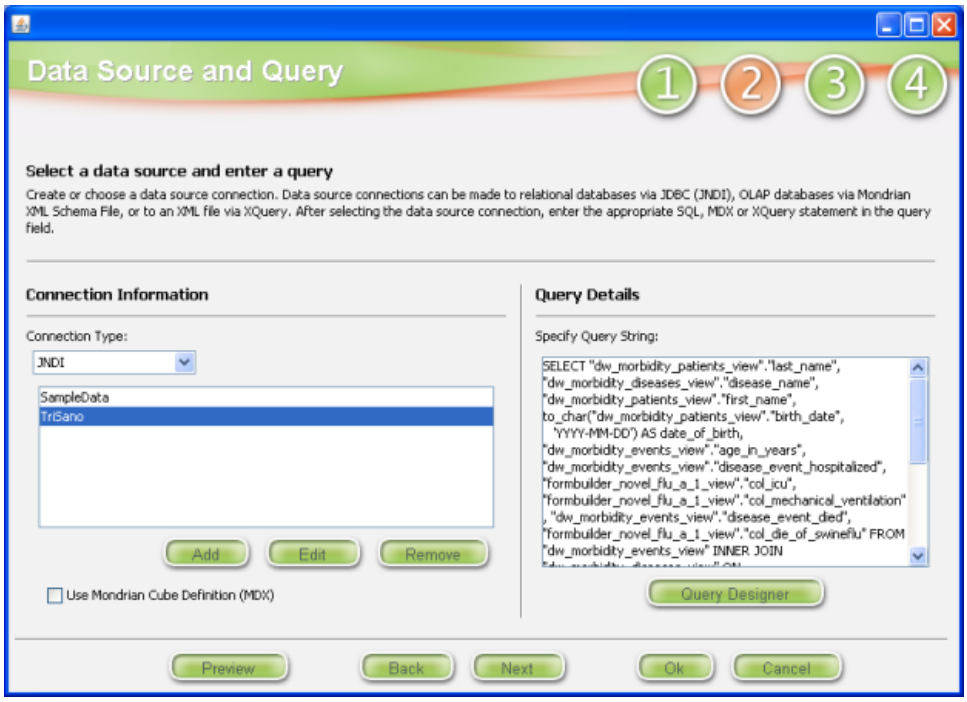
"dw_morbidity_patients_view"."birth_date",

to

to_char("dw_morbidity_patients_view"."birth_date",'YYYY-MM-DD') as date_of_birth,

as shown in the screen below. (You can omit this step if you wish. The birth date will just display with a timestamp of 12:00 AM.)

Copyright © 2009 Collaborative Software Initiative, Inc.



Next Topic:

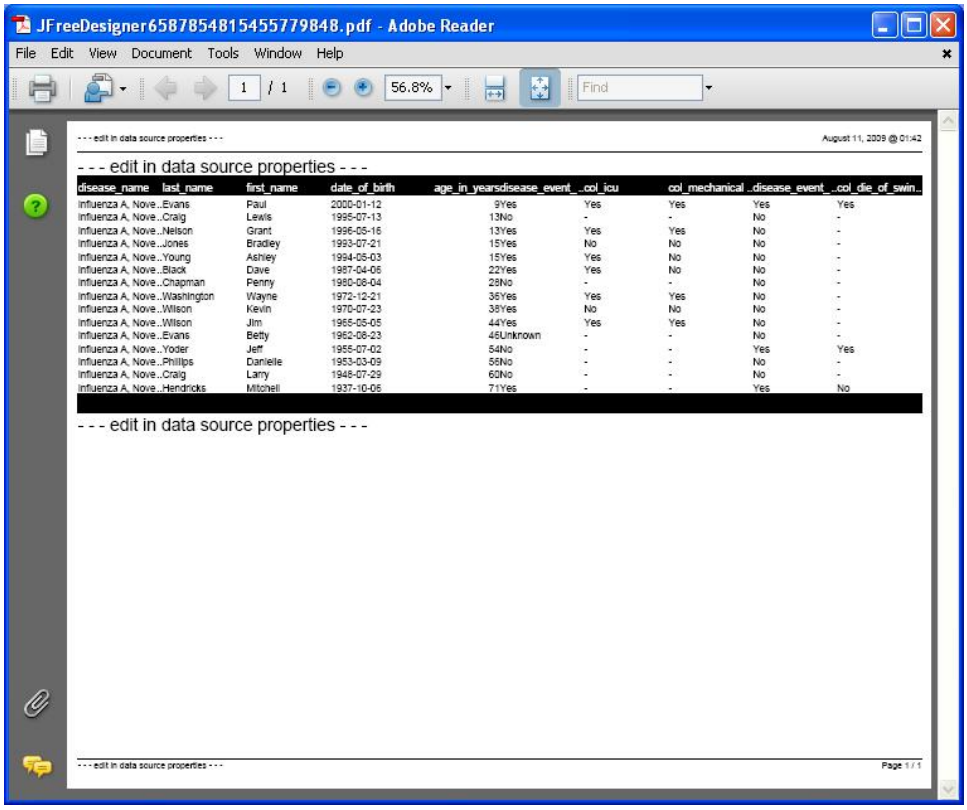
Preview the Results of the Query

Copyright © 2009 Collaborative Software Initiative, Inc.

Preview the Results of the Query

Preview the Results of the Query

At any time, you can **Preview** the report in a separate window by clicking on the **Preview** button, as shown below.



Next Topic:

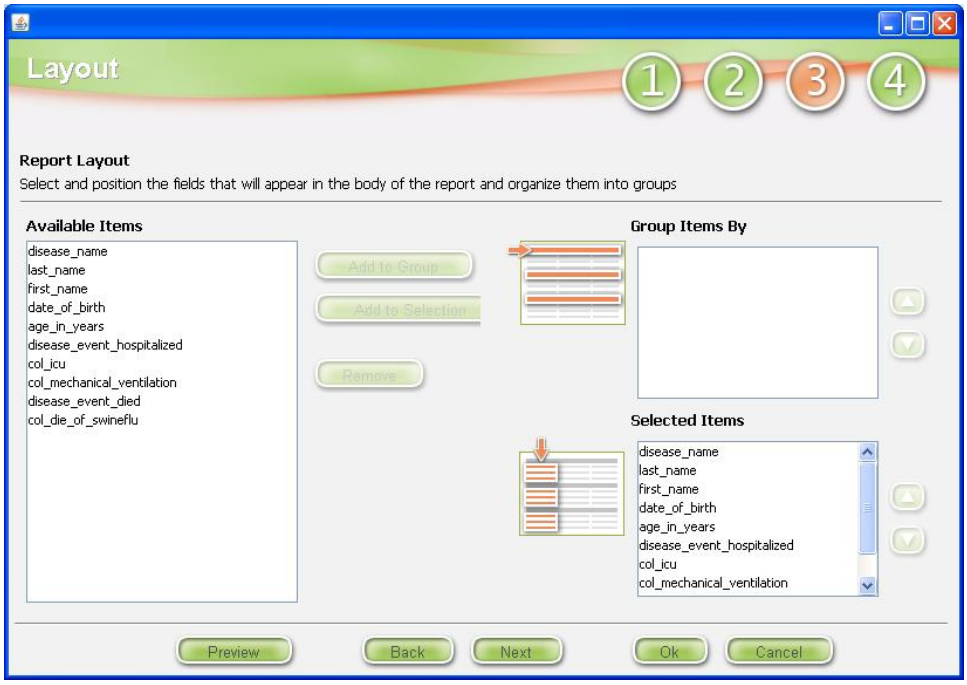
Report Layout

Copyright © 2009 Collaborative Software Initiative, Inc.

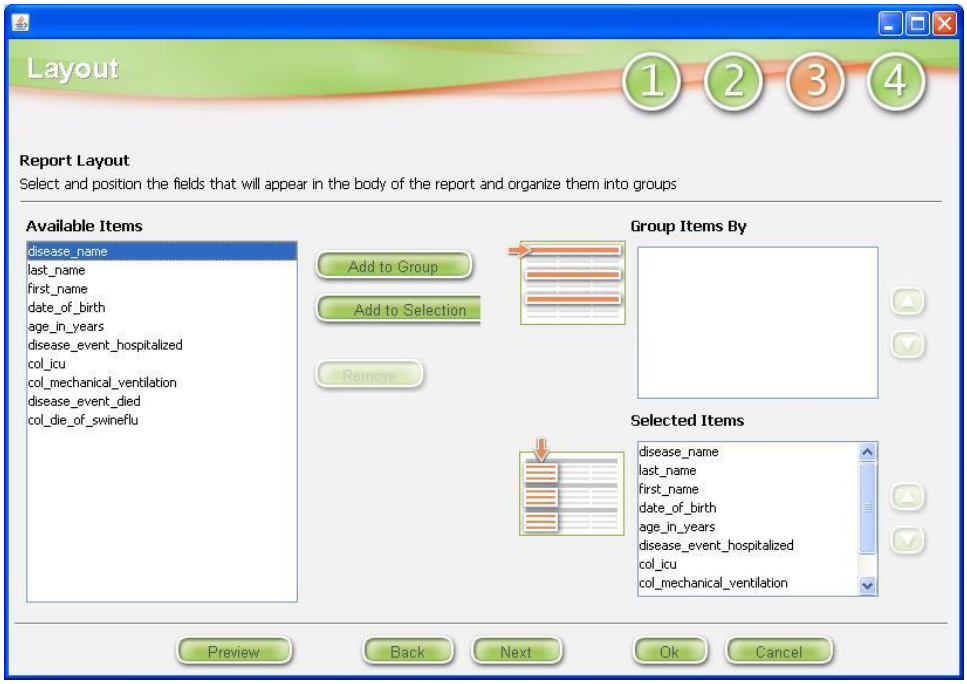
Report Layout

Report Layout

After reviewing the report, close the **Preview** window and return to the **Report Wizard** screen, then click on **Next** to get to the next screen in the **Report Wizard**, the **Layout** screen, shown below.



On this screen, you can select a column or columns to group the report by, and select which columns will display in the report. This means you can query more columns than you want to see on the report and remove them at this point.

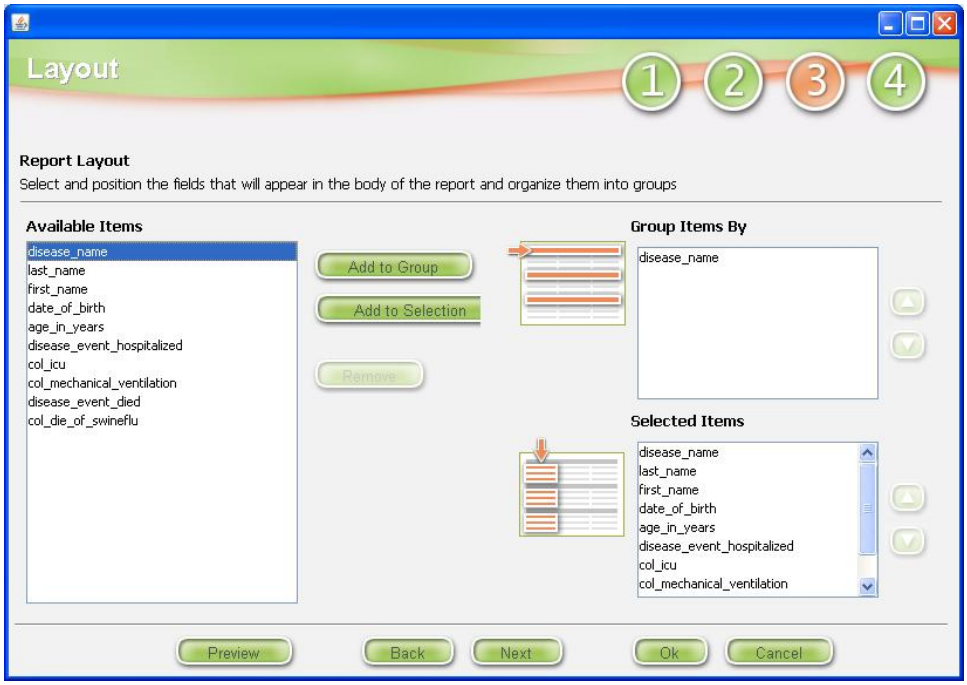


For this report, we want to Group the report by **disease_name**. To do that, click on **disease_name** in the screen above and then click on **Add to Group** to add the **disease_name** to the **Group Items By** box.

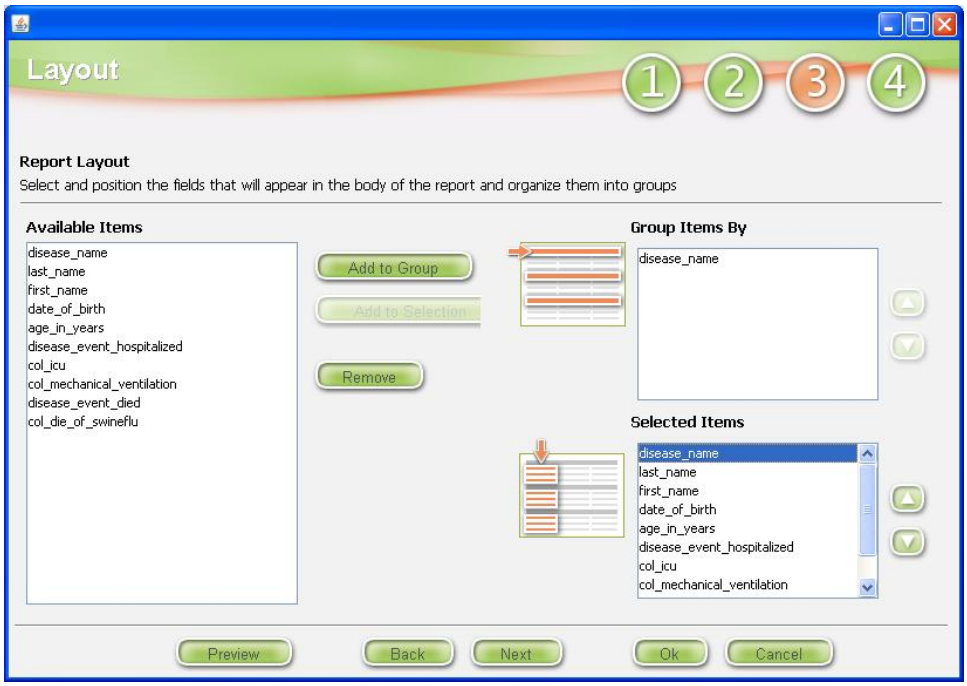
Copyright © 2009 Collaborative Software Initiative, Inc.

Normally, this should to be the same as the **ORDER BY** columns in the query. The Report Designer will automatically generate breaks and totals in the report based on the columns in the **Group Items By** columns.

Your screen should look like the one shown below.



Now that the disease name will be used to grouped the rows, it is not needed to be shown in the details rows of the report. It can be removed from the report by clicking on **disease_name** in the **Selected Items** box as shown below.



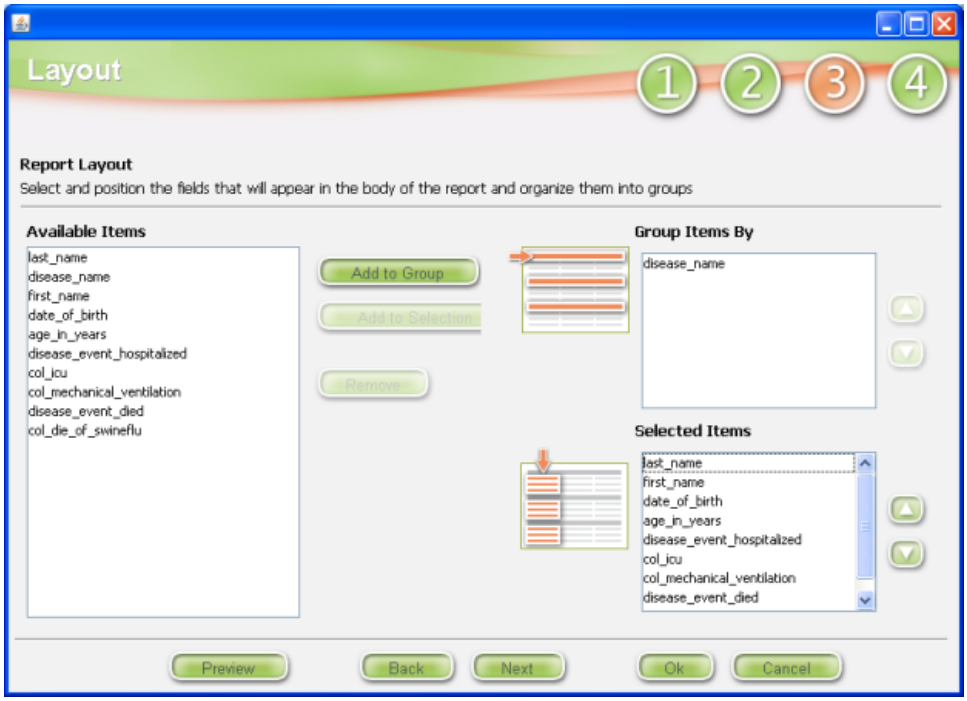
Next, click on the **Remove** button to remove the **disease_name** column from the detail lines of the report.

Note:

Sometimes this did not work the first time. I had to re-select the **disease_name** column and click on **Remove** again.

Copyright © 2009 Collaborative Software Initiative, Inc.

Your screen should look like the one shown below.



Next Topic:

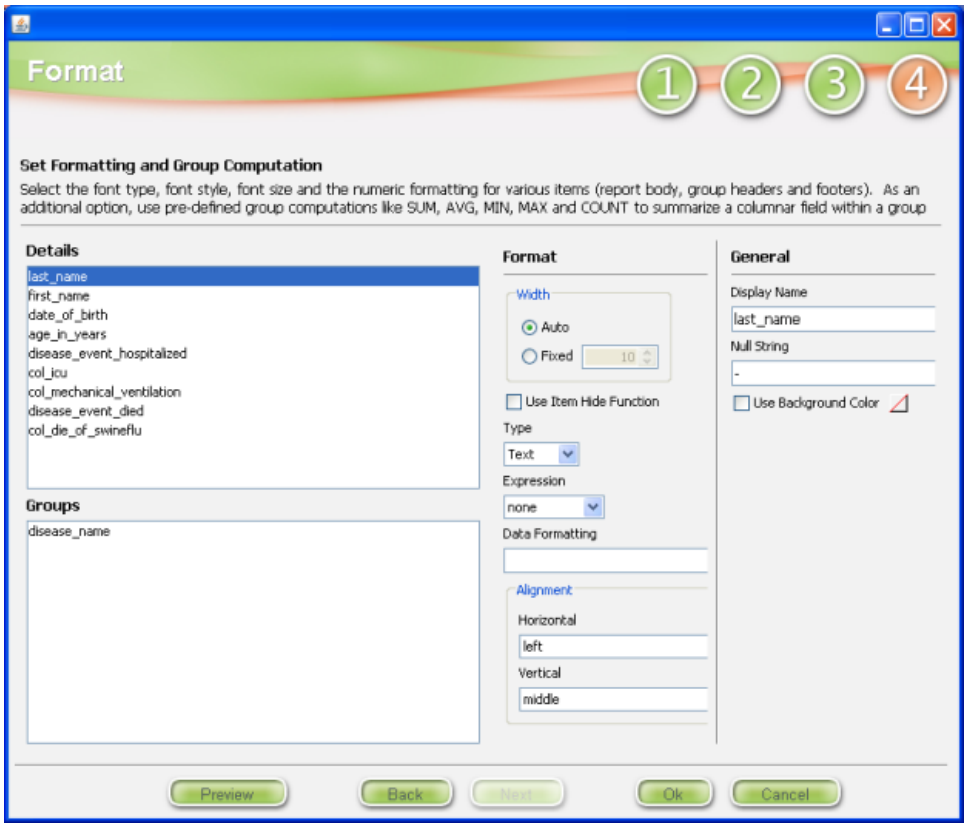
Report Formatting

Copyright © 2009 Collaborative Software Initiative, Inc.

Report Formatting

Report Formatting

For the final step, click on **Next** to move to the **Format** step as shown below.



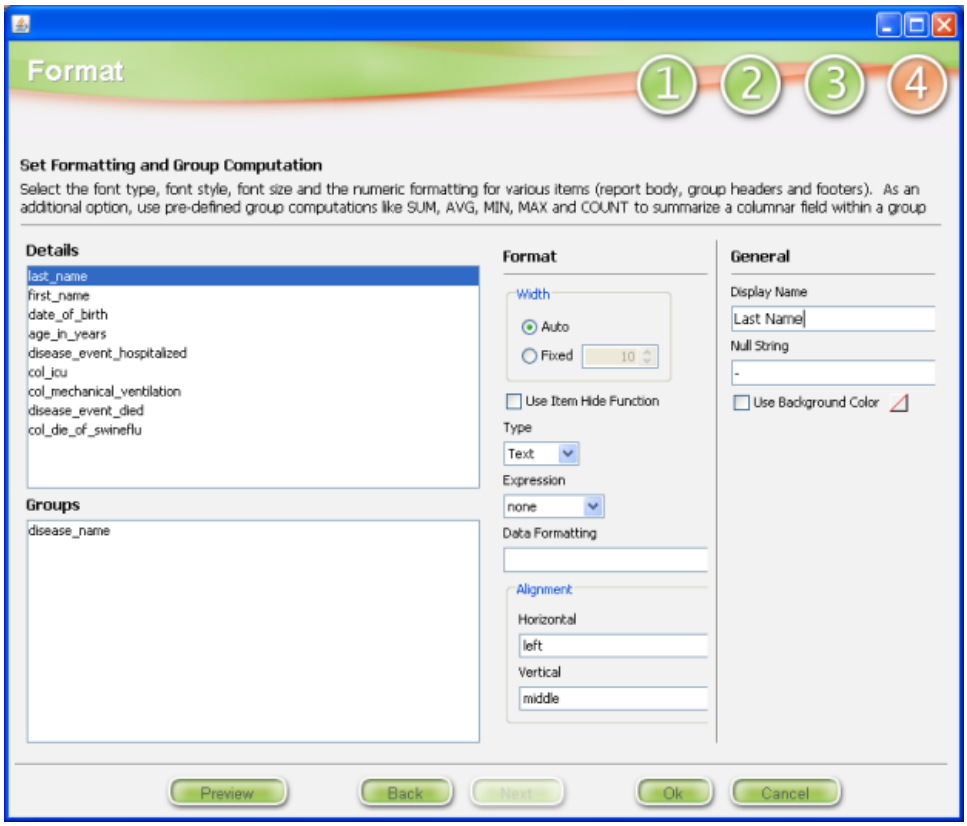
In this step, you can change the formatting for the columns in the report, including the heading for the column of the report, and other formatting, such as width, expressions, data formatting, and alignment.

Note:

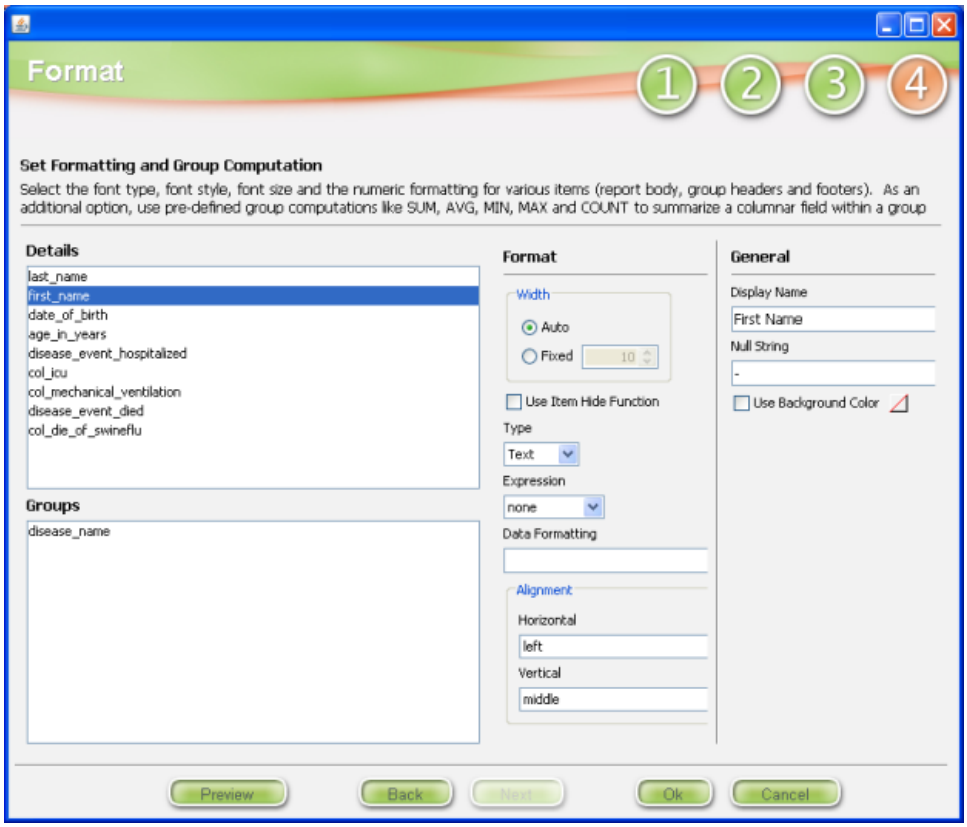
When this screen first displayed, the window was not tall enough to display all the information, and there is no scroll bar on the right to indicate that there is more to be seen. I have resized the window in this guide to show you all the information in the window. You can do the same thing.

The default for the column heading is the column name. For instance, the last name column would be **last_name.** To change that to **Last Name** (a better heading), click on **last_name** in the **Details** box, then change the **Display Name** from **last_name** to **Last Name**. Your screen should look like the one shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.

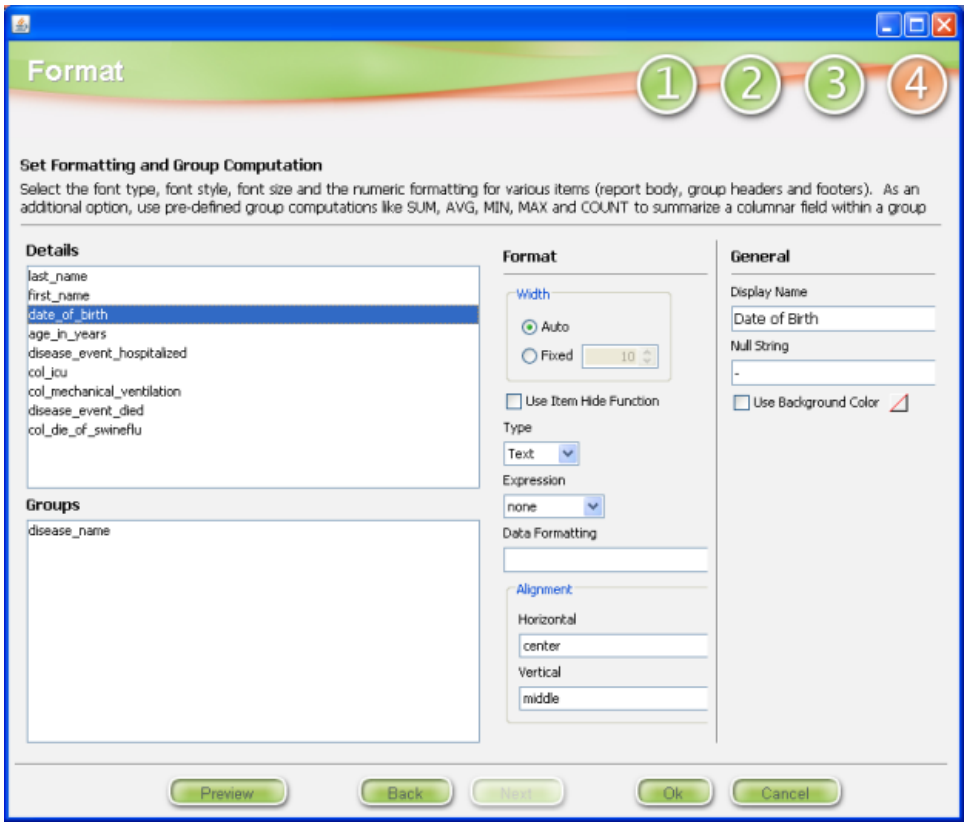


The following screen shows changing the name of the **first_name** column to **First Name**.



Next, change **date_of_birth** to **Date of Birth**.

Copyright © 2009 Collaborative Software Initiative, Inc.



Continue to change columns name:

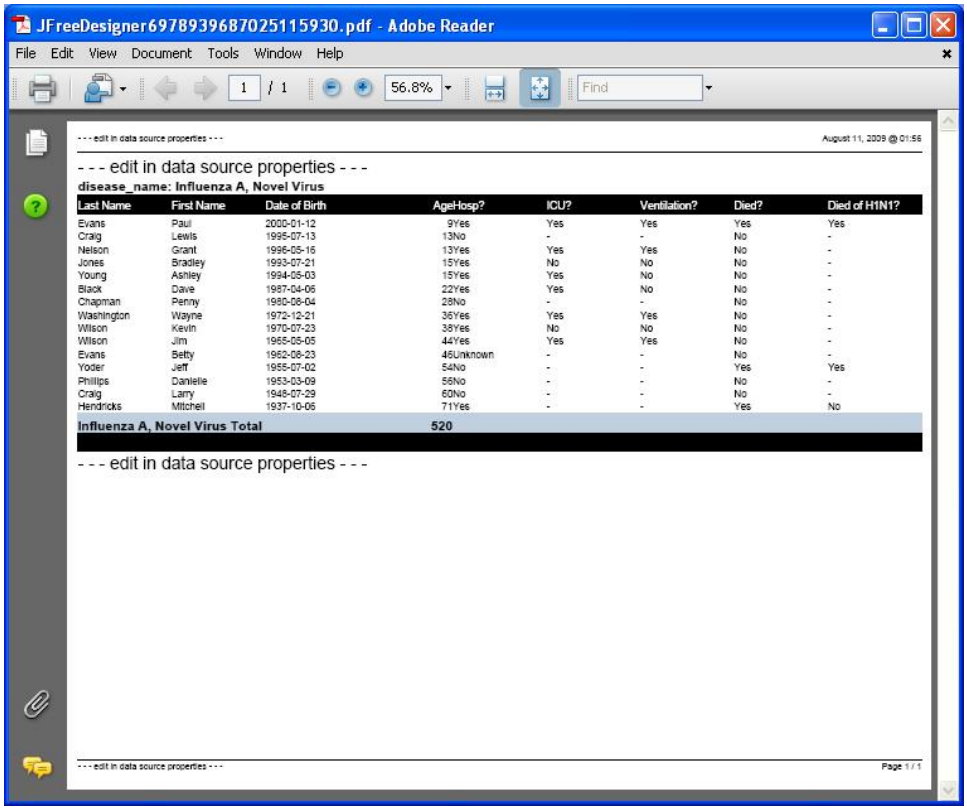
- Change **age_in_years** to **Age**
- Change **disease_event_hospitalized** to **Hosp?**
- Change **col_icu** to **ICU?**
- Change **col_mechanical_ventilation** to **Ventilation?**
- Change **disease_event_died** to **Died?**
- Change **col_die_of_swineflu** to **Died of H1N1?**

When you are finished, click on Preview to see what your report would look like at this point.

Note:

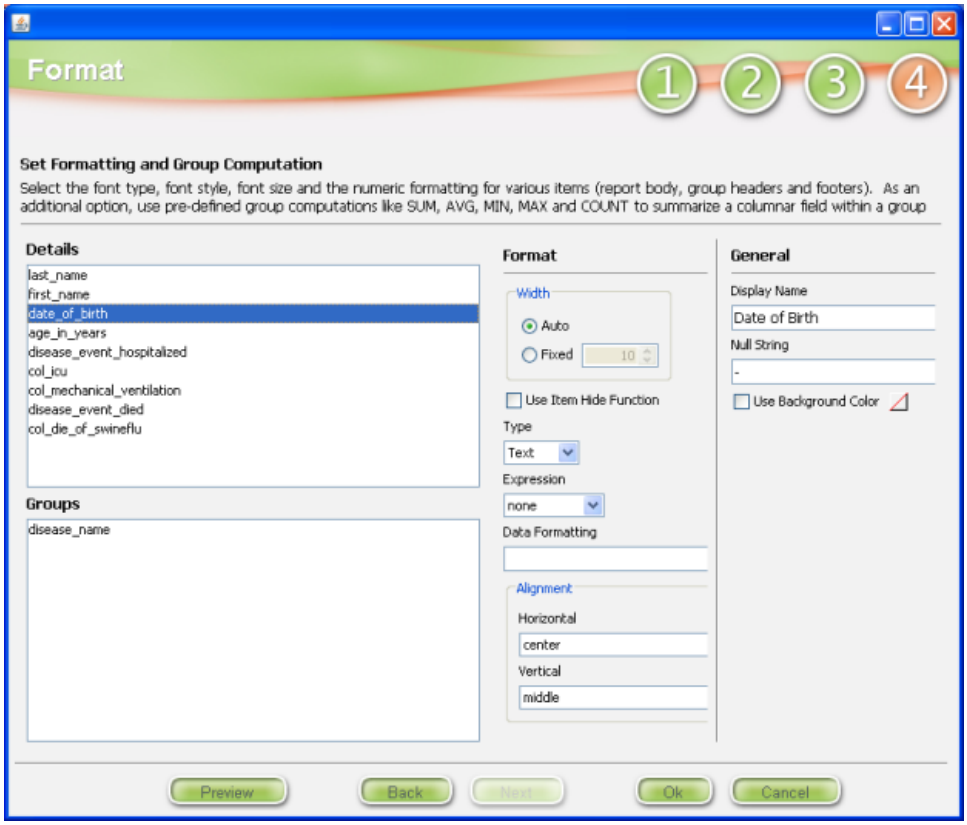
You'll see parts of the report that say "**edit in data source properties**". These are page/report headings and footers that will be changed in the Report Designer, which we will get to very soon.

Copyright © 2009 Collaborative Software Initiative, Inc.



When finished, close the report window and return back to the Report Wizard.

The last thing to do is change the **Alignment** of the **date_of_birth** column to center so that is separated from the **Hosp?** column in the report. Do do this, click on the **date_of_birth** column in the **Details** box and click on the **Horizontal** field under **Alignment** and choose **center**. Your screen should look like the one shown below.



Copyright © 2009 Collaborative Software Initiative, Inc.

You have seen a lot that can be done in the Report Wizard, so click on **OK** to end the Report Wizard to return to the Report Designer.

Next Topic:

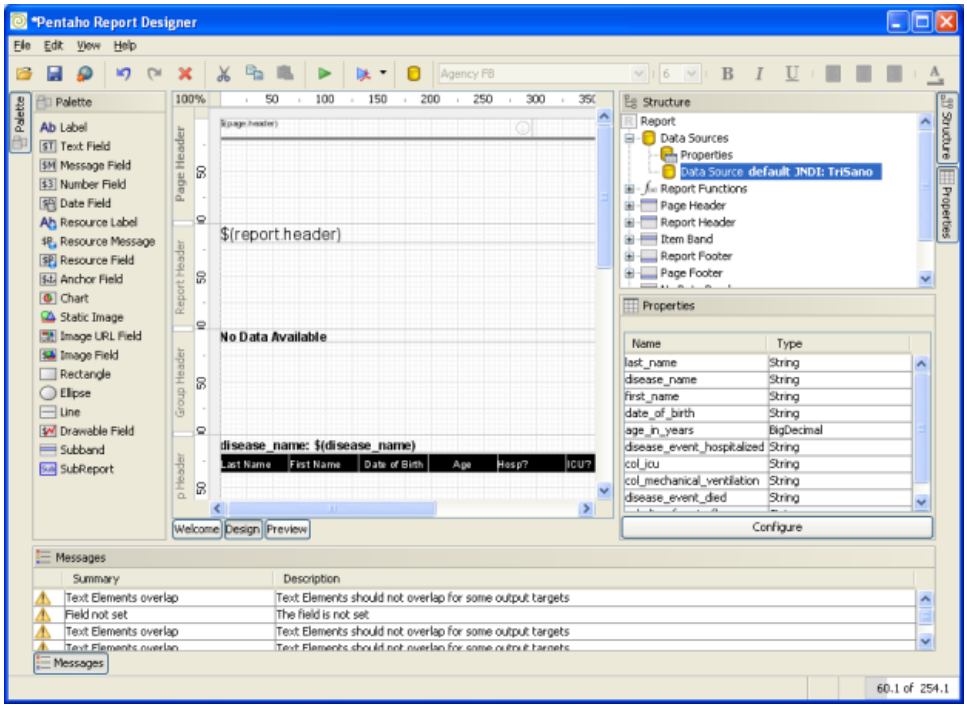
Using the Report Designer

Copyright © 2009 Collaborative Software Initiative, Inc.

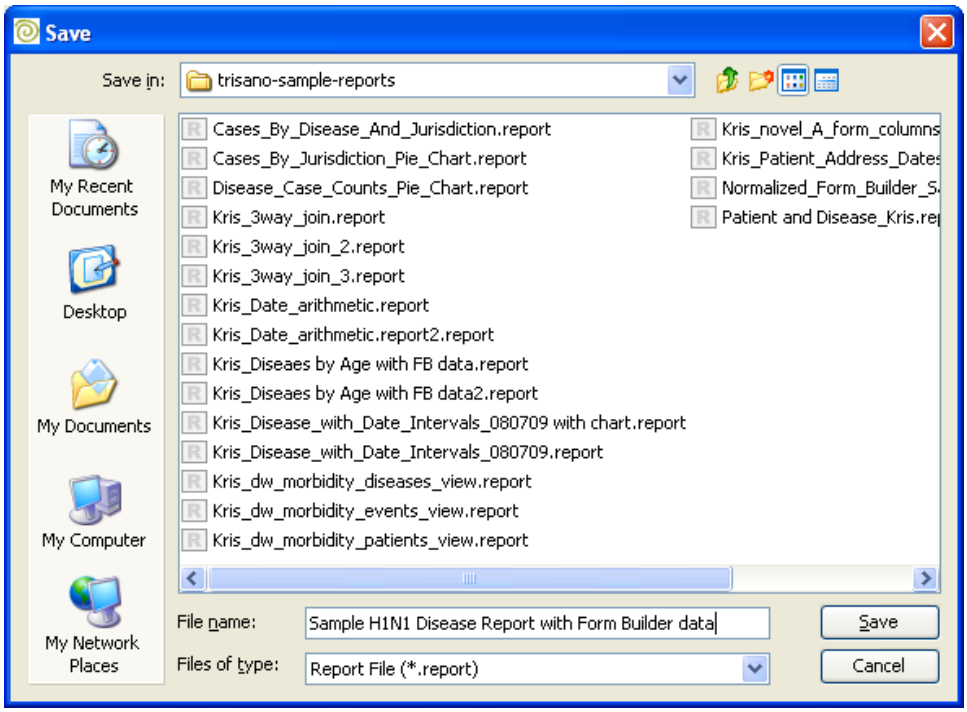
Using the Report Designer

Using the Report Designer

You have now left the **Report Design Wizard** and returned to the **Report Designer**. Your screen should look like the one shown below. The **Pentaho™ Report Designer** is a very powerful, full feature report designer and may seem intimidating at first, but the use of the **Report Design Wizard** has already done a lot of the work for you.



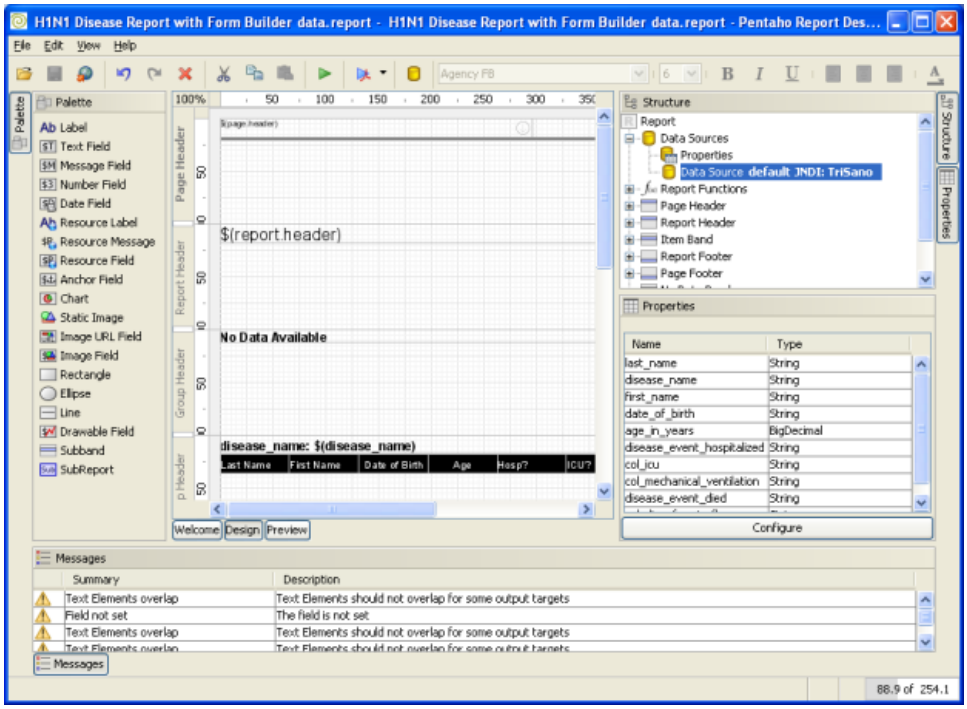
Let's save what we have done so far. So click on **File** and then **Save** in the screen above and **Save** your report. Here is an example of doing that below.



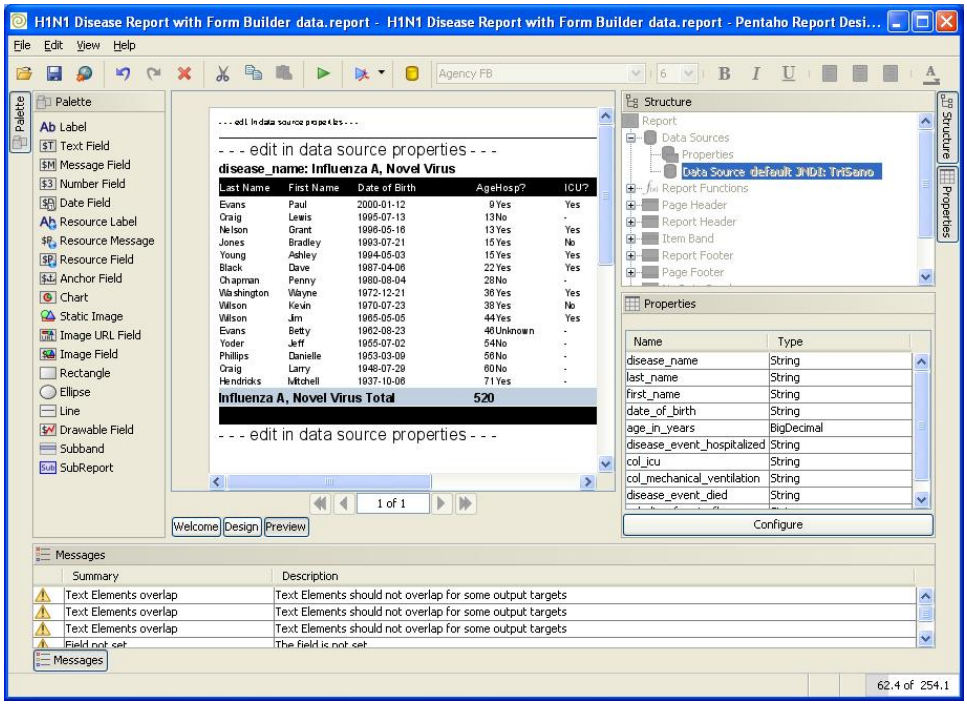
Copyright © 2009 Collaborative Software Initiative, Inc.

After saving the query and report and returning to the Report Designer, you will see three tabs near the bottom of the screen. They are:

Tab	Description
Welcome	This is the screen you started on in the Report Designer at the beginning of this report creation.
Design	This is the screen you are on now and where you will do all the design and formatting of the report.
Preview	This will show you a preview of the report, much like the Preview button in the Report Wizard

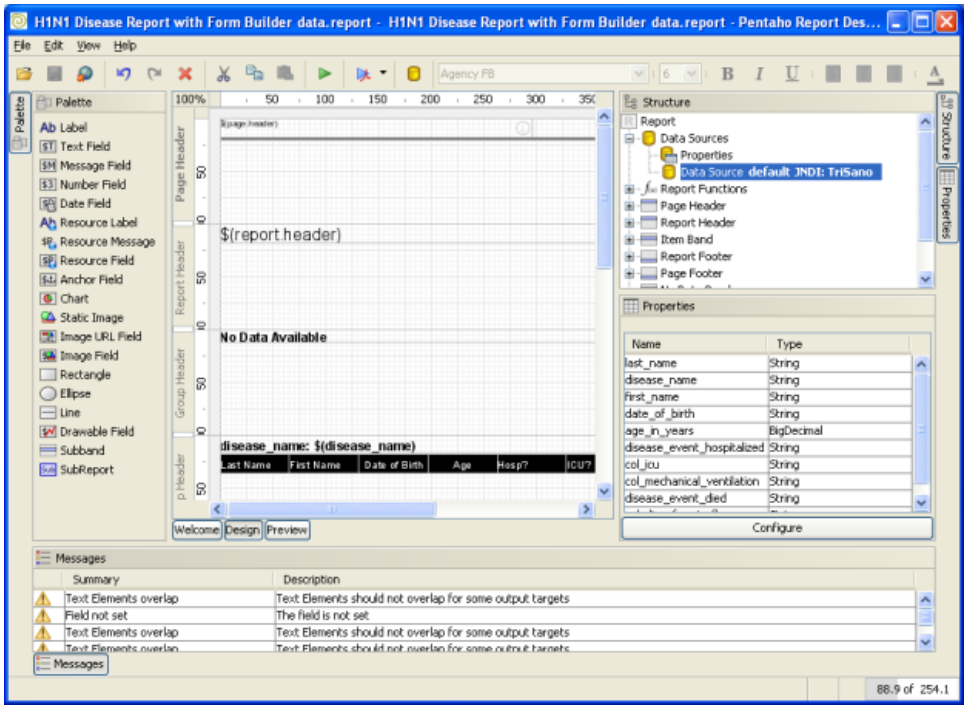


Click on **Preview** to see what your report looks like so far.



Copyright © 2009 Collaborative Software Initiative, Inc.

Click on **Design** to return back to the **Design** screen, as shown below.



Next Topic:

Returning to the Query

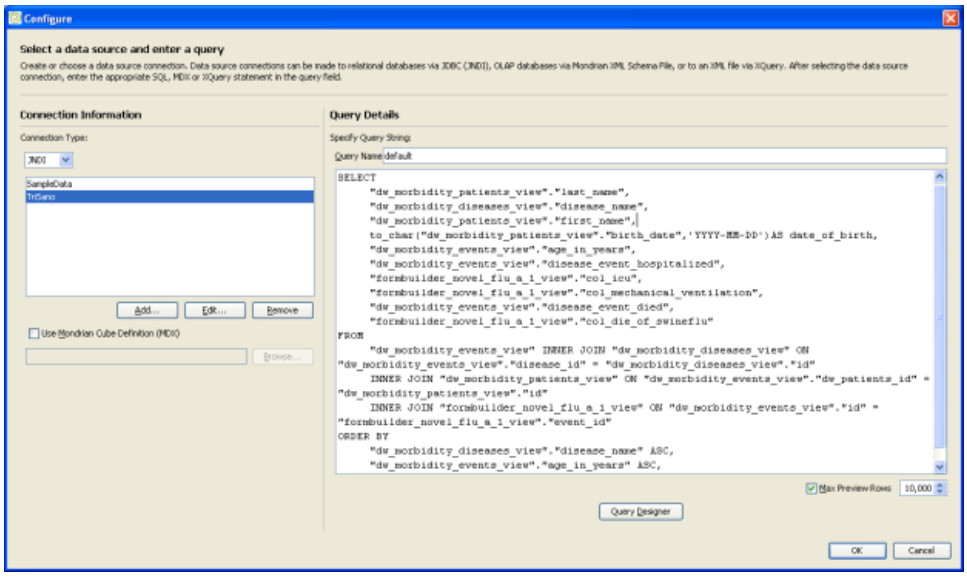
Copyright © 2009 Collaborative Software Initiative, Inc.

Returning to the Query

Returning to the Query

While you are in the Report Designer, you can return to the query you just created at any time. First, click the **+ (plus)** sign to the left of the **Data Source** line in the **Structure** section in the upper right hand corner of the Report Designer. Then click on the line **Data Source: default JNDI: TriSano**. (Your Data Source name might be different. When you click on this, a **Properties** box appears in the lower right hand corner of the screen with a **Configure** button displayed. Click on the **Configure** button at the bottom of the **Properties** section in the lower right hand corner of the screen.

The **Configure** window will pop up as shown below. You can edit the query right in the **Query Details** window, as we did earlier to change the Date format, or you can click on **Query Designer** and go back into the graphical **Query Designer** to modify the query.



Here is the query copied from the Query Details window above:

SELECT
"dw_morbidity_patients_view"."last_name",
"dw_morbidity_diseases_view"."disease_name",
"dw_morbidity_patients_view"."first_name",
to_char("dw_morbidity_patients_view"."birth_date','YYYY-MM-DD') AS date_of_birth,
"dw_morbidity_events_view"."age_in_years",
"dw_morbidity_events_view"."disease_event_hospitalized",
"formbuilder_novel_flu_a_1_view"."col_icu",
"formbuilder_novel_flu_a_1_view"."col_mechanical_ventilation",
"dw_morbidity_events_view"."disease_event_died",
"formbuilder_novel_flu_a_1_view"."col_die_of_swineflu"
FROM
"dw_morbidity_events_view" INNER JOIN "dw_morbidity_diseases_view" ON
"dw_morbidity_events_view"."disease_id" = "dw_morbidity_diseases_view"."id"
INNER JOIN "dw_morbidity_patients_view" ON "dw_morbidity_events_view"."dw_patients_id" =
"dw_morbidity_patients_view"."id"
INNER JOIN "formbuilder_novel_flu_a_1_view" ON "dw_morbidity_events_view"."id" =
"formbuilder_novel_flu_a_1_view"."event_id"
ORDER BY
"dw_morbidity_diseases_view"."disease_name" ASC,
"dw_morbidity_events_view"."age_in_years" ASC,
"dw_morbidity_patients_view"."last_name" ASC

Next Topic:

Returning to the Report Designer

Copyright © 2009 Collaborative Software Initiative, Inc.

Returning to the Report Designer

Returning to the Report Designer

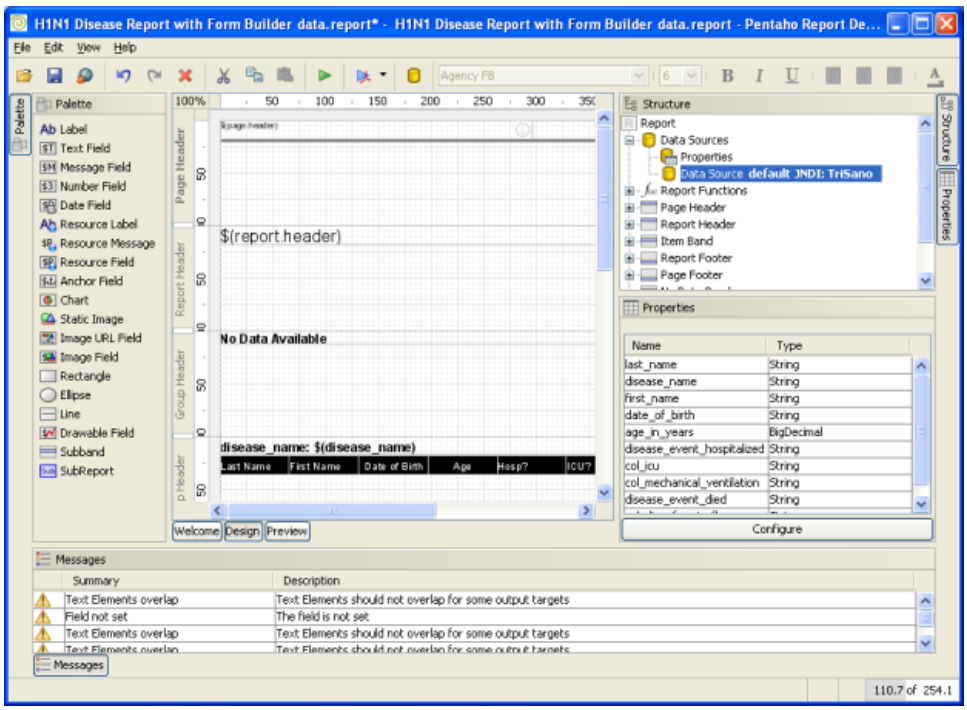
Click on **Cancel** to return back to the **Report Designer** and the **Design** window.

Let's talk about the **Design** window in the middle of the screen. In it are all the different parts of the report layout listed vertically down the left hand side of the **Design** window. From the top to the bottom, you see:

- Page Header
- Report Header
- Group Header

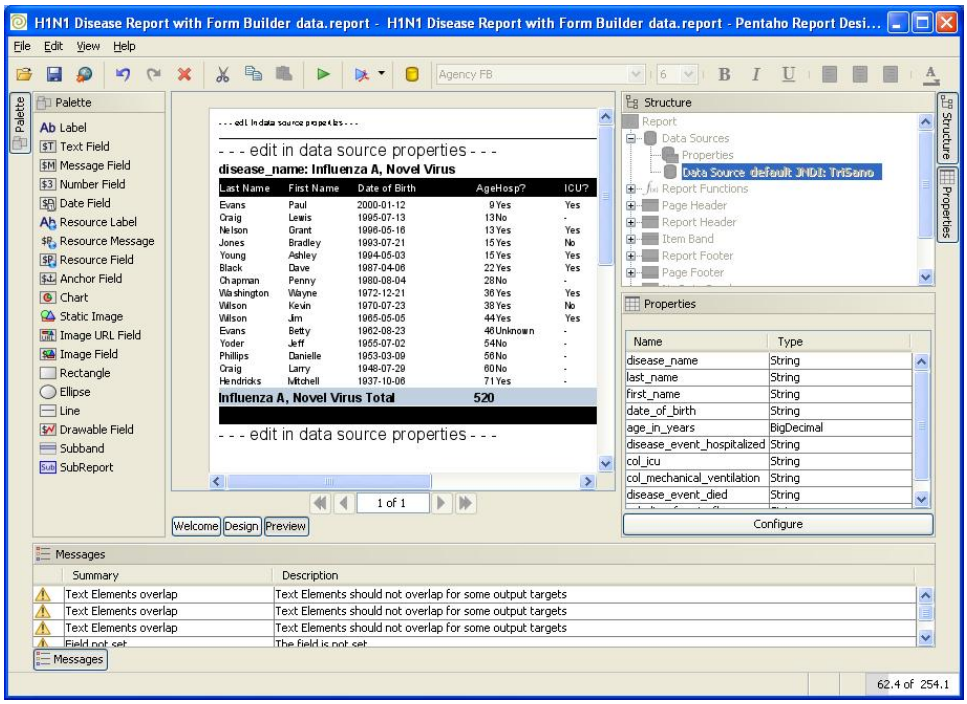
and if you scoll down the window, you will see

- A second Group Header
- Item Band
- Group Footer
- A second Group Footer
- Report Footer
- Page Footer

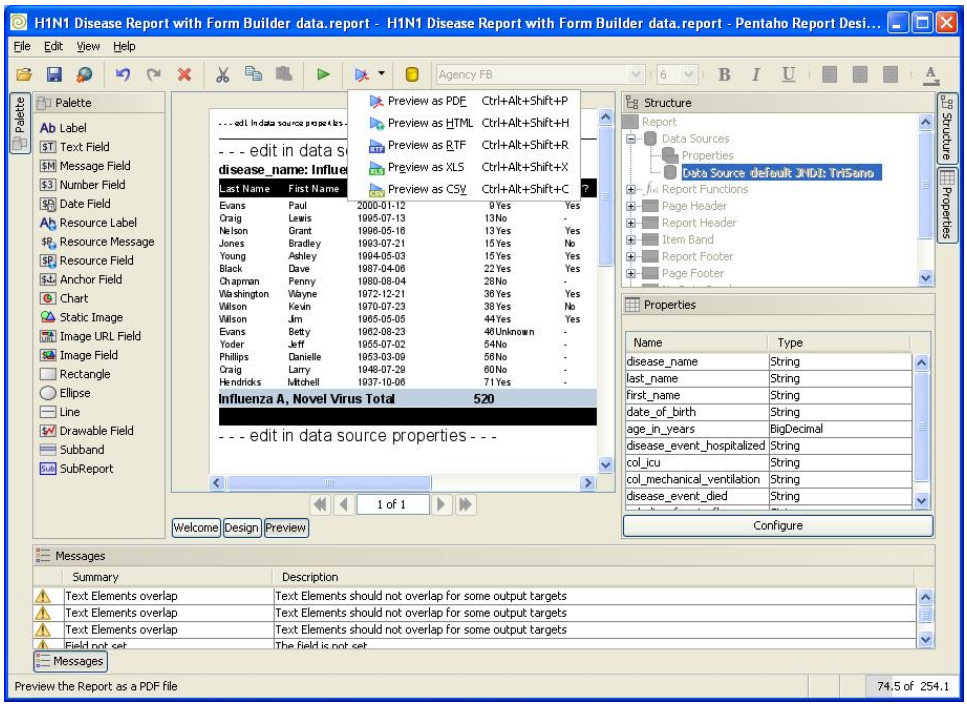


Click on **Preview** again to see the report again as shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.

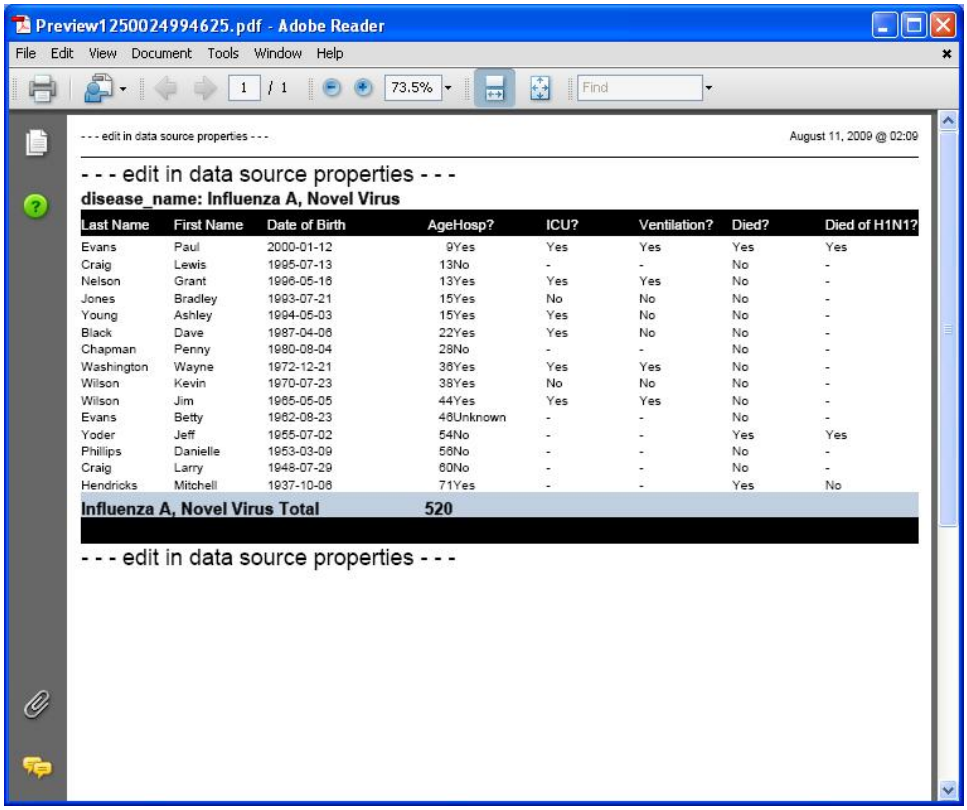


You can also **Preview** the report at any time, even in the **Design** screen, in PDF, HTML, RTF, XLS or CSV format. To see these options, click on the very small black triangle to the right of the bigger blue triangle as shown below, and then select **Preview as PDF**.



Here is the PDF report. When finished reviewing, just close the PDF window and return back to the Report Designer.

Copyright © 2009 Collaborative Software Initiative, Inc.



Preview1250024994625.pdf - Adobe Reader

File Edit View Document Tools Window Help

1 / 1 73.5% Find

- - - edit in data source properties - - - August 11, 2009 @ 02:09

- - - edit in data source properties - - -

disease_name: Influenza A, Novel Virus

Last Name	First Name	Date of Birth	AgeHosp?	ICU?	Ventilation?	Died?	Died of H1N1?
Evans	Paul	2000-01-12	9Yes	Yes	Yes	Yes	Yes
Craig	Lewis	1995-07-13	13No	-	-	No	-
Nelson	Grant	1996-05-16	13Yes	Yes	Yes	No	-
Jones	Bradley	1993-07-21	15Yes	No	No	No	-
Young	Ashley	1994-05-03	15Yes	Yes	No	No	-
Black	Dave	1987-04-06	22Yes	Yes	No	No	-
Chapman	Penny	1980-08-04	28No	-	-	No	-
Washington	Wayne	1972-12-21	36Yes	Yes	Yes	No	-
Wilson	Kevin	1970-07-23	38Yes	No	No	No	-
Wilson	Jim	1965-05-05	44Yes	Yes	Yes	No	-
Evans	Betty	1962-08-23	46Unknown	-	-	No	-
Yoder	Jeff	1955-07-02	54No	-	-	Yes	Yes
Phillips	Danielle	1953-03-09	56No	-	-	No	-
Craig	Larry	1948-07-29	60No	-	-	No	-
Hendricks	Mitchell	1937-10-06	71Yes	-	-	Yes	No
Influenza A, Novel Virus Total			**520**				

- - - edit in data source properties - - -

Next Topic:

Changing Headers and Footers using the Report Designer

Copyright © 2009 Collaborative Software Initiative, Inc.

Changing Headers and Footers using the Report Designer

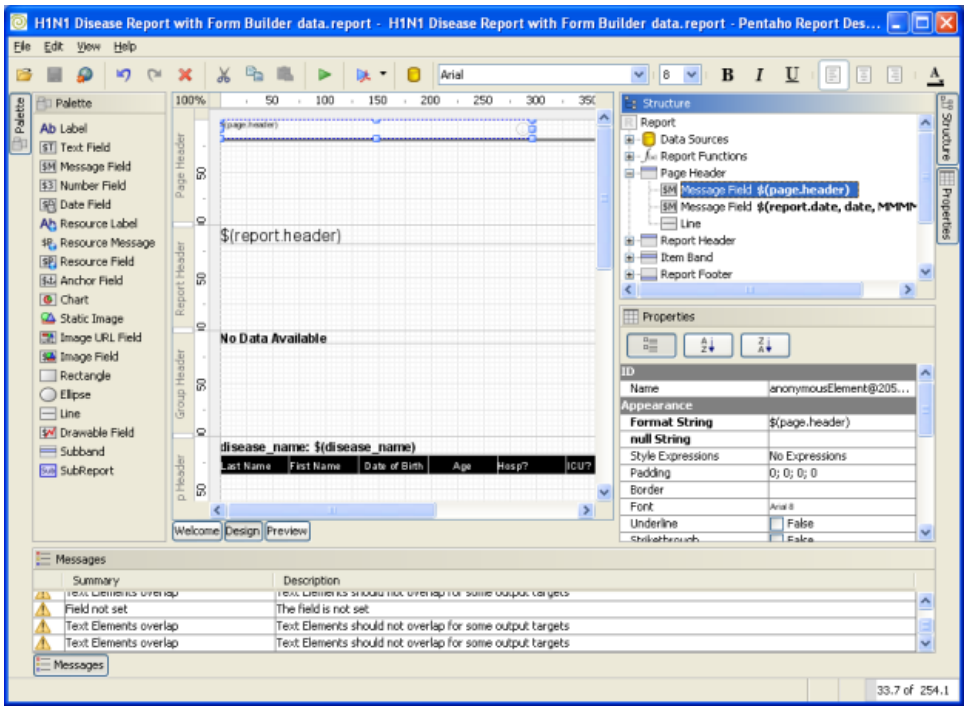
Changing Headers and Footers using the Report Designer

Now, let's start changing some of the headers and footers in the report.

To start, click on the **Design** tab if you are not already on the **Design** screen.

Then, click on the **$(page_header)** field at the top of the screen. You will see an outline appear with blue squares around it. Also, in the **Structure** section, the same field in the **Page Header** section will be highlighted.

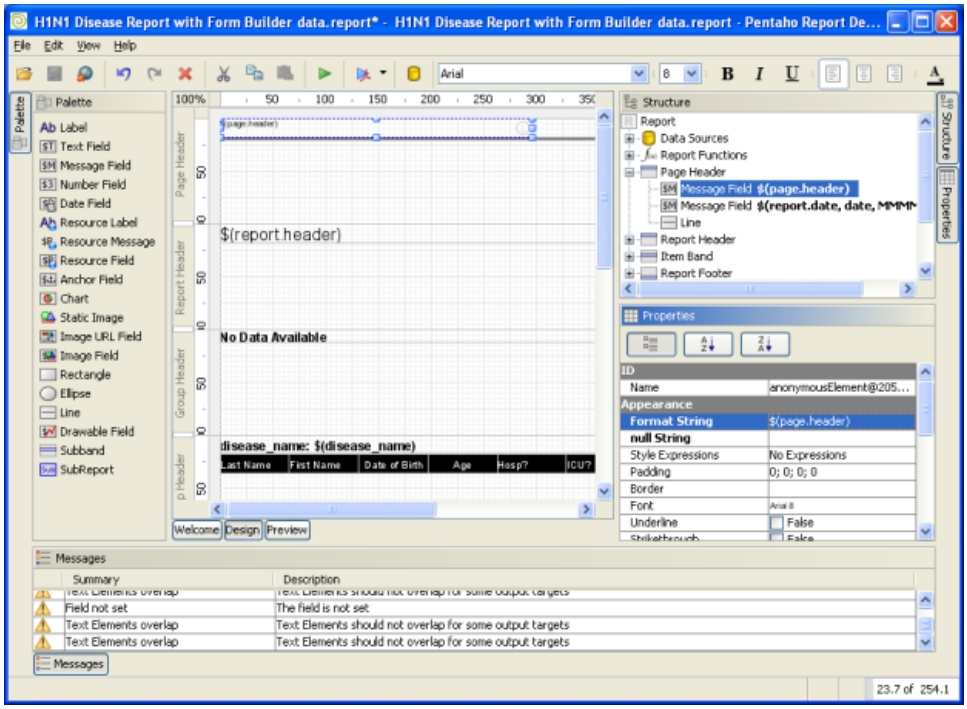
Your screen should look like the one shown below.



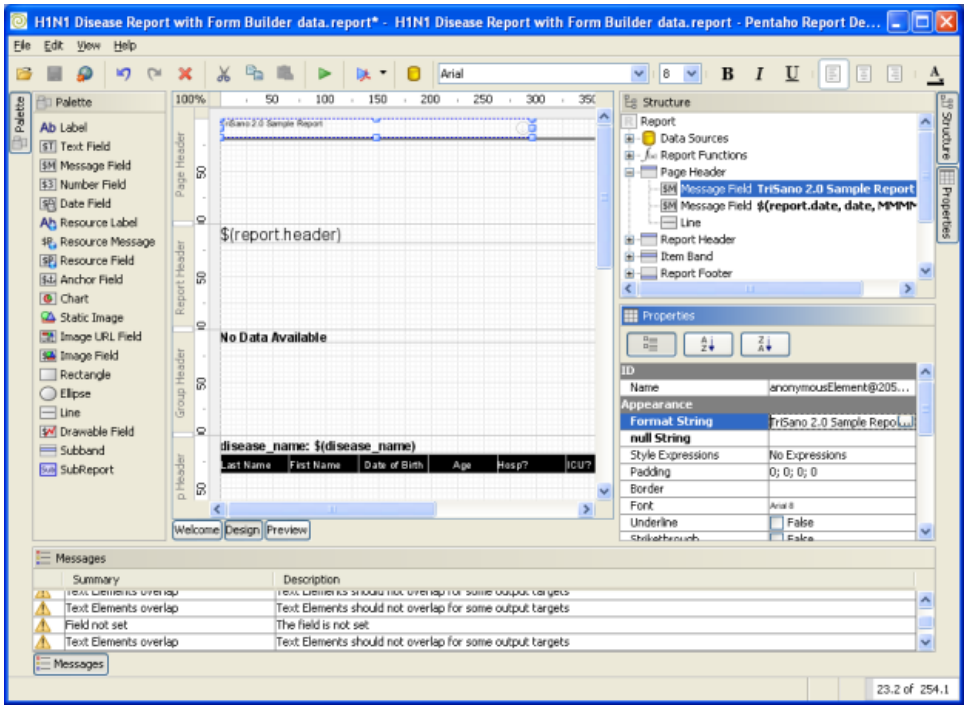
As you click on one of the elements in the **Structure** section or one is highlghted, a set of **Properties** will display in the lower right hand corner of the screen, including information about the **Appearance** of that element. There are many options in the **Properties** section, and they will vary depending on the element that is highlighted in the **Structure** section.

One of the elements under **Appearance** for the Page Header is **Format String** whch for now says **$(page_header)**. To change this to a more acceptable Page Header, start by clicking **Format String** in the **Properties** section, as shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.

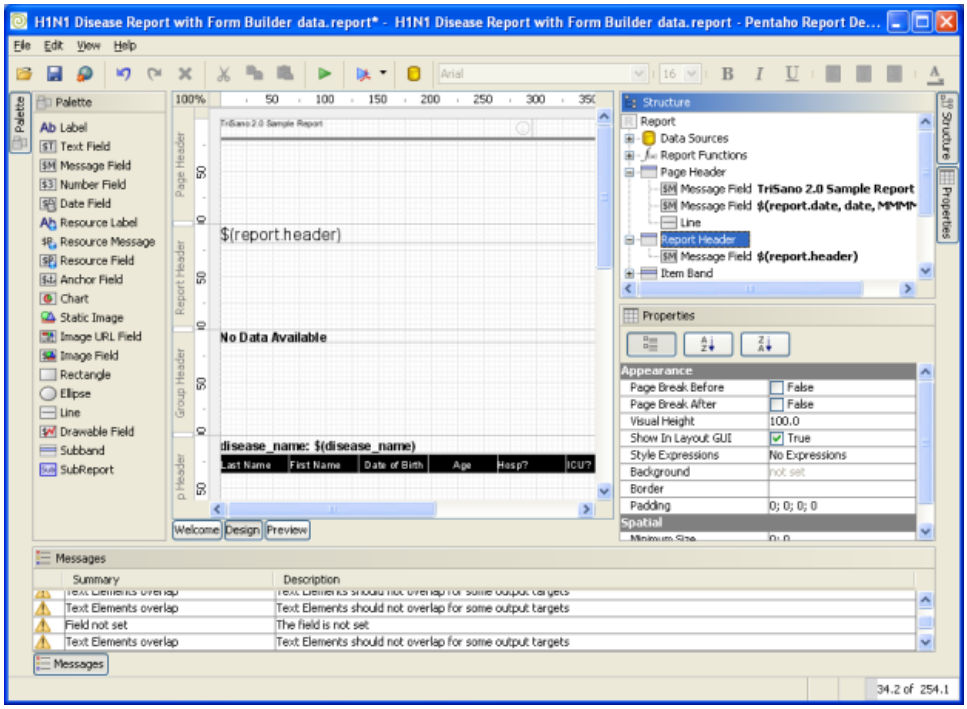


Now, if you click on **$(page_header)**, you can just delete that text, and type in the text you would like to have for the header. In this example, I typed **TriSano 2.0 Sample Report**. Once you hit **Enter**, the text will appear in the Design window, as shown below.

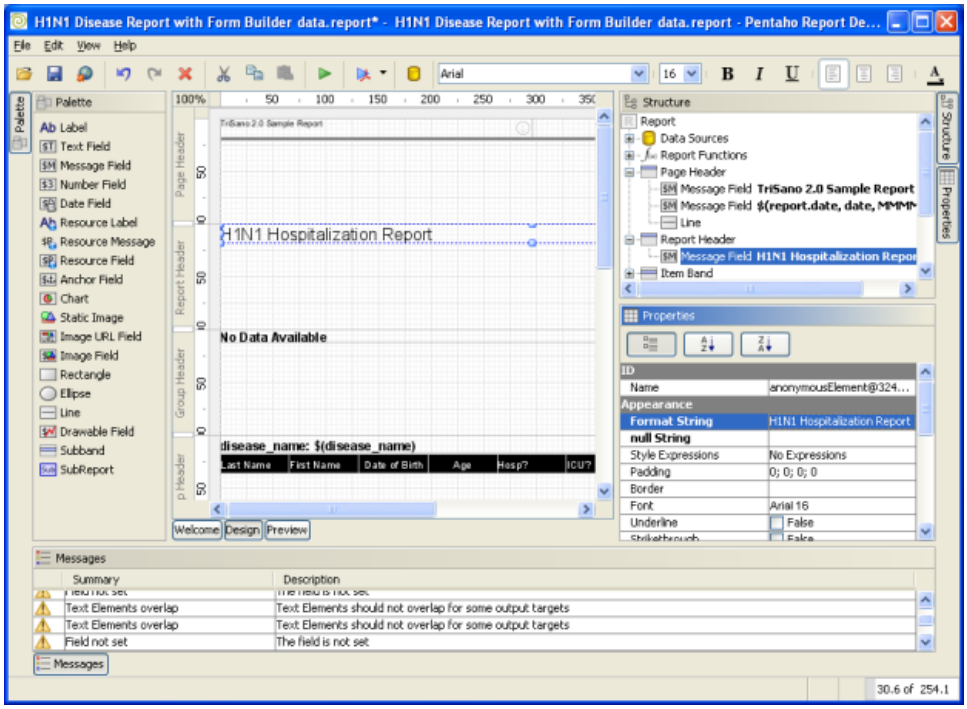


You can also access the different elements in the report by clicking on the tree structure in the Structure section. To replace the Report Header, which now reads "$(report_header)", click on the + sign to the left of **Report Header** and the tree structure will expand to show the elements that make up the **Report Header**, as shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.



Next, click on the line that says **Message Field $(report_header)** in the **Structure** section. This will cause the **Properties** for this element to be displayed. Now, type a new report header in the **Format String** property. I used **H1N1 Hospitalization Report** for this example, as shown below. Once you hit **Enter**, the new report header will appear in the Design window.



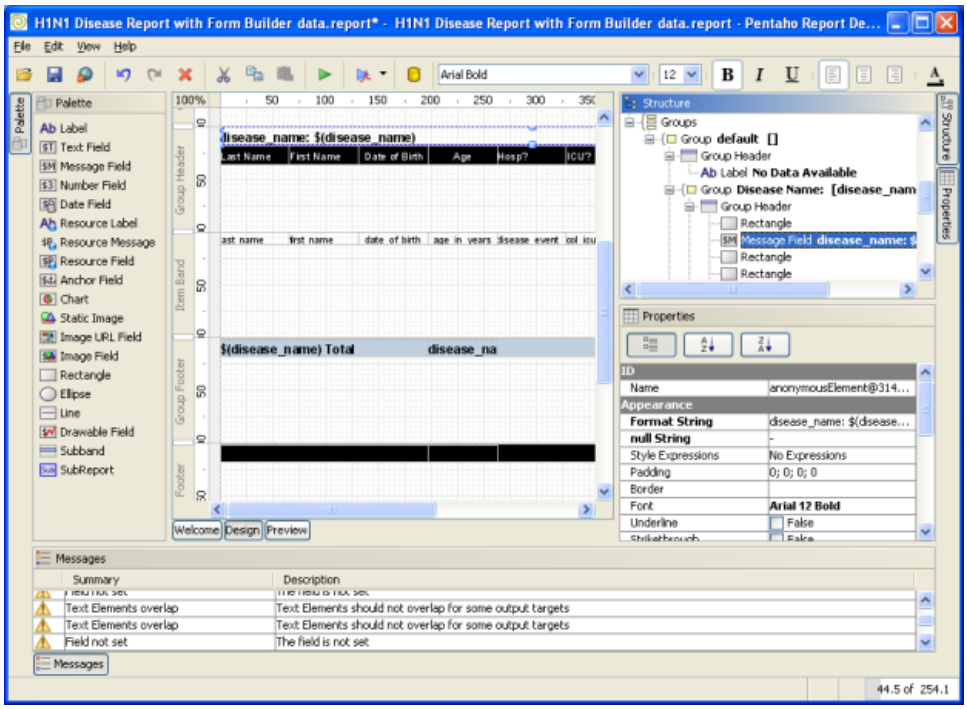
Next, let's change the **Group Header**. You may need to scroll down to see it.

It now reads **disease_name: $(disease_name)** in the **Design** window. When you preview the report, you see that **$(disease_name)** is replaced by the actual **disease_name**, such as **Influenza Virus, Novel A** from the query. That is OK, but the first part of the Group Header, **disease_name**:, would look better if it said: **Disease Name:.**

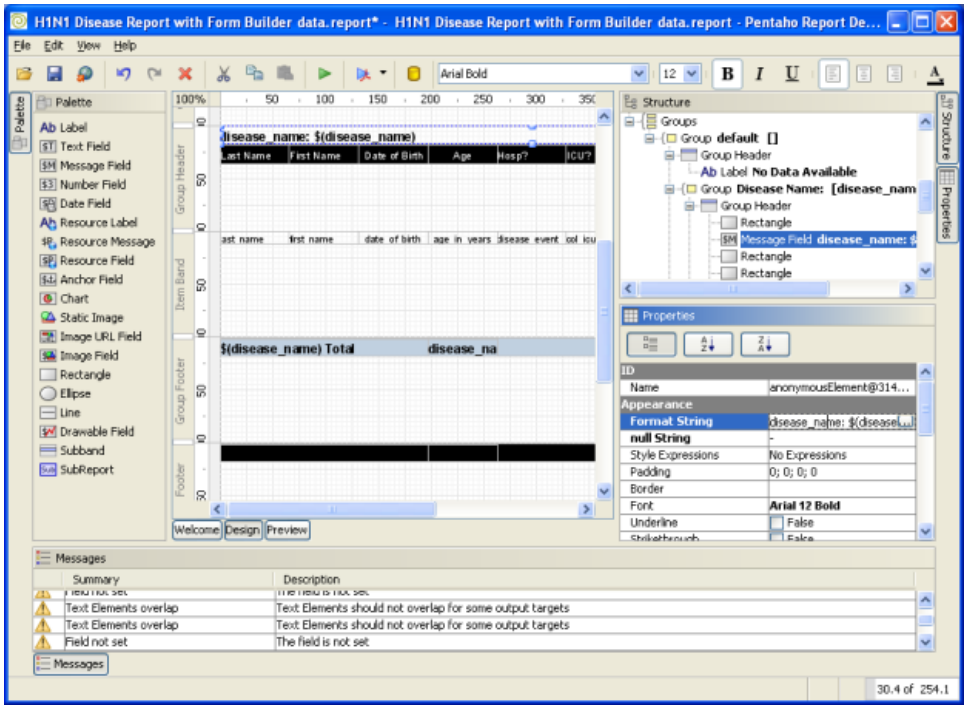
To change this, you can click on the **$disease_name: $(disease_name)** text in the **Design** window, or click on the + sign to the left of **Groups** in the **Structure** section, and then keep expanding that section until you

Copyright © 2009 Collaborative Software Initiative, Inc.

see the line **Message Field disease_name: $(disease_name)**. Click on that line in the **Structure** section and the **Properties** for that will appear in the **Properties** section, as shown in the screen below.

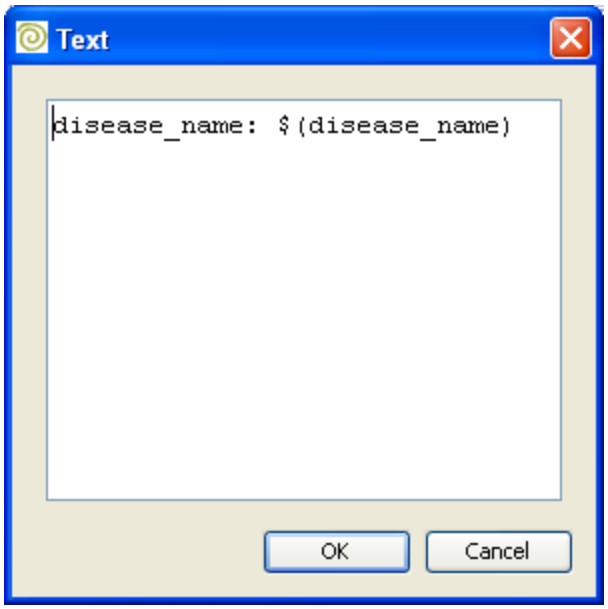


Instead of just keying over the text in the Format String element, there is a different option that can be used. Notice that when you click on **disease_name: $(disease_name)** to the right of **Format String**, there is a small pushbutton at the right end of the entry field. You should see it in the screen shown below right after **disease_name: $(disease**.

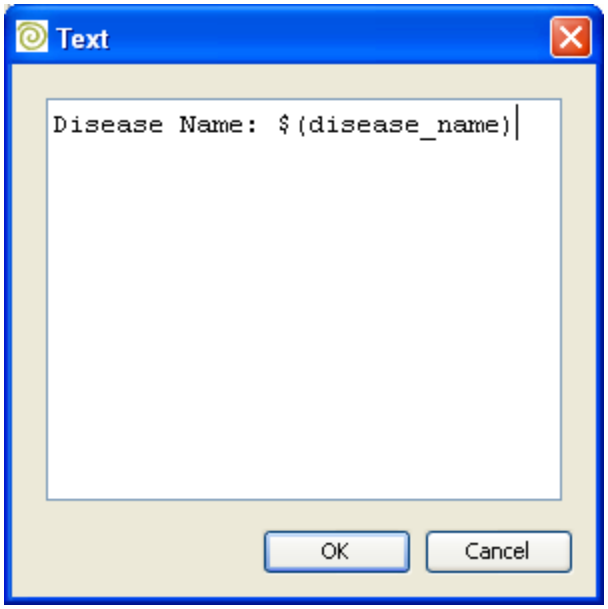


Click on that **pushbutton** and a entry window, shown below, will pop up where you can update the text in the field.

Copyright © 2009 Collaborative Software Initiative, Inc.

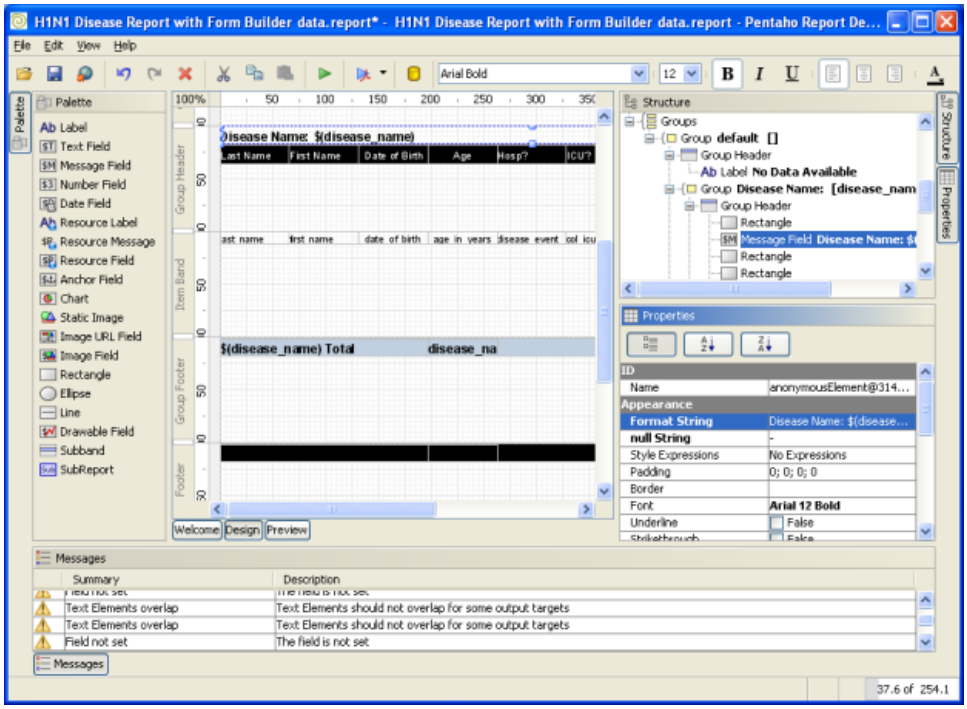


To change the text, just replace the first part so that it reads:



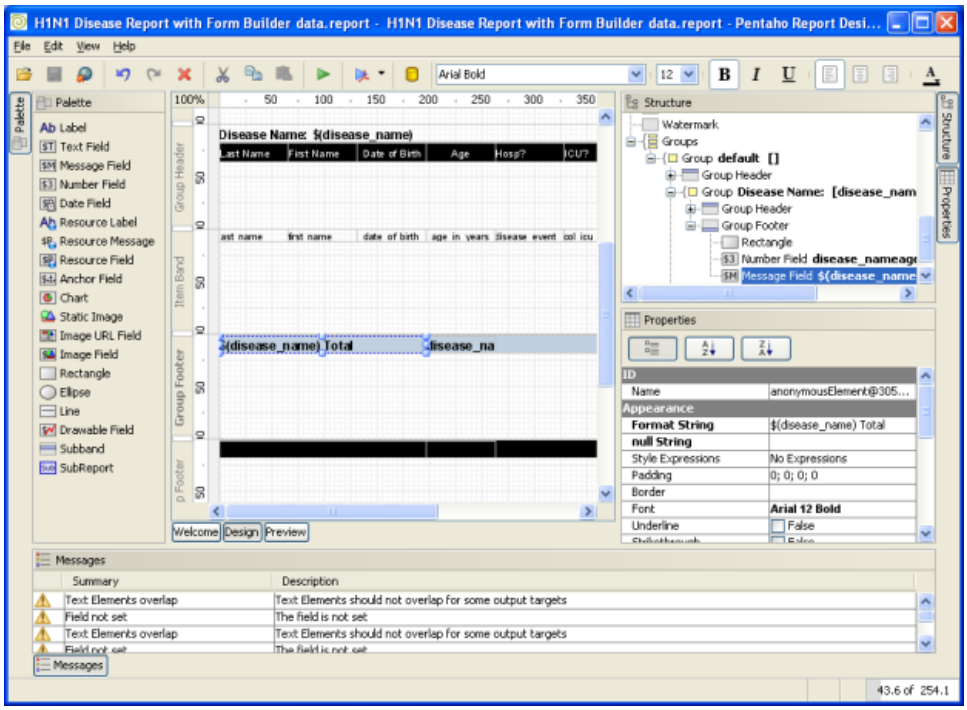
Close the window by clicking on **OK** and you should see your new text in the **Group Header** section of the **Design** window (shown near the top of the screen below).

Copyright © 2009 Collaborative Software Initiative, Inc.



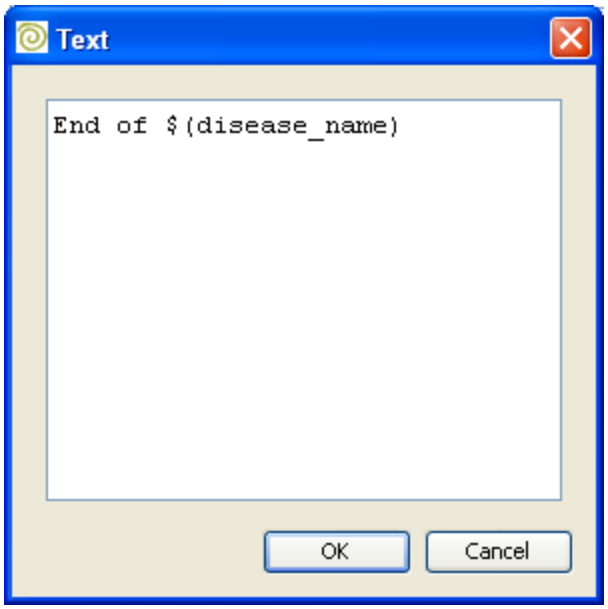
Next, you are going to change the **Group Footer**, which right now is totaling the **Age** column, which we don't want it to do. You may need to scroll the Design window to see it.

To start, click on the "**$(disease_name) Total**" in the **Group Footer** section of the **Design** window. It should have a highlighted box around it with blue squares as shown below, and the same element will be highlighted in the **Structure** section and its properties shown in the **Properties** section as shown below.

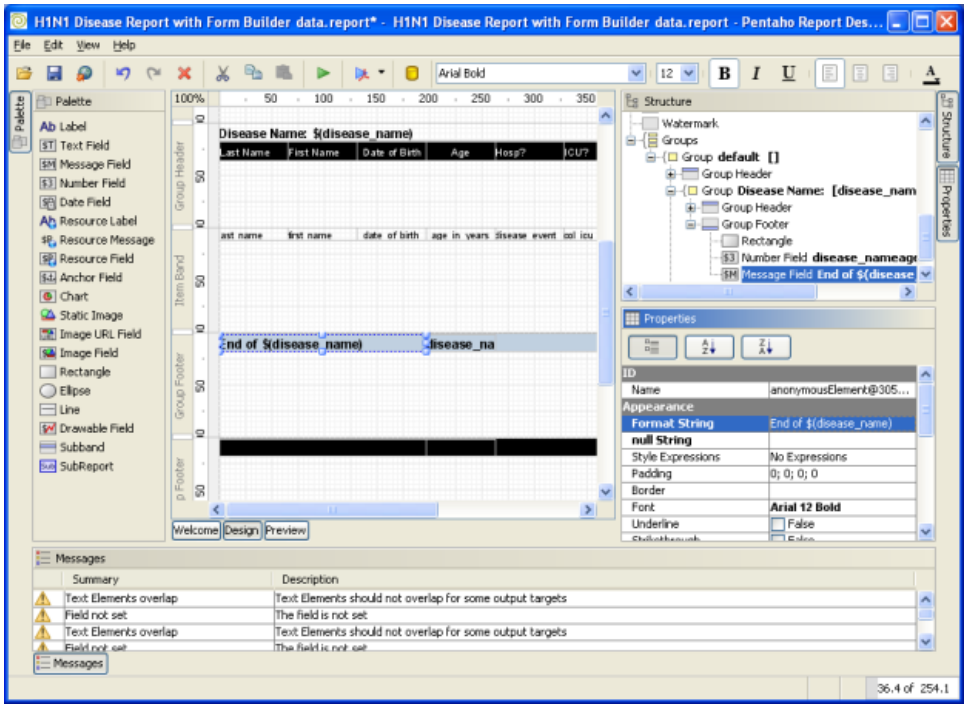


Click on the small pushbutton at the end of the Format String line in the Properties section, and change the text from **$(disease_name) Total** to **End of $(disease_name)** as shown in the pop up window shown below.

Copyright © 2009 Collaborative Software Initiative, Inc.



Click **OK** to close the window and return to the Report Designer. Your screen should look like the one shown below.



Next Topic:

Removing a Column from the Report

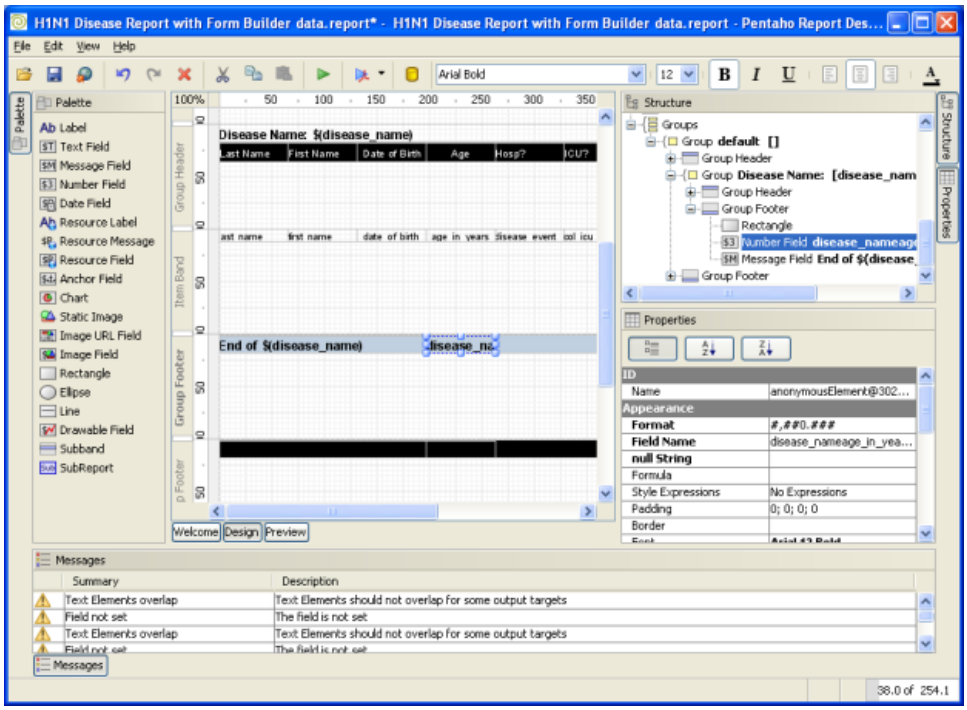
Copyright © 2009 Collaborative Software Initiative, Inc.

Removing a Column from the Report

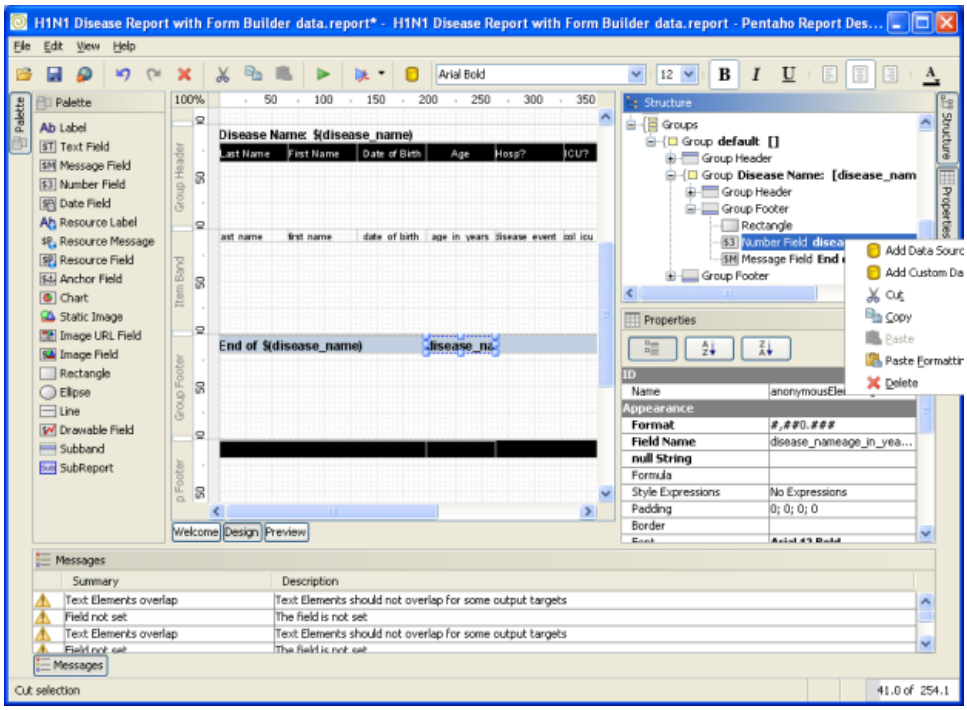
Removing a Column from the Report

Now click on the **"disease_name"** field to the right of the field you were just working with. This is were the sum of the Ages is being calculated. The element in the **Structure** section for this total is highlighted.

Your screen should look like the one shown below.

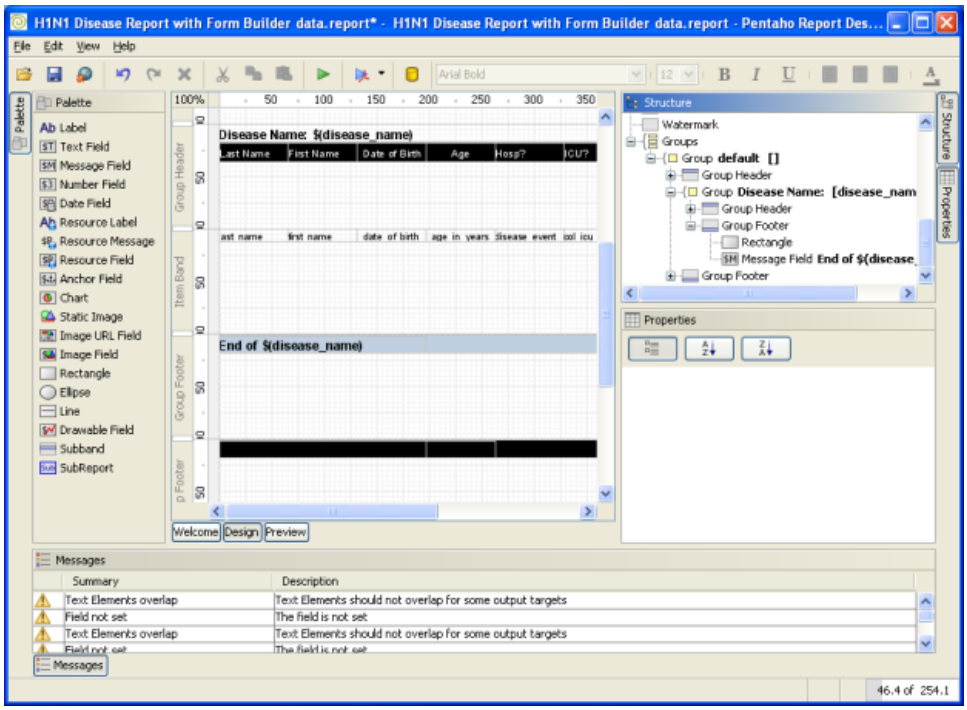


You don't need this element, so you can just delete it from the report by right clicking on that element in the **Structure** section and selecting either **Cut** or **Delete**, as shown below.

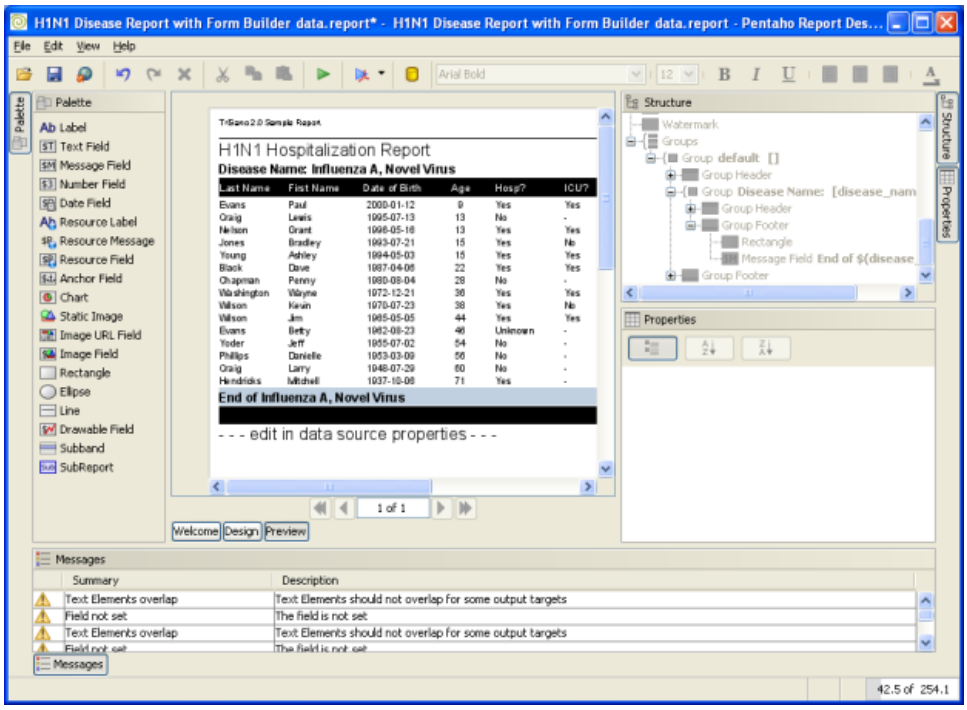


As you can see in the screen below, that element is no longer part of the report.

Copyright © 2009 Collaborative Software Initiative, Inc.



Let's see what our report looks like now. Click on **Preview**. Your screen will look like the one shown below.



Next Topic:

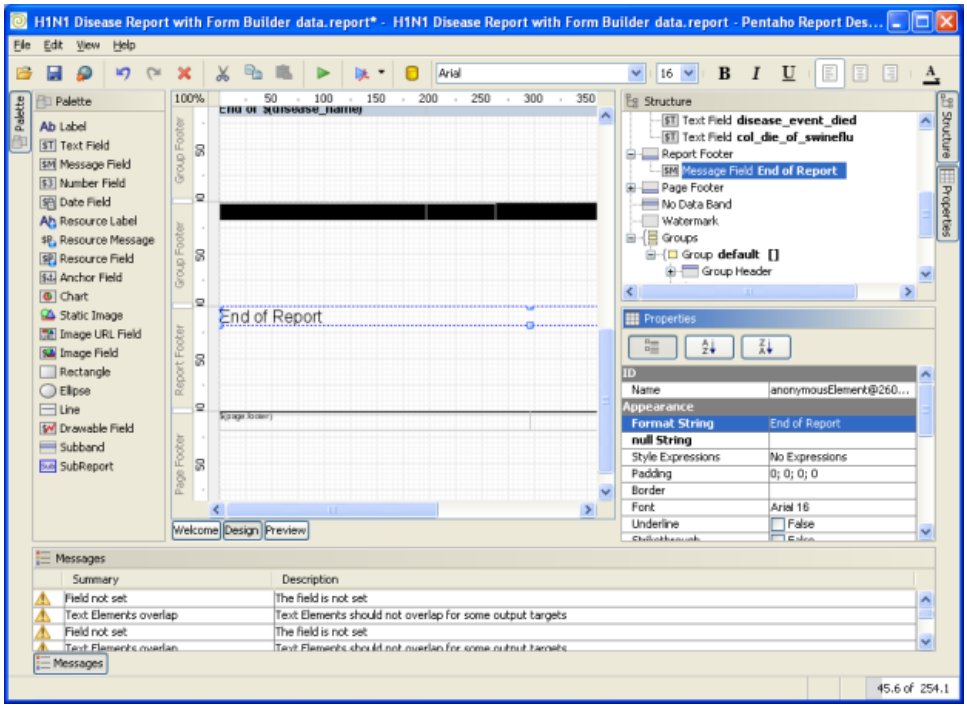
Changing the Report Footer

Copyright © 2009 Collaborative Software Initiative, Inc.

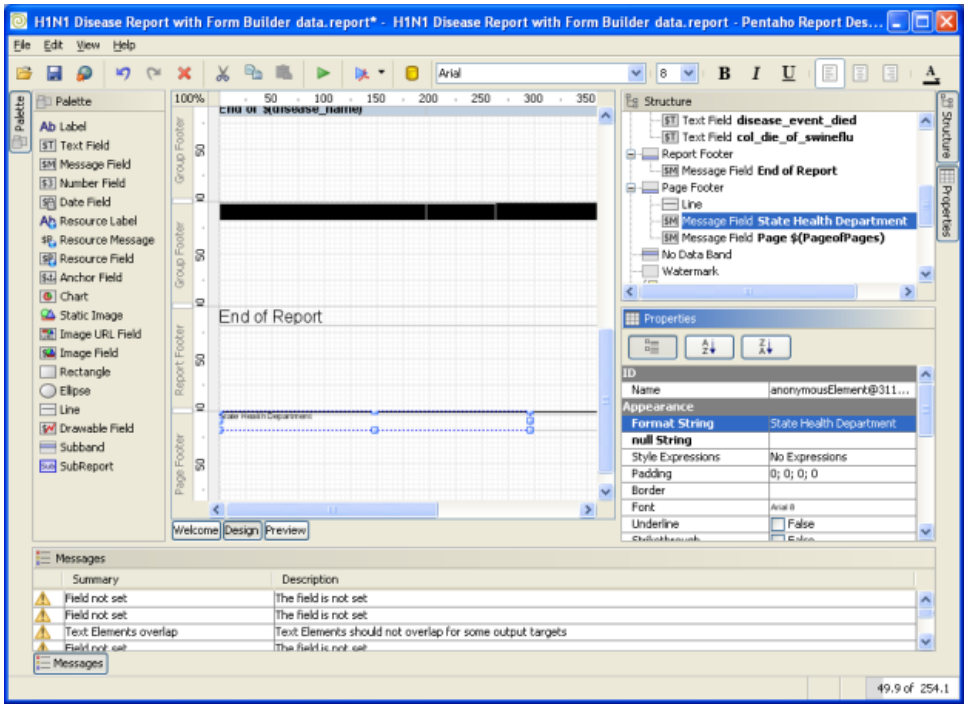
Changing the Report Footer

Changing the Report Footer

Notice that we have not yet changed the **Report Footer.** It still says **"edit in date source properties"**. To change that, first return to the **Design** window by clicking on **Design**, just click on **"$(report footer)"** in the **Report Footer** section of the **Design** window, update the text in the **Properties** section next to **Format String** to say "**End of Report**" and press **Enter**. Your screen should look like the one shown below.



In the same way, update the **Page Footer** (you may need to scroll down) to read **State Health Department**. When finished, your screen should look like this,



Copyright © 2009 Collaborative Software Initiative, Inc.

Next Topic:

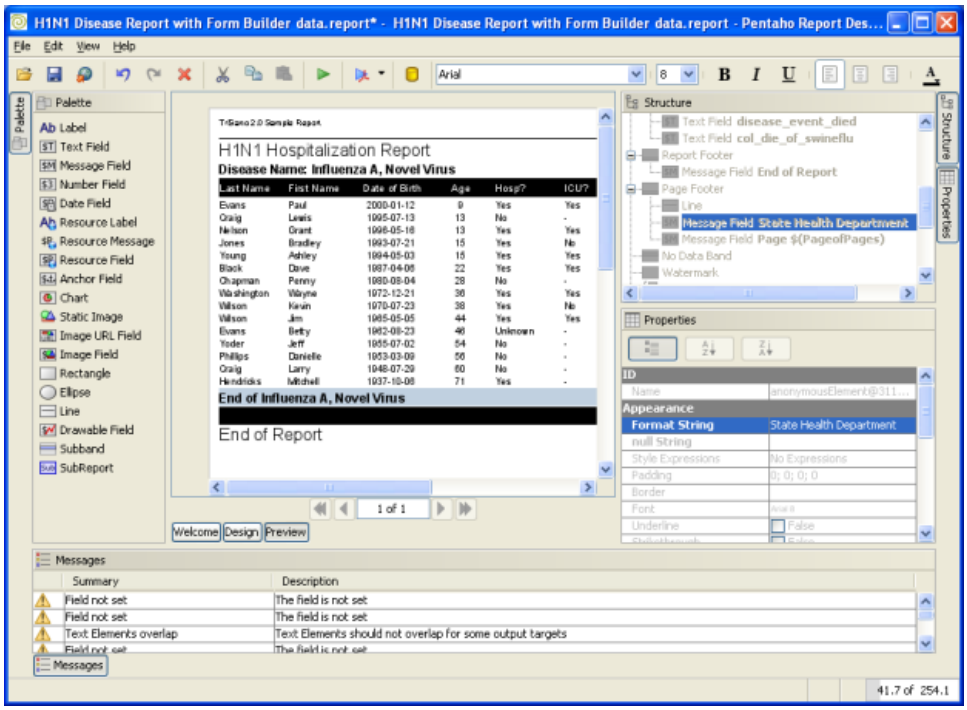
Final Report

Copyright © 2009 Collaborative Software Initiative, Inc.

Final Report

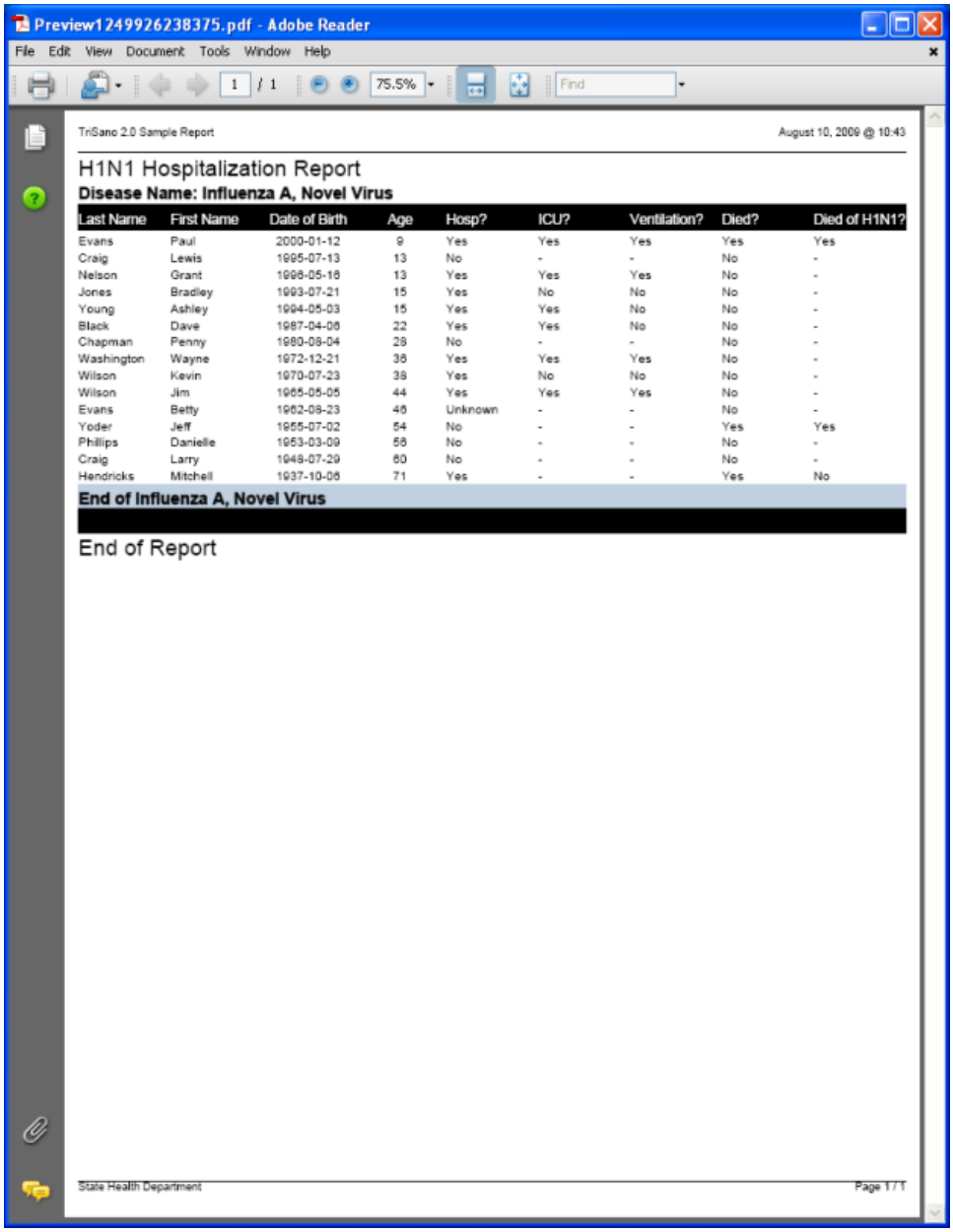
Final Report

By clicking of **Preview**, your report should look like this,



and the finished report in PDF format should look like ths.

Copyright © 2009 Collaborative Software Initiative, Inc.



Preview1249926238375.pdf - Adobe Reader

File Edit View Document Tools Window Help

TriSano 2.0 Sample Report — August 10, 2009 @ 10:43

H1N1 Hospitalization Report

Disease Name: Influenza A, Novel Virus

Last Name	First Name	Date of Birth	Age	Hosp?	ICU?	Ventilation?	Died?	Died of H1N1?
Evans	Paul	2000-01-12	9	Yes	Yes	Yes	Yes	Yes
Craig	Lewis	1995-07-13	13	No	-	-	No	-
Nelson	Grant	1996-05-18	13	Yes	Yes	Yes	No	-
Jones	Bradley	1993-07-21	15	Yes	No	No	No	-
Young	Ashley	1994-05-03	15	Yes	Yes	No	No	-
Black	Dave	1987-04-06	22	Yes	Yes	No	No	-
Chapman	Penny	1980-08-04	29	No	-	-	No	-
Washington	Wayne	1972-12-21	38	Yes	Yes	Yes	No	-
Wilson	Kevin	1970-07-23	38	Yes	No	No	No	-
Wilson	Jim	1965-05-05	44	Yes	Yes	Yes	No	-
Evans	Betty	1962-08-23	46	Unknown	-	-	No	-
Yoder	Jeff	1955-07-02	54	No	-	-	Yes	Yes
Phillips	Danielle	1953-03-09	56	No	-	-	No	-
Craig	Larry	1948-07-29	60	No	-	-	No	-
Hendricks	Mitchell	1937-10-06	71	Yes	-	-	Yes	No

End of Influenza A, Novel Virus

End of Report

State Health Department — Page 1 / 1

Next Topic:

Pentaho™ Report Designer Tool Summary

Copyright © 2009 Collaborative Software Initiative, Inc.

Pentaho™ Report Designer Tool Summary

Pentaho™ Report Designer Tool Summary

It's probably a good idea to save your query and report at this time.

There is lot's more to the Report Designer for you to explore, but I hope this gives you the basic knowledge you need to get started.
For additional information, click on **Help** at the top of the screen, then click on **Getting Started...**. This will open up the **Pentaho Report Designer User Guide** in PDF format. You can view or print all or part of the guide.

Have fun exploring!

Next Topic:

Pentaho™ Analysis - OLAP Tool

Copyright © 2009 Collaborative Software Initiative, Inc.

Pentaho™ Analysis - OLAP Tool

The following is an example of using the Pentaho™ Analysis Services for data cube/OLAP support.

Introduction

Pentaho's OLAP function is useful but non-intuitive. It's the purpose of this section to give you a demonstration of its capabilities and use.

OLAP databases resemble SQL databases in many ways, but have some notable differences. This blog post gives more details on the subject. This introduction assumes you're familiar with the concepts of measures and their aggregators, dimensions, and cubes, all topics introduced in the blog post.

Next Topic:

Starting an OLAP Session

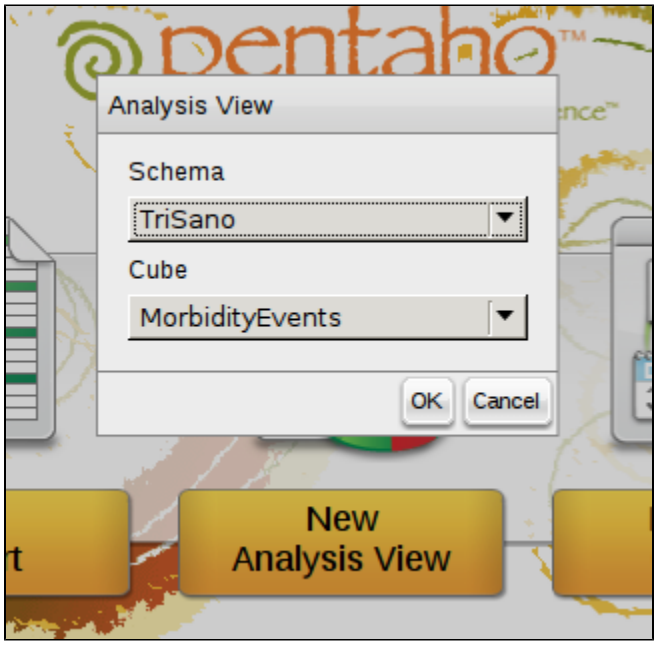
Copyright © 2009 Collaborative Software Initiative, Inc.

Starting an OLAP Session

Starting an OLAP Session

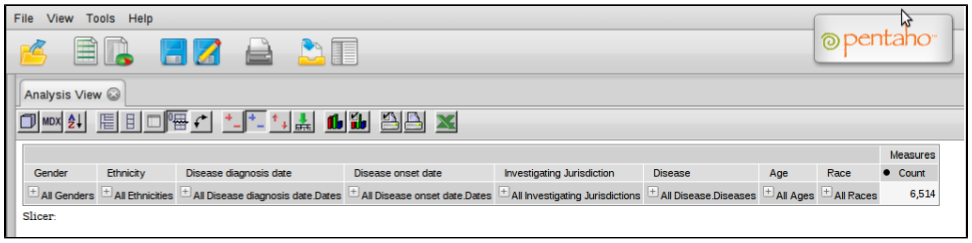
To begin an OLAP session from the Pentaho User Console, click **"New Analysis View"**.

Pentaho then prompts the user to choose a cube.



In CSI TriSano™ there are two OLAP cubes: one for **Morbidity Events**, and one for **Contact Events**. These are effectively identical, except that the **Morbidity Events** have more details. For now, choose the **Morbidity Events** cube.

Having chosen a cube, Pentaho™ shows the initial **Analysis View** page.

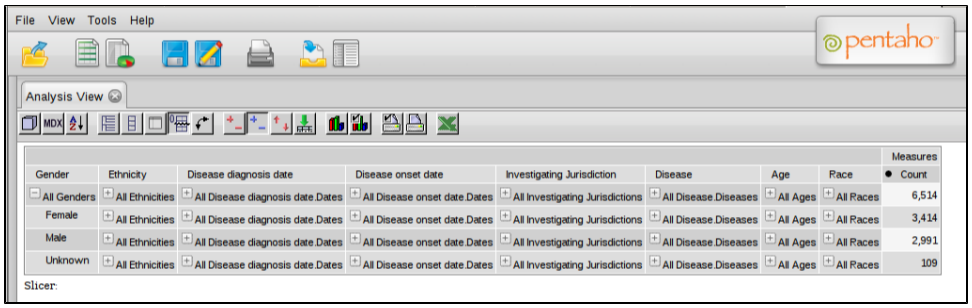


This is a spreadsheet-like interface, with the cube's dimensions on the left side:

- Gender
- Ethnicity
- Disease diagnosis date
- Disease onset date
- Investigating Jurisdiction
- Disease
- Age
- Race

and the default "Count" measure as the one column in the result. Each dimension has a small **""** symbol next to it. Clicking that symbol will expand the dimension's hierarchy, and calculate the visible measures for each possible value for the expanded dimension. Here's an example of clicking on the "" sign to the left of **All Genders**.

Copyright © 2009 Collaborative Software Initiative, Inc.



In the screen shown above, expanding the **Gender** dimension results in a display of the total number of morbidity events, plus the totals for male patients, female patients, and patients of unknown birth gender.

Next Topic:

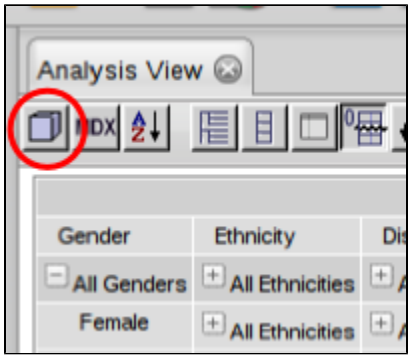
Showing and Hiding Measures and Dimensions

Copyright © 2009 Collaborative Software Initiative, Inc.

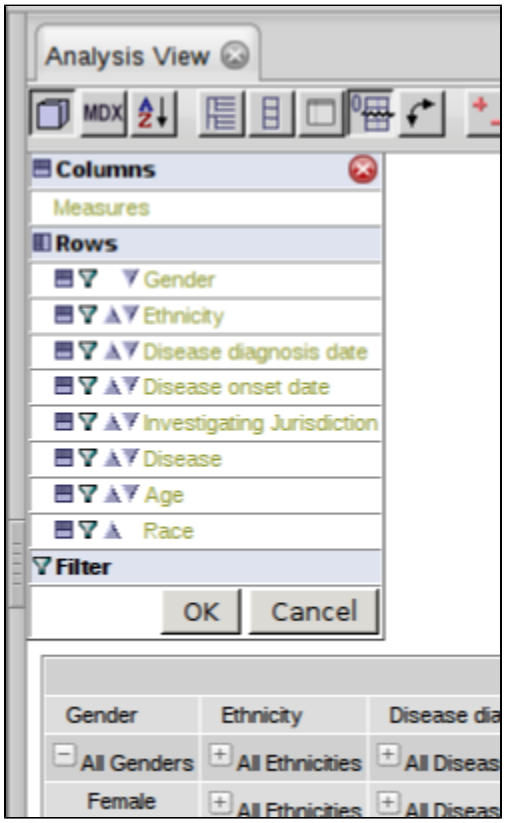
Showing and Hiding Measures and Dimensions

Showing and Hiding Measures and Dimensions

The leftmost button in the Analysis View toolbar opens the **OLAP Navigator**, which controls most aspects of the OLAP view, including which measures and dimensions are visible, whether each makes up a column or row of the database, what filters are applied to the data, etc.

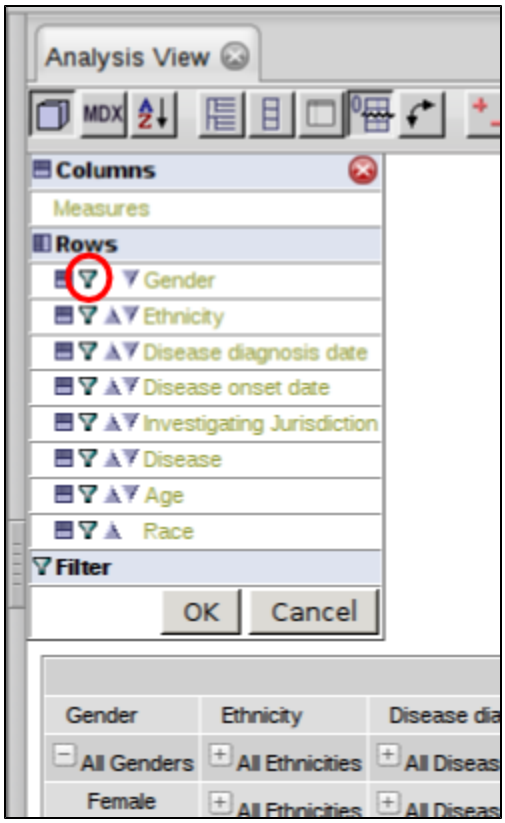


For purposes of this example, assume the user is uninterested in using the gender to filter or analyze data, and would like to remove it from the display to decrease confusion or conserve screen real estate. Click the leftmost toolbar button to open the **OLAP Navigator**.

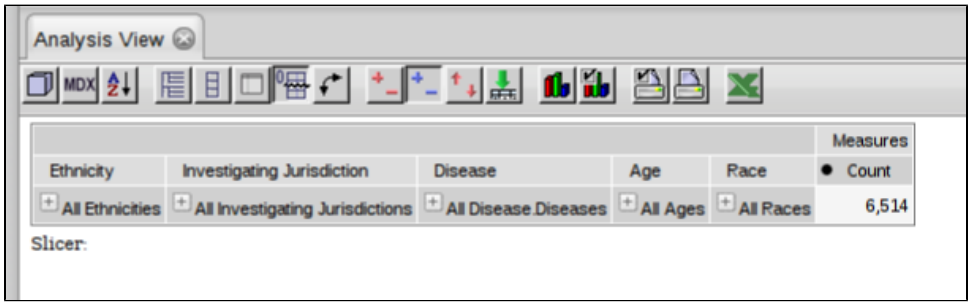


This page allows dimensions, such as **Gender**, to be moved between rows and columns of the OLAP view, and to be used as "filters". Filter dimensions aren't shown on the OLAP screen, so to hide the Gender dimension, make it a filter by clicking the funnel-like icon next to the **Gender** dimension.

Copyright © 2009 Collaborative Software Initiative, Inc.

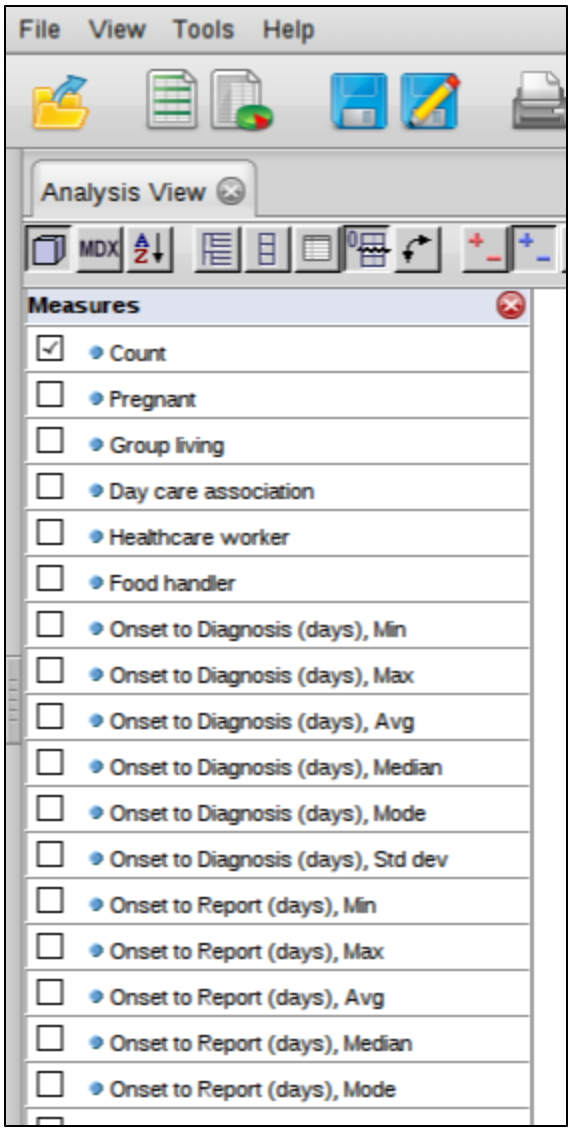


This moves the **Gender** dimension into the Filter category. Doing the same to the **Disease diagnosis date** and **Disease onset date** will also make them filters, and clicking **OK** shows that those dimensions are no longer visible on the Analysis View screen.

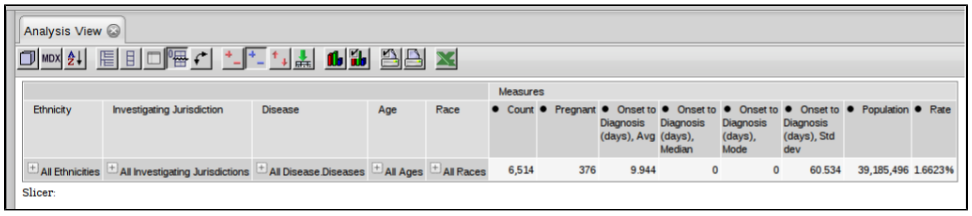


The **Morbidity Events** cube features many other measures, accessible from the **OLAP navigator**. Open the **OLAP navigator** again and click on **Measures** to open the list.

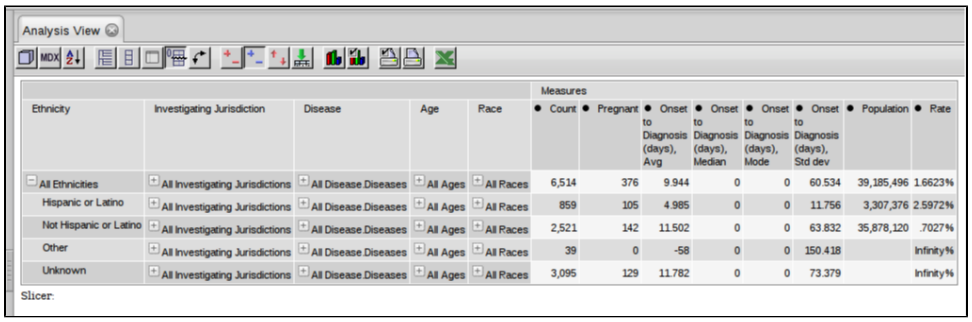
Copyright © 2009 Collaborative Software Initiative, Inc.



Select several measures including the **Population** and **Rate** measures at the bottom of the list, and press **OK**. Click **OK** again to close the **OLAP Navigator** and apply the changes.



Clicking on the **+** next ot **All Ethnicities** recalculates each of the measures for the newly-visible levels of that dimension.



Copyright © 2009 Collaborative Software Initiative, Inc.

Next Topic:

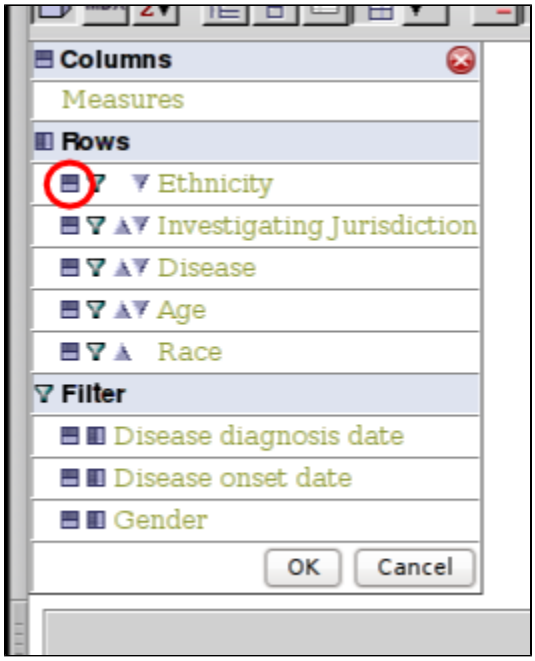
Adjusting Rows and Columns

Copyright © 2009 Collaborative Software Initiative, Inc.

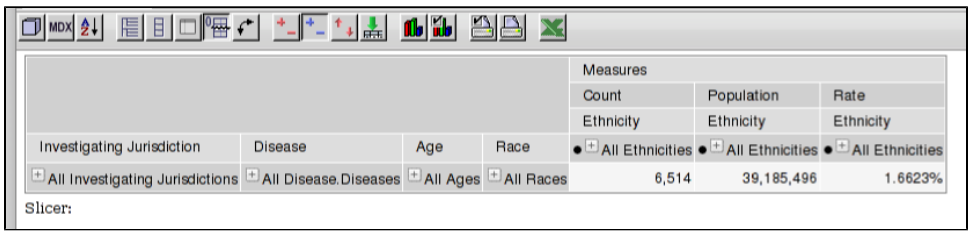
Adjusting Rows and Columns

Adjusting Rows and Columns

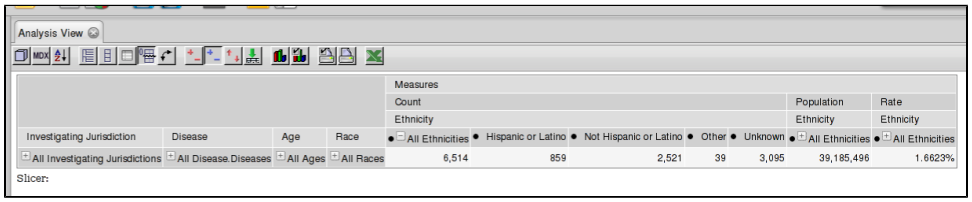
The screen is beginning to look cluttered. The **OLAP Navigator** allows users to move dimensions from rows to columns and back. Open the OLAP Navigator and click on the square icon next to the **Ethnicity** dimension.



This moves the **Ethnicity** up under "Columns". Click **OK**. Note that this next screen showing the result has had all the other measures manually removed for clarity. (Click om **Measures** and "uncheck" the checked dimension, and click **OK** to remove them.)

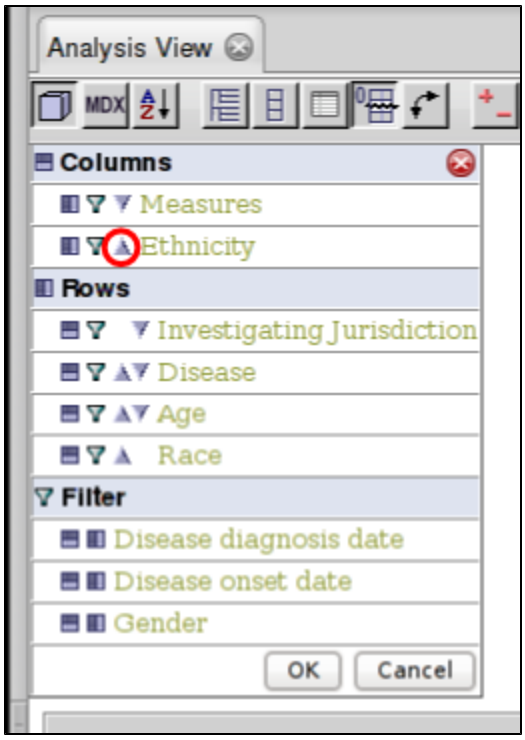


You can move dimensions between rows, columns, and filters as desired. Although measures can't be made to disappear as a filter (since the resulting report would show no data), they can also move between rows and columns. Note that in the image above, if you were to expand the **Ethnicity** dimension in the first column, it would only show the separate ethnicities' count measure, as shown here:

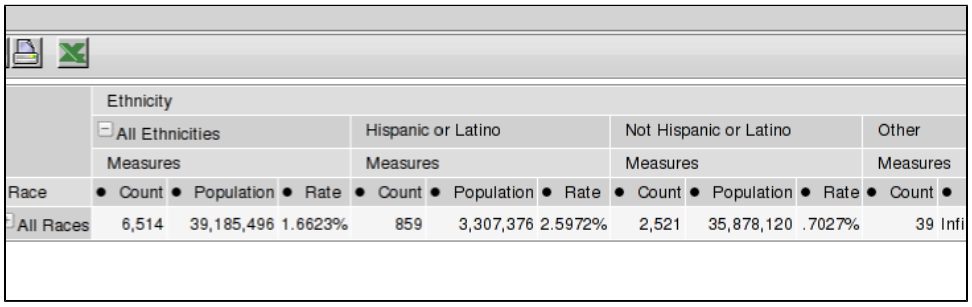


The **OLAP Navigator** allows reordering of the items along the row axis or column axis. Open the **OLAP Navigator** again, and click the up arrow next to **Ethnicity** to move it above **Measures**.

Copyright © 2009 Collaborative Software Initiative, Inc.



This tells the Analysis View to display all visible measures, those that have been selected, underneath each hierarchy, as shown here:



	Ethnicity										
	All Ethnicities			Hispanic or Latino			Not Hispanic or Latino			Other	
	Measures			Measures			Measures			Measures	
Race	Count	Population	Rate	Count	Population	Rate	Count	Population	Rate	Count	
All Races	6,514	39,185,496	1.6623%	859	3,307,376	2.5972%	2,521	35,878,120	.7027%	39	Infi

Next Topic:

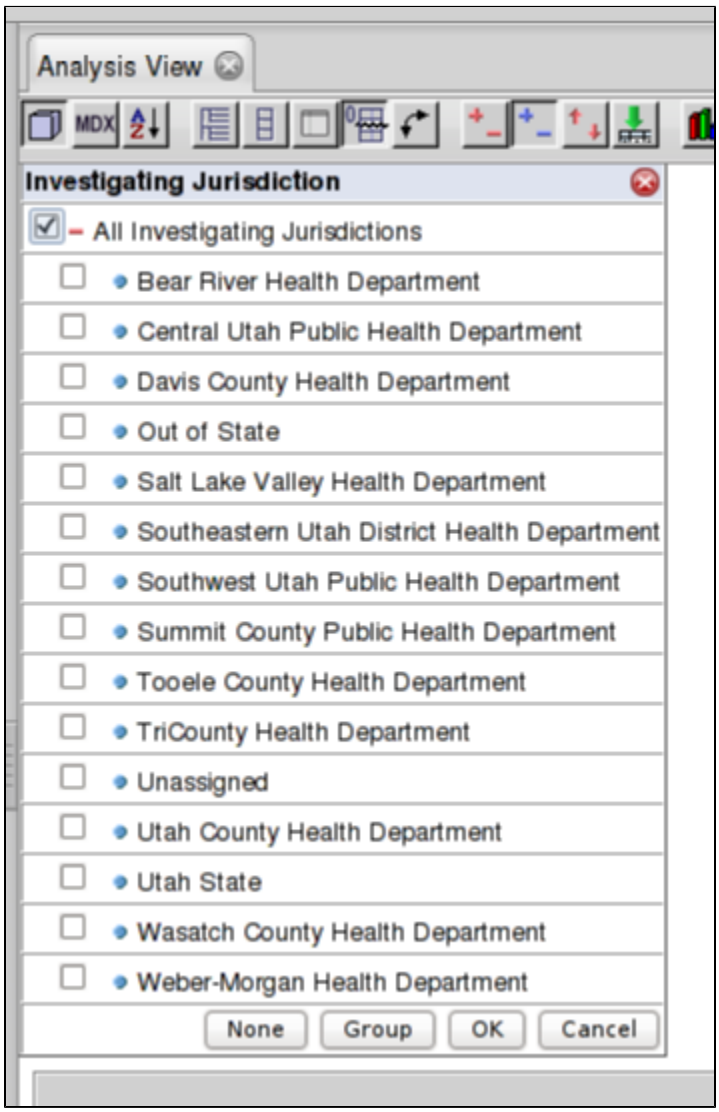
Filtering

Copyright © 2009 Collaborative Software Initiative, Inc.

Filtering

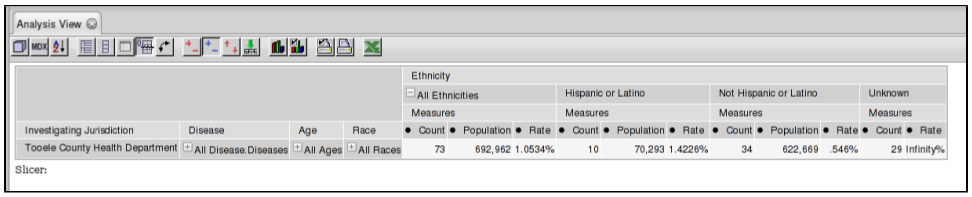
Filtering

The OLAP Navigator also allows filtering of the whole dataset. For instance, a user from the Tooele County Health Department might be interested only in results from his or her own jurisdiction. One way to see the data split as needed would be to expand the Jurisdiction dimension and look only at the row of interest, but the resulting screen is unnecessarily cluttered, and the OLAP tool must recalculate the value for each visible cell, whether it's interesting or not. A better way would be to tell the OLAP tool only to display data from the appropriate jurisdiction. To accomplish this, open the OLAP Navigator and click the **Jurisdiction** dimension. A list of all possible jurisdictions appears (if the resulting list contains only one line saying **All Investigating Jurisdictions**, click the red "**+**" sign next to it to expand the list).

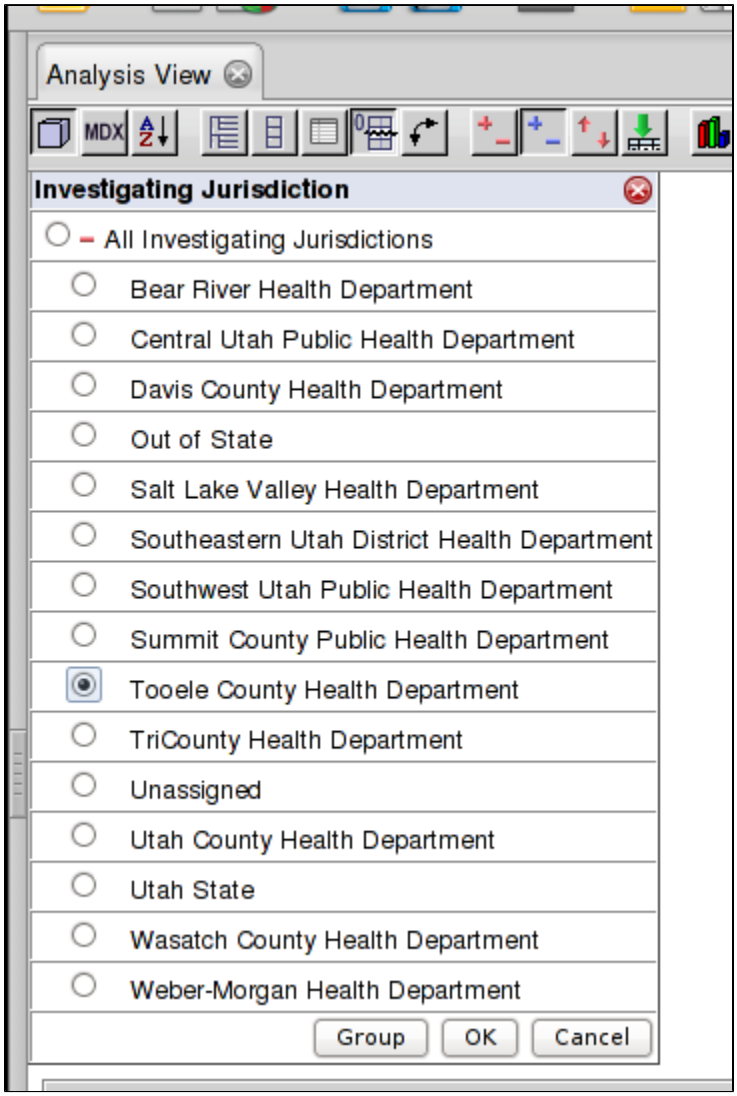


Uncheck **All Investigating Jurisdictions** and check **Tooele County Health Department**. Note that if more than one jurisdiction were of interest, multiple jurisdictions can be selected on this page. Click **OK** twice to save these changes and close the OLAP Navigator. Only the **Tooele County Health Department** jurisdiction is shown.

Copyright © 2009 Collaborative Software Initiative, Inc.

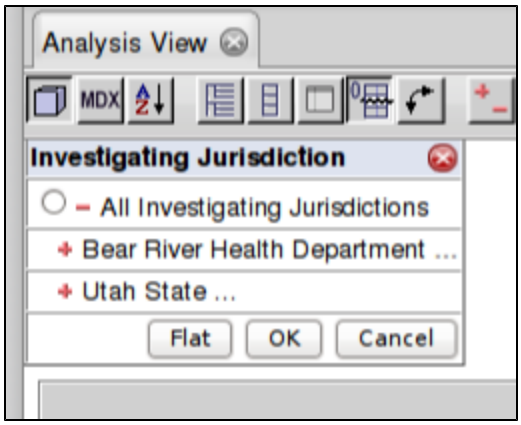


As mentioned above, it would have been possible to select multiple jurisdictions in the OLAP Navigator. In this case, only one jurisdiction is selected, and the screen would read more easily if the jurisdiction dimension weren't visible at all, provided some indication remained that only the Tooele County data was displayed. To achieve this, open the OLAP Navigator again, and move the Investigating Jurisdiction dimension to the set of filters. Then click on Investigating Jurisdiction, and select Tooele County Health Department again. Note that Pentaho™ allows only one selection now that Investigating Jurisdiction is a filter.



When trying to select a specific jurisdiction, you may see something like this:

Copyright © 2009 Collaborative Software Initiative, Inc.



Here, Pentaho™ has tried to group the dimension into multiple levels (see "Multi-Level Dimension Hierarchies", below). Hopefully Pentaho™ can be told somehow not to do this, but in the meantime, simply click "Flat" to indicate that these levels shouldn't be grouped. Then select **Tooele County Health Department**.

Close the OLAP Navigator. Notice that now the "Slicer" text at the bottom of the OLAP spreadsheet shows Investigating Jurisdiction is limited to **Tooele County Health Department**.

Next Topic:

Multi-Level Dimension Hierarchies

Copyright © 2009 Collaborative Software Initiative, Inc.

Multi-Level Dimension Hierarchies

Multi-Level Dimension Hierarchies

OLAP dimensions can be hierarchical, with multiple distinct levels, as used by the Age dimension, which has an **Age Group** level above the actual age. To see these levels, expand the Age dimension. The first level displayed is the **Age Group**. If you expand an age group, Pentaho™ will display the individual ages that make up that group.

			Measure
Disease	Age	Race	Count
All Disease.Diseases	All Ages	All Races	73
	1-4 years	All Races	3
	10-14 years	All Races	1
	15-19 years	All Races	17
	15	All Races	3
	16	All Races	3
	17	All Races	6
	18	All Races	4
	19	All Races	1
	20-24 years	All Races	13
	25-29 years	All Races	9

Note that Pentaho™ sorts the age groups non-intuitively. This also will be fixed eventually.

Next Topic:

Miscellaneous Toolbar Options

Copyright © 2009 Collaborative Software Initiative, Inc.

Miscellaneous Toolbar Options

Miscellaneous Toolbar Options

The Analysis View toolbar, shown below, features several other interesting options:

Icon	Icon Name	Description
	Open OLAP Navigator	The OLAP Navigator is used to define the overall layout of your query, including rows and columns to display, and what to filter on.
	Show MDX Editor	The OLAP application provides a graphical interface for an OLAP database. It creates a query, called an MDX query, which is sent to the database for processing. This button allows users who understand MDX to modify the query directly.
	Configure OLAP Table	Allows you to specify sort options, apply filters based on rank, and show or hide member properties
	Show Parent Members	Changes display to show parent members of a query in their row or column
	Hide Spans	Hides or displays repeating header information
	Show Properties	Toggles the display of member properties
	Suppress Empty Rows/Columns	This button, selected by default, determines whether dimensions containing no data will be displayed.
	Swap Axes	Swaps rows and columns
	Drill Members	All instances of that member will be expanded (multiple instances of a member would be displayed when you have nested dimensions on an axis).
	Drill Position	)nly the instance of the member clicked on will be expanded, even if there are other instances of the same member.
	Drill Replace	The member drilled on will be replaced by its children members
	Drill Through	This provides some ability to view the actual rows of the database.
	Show Chart	This will put a constantly-updated graph on the OLAP page. Careful dimension selection can make them quite informative. The neighboring button, picturing a graph with a check box, allows the user to configure graph options.

Copyright © 2009 Collaborative Software Initiative, Inc.

	Chart Config	Allows you to set chart properties, such as Chart Type, Title, fonts, legends, etc.
	Configure Print Settings	Alllows you to configure printing properties such as Report Title, Page Orientation, etc.
	Print this page via PDF	Print analysis as a PDF file
	Start Excel	This exports the currently visible data to a spreadsheet.

Next Topic:

CSI TriSano™ Data Warehouse Data Dictionary

Copyright © 2009 Collaborative Software Initiative, Inc.

CSI TriSano™ Data Warehouse Data Dictionary

This section contains detail information on the contents of the CSI TriSano™ Data Warehouse database.

- TriSano Data Dictionary
  - population.newpop
  - population.population
  - population.population_dimensions
  - population.population_tables
  - population.racecombined
  - trisano.current_schema_name
  - trisano.dw_contact_answers_view
  - trisano.dw_contact_clinicians_view
  - trisano.dw_contact_diagnostic_facilities_view
  - trisano.dw_contact_diseases_view
  - trisano.dw_contact_events_view
  - trisano.dw_contact_hospitals_view
  - trisano.dw_contact_jurisdictions_view
  - trisano.dw_contact_lab_results_view
  - trisano.dw_contact_patients_races_view
  - trisano.dw_contact_patients_view
  - trisano.dw_contact_questions_view
  - trisano.dw_contact_secondary_jurisdictions_view
  - trisano.dw_contact_treatments_events_view
  - trisano.dw_contact_treatments_view
  - trisano.dw_date_dimension_view
  - trisano.dw_encounter_answers_view
  - trisano.dw_encounter_patients_view
  - trisano.dw_encounter_questions_view
  - trisano.dw_encounters_lab_results_view
  - trisano.dw_encounters_treatments_events_view
  - trisano.dw_encounters_treatments_view
  - trisano.dw_encounters_view
  - trisano.dw_events_diagnostic_facilities_view
  - trisano.dw_events_hospitals_view
  - trisano.dw_events_reporters_view
  - trisano.dw_events_reporting_agencies_view
  - trisano.dw_events_treatments_view
  - trisano.dw_lab_results_view
  - trisano.dw_morbidity_answers_view
  - trisano.dw_morbidity_clinicians_view
  - trisano.dw_morbidity_diagnostic_facilities_view
  - trisano.dw_morbidity_diseases_view
  - trisano.dw_morbidity_events_view
  - trisano.dw_morbidity_hospitals_view
  - trisano.dw_morbidity_jurisdictions_view
  - trisano.dw_morbidity_lab_results_view
  - trisano.dw_morbidity_patients_races_view
  - trisano.dw_morbidity_patients_view
  - trisano.dw_morbidity_questions_view
  - trisano.dw_morbidity_reporters_view
  - trisano.dw_morbidity_reporting_agencies_view
  - trisano.dw_morbidity_secondary_jurisdictions_view
  - trisano.dw_morbidity_treatments_events_view
  - trisano.dw_morbidity_treatments_view
  - trisano.dw_patients_races_view
  - trisano.dw_place_answers_view
  - trisano.dw_place_events_view
  - trisano.dw_place_questions_view
  - trisano.dw_secondary_jurisdictions_view
  - trisano.etl_success

TriSano Data Dictionary

Copyright © 2009 Collaborative Software Initiative, Inc.

population.newpop

column	type	details	description
year	integer		
population	integer		
age	integer		
age_group	text		
sex	text		
county	text		
jurisdiction	text		

population.population

column	type	details	description
race	text		
jurisdiction	text		
year	integer		
population	integer		

population.population_dimensions

Describes each of the OLAP dimensions usable for population calculation

column	type	details	description
dim_name	text	NOT NULL	Names of each OLAP dimension for which population divisions are known
dim_cols	text[]		Column names to be searched for population counts. Each array element corresponds to a different level of the dimension hierarchy
mapping_func	text[]		Functions to map OLAP values to something the underlying population tables will understand. Each element correspond to a level of the dimension hierarchy

Constraints
PRIMARY KEY (dim_name)

population.population_tables

Records the tables in the population schema that record population information

column	type	details	description
table_name	text	NOT NULL	Name of the population table
table_rank	integer	NOT NULL	When multiple tables will provide the required information, this rank is used to find the desired table. Lowest ranks are selected first (e.g. the table with rank 1 is the best one to use, all else being equal

Copyright © 2009 Collaborative Software Initiative, Inc.

Constraints
PRIMARY KEY (table_name)
UNIQUE (table_rank)

population.racecombined

column	type	details	description
year	integer		
age_group	text		
population	integer		
sex	text		
race	text		
ethnicity	text		
county	text		
jurisdiction	text		

trisano.current_schema_name

Internal – used so ETL process knows which schema can be dumped and reloaded

column	type	details	description
schemaname	text	NOT NULL	Contains the name of the schema currently in use by the warehouse, and referenced by all the views

trisano.dw_contact_answers_view

Formbuilder answers applicable to contact events. See transactional database, answers table for more information

column	type	details	description
id	integer		
event_id	integer		
question_id	integer		
text_answer	character varying		
export_conversion_value_id	integer		
code	character varying(255)		

trisano.dw_contact_clinicians_view

Clinician information applicable to contact events

column	type	details	description
id	integer		Primary key
dw_contact_events_id	integer		The contact event ID this answer refers to

Copyright © 2009 Collaborative Software Initiative, Inc.

first_name	character varying(25)		
last_name	character varying(25)		
middle_name	character varying(25)		

trisano.dw_contact_diagnostic_facilities_view

Diagnostic facility information applicable to contact events

column	type	details	description
id	bigint		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		The contact event ID this answer refers to
name	character varying(255)		Name of the diagnostic facility
place_type	character varying(100)		Type of facility
place_id	integer		Places table record related to this record

trisano.dw_contact_diseases_view

Diseases related to contact events. See diseases table for more information

column	type	details	description
id	integer		Primary key
disease_name	character varying(100)		
contact_lead_in	text		
place_lead_in	text		
treatment_lead_in	text		
active	boolean		
cdc_code	character varying(255)		

trisano.dw_contact_events_view

Table of contact events. Most fields match the events table

column	type	details	description
id	integer		Primary key
parent_id	integer		
dw_patients_id	integer		ID of patient record related to this event
disease_id	integer		
investigating_jurisdiction	character varying(255)		Name of the investigating jurisdiction
investigating_jurisdiction_id	integer		Places table record for this investigating jurisdiction
jurisdiction_of_residence	character varying(255)		Name of jurisdiction of residence

Copyright © 2009 Collaborative Software Initiative, Inc.

jurisdiction_of_residence_id	integer		Places table record for this jurisdiction of residence
actual_age_at_onset	integer		
actual_age_type	character varying(100)		
age_in_years	numeric		Calculated age in years, based on actual_age_at_onset and actual_age_type
estimated_age_at_onset	integer		
estimated_age_type	character varying(100)		
food_handler	character varying(100)		
healthcare_worker	character varying(100)		
group_living	character varying(100)		
day_care_association	character varying(100)		
pregnant	character varying(100)		
pregnancy_due_date	date		
additional_risk_factors	character varying(255)		
risk_factor_details	text		
occupation	character varying(255)		
other_data_1	character varying(255)		
other_data_2	character varying(255)		
disease_event_hospitalized	character varying(100)		
investigator	text		Name of investigator
event_queue_id	integer		
street_number	character varying(10)		
street_name	character varying(50)		
unit_number	character varying(10)		
city	character varying(255)		
county	character varying(100)		
state	character varying(100)		
postal_code	character varying(10)		

Copyright © 2009 Collaborative Software Initiative, Inc.

latitude	numeric(9,6)		
longitude	numeric(9,6)		
date_disease_onset	date		
date_disease_diagnosed	date		
date_entered_into_system	date		
date_investigation_started	date		
date_investigation_completed	date		
review_completed_by_state_date	date		
date_created	timestamp without time zone		
date_updated	timestamp without time zone		
date_deleted	timestamp without time zone		
disposition	character varying(100)		
contact_type	character varying(100)		
imported_from_code	character varying(100)		
sent_to_ibis	boolean		
ibis_updated_at	date		
disease_event_died	character varying(100)		

trisano.dw_contact_hospitals_view

Information about hospital admissions related to contact events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
hospital_name	character varying(255)		
admission_date	date		
discharge_date	date		
medical_record_number	character varying(255)		
hospital_record_number	character varying(100)		

trisano.dw_contact_jurisdictions_view

Jurisdictions related to contact events

column	type	details	description

Copyright © 2009 Collaborative Software Initiative, Inc.

id	integer		Primary key
entity_id	integer		
created_at	timestamp without time zone		
updated_at	timestamp without time zone		
name	character varying(255)		
short_name	character varying(255)		

trisano.dw_contact_lab_results_view

Lab results for contact events. See lab results view for more information

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
dw_encounter_events_id	integer		
name	character varying(255)		
lab_type	character varying(100)		
test_type	character varying(255)		
test_detail	character varying(255)		
lab_result_text	character varying(255)		
reference_range	character varying(255)		
interpretation	character varying(100)		
specimen_source	character varying(100)		
collection_date	date		
lab_test_date	date		
specimen_sent_to_uphl	character varying(100)		

trisano.dw_contact_patients_races_view

Races of patients involved in contact events. See patients_races table for more information

column	type	details	description
id	bigint		Primary key
race	character varying(100)		
person_id	integer		

trisano.dw_contact_patients_view

Patients involved in contact events. See people table for more information

column	type	details	description

Copyright © 2009 Collaborative Software Initiative, Inc.

id	integer		Primary key
entity_id	integer		
birth_gender	character varying(100)		
ethnicity	character varying(100)		
primary_language	character varying(100)		
first_name	character varying(25)		
middle_name	character varying(25)		
last_name	character varying(25)		
birth_date	date		
date_of_death	date		
first_name_soundex	character varying(255)		
last_name_soundex	character varying(255)		

trisano.dw_contact_questions_view

Questions on forms related to contact events. See questions table for more information

column	type	details	description
id	integer		Primary key
form_element_id	integer		
question_text	character varying		
help_text	character varying		
data_type	character varying(50)		
size	integer		
is_required	boolean		
created_at	timestamp without time zone		
updated_at	timestamp without time zone		
core_data	boolean		
core_data_attr	character varying(255)		
short_name	character varying(255)		
style	character varying(255)		

trisano.dw_contact_secondary_jurisdictions_view

Secondary jurisdictions of contact events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
jurisdiction_id	integer		Places record related to this jurisdiction

Copyright © 2009 Collaborative Software Initiative, Inc.

name	character varying(255)		Name of the jurisdiction

trisano.dw_contact_treatments_events_view

Treatment events related to contact events. See treatment_events table for more information

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
dw_encounter_events_id	integer		
treatment_id	integer		
treatment_given	character varying(100)		
date_of_treatment	date		
treatment_notes	character varying(255)		

trisano.dw_contact_treatments_view

Treatments related to contact events. See treatments table for more information

column	type	details	description
id	integer		Primary key
treatment_type_id	integer		
treatment_name	character varying(100)		

trisano.dw_date_dimension_view

Dates used in various records. This provides functionality for the date-related dimensions in the OLAP system

column	type	details	description
fulldate	date		
day_of_week	integer		
day_of_month	integer		
day_of_year	integer		
month	integer		
quarter	integer		
week_of_year	integer		
year	integer		

trisano.dw_encounter_answers_view

Formbuilder answers provided in relation to encounter events. See answers table for more information

column	type	details	description

Copyright © 2009 Collaborative Software Initiative, Inc.

id	integer		Primary key
event_id	integer		
question_id	integer		
text_answer	character varying		
export_conversion_value_id	integer		
code	character varying(255)		

trisano.dw_encounter_patients_view

Patients involved in encounter events. See people table for more information

column	type	details	description
id	integer		Primary key
entity_id	integer		
birth_gender	character varying(100)		
ethnicity	character varying(100)		
primary_language	character varying(100)		
first_name	character varying(25)		
middle_name	character varying(25)		
last_name	character varying(25)		
birth_date	date		
date_of_death	date		
first_name_soundex	character varying(255)		
last_name_soundex	character varying(255)		

trisano.dw_encounter_questions_view

Questions asked on forms for encounter events. See questions table for more information

column	type	details	description
id	integer		Primary key
form_element_id	integer		
question_text	character varying		
help_text	character varying		
data_type	character varying(50)		
size	integer		
is_required	boolean		
created_at	timestamp without time zone		
updated_at	timestamp without time zone		
core_data	boolean		
core_data_attr	character varying(255)		

Copyright © 2009 Collaborative Software Initiative, Inc.

short_name	character varying(255)		
style	character varying(255)		

trisano.dw_encounters_lab_results_view

Lab results for encounter events. See lab_results table for more information

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
dw_encounter_events_id	integer		
name	character varying(255)		
lab_type	character varying(100)		
test_type	character varying(255)		
test_detail	character varying(255)		
lab_result_text	character varying(255)		
reference_range	character varying(255)		
interpretation	character varying(100)		
specimen_source	character varying(100)		
collection_date	date		
lab_test_date	date		
specimen_sent_to_uphl	character varying(100)		

trisano.dw_encounters_treatments_events_view

Treatment events for encounter events. See treatment_events table for more information

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
dw_encounter_events_id	integer		
treatment_id	integer		
treatment_given	character varying(100)		
date_of_treatment	date		
treatment_notes	character varying(255)		

trisano.dw_encounters_treatments_view

Treatments for encounter events. See treatments table for more information

Copyright © 2009 Collaborative Software Initiative, Inc.

column	type	details	description
id	integer		Primary key
dw_encounters_id	integer		
dw_events_treatments_id	integer		

trisano.dw_encounters_view

Encounter events. See events table for more information, but note that many fields have been removed because they do not relate to encounter events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		The morbidity event parent of this event
encounter_event_id	integer		
investigator_id	text		Users table record containing the investigator for this event
encounter_date	date		
location	character varying(40)		
description	text		

trisano.dw_events_diagnostic_facilities_view

Diagnostic facilities for various event types

column	type	details	description
id	bigint		Primary key
dw_morbidity_events_id	integer		Morbidity event ID, if any, related to this diagnostic facility
dw_contact_events_id	integer		Contact event ID, if any, related to this diagnostic facility
name	character varying(255)		Name of the diagnostic facility
place_type	character varying(100)		Type of facility
place_id	integer		Place record for this facility

trisano.dw_events_hospitals_view

Hospitalizations related to various events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		Morbidity event ID, if any, related to this diagnostic facility
dw_contact_events_id	integer		Contact event ID, if any, related to this diagnostic facility
hospital_name	character varying(255)		
admission_date	date		
discharge_date	date		
medical_record_number	character varying(255)		

Copyright © 2009 Collaborative Software Initiative, Inc.

hospital_record_number	character varying(100)		

trisano.dw_events_reporters_view

Reporters for various event types

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		Morbidity event ID, if any, related to this diagnostic facility
dw_contact_events_id	integer		Contact event ID, if any, related to this diagnostic facility
first_name	character varying(25)		
last_name	character varying(25)		
middle_name	character varying(25)		

trisano.dw_events_reporting_agencies_view

Reporting agencies for various event types

column	type	details	description
id	bigint		Primary key
dw_morbidity_events_id	integer		Morbidity event ID, if any, related to this diagnostic facility
dw_contact_events_id	integer		Contact event ID, if any, related to this diagnostic facility
name	character varying(255)		
place_type	character varying(100)		
place_id	integer		

trisano.dw_events_treatments_view

Treatments events related to morbidity, encounter, or contact events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		Morbidity event ID, if any, related to this diagnostic facility
dw_contact_events_id	integer		Contact event ID, if any, related to this diagnostic facility
dw_encounter_events_id	integer		Encounter event ID, if any, related to this diagnostic facility
treatment_id	integer		
treatment_given	character varying(100)		
date_of_treatment	date		
treatment_notes	character varying(255)		

Copyright © 2009 Collaborative Software Initiative, Inc.

trisano.dw_lab_results_view

Lab result records for various event types. See lab results table for more information

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		Morbidity event ID, if any, related to this diagnostic facility
dw_contact_events_id	integer		Contact event ID, if any, related to this diagnostic facility
dw_encounter_events_id	integer		Encounter event ID, if any, related to this diagnostic facility
name	character varying(255)		
lab_type	character varying(100)		
test_type	character varying(255)		
test_detail	character varying(255)		
lab_result_text	character varying(255)		
reference_range	character varying(255)		
interpretation	character varying(100)		
specimen_source	character varying(100)		
collection_date	date		
lab_test_date	date		
specimen_sent_to_uphl	character varying(100)		

trisano.dw_morbidity_answers_view

Formbuild answers related to morbidity events. See answers table for more information

column	type	details	description
id	integer		Primary key
event_id	integer		
question_id	integer		
text_answer	character varying		
export_conversion_value_id	integer		
code	character varying(255)		

trisano.dw_morbidity_clinicians_view

Clinicians related to morbidity events

Copyright © 2009 Collaborative Software Initiative, Inc.

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
first_name	character varying(25)		
last_name	character varying(25)		
middle_name	character varying(25)		

trisano.dw_morbidity_diagnostic_facilities_view

Diagnostic facilities asscoiated with morbidity events

column	type	details	description
id	bigint		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
name	character varying(255)		
place_type	character varying(100)		
place_id	integer		

trisano.dw_morbidity_diseases_view

Diseases related to morbidity events. See diseases table for more information

column	type	details	description
id	integer		Primary key
disease_name	character varying(100)		
contact_lead_in	text		
place_lead_in	text		
treatment_lead_in	text		
active	boolean		
cdc_code	character varying(255)		

trisano.dw_morbidity_events_view

Morbidity event records. See events table for more information

column	type	details	description
id	integer		Primary key
parent_id	integer		
dw_patients_id	integer		Patient record for this event
disease_id	integer		
investigating_jurisdiction	character varying(255)		

Copyright © 2009 Collaborative Software Initiative, Inc.

investigating_jurisdiction_id	integer		
jurisdiction_of_residence	character varying(255)		
jurisdiction_of_residence_id	integer		
state_case_status_code	character varying(100)		
lhd_case_status_code	character varying(100)		
mmwr_week	integer		
mmwr_year	integer		
event_name	character varying(100)		
record_number	character varying(20)		
actual_age_at_onset	integer		
actual_age_type	character varying(100)		
age_in_years	numeric		
estimated_age_at_onset	integer		
estimated_age_type	character varying(100)		
parent_guardian	character varying(255)		
food_handler	character varying(100)		
healthcare_worker	character varying(100)		
group_living	character varying(100)		
day_care_association	character varying(100)		
pregnant	character varying(100)		
pregnancy_due_date	date		
additional_risk_factors	character varying(255)		
risk_factor_details	text		
occupation	character varying(255)		
other_data_1	character varying(255)		
other_data_2	character varying(255)		
disease_event_hospitalized	character varying(100)		
outbreak_associated_code	character varying(100)		
outbreak_name	character varying(255)		
investigator	text		
event_queue_id	integer		
acuity	character varying(255)		
street_number	character varying(10)		
street_name	character varying(50)		
unit_number	character varying(10)		
city	character varying(255)		
county	character varying(100)		
state	character varying(100)		

Copyright © 2009 Collaborative Software Initiative, Inc.

postal_code	character varying(10)		
latitude	numeric(9,6)		
longitude	numeric(9,6)		
date_disease_onset	date		
date_disease_diagnosed	date		
results_reported_to_clinician_date	date		
date_reported_to_public_health	date		
date_entered_into_system	date		
date_investigation_started	date		
date_investigation_completed	date		
review_completed_by_state_date	date		
date_created	timestamp without time zone		
date_updated	timestamp without time zone		
date_deleted	timestamp without time zone		
sent_to_cdc	boolean		
disposition_if_once_a_contact	character varying(100)		
contact_type_if_once_a_contact	character varying(100)		
imported_from_code	character varying(100)		
sent_to_ibis	boolean		
ibis_updated_at	date		
disease_event_died	character varying(100)		

trisano.dw_morbidity_hospitals_view

Hospitalizations related to morbidity events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
hospital_name	character varying(255)		
admission_date	date		
discharge_date	date		
medical_record_number	character varying(255)		
hospital_record_number	character varying(100)		

trisano.dw_morbidity_jurisdictions_view

Jurisdictions of morbidity events

column	type	details	description

Copyright © 2009 Collaborative Software Initiative, Inc.

id	integer		Primary key
entity_id	integer		
created_at	timestamp without time zone		
updated_at	timestamp without time zone		
name	character varying(255)		
short_name	character varying(255)		

trisano.dw_morbidity_lab_results_view

Lab results for morbidity events. See lab_results table for more infomrmation

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
dw_encounter_events_id	integer		
name	character varying(255)		
lab_type	character varying(100)		
test_type	character varying(255)		
test_detail	character varying(255)		
lab_result_text	character varying(255)		
reference_range	character varying(255)		
interpretation	character varying(100)		
specimen_source	character varying(100)		
collection_date	date		
lab_test_date	date		
specimen_sent_to_uphl	character varying(100)		

trisano.dw_morbidity_patients_races_view

Races of patients in morbidity events

column	type	details	description
id	bigint		
race	character varying(100)		
person_id	integer		

trisano.dw_morbidity_patients_view

Patients associated with morbidity events. See people table for more information

column	type	details	description

Copyright © 2009 Collaborative Software Initiative, Inc.

id	integer		Primary key
entity_id	integer		
birth_gender	character varying(100)		
ethnicity	character varying(100)		
primary_language	character varying(100)		
first_name	character varying(25)		
middle_name	character varying(25)		
last_name	character varying(25)		
birth_date	date		
date_of_death	date		
first_name_soundex	character varying(255)		
last_name_soundex	character varying(255)		

trisano.dw_morbidity_questions_view

Formbuilder questions related to morbidity events. See questions table for more information

column	type	details	description
id	integer		Primary key
form_element_id	integer		
question_text	character varying		
help_text	character varying		
data_type	character varying(50)		
size	integer		
is_required	boolean		
created_at	timestamp without time zone		
updated_at	timestamp without time zone		
core_data	boolean		
core_data_attr	character varying(255)		
short_name	character varying(255)		
style	character varying(255)		

trisano.dw_morbidity_reporters_view

Reporter records for morbidity events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
first_name	character varying(25)		

Copyright © 2009 Collaborative Software Initiative, Inc.

last_name	character varying(25)		
middle_name	character varying(25)		

trisano.dw_morbidity_reporting_agencies_view

Reporting agencies for morbidity events

column	type	details	description
id	bigint		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
name	character varying(255)		
place_type	character varying(100)		
place_id	integer		

trisano.dw_morbidity_secondary_jurisdictions_view

Secondary jurisdictions for morbidity events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
jurisdiction_id	integer		
name	character varying(255)		

trisano.dw_morbidity_treatments_events_view

Treatment events for morbidity events

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
dw_encounter_events_id	integer		
treatment_id	integer		
treatment_given	character varying(100)		
date_of_treatment	date		
treatment_notes	character varying(255)		

trisano.dw_morbidity_treatments_view

Treatments related to morbidity events

Copyright © 2009 Collaborative Software Initiative, Inc.

column	type	details	description
id	integer		Primary key
treatment_type_id	integer		
treatment_name	character varying(100)		

trisano.dw_patients_races_view

Races of patients

column	type	details	description
id	bigint		Primary key
race	character varying(100)		
person_id	integer		

trisano.dw_place_answers_view

Answers to formbuilder questions related to place events. See answers table for more information

column	type	details	description
id	integer		Primary key
event_id	integer		
question_id	integer		
text_answer	character varying		
export_conversion_value_id	integer		
code	character varying(255)		

trisano.dw_place_events_view

Place events. See events table for more information, keeping in mind that many fields have been removed

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		Morbidity event record that is the parent for this record
name	character varying(255)		
place_type	character varying(100)		
street_number	character varying(10)		
street_name	character varying(50)		
unit_number	character varying(10)		
city	character varying(255)		
state	character varying(100)		
county	character varying(100)		
postal_code	character varying(10)		

Copyright © 2009 Collaborative Software Initiative, Inc.

latitude	numeric(9,6)		
longitude	numeric(9,6)		

trisano.dw_place_questions_view

Formbuilder questions related to place events. See questions table for more information

column	type	details	description
id	integer		Primary key
form_element_id	integer		
question_text	character varying		
help_text	character varying		
data_type	character varying(50)		
size	integer		
is_required	boolean		
created_at	timestamp without time zone		
updated_at	timestamp without time zone		
core_data	boolean		
core_data_attr	character varying(255)		
short_name	character varying(255)		
style	character varying(255)		

trisano.dw_secondary_jurisdictions_view

Secondary jurisdictions for various event typesj

column	type	details	description
id	integer		Primary key
dw_morbidity_events_id	integer		
dw_contact_events_id	integer		
jurisdiction_id	integer		
name	character varying(255)		

trisano.etl_success

Internal - records results of ETL processes

column	type	details	description
success	boolean		
entrydate	timestamp with time zone	DEFAULT now()	

Copyright © 2009 Collaborative Software Initiative, Inc.