

Applenti - Report Designer

In this documentation we have provided the steps to follow to set up the Report designer

Our Report Designer offers a complete framework for producing daily reports from any database or any No SQL source.

The product focuses on easy installation and report design: Once setup, reports can be built and published in a minute.

Main Features

- **Dynamic SQL sources:** Use either your SQL or let the Seal engine build dynamically the SQL used to query your database.
- **LINQ queries:** Join and query any Data Sources (SQL, Excel, XML, Olap Cube, HTTP JSON, etc.) with the power of LINQ queries.
- **Native Pivot Tables:** Simply drag and drop elements directly in a pivot table (Cross Tab) and display them in your report.
- **HTML 5 Charts:** Define and display Chart Series in two mouse clicks (Support of ChartJS, NVD3 and Plotly libraries).
- **Fully Responsive and HTML Rendering using Razor engine:** Use the power of HTML5 in the report result (Bootstrap layout, Responsiveness, Tables sorting and filtering). Customize your report presentation in HTML with the Razor engine parsing.
- **Web Report Server:** Publish and edit your reports on the web (Support of Windows and Linux OS with .NET Core).
- **Drill Down navigation and Sub Reports:** Navigate in your report result to drill to a detail or to execute another report.
- **Low TCO (total cost of ownership):** The product is designed for minimal ongoing maintenance

Main Entities: Repository and Reports

The **Seal Repository** stores all the entities managed by Seal Report: Data Sources, Devices, and Reports.

In addition, the repository contains extra folders for the Settings (configuration and dictionary for translations), the Views (used to render report result), and the Security (configuration and providers).

By default, the repository root folder is located in "*C:\ProgramData\Seal Report Repository*".

A report is composed of Data Sources, Models, and Views. The report may also reference Views Templates located in the repository.

- **Data Sources** contain the descriptions of database connections, tables, joins and columns.
- **Models** define how to generate the Result Set (Data Table) and Series from a single SQL Statement.
- The **Views** are used to generate a HTML document from the **Models** using Razor parsing and rendering.
- **Tasks** may be defined to perform SQL or Script tasks.

Product Components

Seal Report is composed of the **Server Manager** application, the **Report Designer** application, the **Web Report Server**, the **Task Scheduler** and the **Seal Report Scheduler**.

- The **Server Manager** (a Windows application) edits repository Data Sources, Devices and is used to perform administrative tasks (checking sources and reports, publishing web site, configuring the server, editing the security, etc.).
- The **Report Designer** (a Windows application) creates, edits and executes reports.

Repository Folders

Several sub-folders are located from the Repository Root folder. These folders contain all files necessary to run Seal Report.

- **Databases:** dedicated folder for local database files (e.g. an MS Access file, an Excel file).
- **Reports:** the reports (*.srex), files and sub-folders published by the Web Report Server. The Reports root folder is the parent of all folders, files and reports published.
- **Settings:** the Server configuration (Configuration.xml) and the translation files (*.csv).
- **Sources:** the data source files (*.scfx) available for the reports. These data sources are shared amongst all the reports.
- **Sources\TableTemplates:** table templates (*.cshtml) referenced by No SQL tables defined in LINQ Data Sources.
- **SubReports:** the sub-reports (*.srex) referenced by an element in a data source. Sub-Reports allow master-detail navigation from a report result.
- **Views:** the view templates (*.cshtml) used by the report views with their configuration and their related JavaScript, CSS and image files.

Setup

Run the executable file (provided under releases in the Github link-

https://github.com/nehanandank/applenti_sealReport)

The setup will install the Report Designer, the Server Manager with a default repository.

All executables, libraries and configuration files are installed by default in C:\Program Files\Seal Report.

All repository files are installed by default in C:\ProgramData\Seal Report Repository.

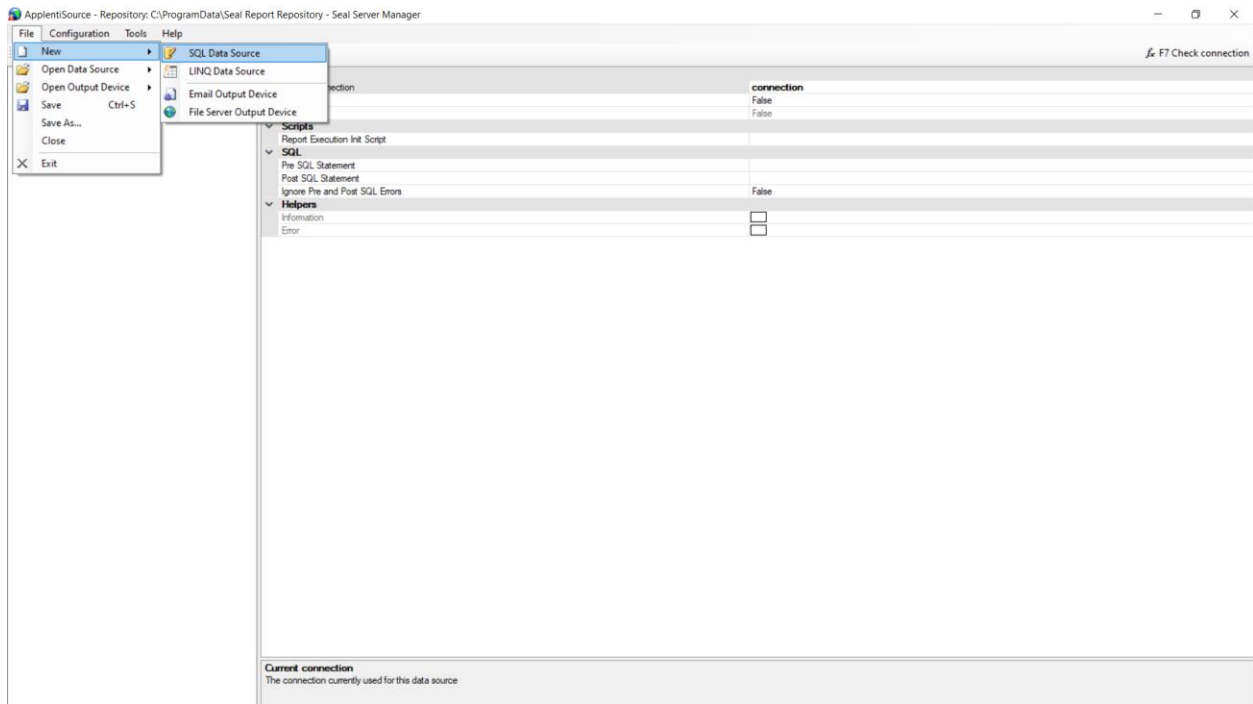
Steps to follow to make a report

Step 1 – Adding data source

This can be done either in Report designer or Server manager.

In server manager the following steps are to be followed

File -> New -> SQL Data source



We get the following page after we create our data source.

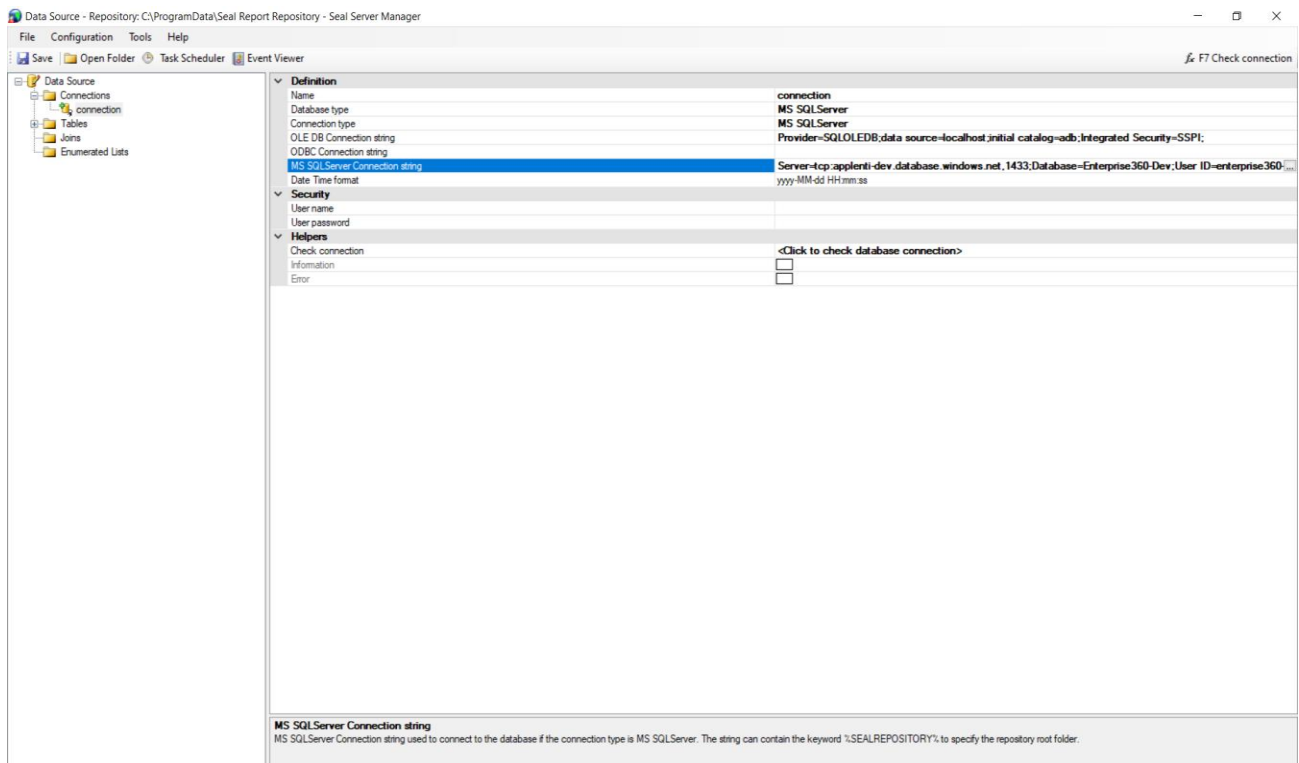
After a Data source is created, we can add connections to the Database, Add tables, Specify the Joins that we may require and Different types of Enumerated Lists that we can use.

To add a connection, Under the Data source that we created, in Connections -> connection we can set the following values and these values depend on the type of database server i.e. whether it is MS SQL Server or OLE DB etc.

Database type -> 'MS SQL Server'

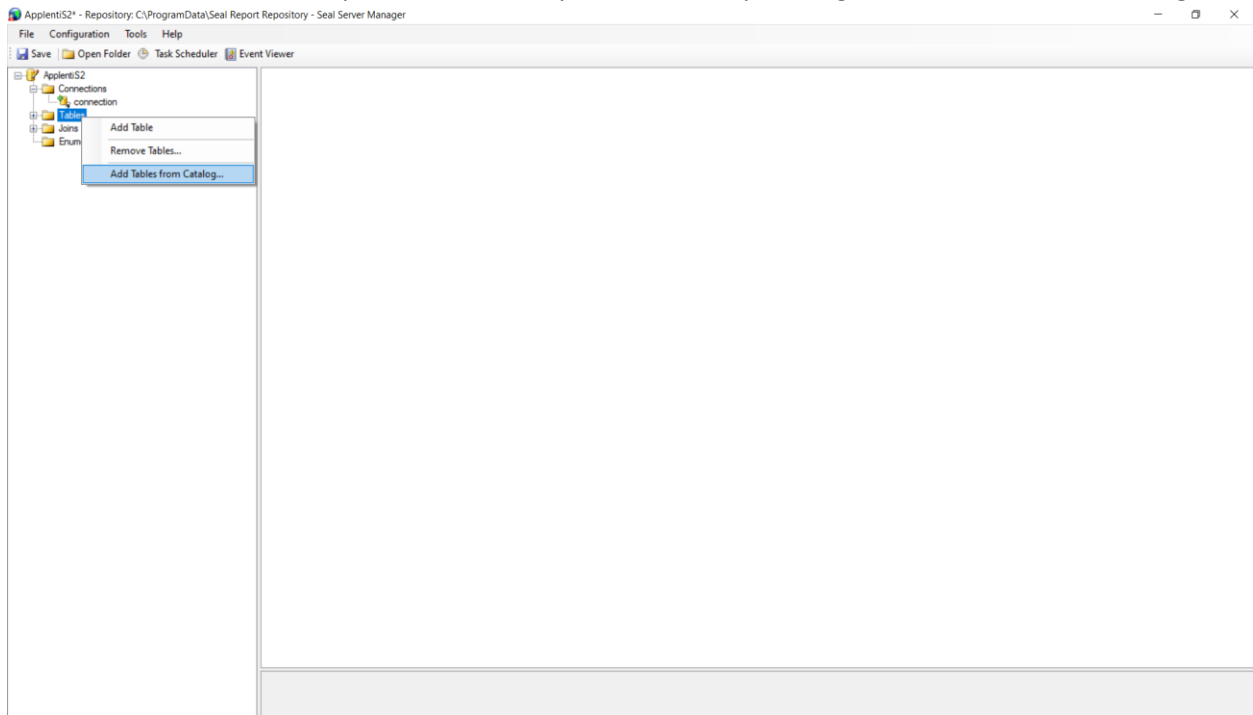
Connection type -> 'MS SQL Server'

MS SQLServer connection string -> Server=tcp:applenti-dev.database.windows.net,1433;Database=Enterprise360-Dev;User ID=enterprise360-dev;Password=3MBzrk!tk#ymnxkFXeBNf2pZY%J8E!FQMkVyS7u8GrqevBsxattEvaqtbxAESz^R;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;

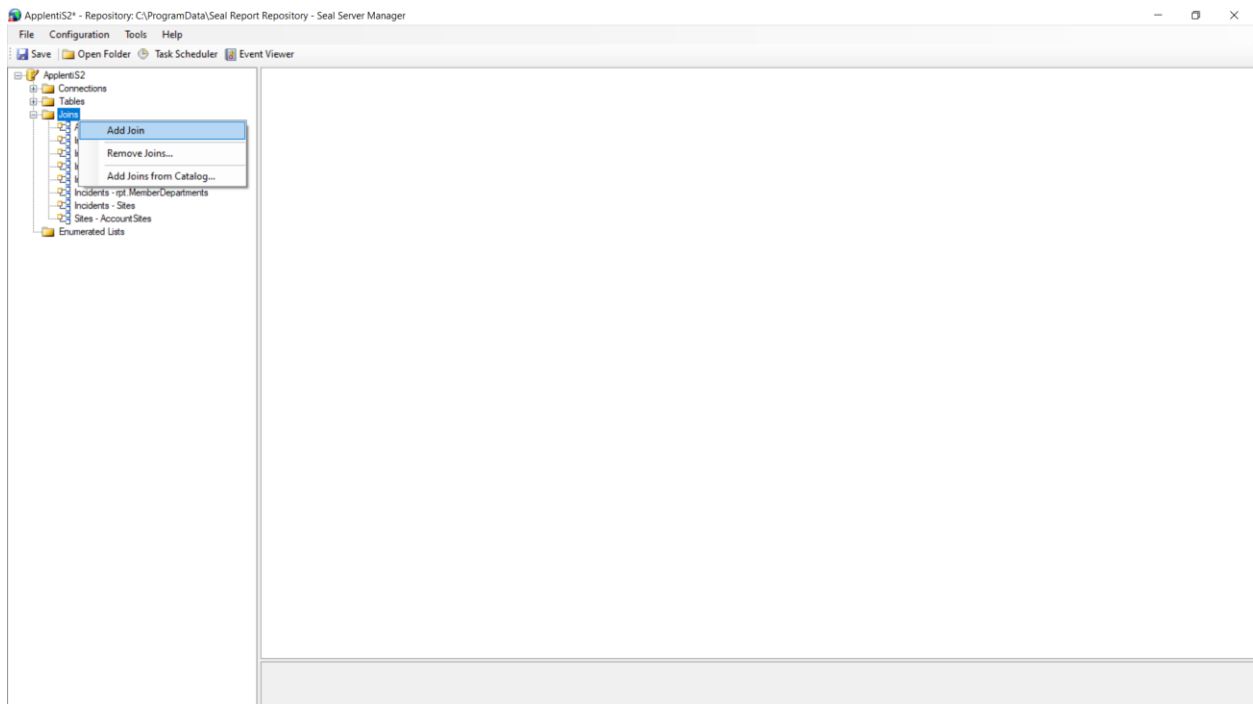


We can check the connection by clicking on "Check connection" present below.

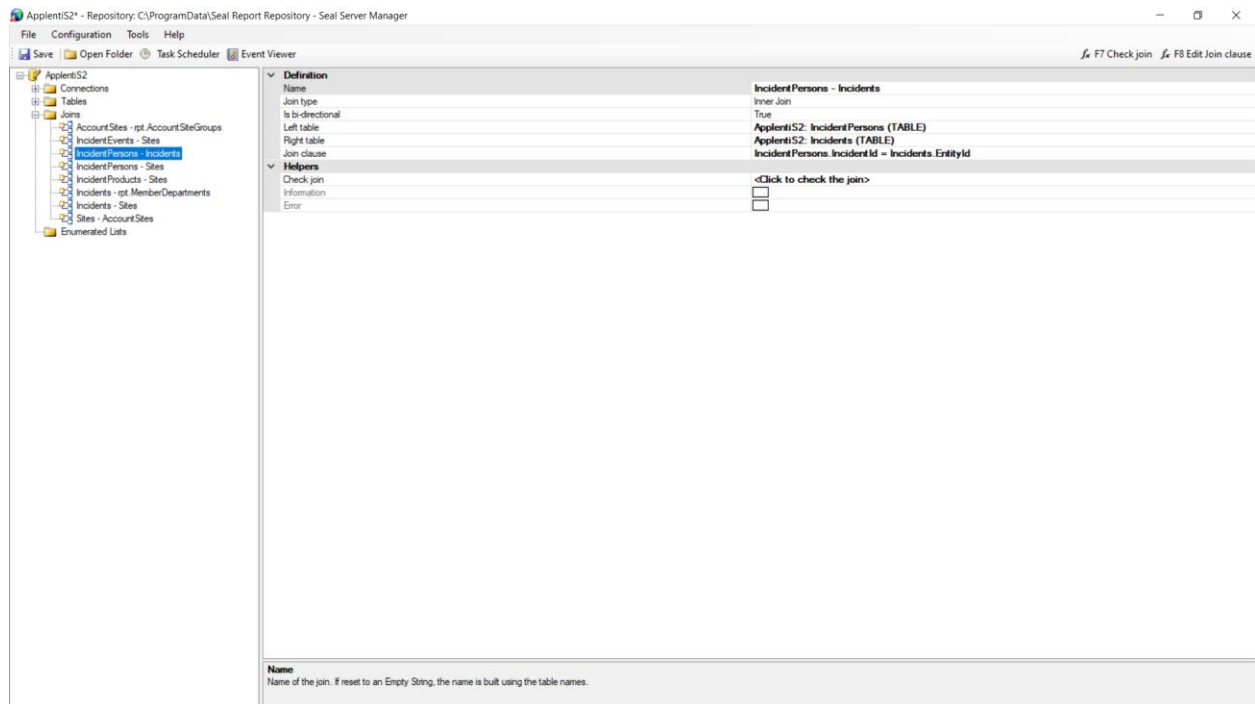
We can add new table or import all the tables present in DB by clicking on 'Add Tables from catalog'



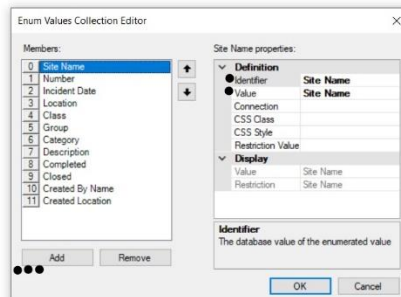
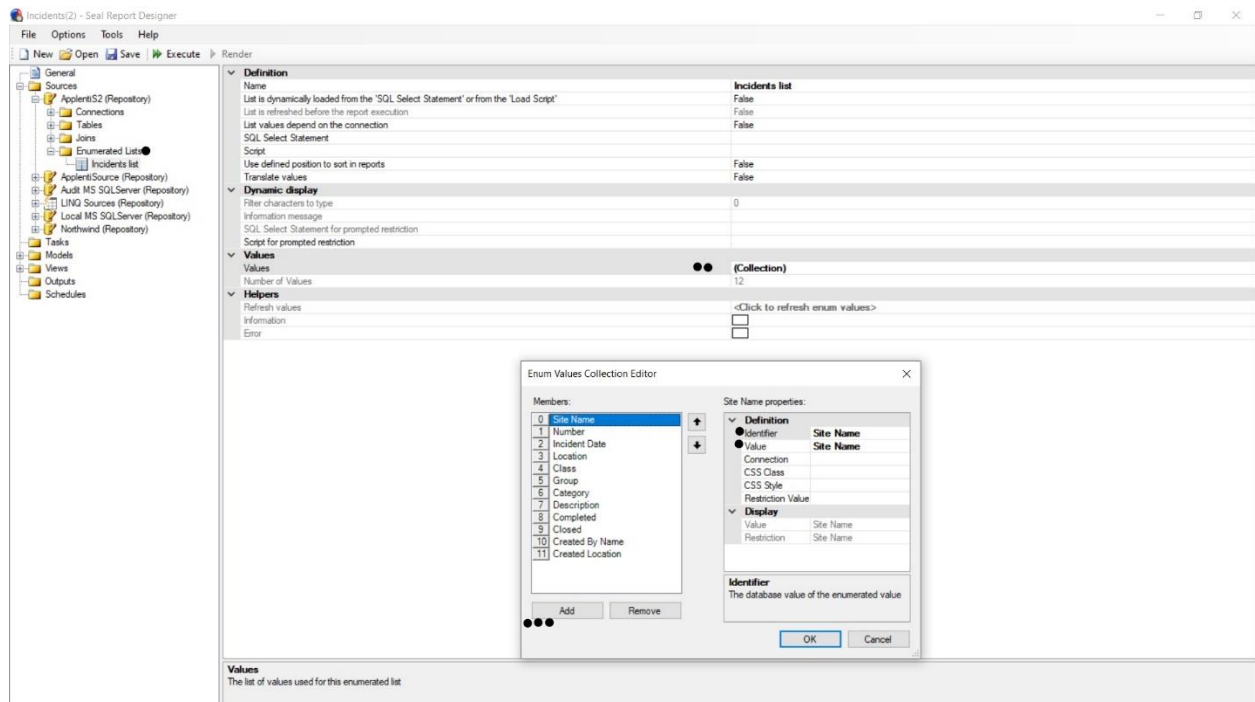
We can also add join or import existing joins by right clicking on Joins



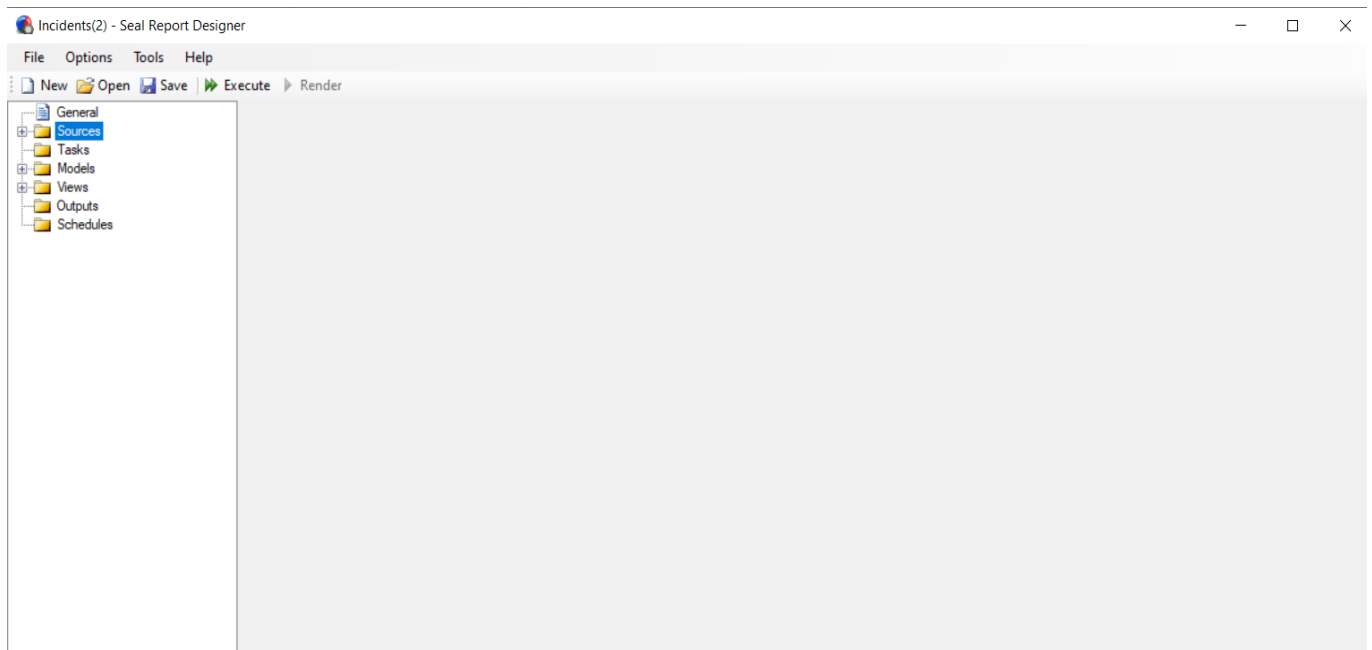
A sample join looks like this. We can specify the type of join in Join type, the tables involved and join clause.



In the Enumerated Lists, we can add a new list in which we can specify the values that we can later use in 'Columns displayed' or 'group by' etc.

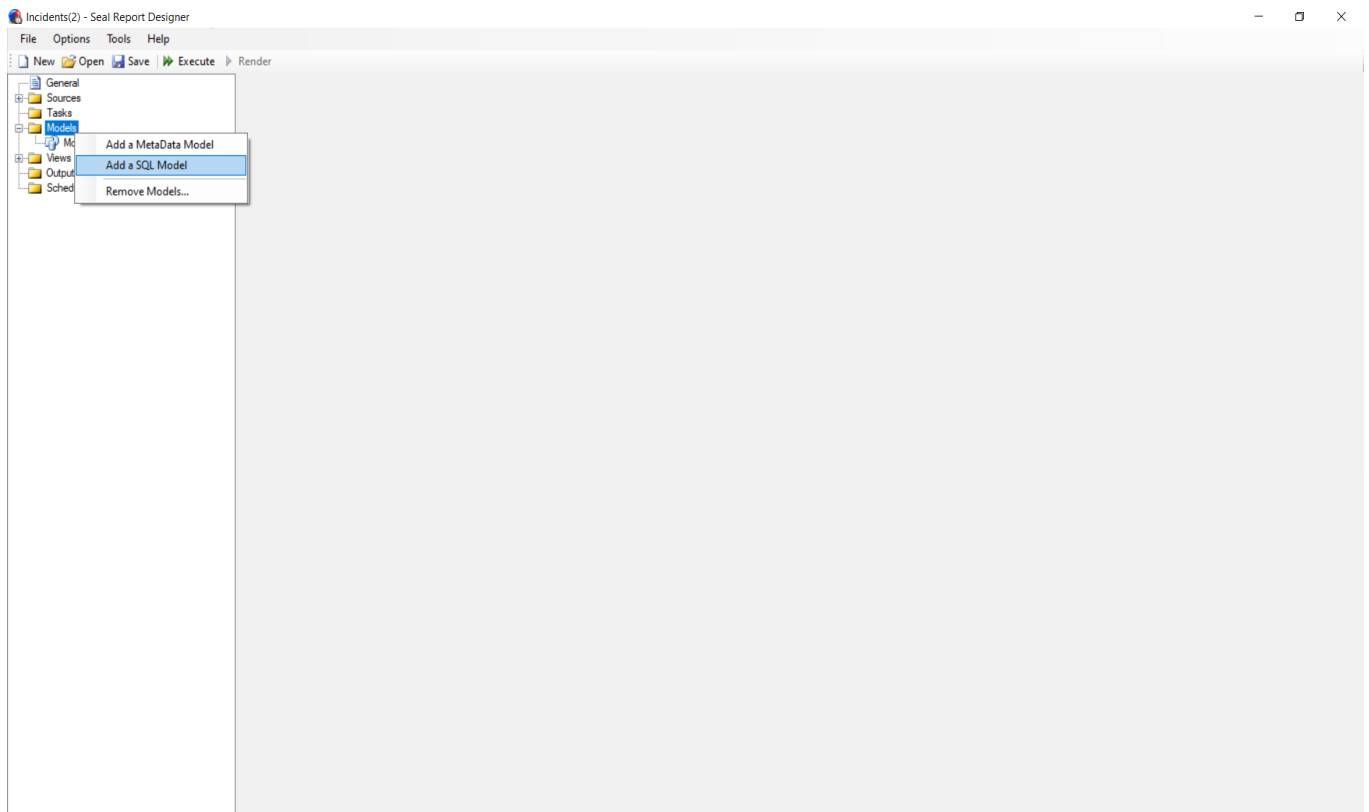


Similar procedure can be followed in Seal Report designer too to create a Source



Step 2 – Creating a Model

Right click on 'Models' and add the type of model we wish for



After we create our model, the source and connection that the model is gonna use can be specified in Model Definition part.

All the columns that can be used are displayed. We can drag and drop the columns that we require in the Elements section. Depending on how the report should look like we drop the columns in either 'Page elements' or 'Column elements' or 'Row elements' or 'Data elements'

We have added a Pre load script and Post load script in the Model Definition section in the concerned places for “Group by” feature

The following are the scripts we wrote

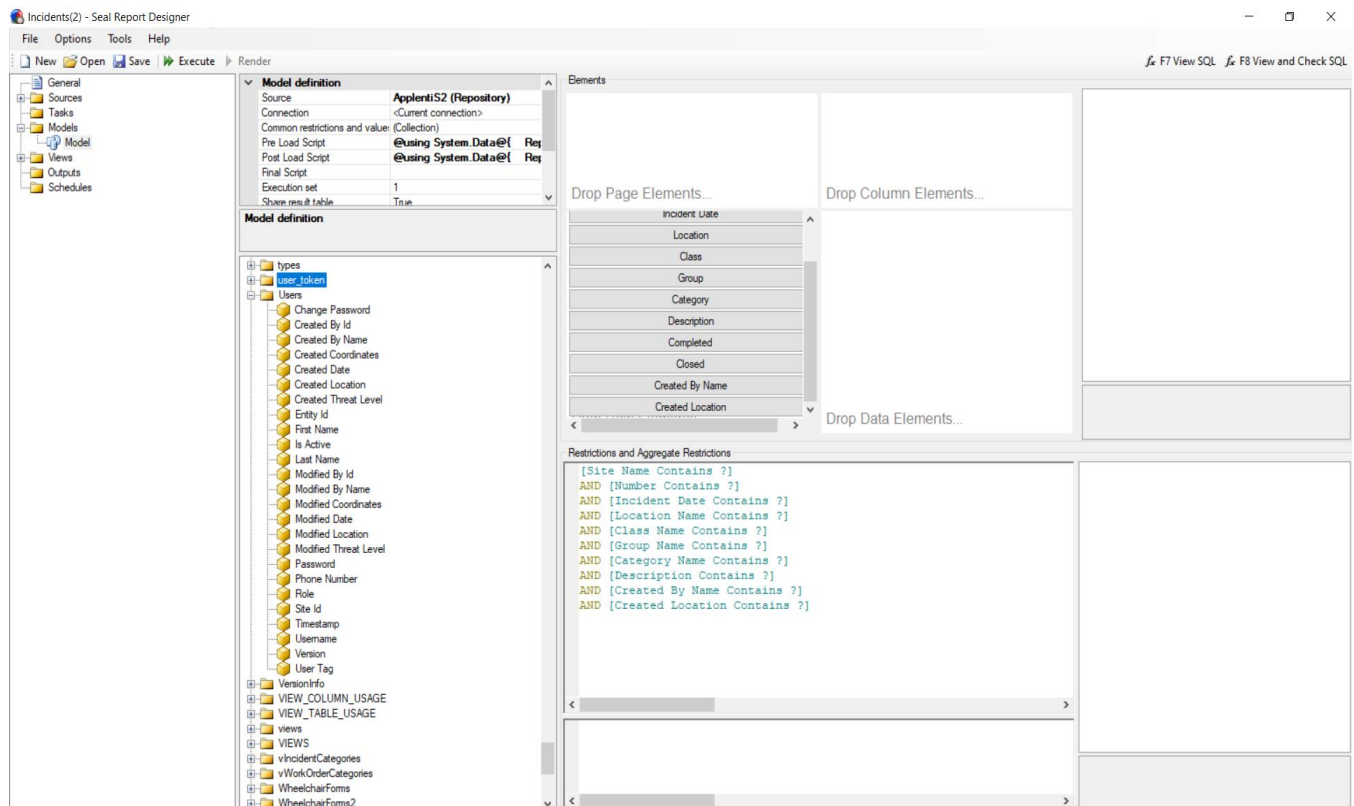
Pre Script

- Get the restriction from the model
- Check for the previous group by elements if they exist
- Get the new group by values from the restriction
- Find the difference between these 2 arrays
- Now in the last part here we bring these new group by rows to the front of the schema.

Post Script

- First we follow the above approach to find the new group by values
- Then we find the names of these columns in the actual schema
- Using these names we give a sort order to the result
- Finally we iterate over the result data and group elements until we find new values for the grouped columns since it has already been sorted

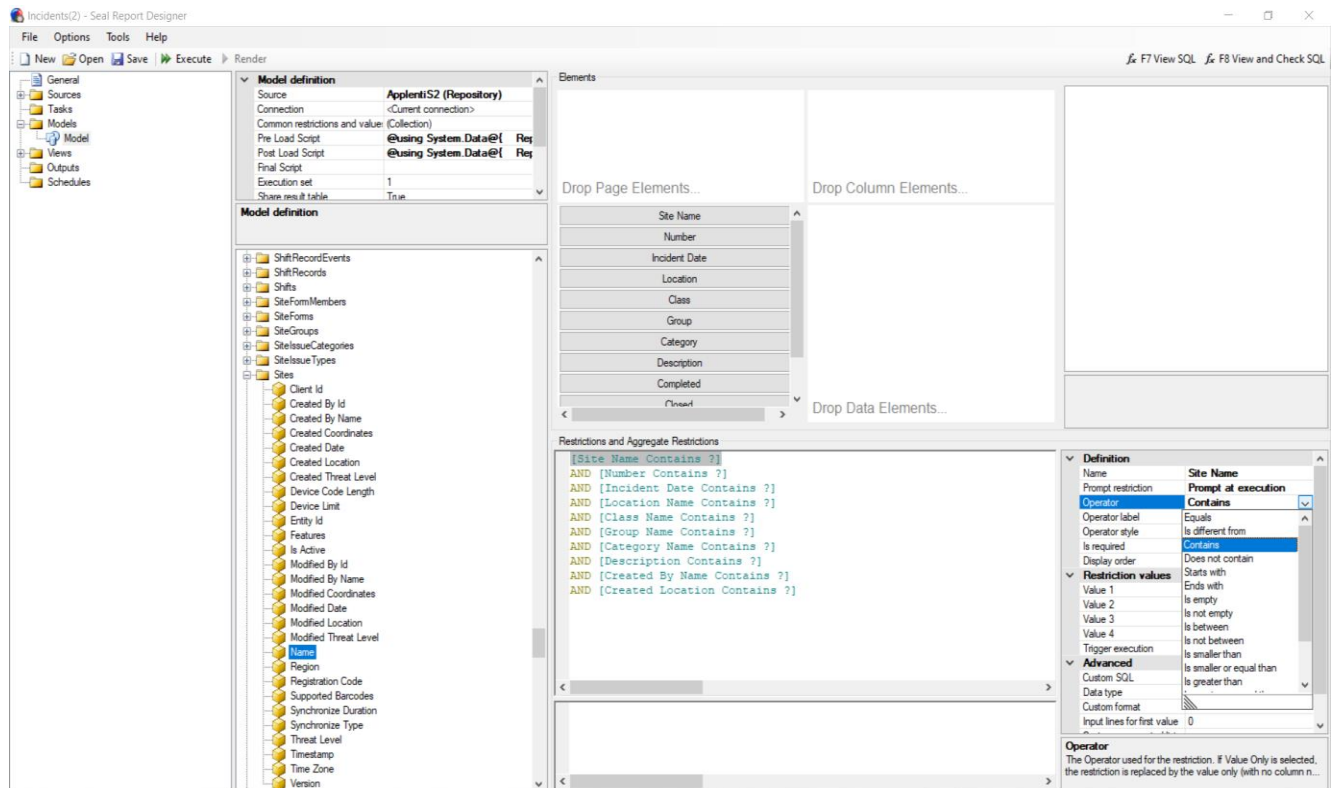
Below Screenshot is for the Incidents report where all the columns are placed appropriately.



The Restrictions on columns can be added by drag and drop of concerned column in the 'Restrictions and Aggregate Restrictions' section.

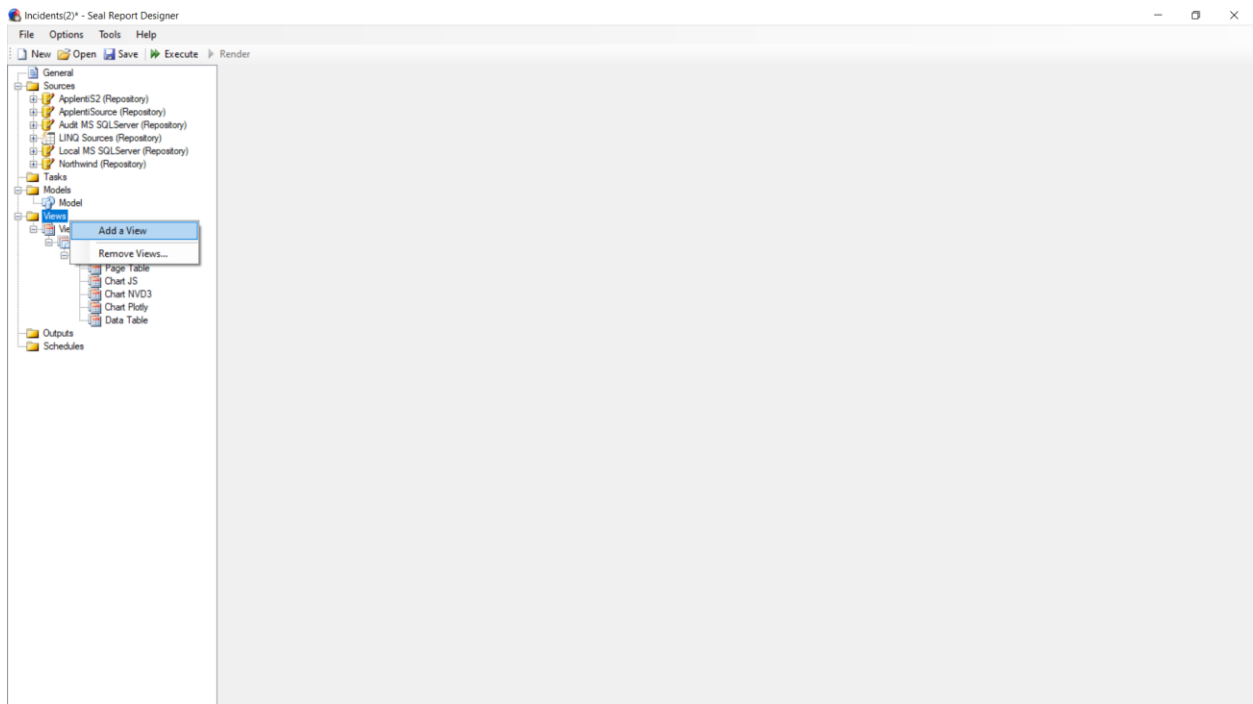
We can set the restriction like 'contains' or 'equals' or 'starts with' etc. as shown in screenshot below.

Also we can make them prompt at execution time and this basically works as a Filter on that column.

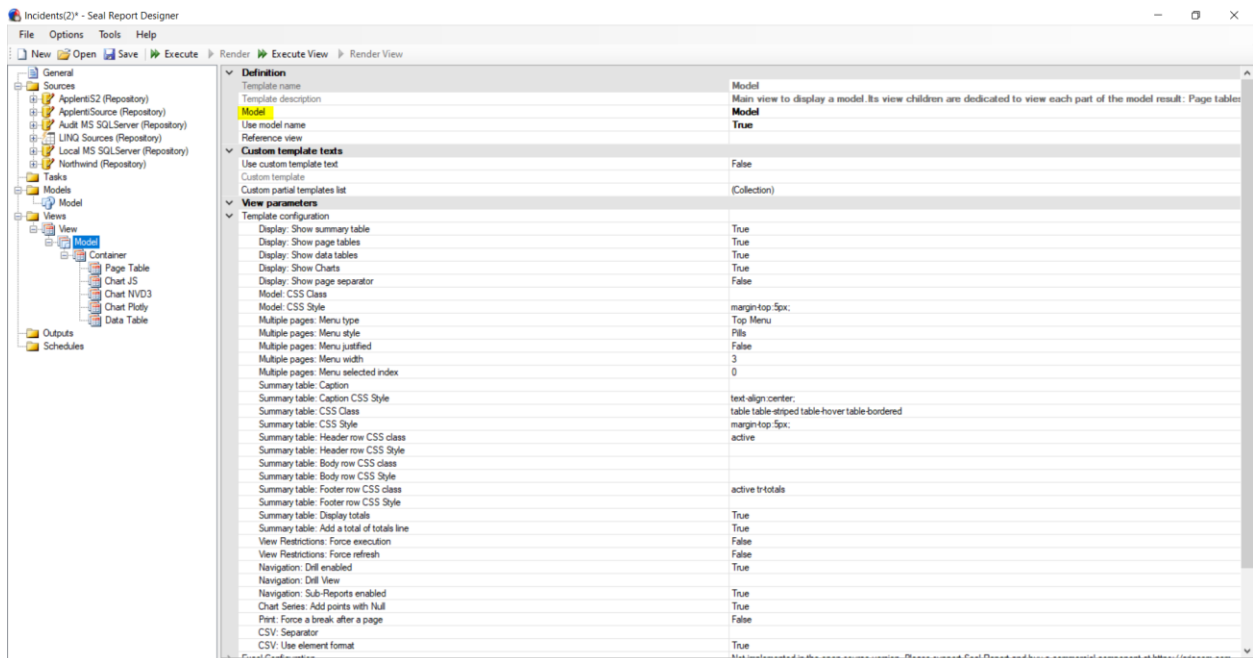


Step 3 – Create a View

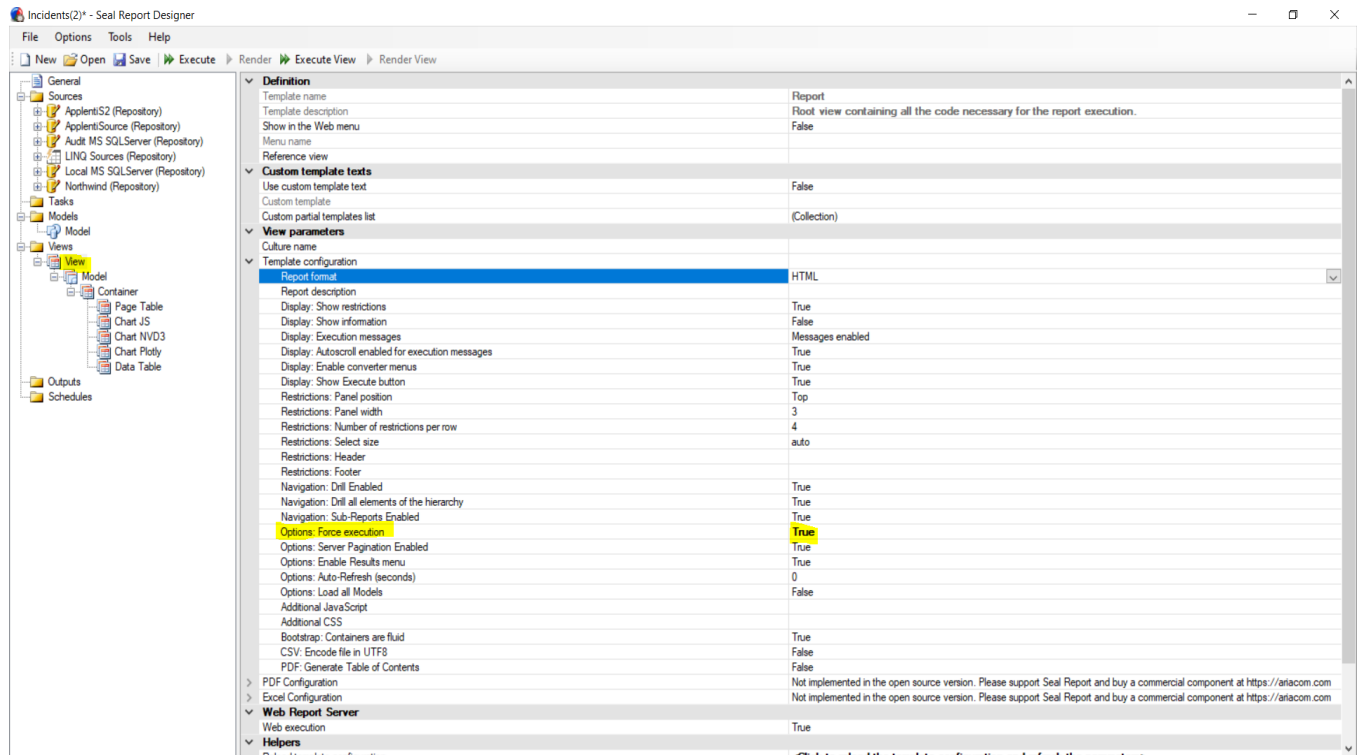
Right click on the view to create a view



Under View -> Model we can select the model we wish to create view for



In View if the option Force execution is set to true then the execution is done when the view loaded without clicking on execute button



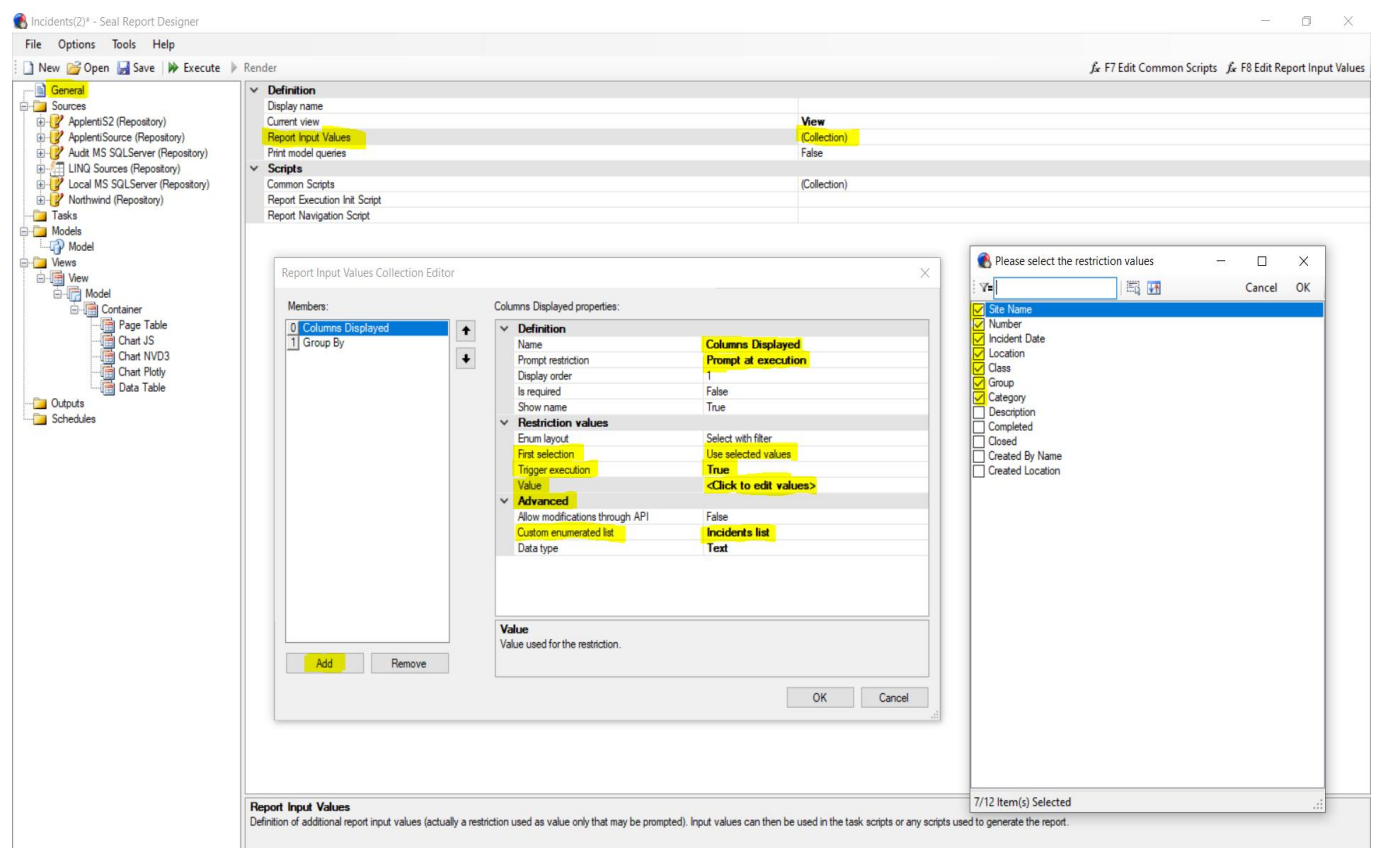
For different views to use some of the common features like 'columns displayed' and 'group by', We can go to General -> Click on 'Collection' under Report Input values -> Add the common elements here.

The values then are set as shown in screenshot.

For the view to load with some default columns, we can select any one of the enumerated lists that we created under 'Advanced' -> Custom enumerated list

After selecting the proper list, under Restriction values -> Value -> <Click to edit values> -> here we can select the columns that are used as default when a view loads

By setting the 'First selection' to 'Use selected values' we will be using only the values that we select and setting 'Trigger execution' to True makes sure the view loads/refreshes whenever a new value is selected.



Drill down –

In the incident report, if we hover over the incident no, we can see a drill down option to view the ‘information’ related to that incident

Incidents(2) - Seal Report Viewer

Incidents(2)

Columns Displayed: Site Name, Number, Incident Date, Location, Clas

Group By: Nothing selected

Site Name Contains:

Number Contains:

Incident Date Contains:

Location Name Contains:

Class Name Contains:

Group Name Contains:

Category Name Contains:

Description Contains:

Created By Name Contains:

Created Location Contains:

Show 25 rows

Filter:

Site Name	Number	Incident Date	Location	Class	Group	Category
Sample Site 1 - Dev		0001-01-01 00:00:00.0000000	Test8	Class	Group	Category
Sample Site 1 - Dev	IN-0000003	2014-09-15 05:55:08.0000000	Lance - Location 02	Security	Alarms	Alarm Malfunction
Sample Site 1 - Dev	Information	2014-09-15 06:03:36.0000000	Lance - Location 02	Security	Conflict	Assault
Sample Site 1 - Dev	IN-0000041	2014-11-06 13:56:00.0000000	Building 130 - Room 5	Security	Alarms	Alarm
Sample Site 1 - Dev	IN-0000049	2014-11-17 12:07:41.0000000	Lance - Location 02	Security	Conflict	Verbal Abuse
Sample Site 1 - Dev	IN-0000054-001	2014-12-04 14:30:00.0000000	Building 130 - Room 5	Security	Conflict	Injury
Sample Site 1 - Dev	IN-0000077	2014-12-23 07:16:00.0000000	Aztec 130 - Front car park	Security	Alarms	Alarm Malfunction
Sample Site 1 - Dev	IN-0000085-001	2015-01-09 00:01:07.0000000	Aztec 130 - Front car park	Emergency Rescue	Inner region	Floor
Sample Site 1 - Dev	IN-0000100	2015-11-06 21:31:48.7239170	Building 130 - Reception	Emergency Rescue	Outer region	Fire extinguisher
Sample Site 1 - Dev	IN-0000101	2015-11-11 14:01:03.0120060	Building 130 - Reception	Security	Conflict	Assault
Sample Site 1 - Dev	IN-0000102	2015-11-11 17:04:27.0804950	1000KVA Generator Diesel	DK Test	Dk Test - Groop	Dk Test - Category1
Sample Site 1 - Dev	IN-0000103	2015-11-11 20:05:40.1409890	Building 130 - Room 3	DK Test	Dk Test - Groop	Dk Test - Category1
Sample Site 1 - Dev	IN-0000104	2015-11-11 20:20:49.1873450	Building 130 - Reception	Emergency Rescue	Inner region	Floor
Sample Site 1 - Dev	IN-0000105	2015-01-28 03:28:00.0000000	Emergency exit	Security	Alarms	Alarm Vandalized

This is done as shown below

In the model that we created, if we click on the columns that we dropped the elements section then that shows us the table under which the column is actually chosen from. In this case the Incident number column is chosen from the table Incidents, which is highlighted in blue when we click on Number

Incidents(2)* - Seal Report Designer

File Options Tools Help

New Open Save Execute Render

Model definition

Source: AppletIS2 (Repository)

Connection: <Current connection>

Post Load Script

Optional Razor Script to modify the result table of the model just after the...

IncidentProducts

Incidents

Category Name

Class Name

Closed By Id

Closed By Name

Closed Coordinates

Closed Date

Closed Signature

Closed Signature Url

Closed Threat Level

Completed By Id

Completed By Name

Completed Coordinates

Completed Date

Completed Signature

Completed Signature Url

Completed Threat Level

Created By Id

Created By Name

Created Coordinates

Created Date

Created Location

Created Threat Level

Description

Entity Id

Group Name

Incident Date

Is Active

Is Closed

Is Completed

Is Revised

Location Name

Modified By Id

Modified By Name

Modified Coordinates

Modified Date

Modified Location

Modified Threat Level

Number

Parent Id

Parent Type

Elements

Drop Page Elements...

Drop Column Elements...

Drop Data Elements...

Restrictions and Aggregate Restrictions

[Site Name Contains ?]

AND [Number Contains ?]

AND [Incident Date Contains ?]

AND [Location Name Contains ?]

AND [Class Name Contains ?]

AND [Group Name Contains ?]

AND [Category Name Contains ?]

AND [Description Contains ?]

AND [Created By Name Contains ?]

AND [Created Location Contains ?]

Definition

Name: Number

Sort order: Automatic Ascendant

Chart

Options

Data type: Default

Show sub-totals: False

Custom format

Advanced

Custom SQL

Cell script

Cell navigation script

Custom enumerated list

Force aggregate: Default

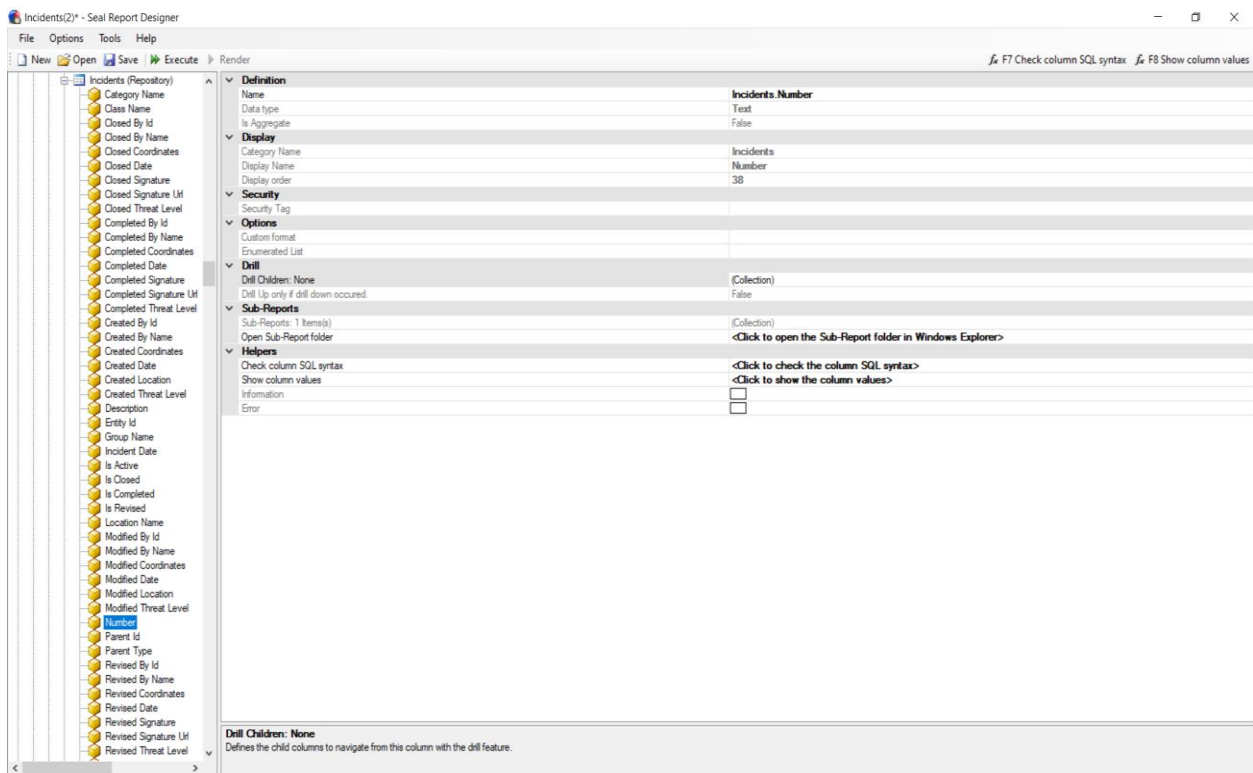
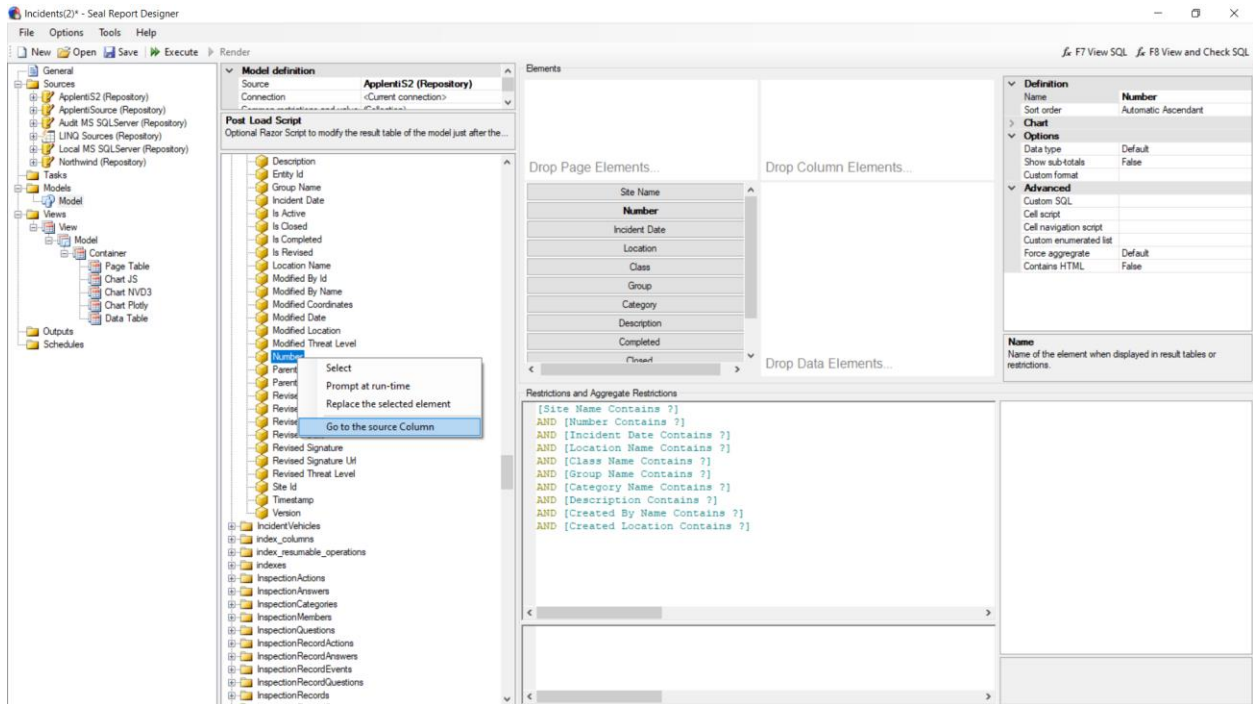
Contains HTML: False

Name

Name of the element when displayed in result tables or restrictions.

Right click on the column and click on 'Go to Source Column'

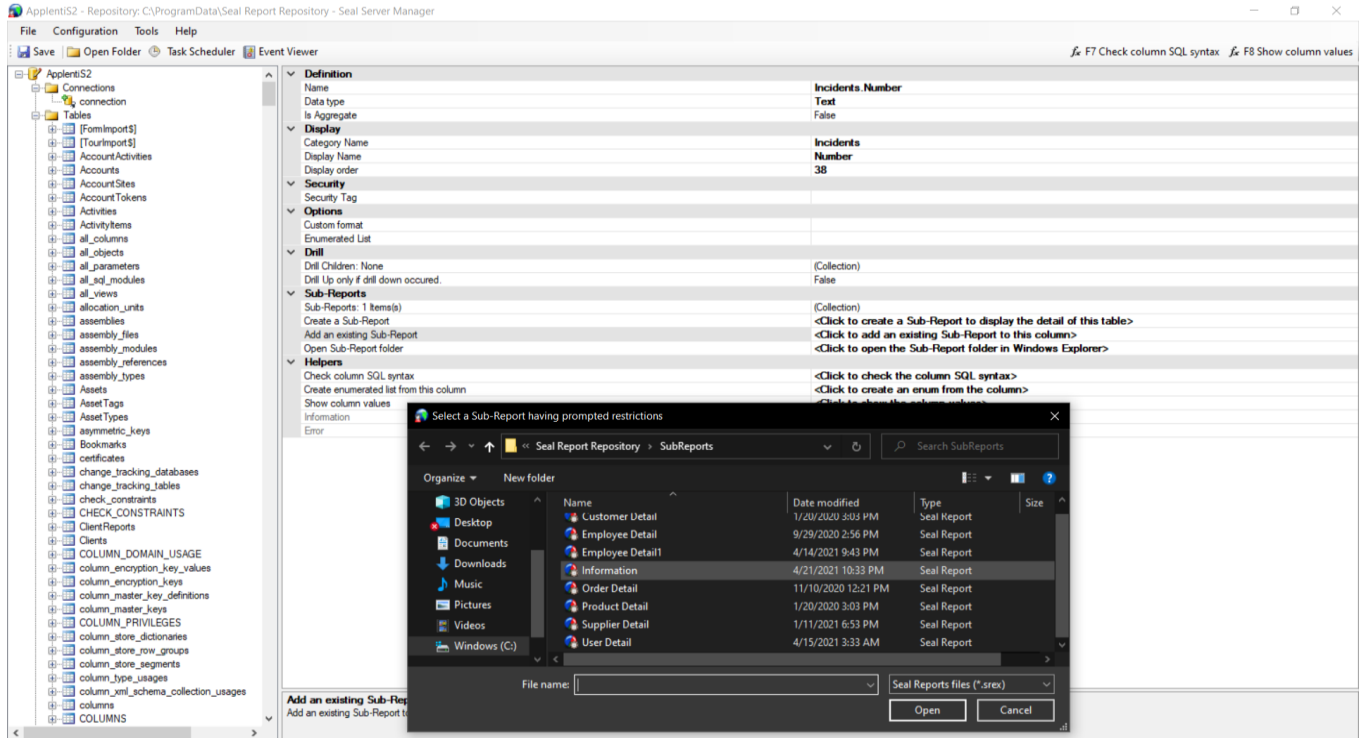
This takes us to the data source under which the table of the selected column exists



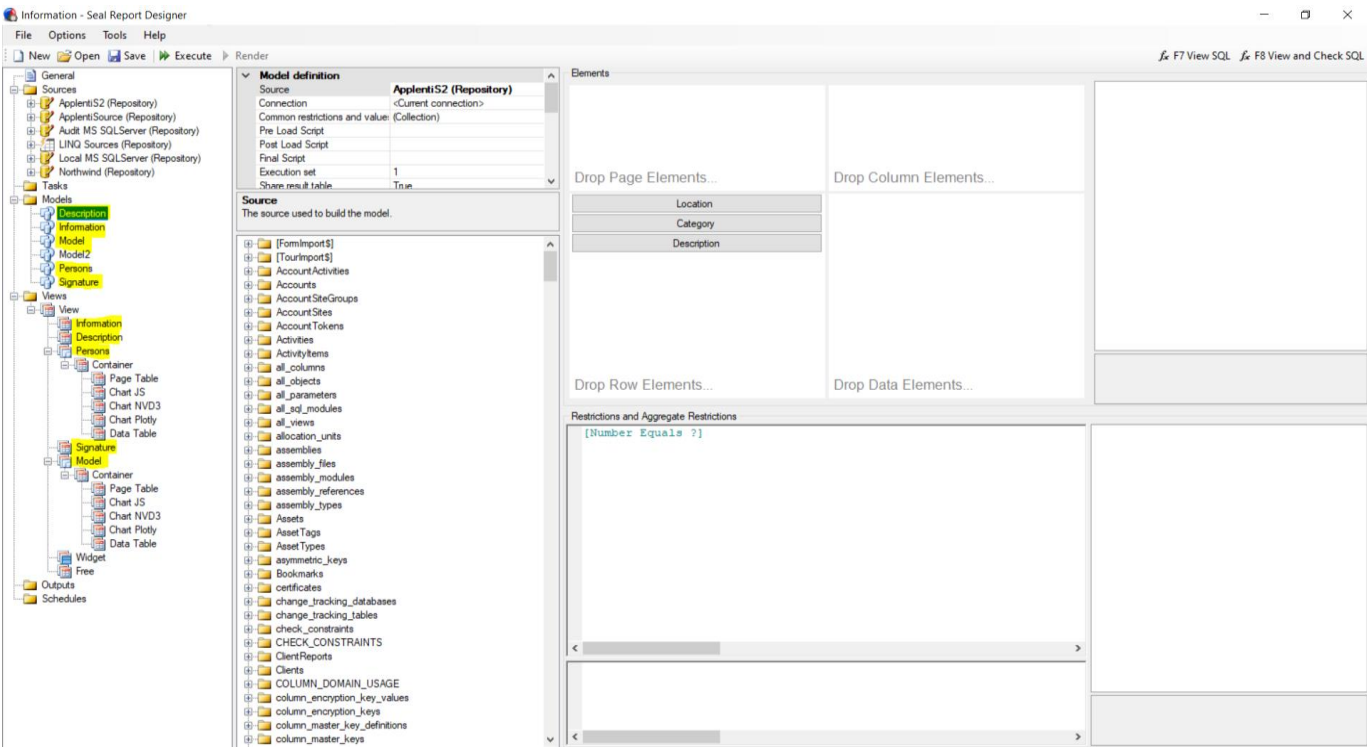
So to generate a sub report when we click on the 'Information' that we see in our view, we have to add a 'Sub report' under 'Sub reports' as seen in above screenshot. This is done in Server manager

In Sever manager -> open the data source that we created -> go to tables -> Incidents -> Number

Here we can just attach the report we want to be shown when we drill down in to the Incident number of our previous report. This is done by creating another report and adding it under 'Sub-reports' -> Add an existing sub report -> and in this case it is the 'Information' report.



The Information report which is our sub report looks like this i.e. the model looks like this



As the sub report contains a number of views combined in to 1 report, different models are created under Models and under Views -> view -> Different types of views required to generate a report are placed.

The following report is generated when we click on Incident number -> Information of our report

Incidents(2) - Seal Report Viewer

Information > IN-0000003

Execute

Site Name:	Sample Site 1 - Dev
Number:	IN-0000003
User:	Lance Sologub
Incident Date:	2014-09-15 05:55:08.0000000
Created Date:	2014-09-15 05:55:24.0000000
Modified Date:	2014-10-29 19:30:34.0000000

Show 25 rows

Filter:

Role	First Name	Last Name	Height (Inches)	Gender	Ethnicity	Build	Hair Color	Hair Style	Hair Length	Phone Number	Number	Address	City	State	Postal Code	Country	Email Address	Notes
Customer	Dddd	Eeee	0								IN-0000003-001							
Suspect	Adam	Frost	0	Male	IC1 - White Person						IN-0000003							
Suspect	mr	test	72	Male	IC0 - Origin Unknown	Large	Dark Brown	Unkempt	Long	158181818181	IN-0000003							Person notes
Suspect	Vicente	Guzman	0								IN-0000003-001							

Showing 1 to 4 of 4

Show 25 rows

Filter:

Image Uri	Thumbnail Uri	Notes	Number
https://e360dev.blob.core.windows.net/media/images/ae76bea9-0797-45f7-ab85-c39240e3422f-image.jpg	e360dev.blob.core.windows.net/media/thumbnails/ae76bea9-0797-45f7-ab85-c39240e3422f-thumbnail.jpg	Notes	IN-0000003