Notice

The contents of this document pertain to technology developed **prior to January 2017**. This information is subject to the 1Schema.com Terms of Service (www.1schema.com/Home/TermsOfService), as are all files (including files in sub-directories) contained within this directory.

The Need

When using traditional Reporting Engines such as SQL Server Reporting Services or Crystal Reports, the User must spend a lot of effort to template of the report to the appropriate data source and to query the correct information.

Additional problems when designing these reports include:

- Poor customization of template blocks
- Poor ability to control multidimensional aggregation
- Grouping of information is integrated into Report layout
- Resulting output is quite static, like a PDF

1Schema circumvents these challenges by leveraging its automated querying engine to automate the challenging aspects of reporting.

Because 1Schema stores Variables along with metadata about their Dimensional hierarchy and prior dependencies, 1Schema easily infer default report settings for Variables, as well as allow Users to override those settings with marginal effort.

To start designing a Report, Users need only drag 1Schema Variables onto the Report Designer surface.

1Schema automatically detects whether the Variable includes is at least as general as the Report itself (e.g. Company information in a Company report). If so, it will add the Variable directly to the surface of the Report.

If the information is less general, then it must have additional Dimensions beyond those inherent to the Report. In this case, 1Schema adds the Variable to a Dimensional Table (a report element most akin to an Excel Pivot Table).

In the Dimensional Table, 1Schema presents Users with information organized such that the Dimensions common to all Variables (aka the Common Dimensions) are presented along one axis (X or Y). The Variables, along with their, Variable-specific Dimensions are presented along the other axis.

The axis of the Common Dimensions is called the Common Axis. The other axis is called the Variable Axis.

By organizing information in this manner, grouping becomes quite simple. The Dimensional Table stores a "Repeat Group" name for each Variable in the table, and at the time of rendering, Variable values are automatically presented as grouped based on the Repeat Group name.

Furthermore, the Repeat Group can be specified in such a manner as to build a tree of groupings based on the Dimensions specific to each Variable in the Variable Axis. For instance, Repeat Group values of "Company" for one Variable and "Company.Product" for another Variable would yield a "Company" group with a nested "Product" group inside.

Also, note that the names of Repeat Groups do not have to use Dimensions, as the resulting Dimensions for the Groups are automatically determined by 1Schema. However, it may be easiest for the User to use Dimension names within the Repeat Group names.

Dimensional Reports

In 1Schema, the challenge for Reports is to "project" N-Dimensional information into the 3-Dimensional Workbook Structure (i.e. Sheet, Row, Column = 3 Dimensions).

1Schema detects the Dimensionality of the information contained within the Report and ensures that all necessary Dimensions are either directly presented or aggregated away.

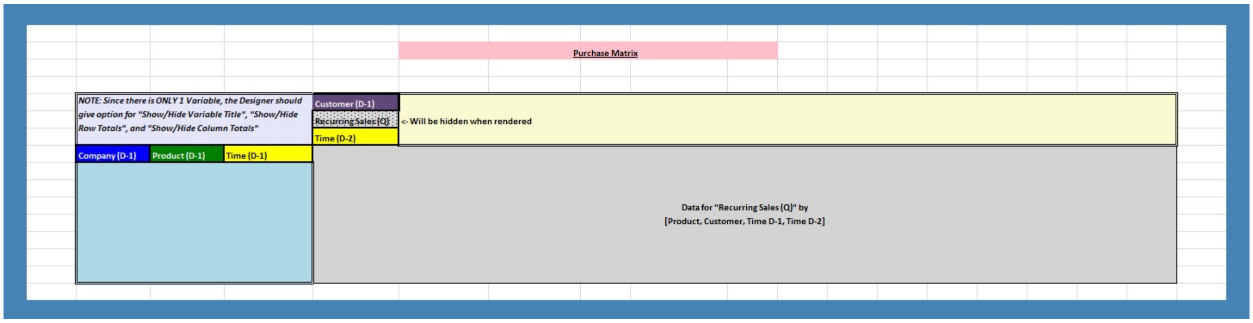
To accomplish this, the User starts by defining the Sheet-level (i.e. "Report) requirements for the Report, including the Structural Type that determines the Multiplicity of the Report. For instance, if the Report is tied to the "Product" Structural Type, an Report will be rendered for each Product in a given Model.

Additionally, the Report Builder provides the following 1Schema Object Types:

- Cell:
  - An Element whose multiplicity is tied to the Structural & Time Context of the Report.
  - The value of this Element can be a Variable, a Formula, or Direct Input.
- Container:
  - A layout Element for presenting information in a highly customized display.
- Dimensional Table:
  - An Element most analogous to an Excel "Pivot Table".
  - It allows flexible presentation of multiple Variables, along with their Structural and Time Dimensions.

Dimensional Tables

Below is an example of a Dimensional Table in Design Mode.



Purchase Matrix

NOTE: Since there is ONLY 1 Variable, the Designer should give option for "Show/Hide Variable Title", "Show/Hide Row Totals", and "Show/Hide Column Totals"

Customer (D-1)

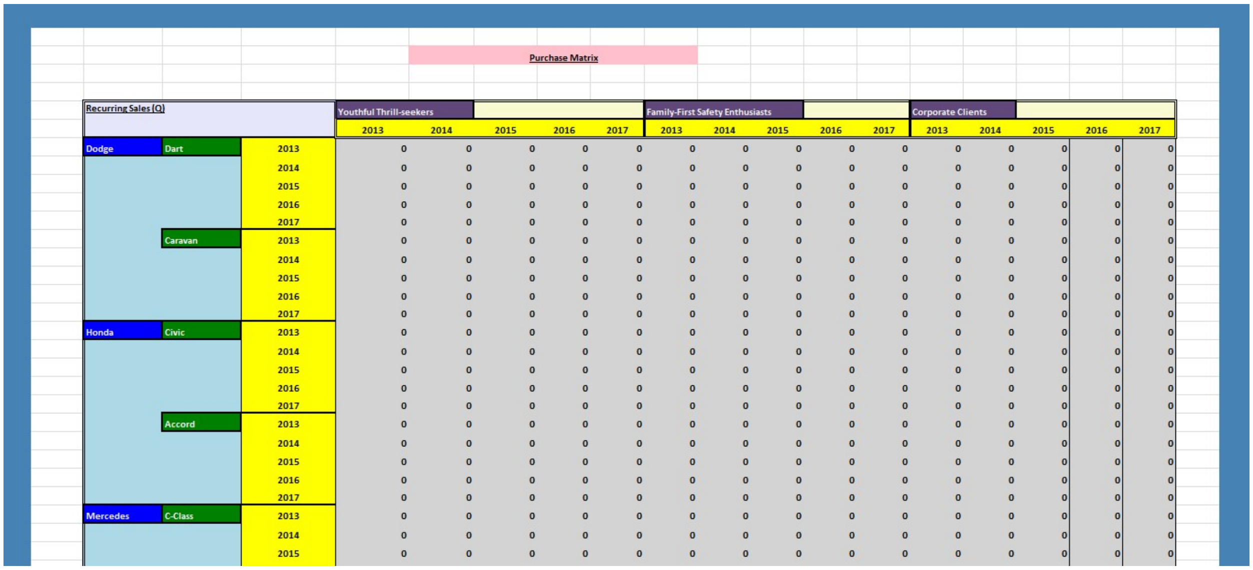
Recurring Sales (Q) <- Will be hidden when rendered

Time (D-2)

Company (D-1) Product (D-1) Time (D-1)

Data for "Recurring Sales (Q)" by
[Product, Customer, Time D-1, Time D-2]

Below is an example of that same Dimensional Table rendered with actual data.



Purchase Matrix

Recurring Sales (Q)			Youthful Thrill-seekers					Family-First Safety Enthusiasts					Corporate Clients				
			2013	2014	2015	2016	2017	2013	2014	2015	2016	2017	2013	2014	2015	2016	2017
Dodge	Dart	2013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2015	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Caravan	2013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2015	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Honda	Civic	2013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2015	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Accord	2013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2015	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mercedes	C-Class	2013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		2015	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*** All companies and products used above are registered trademarks of those respective companies. These companies do not have any relationship with Decia LLC and are used purely for explanatory purposes.**

Report Layout and Styling

In contrast to current reporting engines, which are more focused on rendering text and documents, the 1Schema reporting engine uses flexible Layout and Styling data structures that yield a superior user experience.

In 1Schema, the Report Designer UI feels and acts more like a "Windowing" graphical sub-system. Each object has its own Layout and Style, as well as a "Default Layout" and "Default Style". Layouts and Styles can be chained together. The User can define a library of Styles, such that Styles are shared across multiple objects. The User can also specify how elements "Repeat" together, which is particularly useful when objects have overlapping Dimensionality.

When rendered, the Dimensionality, Layout, and Style settings are combined to determine whether Report Elements use Absolute or Relative sizes and positions and whether the Report Elements are rendered once, or multiple times due to Dimensional Multiplicity.

Some of the settings available in the Layout data structure, for each Dimension (X=Column, Y=Row, Z=Sheet, etc.), include:

- **Dimension**: The Excel Dimension that the Layout object corresponds to
- **Default Layout**: Another instance of the Layout data structure to get default values from
- **Anchor Mode**: How the layout is positioned relative to its Parent Element
- **Size Mode**: Whether the Size is specified in Cells, a Ratio, or a Percentage
- **Range Size**: The size in of the object for that Dimension. Sizes are ultimately computed as number of Cells
- **Stack Contents**: Whether to automatically Stack Contents for that Dimension for child Report Elements
- **Rank**: Given the parent Report Element stacks objects, the Rank specifies the Order for each particular child Report Element
- **Minimum Range Size**: The minimum number of Cells that the object's size can be set to
- **Maximum Range Size**: The maximum number of Cells that the object's size can be set to, or NULL if the object is unconstrained (i.e. there is no Maximum Size)
- **Margin**: The number of Cells used around the exterior of the object to space it from its Parent
- **Padding**: The number of Cells used on the interior of the object to space its contents from its border
- **Cell Size**: A way to control the Size of the Row or Column in Excel such that the Size can vary
- **Override Cell Size in Padding**: Whether to use the Cell Size value to control the Rows or Columns for the Padding, as well as the actual Contents
- **Merge Interior Area Cells**: Controls whether the Cells for the actual contents are Merged in the Excel export

Some of the settings available in the Style data structure, for each Element, include:

- **Default Style**: Another instance of the Style data structure to get default values from
- **Background Color**: The color of the background of the Element
- **Foreground Color**: The color of the foreground of the Element
- **Font Name**: The name of the Font to use for rendering the Element's contents
- **Font Size**: The size of the Font to use for rendering the Element's contents
- **Font Style**: The style of the Font to use for rendering the Element's contents
- **Horizontal Alignment**: The horizontal setting for aligning the text in the Element's contents
- **Vertical Alignment**: The vertical setting for aligning the text in the Element's contents
- **Indent**: The number of units to indent the text in the Element's contents
- **Border Color**: The color of the Border for the Element
- **Border Style**: The Excel line style of the Border for the Element