

# Formulize 4



## Getting Started Guide

August 28, 2012



This document has been prepared by Freeform Solutions. It is licensed under a Creative Commons license (attribution – non-commercial – no derivatives).

## Table of Contents

Overview.....	3
How can you help? How can you get help?.....	3
Some of the major new features in Formulize 4.....	4
Creating a Form.....	5
Configuring Formulize and Accessing Forms.....	6
Controlling how the forms and the lists of entries look.....	6
Relationships Between Forms.....	6
Other Ways of Accessing Forms Besides the Menu.....	7
Through the URL.....	7
Through PHP code.....	7
How do I change the way the "list of entries" screen looks? .....	8
Why do I see a list of entries instead of my form, or vice versa?.....	9
Saving and Loading Views of Your Data.....	9
Troubleshooting Tips.....	10
I'm getting some strange database errors, any suggestions? .....	10
Help, my page is blank, or not drawn completely on the screen!.....	10

I've changed my derived value formula, but the actual values aren't changing. Why?....	10
My {TODAY} search isn't working. Why?.....	11
I have marked some elements as "disabled" for some groups, but some of those people still see the elements. Why?.....	11
<b>Search Terms.....</b>	<b>11</b>
The basics.....	11
Numbers.....	12
More than one search on the same column.....	12
Wildcard terms {TODAY}, {USER} and {BLANK}.....	13
"OR" searches on multiple columns.....	14
Advanced developer-focused search terms.....	14
<b>Relationships Between Forms.....</b>	<b>15</b>
Creating Relationships.....	15
Using and Accessing Relationships.....	15
Screens.....	15
API.....	16
URL.....	16
Where to find the frid.....	16
Scenarios for using Relationships.....	16
Unified Data Viewing.....	16
Unified Data Entry – Subforms .....	17
One to Many Relationships and Changing the Main Form: Reversing the Point of View .....	18
Unified Data – "one-to-one" Forms.....	18

## Overview

Formulize is a tool that lets you create forms in your website, users can fill them in, and you can then view, search, sort and analyze the data afterwards. There is also a report writing feature which lets you save “views” of the data for future reference, or even publish views to other users in your site.

Formulize also has a "screen" system that lets you customize the appearance of the module, to suit whatever applications you are trying to build. Read the section below on changing the "list of entries" screen to get started.

Formulize includes several functions that can be included from outside the module. So if you are a developer or webmaster, and you want to create a form-driven, custom application in your website, and the screen system is not enough for you, then this “API” lets you rapidly create an application on top of a standard Formulize form. For more information about this, please see the PDF called *Using Formulize and Pageworks to Make Custom Applications* available from: <http://www.freeformsolutions.ca/formulize>

Essentially, if there is no existing tool specifically made to solve your problem, and your problem can be boiled down to a data input/data output description, then you can probably create an application with Formulize that will meet your needs more than adequately.

For installation instructions, consult the readme file that came with your copy of Formulize.

## How can you help? How can you get help?

**If you want step-by-step help, there are tutorial videos for Formulize. Follow the link for them, from the *Downloads and Docs* page at:  
<http://www.freeformsolutions.ca/formulize>**

Freeform Solutions is actively developing Formulize, with support from the community. You can post issues, questions, feedback, problems, etc, on support forums at <http://www.freeformsolutions.ca/formulize>. We will attempt to respond to all postings as soon as possible. Further documentation is available at that website.

## Some of the major new features in Formulize 4

- **Completely new administration UI!** Everything is a lot easier to find and it makes more sense how things are related together.
- **A module for integration with Drupal and other CMSs!** You can use the Formulize Standalone version together with Drupal, to publish Formulize screens inside your Drupal site. You can also embed Formulize screens in other CMSs too, using the same principles and the API.
- **Autocomplete boxes!** Anywhere that you'd use a drop down list in the past, you can convert it to an autocomplete box now.
- **Add file-upload elements to your forms!** In part created with funding from the community, there is now a file-upload element available for adding to forms. Let your users attach files to their form entries.
- **Conditional elements show and hide automatically!** As soon as a user types or clicks something that matches the condition for an element, that element is automatically added to the form without saving or reloading.
- **New graphics for the buttons!** Thanks to Francois at [www.eorthodontie.com](http://www.eorthodontie.com)
- **Embed entire forms inside other forms!** You used to be restricted to showing only a few elements from another form, if you needed to include one form inside another. Now you can embed the entire form if you want to. It appears inside a rolled up “accordion” section so it doesn't take up space until the user clicks on it.
- **Track revision history for entries in a form!** Now you can have a copy in the database of every version of every entry, if that kind of audit trail is important.
- **No more framework handles!** Now there's one way to refer to elements, and that's it. No more doing things one way sometimes, and another way other times.
- **Create your own element types!** Now you can write one PHP file, and make your own element types. Use your own elements to enforce custom rules and logic that are unique to your site – or create something useful for everyone and share the files with the community.
- **Major performance improvements when using complex, multi-form relationships!** Got 20 subforms attached to a main form? Not a problem. It might have timed out before, but now it only takes seconds to gather all your data together. Any set of 3 or more forms will perform much better now.

- **Lots more permissions and options throughout!** There's all kinds of new ways to configure various features and settings throughout Formulize, so you have even more control over your applications. Some of the cool new things:
  - send notifications to e-mail addresses users type in, and other new notification options
  - improved export process, which auto-starts downloads now
  - lock columns on screen so they stay on the left as you scroll in a list of entries
  - activate form elements in a list of entries screen, using icons and Ajax requests
  - control which groups see which custom buttons
  - link together multiple forms and screens using new template functions
  - specify form handles, so the database table names are easier to read
  - add date range searches to a screen using \$quickDateRangehandle
  - use {PERGROUPFILTER} to make a saved-view search with the active user's group permissions on a form

## Creating a Form

The most basic thing you can do is create a form. You can do this easily by going to the main admin page for Formulize, and clicking where it says “Create a new form”

At that point, you will be asked for the name of the form and also how many entries people are allowed to make in it.

Next, specify the elements that you want in the form, such as text boxes, radio buttons, dropdown lists, etc. There are a lot of configuration options for each element. There is helpful descriptive text shown on screen near most options.

Lastly, you must specify the permissions for the form so that people can access it. Use the Permissions tab to do this for each form. On that tab, you can specify the group or groups you want to modify permissions for.

Your form is now ready for use. Simply go to the front page of the site and click on the form in the Form Menu, or wherever else you have setup navigation to your forms.

## Configuring Formulize and Accessing Forms

Formulize 4 does not require much configuration. In the area where you manage groups (probably System—Groups, but it depends on which exact software you're using), you should make sure that all the groups that need to access forms have Module Access permission for Formulize. Usually giving Registered Users this level of access, is enough.

Formulize also has some preferences that you can use to control the default settings for various form elements, as well as the default number formatting options for textboxes. There's a link for the preferences on the main Formulize admin page.

If you are using the Main Menu in a XOOPS or ImpressCMS site, then a link for Forms will appear there and it should provide a list of "General Forms" that includes all the forms you have created in your site. You can customize this list (hide certain forms from it) by removing the menu text for each form, on the Settings page.

## Controlling how the forms and the lists of entries look

Each form can have one or more “screens.” A screen is a way of presenting the form to people. You can have different screens for the same form, so that you can show different versions of the form to different people and/or in different situations.

See the section called *How do I change the way the "list of entries" screen looks?* below for more information on what screens are and how to create them. There are many settings for the different screen types, especially the list of entries screens. Experiment with them to learn more, or watch the tutorial video about altering screens.

## Relationships Between Forms

You can specify that two or more forms are connected in a relationship. This can let you embed one form inside another, such as having a Final Report form, and an Activity Form, and people should fill in several activity records as part of the final report.

To do this, you can create a one-to-many relationship between the Final Report form and the Activity Form. Use the Relationships tab on the Application's admin page. You will need a way of joining the forms together. Usually you would do that by creating a selectbox in the Activity Form, and linking it to an identifying element of the Final Report. That linked selectbox can then be the basis for the relationship between the forms.

Each screen can use a relationship as the underlying data set that it works with. By default screens are based on a single form by itself. But if you use a relationship, then you can start to work with all the data from all the forms that are joined to this one. For more information, see **Appendix C – Relationships Between Forms**

# Other Ways of Accessing Forms Besides the Menu

## Through the URL

You can access forms via the URL simply by adding a "fid" parameter to the end of the URL and including a valid form ID. You can learn the form ID numbers from the main admin page, where they're listed in ( ) after the name of each form. This URL will access form 5:

```
http://www.yoursite.com/modules/formulize/index.php?fid=5
```

You can access specific entries in a form by adding a "ve" parameter to the end of the URL (view entry). You can learn the correct value for the "ve" parameter by looking in the standard List of Entries page for a form, and hovering the mouse over the page and magnifying glass icon. The status bar of your web browser will show a URL for the entry, including a "ve" parameter. Use that number to access that entry directly. ie, this URL will access entry number 99 from form 5:

```
http://www.yoursite.com/modules/formulize/index.php?fid=5&ve=99
```

## Through PHP code

You can also refer to certain things in PHP code, which gives you a whole lot of flexibility for controlling how people interact with different parts of your Formulize applications.

You can use PHP to include any "screen" in a page. See the section called *How do I change the way the "list of entries" screen looks?* below for more information on what screens are and how to create them. You can also include any Pageworks page, if you are using the companion Pageworks module (Pageworks is useful for creating highly custom interfaces and systems based on Formulize data structures, since it gives you an easy way of managing PHP pages in your website, and it provides easy access to the Formulize API).

To include a screen using PHP, just add this PHP code to your page:

```
include "path/to/mainfile.php";
$formulize_screen_id = 1;
include XOOPS_ROOT_PATH . "/modules/formulize/index.php";
```

That will cause screen number 1 to appear in your webpage.

To include a Pageworks page using PHP, just add this PHP code to your page:

```
include "path/to/mainfile.php";

$page = 1;
include XOOPS_ROOT_PATH . "/modules/pageworks/index.php";
```

That will cause Pageworks page number 1 to appear in your webpage.

## How do I change the way the "list of entries" screen looks?

Since Formulize 2.3, you can define multiple screens that control the appearance and behaviour of the "list of entries" page that people use to work with your form. You can make other types of screens too, not just lists of entries. Each screen is a different version of the form and the data in it, which you can make available to people by giving them the screen's URL, or adding a link to it in your website.

You create and configure screens using the Screens tab on the form's admin page. Add a new list of entries screen, and try changing the number of entries shown on each page. Try changing the text for certain buttons, or removing the buttons altogether by leaving the text blank.

Save your screen, and then note its ID number. With that, you can access your screen by going to this URL:

<http://www.yoursite.com/modules/formulize/index.php?sid=1>

You can also set a screen to be the default screen for a form. In that case, it will be used whenever someone goes to the main URL for the form, ie: the ?fid=1 page.

You can go back and make more changes to the screen and they will be reflected at that URL, after you have saved the changes.

You can completely alter the appearance of the screen by using the custom template options near the bottom of the screen admin interface. You can customize where you want buttons and search boxes to appear, by using the variables listed and using PHP code to print them out wherever you wish on the screen.

You can even customize the appearance of each row in the list, so that it doesn't look like a table at all. Use the "display" function to print out specific values from each entry in each row. For more information on the display function, consult the *Using Formulize and Pageworks to Make Custom Applications* PDF, available on the Freeform Solutions website.



## Why do I see a list of entries instead of my form, or vice versa?

It all depends on the permissions you have on the form, and how many entries the form is configured to allow.

For instance, if you make a form that is set up to allow more than one entry per user, then when you go to that form, the default screen that you see will be a list of entries in the form.

If you make a form that is setup to allow only one entry per user, then when most people go to that form, they will see the form itself and not a list. However if someone has the ability to see all entries by everyone in a group, or all entries by everyone, period, that will allow them to see multiple entries not just their own. In those cases, people will see the list of entries by default instead. The webmaster account has all permissions, so it should always see a list of entries by default, no matter what.

## Saving and Loading Views of Your Data

You can use the *Save current view* button on the "list of entries" screen to save all the search, sort, calculations and any other selections you have made. Your saved view will show up in the *Current view* drop down list as an available view that you can return to at any time.

Webmasters and other users with sufficient permissions on the form, can publish saved views to other groups of users. This can be a useful way to make reports available to people.

When you make custom screens, any published view can be used as the default set of entries that shows up in the screen. Also, when you are using screens, you can cause certain views to be used as the default view on a page, by putting the view number in the URL like this:

```
http://www.yoursite.com/modules/formulize/index.php?sid=1&loadview=12
```

That will cause screen number 1 to appear, and saved view number 12 will be the default view.

Unfortunately the only place to identify the view numbers currently, is in the MySQL database. Look in the `formulize_saved_views` table for the `sv_id` parameter.

# Troubleshooting Tips

## **I'm getting some strange database errors, any suggestions?**

If you are using MySQL 5, you may need to run in "traditional" mode. To set this mode, run the following SQL statement once:

```
SET @@global.sql_mode='';
```

## **Help, my page is blank, or not drawn completely on the screen!**

This kind of error is most often caused by having the memory limit and/or max execution time of PHP set too low. If you change these values in your PHP ini file, or through other means, it may solve the problem. On a Linux web server running Apache, you can often override them by putting a .htaccess file in the root folder of your website with the following lines in it:

```
php_value memory_limit 128M  
php_value max_execution_time 120
```

## **I've changed my derived value formula, but the actual values aren't changing. Why?**

As of Formulize 3.1, derived values are stored in the database, and they are normally only updated when an entry is saved. If you create a derived value formula, load a list of entries to inspect the values that are created, and then change the formula, the values will not normally be updated again until the entries are saved. This is a performance precaution, since updating derived values can take some time, depending on the formula.

If you are making changes to a derived value formula and need to force the derived values to be updated every time you reload the list of entries, so you can see if your formula is working, then change your URL so it has an extra parameter on the end like this:

```
http://www.yoursite.com/modules/formulize/index.php?fid=1&forceDerivedValueUpdate=1
```

When that special parameter is on the end, it will force the derived values to be updated using the latest formula.

## My {TODAY} search isn't working. Why?

If you are using a + or - number as part of your {TODAY} search, ie: {TODAY+30}, make sure the number is inside the { }. Although it does work in some places to type {TODAY}+30, that will not work in all cases, so you should always include the number inside the { }.

## I have marked some elements as “disabled” for some groups, but some of those people still see the elements. Why?

If a user is a member of a group for which the element is not disabled, then the element will not be disabled for that user. All the user's groups must be specified in order for the element to be disabled for that user. The logic for this feature may be inverted at some point in the future, so you would then specify only the groups for which the element should not be disabled.

## Search Terms

Since Formulize 3.0, you have a lot of options for search terms that you can type in the “quicksearch” boxes at the top of each column in the list of entries pages. Here is a brief summary:

### *The basics*

<b>This search term:</b>	<b>Means this:</b>	<b>It will find:</b>
orange	find “orange” by itself or inside something else	<b>orange</b> <b>orange</b> peels apples and <b>oranges</b>
=orange	find the exact text “orange”, no more, no less	<b>orange</b> (it won't find “oranges”)
!orange	find entries that do not have orange in them	red purple banana peels
!=orange	find entries that do not match the exact text “orange”	red apples and oranges orange peels

<b>This search term:</b>	<b>Means this:</b>	<b>It will find:</b>
15	find the text “15” by itself or inside something else	<b>15</b> baseballs <b>12715</b> <b>15</b>
=15	find the exact number 15	15
!15	find text that does not contain “15”	1848 3 strikes etc
!=15	find anything except the exact number 15	14 16 15 baseballs

### ***Numbers***

<b>This search term:</b>	<b>Means this:</b>	<b>It will find:</b>
>10	find numbers greater than 10	11, 12, 1485
>=1000	find numbers greater than or equal to 1000	1000, 1001, etc
<65	find numbers less than 65	64, 10, -401
<=65	find numbers less than or equal to 65	65, 64, etc

### ***More than one search on the same column***

<b>This search term:</b>	<b>Means this:</b>	<b>It will find:</b>
>10//<100	find numbers between 11 and 99	11, 12, etc
>10//<10	find entries that are both greater than 10 and less than 10	it will find nothing...the search condition is logically impossible

<b>This search term:</b>	<b>Means this:</b>	<b>It will find:</b>
santa/!monica	find entries that contain “santa” and not “monica”	<b>Santa</b> Claus <b>Santa</b> Cruz
>=2008-10-01//<=2008-10-31	find dates within October 2008	2008-10-01 2008-10-15
red//orange//blue	find those three values. You can have as many terms as you want in a single column.	Searches like this will usually only find matches if the field you're searching accepts multiple selections (ie: checkboxes), and red, orange and blue were three of the options.

### ***Wildcard terms {TODAY}, {USER} and {BLANK}***

<b>This search term:</b>	<b>Means this:</b>	<b>It will find:</b>
{TODAY}	find today's date	the text of today's date, in YYYY-mm-dd format
>={TODAY-30}	find dates anytime from the past 30 days into the future	you get the idea
>={TODAY+30}//<={TODAY}	find dates between 30 days ago and today	
>{TODAY+14}	Find dates more than 14 days in the future	
{USER}	Find entries that match the current user's full name, or if no full name is in their profile, then match on their username	
{BLANK}	Find entries that are blank or empty	

***“OR” searches on multiple columns***

<b>This search term:</b>	<b>Means this:</b>	<b>It will find:</b>	
<p><i>on the fruit column:</i> ORapples <i>and at the same time on the vegetable column:</i> ORcarrots</p>	<p>Find entries that match either apples in the fruit column, or carrots in the vegetable column (normally, search terms on multiple columns must all be matched)</p>	<b>Fruits</b>	<b>Vegetables</b>
		Apples	Spinach
		Oranges	Carrots

***Advanced developer-focused search terms***

<b>This search term:</b>	<b>Means this:</b>
{order}	Find entries that match whatever is in \$_POST['order'] or if that's empty, \$_GET['order']. This is meant for use in a complex application where a certain screen might need to show different things at different times. You can save one view with this kind of search term and then make some kind of architecture (with javascript maybe?) to populate \$_POST or \$_GET with the values you need at the right time.
!{order}	Find entries that do not match whatever is in \$_POST['order'] or \$_GET['order']
!orange!	Persist this search term even if this column is not included in the view. Certain columns are visible only to certain groups of users. You might want to use a search term on a certain column that only webmasters have access to, in order to limit the list of entries. Then you could publish that view to other users who do not have that column available, but the list of entries would still be limited by this search term.
!!monica//santa!	When persisting a search that includes multiple terms, put the !! at the very beginning and very end of the terms. Don't get confused by “not” operators (!) that may be part of the terms (as in “not monica” at the beginning of this set of terms).

# Relationships Between Forms

Formerly called Frameworks, a Relationship is simply a series of forms that have some kind of defined relationship. In database terms, it's a join. For instance, in a project management system there might be a client form, and a project form. Each project belongs to one client. So there's a one-to-many relationship between entries in the client form and entries in the project form. Furthermore, there is likely a special field in the project form, a linked selectbox, which points to the client form and contains a list of client names. Linked selectboxes are added to forms just like regular selectboxes, in the administration screens of Formulize. The difference is that instead of the options for the box being typed in manually, a link to an element in another form is specified.

Relationships are useful because sometimes you want to get a whole series of data (such as information about a client plus all their projects) and a Relationship makes it easy to gather that information.

## Creating Relationships

You create Relationships using the Relationships tab on the admin page for your application in Formulize. When you create a Relationship you choose two forms and add them to the Relationship. You can then specify what the nature of their relationship is (one-to-one, one-to-many) and what the key elements are that relate them to one another.

You also have the option of specifying whether the forms should be displayed together when users are entering data. If you specify that there is no "unified display" then the Relationship is simply an advanced way of referring to data in the forms. If you specify that the forms should have a "unified display", then in addition to the data in the forms being related according to the options you specified above, the forms can appear on screen together, one inside the other as a subform (use subform elements in the parent form, to control how the subform appears).

## Using and Accessing Relationships

### *Screens*

Relationships are used by screens. This is their primary purpose. When you make a screen, you can specify what the underlying data set is that the screen relies on. By default, it will be the single form that the screen is associated with. But you can instead tie the screen to any Relationship that the form is involved in. When you do this, that form, plus all the forms its related to in the Relationship, will be available to Formulize.

This means you can include any column/element from any form in the relationship, when you're working on list of entries screens. It also means that you can use subform elements on form screens to embed one form inside another.

## ***API***

You can also use Relationships through the API. Because they used to be called Frameworks, you generally use a parameter called the 'frid' or framework id. Specify a Relationship ID wherever the API asks for a frid.

For more details about the API, check out the PDF called *Using Formulize and Pageworks to Make Custom Applications* available from: <http://www.freeformsolutions.ca/formulize>

## ***URL***

Once you have a Relationship, you can access it simply by putting the Relationship ID and a form ID into the URL like this:

<http://www.yoursite.com/modules/formulize/index.php?frid=3&fid=7>

(Note the legacy frid parameter, framework ID, used to specify the relationship.)

That URL will call up Relationship number 3 with form number 7 as the primary form (mainform).

The primary form, or mainform, is simply the form that you are most interested in getting data from, with all other forms in the Relationship being secondary. For instance, if you wanted to see client data plus all the projects each client has, then the client form would be the mainform. If you wanted to see a list of projects, including client data for each project, then the project form would be the mainform. For more information, see *One to Many Relationships and Changing the Main Form: Reversing the Point of View* on page 18.

## **Where to find the frid**

To get the ID numbers, look in the URL of your browser when you are configuring a Relationship. The frid is included in the URL

## **Scenarios for using Relationships**

### ***Unified Data Viewing***

For instance, imagine a profile form that includes fields like name, age, etc. And imagine an activity log form that people use to record things they have done. The activity log might have fields like task name, description, etc.

The profile form would most likely be setup as a one-entry-per-user form. The activity log is obviously going to be a multiple-entry-per-user form (you would only have one profile for a person, but that person could enter many different activities). This means there's a one-to-many relationship between the forms.



Suppose you want to view all the activity logs someone has entered, plus their profile information. Perhaps you want to search for logs, based on information in a person's profile. This is where a Relationship can provide a way to join the data in the forms so you can view it all together. One "entry" in the Relationship would include a person's profile data, plus data from all their activities. This way you don't have to manually compare and relate information in two views from two forms. All data would be on screen together (and searchable together and so on).

To setup such a Relationship, you would select the two forms, specify the relationship as one-to-many, and specify that the link between the forms is the "user ID of the person who filled them in".

To call up the unified list of entries, you can make a new screen that is based on the Relationship. You could then add a link to that URL to a custom menu module such as iMenu.

### ***Unified Data Entry – Subforms***

Imagine a product list that includes fields like name, price, colour, etc. Imagine that in the data entry form for the product list, you want to include details about the parts that are used in making a product. There will be many parts for each product, and for each part you want to include the same information, such as part number, manufacturer, etc.

There is a one-to-many relationship between the products and the parts. Create a parts list form in addition to the product list form, and add a linked selectbox to the parts list, that refers to the name element in the product form. Then you can relate them in a Relationship based on that link. Turn on the "unified display" display option for this pair of forms in the Relationship, and you will be able to have the parts list embedded into the product list form. Users who go to the product list form will be able to add information about one or more parts used to create the product. All that data will be accessible as a single unit of information; the product data plus all the data about all the parts will be bound together.

Once you have two forms related this way, you can add the Subform element to the parent form. This lets you control where the special Subform features appear in the form. For instance, you might want the "parts list" info to appear right after the product name, so you would add the Subform element as the second element in the form.

You could then access the product list by creating a screen that is based on the Relationship instead of just one of the forms.

## ***One to Many Relationships and Changing the Main Form: Reversing the Point of View***

Each Relationship is simply a description of the links between forms. You must specify the "main form", or the starting point, that you want to use each time you work with the Relationship. In the case of the product list/parts list scenario above, the product list is the main form. Each "entry" in the Relationship would include the product info, plus info about all the parts related to that product. Here's how data from **two** entries in that Relationship would be arranged:

```
Product 1 data      Part 1 data
                   Part 2 data
                   Part 3 data
Product 2 data      Part 4 data
                   Part 5 data
                   Part 6 data
```

Suppose you wanted to view that information, but have it arranged by part, so there would be **six** entries, looking like this:

```
Part 1 data        Product 1 data
Part 2 data        Product 1 data
Part 3 data        Product 1 data
Part 4 data        Product 2 data
Part 5 data        Product 2 data
Part 6 data        Product 2 data
```

Achieving this is simple: whichever form the screen belongs to, that's the main form. If you create a screen on the other form, and use the same relationship as the basis for the screen, then the same data will be shown, just from the other form's point of view. If you are using the URL access method, just change the value of the "fid", or form ID.

### ***Unified Data - "one-to-one" Forms***

Suppose you have an inventory list. For each item in the inventory, there are certain common pieces of information, such as barcode, type of item, price, etc. But for each type of item, you want certain other fields to appear, and only for that type. For instance, you might only want to store information about the intended age groups for items that are toys. Essentially, part of the form should be common to all items in the inventory, and the other part of the form should be customized based on the kind of item this is.

You could create one common inventory form, and one custom inventory form for each type of item, and then create a Relationship for each pairing. The nature of the relationship would be one-to-one between each pair of forms. With the "unified display" option turned on, this would allow you to display "unique" inventory forms for each type of item, but all the common data would be stored in the database through a single common form, while the data unique to each type would be stored in each type's form and database table.