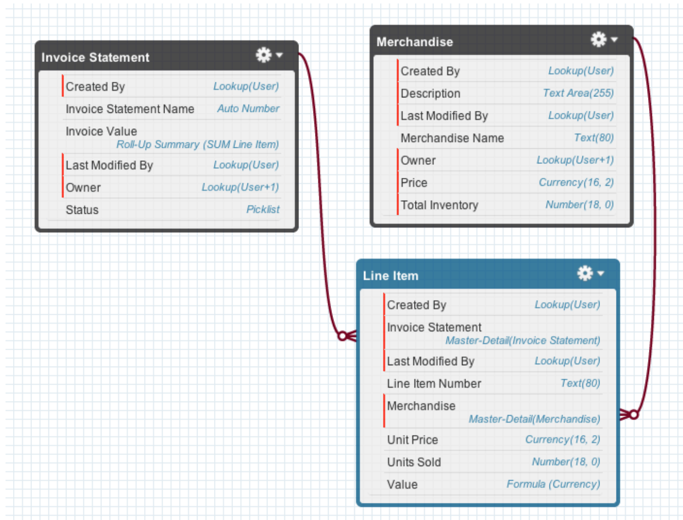
Schema Builder

Force.com Platform Workshop



Schema Builder

Table of Contents

Prerequisites

Force.com

Sign up for a free Force.com Developer Edition (DE) organization (org).

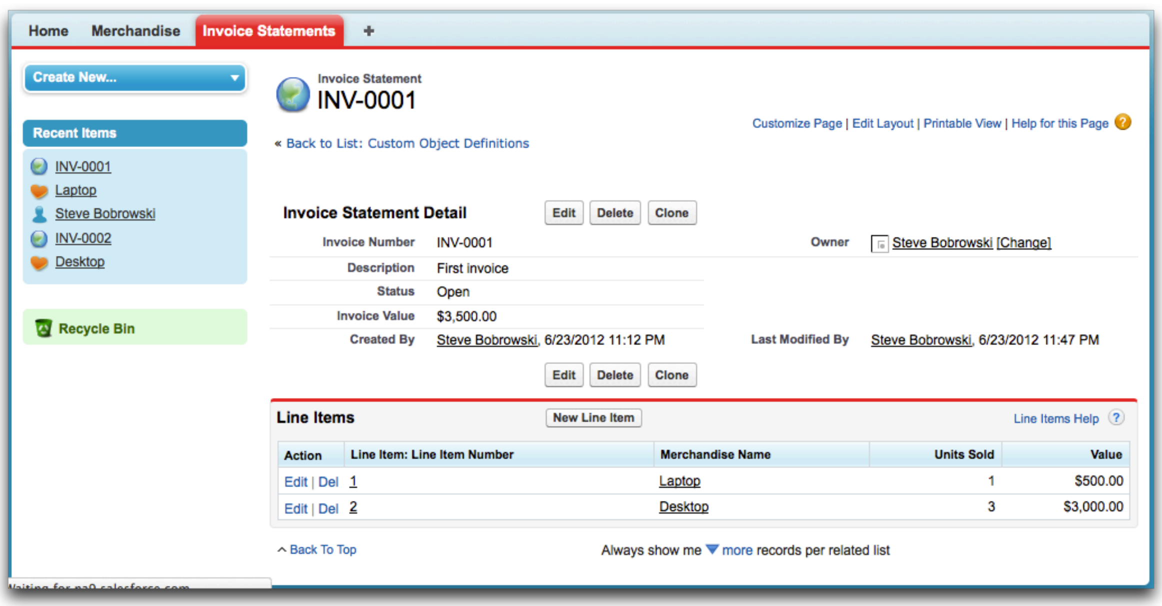
Building Apps in the Cloud

This tutorial teaches you how to use Force.com's declarative interface to create a highly functional app's data model, user interface, and basic business logic.

To build the data model, the tutorial introduces you to the Schema Builder, a tool for creating data models within the Force.com platform in a visual and user-friendly manner. You can use the Schema Builder to quickly create the data structures necessary to support your example Warehouse application and get an overview of the relationships among objects.

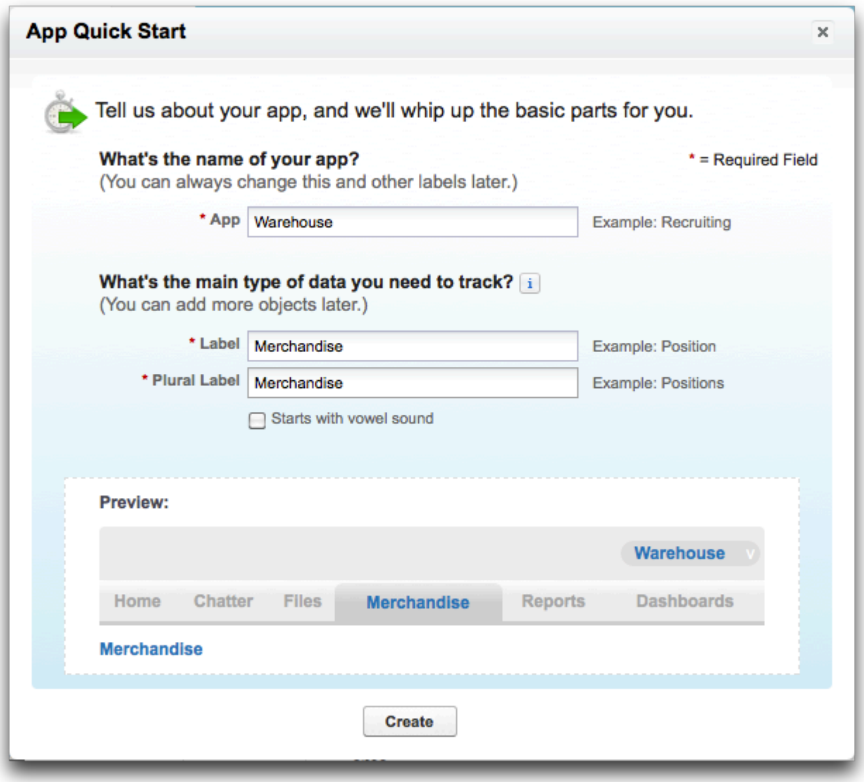
Along the way, the tutorial also shows you how to build a Web-based app related to your data objects, complete with tabs, page layouts, and data entry forms on the Force.com platform. Lessons in this section teach you some basic skills for customizing it.

Security is important for all types of apps, and Force.com has many security features that are easy to use. In this tutorial, you get some hands-on experience configuring the app so that users can work with the data they need.



Step 1: Create the Warehouse Application

To provide our data model with some context, you're going to create an application. An application helps define a set of elements such as custom objects and tabs. Before you jump into schema builder, take advantage of the *Application Quick Start* feature to get things rolling.



1. Launch your browser and go to https://login.salesforce.com.
2. Enter your username (in the form of an email address) and password.
3. Click **Your Name | Setup** in the upper right corner, and then click **Create | Apps** in the sidebar menu.
4. Notice that your DE org comes with many pre-built apps (e.g., Sales, Marketing).
5. Click **Quick Start**.
6. In the *Force.com Quick Start* overlay, enter the app and object details.

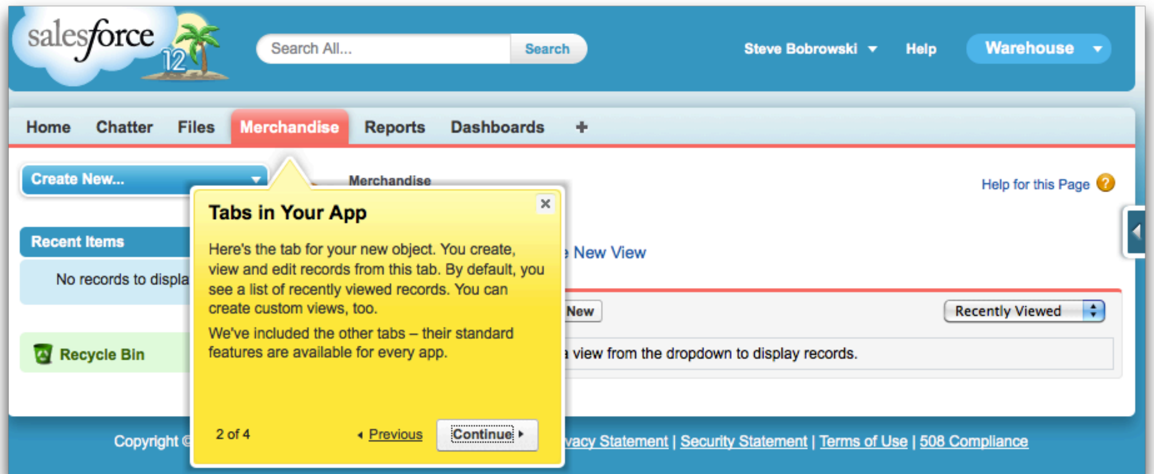
    - For the *App Label*, type `Warehouse`
    - For the *Plural Label*, type `Merchandise`.
    - For the *Singular Label*, type `Merchandise`.

6. Click **Create** to finish creating your new object.
7. Click **Go To My App** to see your new app.

Step 2: Tour Your New App

With just a few clicks of your mouse, you have a working Web-based app for managing Merchandise, albeit very simple. Nonetheless, take a look at your work.

1. Click **Start Tour** and follow along.
2. Get a feel for the structure of the app: tabs, page layouts, etc.
3. Feel free to click **New** and create one or two Merchandise objects (e.g., Laptop, Tablet).

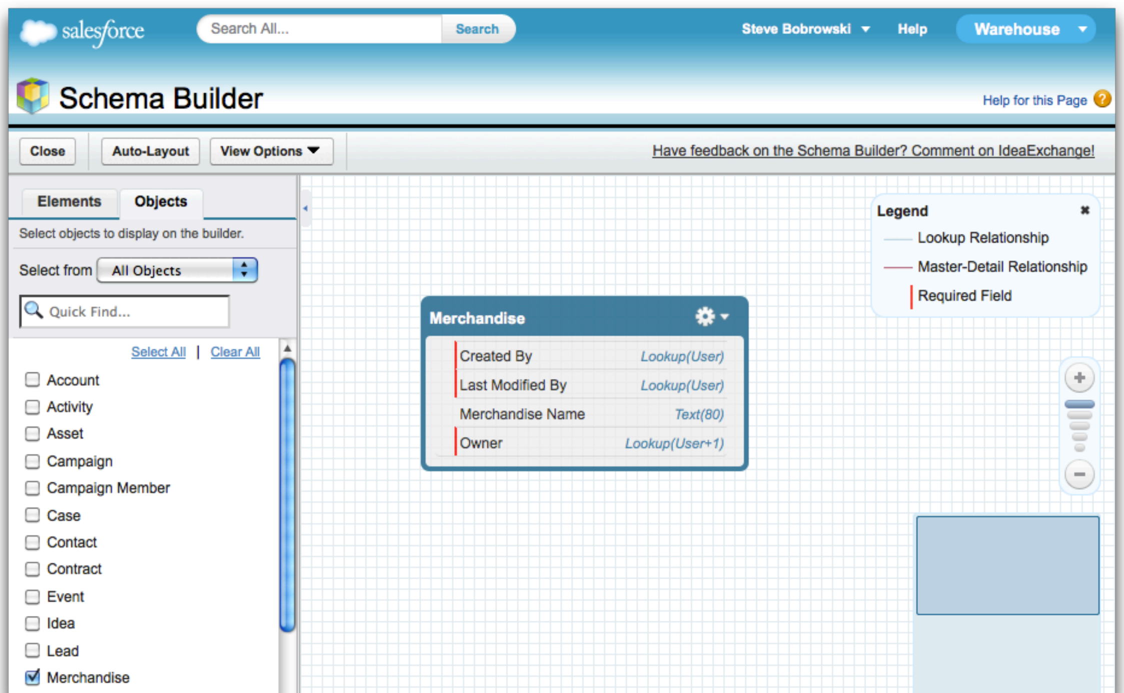


Step 3: Open Schema Builder

Next, enhance the basic Warehouse application so that you can track more data for the Merchandise object. Crack open the Schema Builder to define the details of your intended data model.

1. Go to **Your Name | Setup** in the upper right corner.
2. Click **App Setup | Schema Builder.**
3. Click the **Clear All** link above the list of object types.
4. Click the **Merchandise** checkbox to see the object definition.
5. If necessary, click **Auto Layout** to position the object definition.

Notice that Schema Builder is a visual tool, much like a traditional entity-relationship (ER) diagramming tool.



Step 4: Add More Fields to Merchandise

The Quick Start feature created the basics of the Merchandise object for your application: the object, standard fields, the app, and a tab. Now use Schema Builder to extend that definition.

First, add a *Description* field to the object:

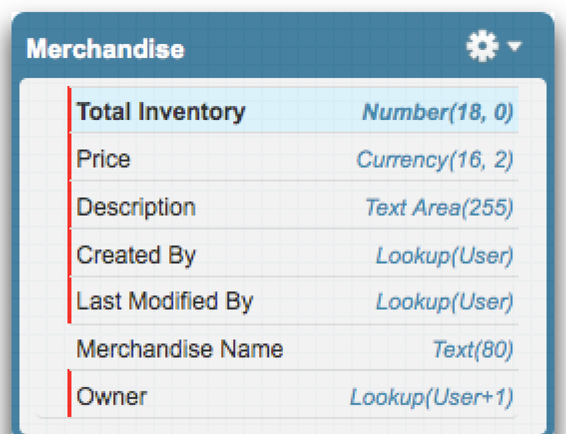
1. Click the **Elements** tab.
2. Drag and drop a *Text Area* element to the *Merchandise* object.
3. Enter `Description` for *Field Label*, and hit `Tab` to see it filled for *Field Name*.
4. Select the **Required** checkbox.
5. Click **Save**.

Next, create a *Price* field.

1. Drag and drop a *Currency* element to the *Merchandise* object.
2. Enter `Price` for *Field Label*, and hit **Tab** to see it filled for *Field Name*.
3. For *Length* enter `16` and for *Decimal Places* enter `2`.
4. Select the **Required** checkbox.
5. Click **Save**.

Finally, create a *Total Inventory* field.

1. Drag and drop a *Number* element to the *Merchandise* object.
2. Enter `Total Inventory` for the *Field Label*, and hit **Tab** to see it filled for *Field Name*.
3. Select the **Required** checkbox.
4. Click **Save**.



Step 5: Examine the Enhanced Merchandise Object

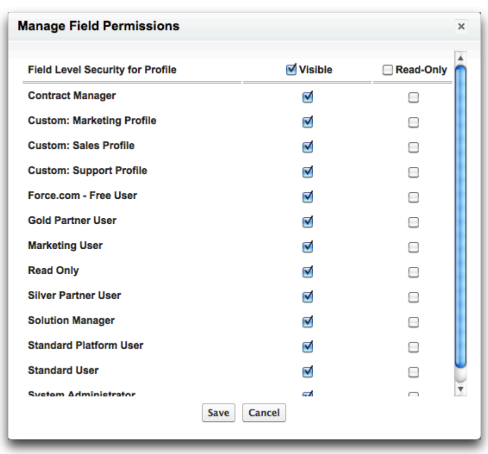
Go back to the Warehouse app to see if the new fields appear in the page layout for Merchandise.

1. Click **Close** (upper left).
2. Click the *Merchandise* tab.
3. Create a new Merchandise object, or edit an existing object.
4. Notice that the new fields appear.

Step 6: Make the New Fields Visible to All Users

You can see the fields you just added because you are a member of the all-powerful System Administrator security profile. However, other types of users will not be as privileged. To introduce yourself to a bit of Force.com security, complete the following steps to make these new fields visible to other types of users.

1. Go to **Your Name** (upper right corner), but then use your alternate mouse button to click **Setup** and choose **Open in a New Tab** so that you have two browser tabs open: one with the app, and another with App Setup.
2. In the new App Setup browser tab, click **Schema Builder**.
3. In the *Merchandise* object definition, alternate click on **Price**.
4. Click **Manage Field Permissions**.
5. Select the checkbox next to *Visible*.
6. Click **Save**.
7. Repeat steps 3-6 for *Description* and *Total Inventory*.



Now users with any security profile can view the new fields in the Merchandise object.

Step 7: Create an Invoice Statement Object

Next, extend the database schema to support the ability to maintain *Invoice Statements*.

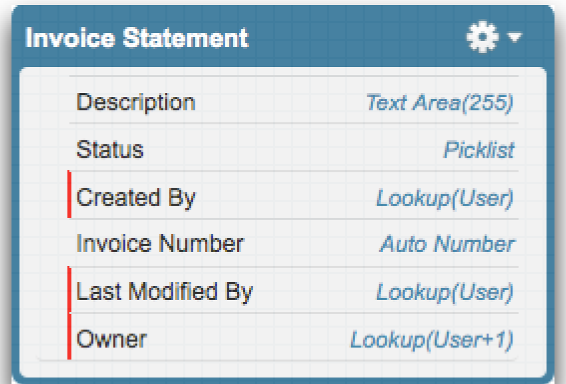
1. In the Schema Builder browser tab, drag and drop a *Object* element next to the *Merchandise* object definition.
2. For *Label* enter `Invoice Statement`.
3. For *Plural Label* enter `Invoice Statements`.
4. Select **Starts with vowel** sound for *Starts With*.
5. For *Record Name*, enter `Invoice Number`.
6. For *Data Type*, select **Auto Number**.
7. For *Display Format*, enter `INV-{0000}`.
8. For *Starting Number*, enter `1`.
9. Select the **Allow Reports** checkbox.
10. Click **Save**.

Next, add a *Status* field to the new Invoice Statement object.

1. Drag and drop a *Picklist* element to the *Invoice Statement* object definition.
2. For *Field Label*, enter Status.
3. Hit **Tab** to see `Status` filled for *Field Name*.
4. For *Values*, enter:
   ```
   Open
   Closed
   Negotiating
   Pending
   ```
5. Select the **Use first value as default value** checkbox.
6. Click **Save**.
7. In the *Invoice Statement* object definition, right click on *Status*.
8. Select **Manage Field Permissions**.
9. Select the **checkbox** next to *Visible*.
10. Select the **checkbox** next to *Read Only*.
11. Click **Save**.

Complete the new Invoice Statement object with a new *Description* field.

1. Drag and drop a *Text Area* to the *Invoice Statement* object definition.
2. Enter `Description` for *Field Label*, and hit **Tab** to see it filled for *Field Name*.
3. Click **Save**.
4. In the *Invoice Statement* object definition, right click on **Description**.
5. Select **Manage Field Permissions**.
6. Select the **checkbox** next to *Visible*.
7. Click **Save**.



Step 8: Create a Line Item Object

Next, add a *Line Item* object to keep track of line items for each *Invoice Statement*.

1. Drag and drop an *Object* element to the Schema Builder layout, near *Merchandise*.
2. For *Field Label*, enter `Line Item`.
3. For *Plural Label*, enter `Line Items`.
4. For *Record Name*, enter `Line Item Number`.
5. Leave *Data Type* as **Text**.
6. Select the **Allow Reports** checkbox.
7. Click **Save**.

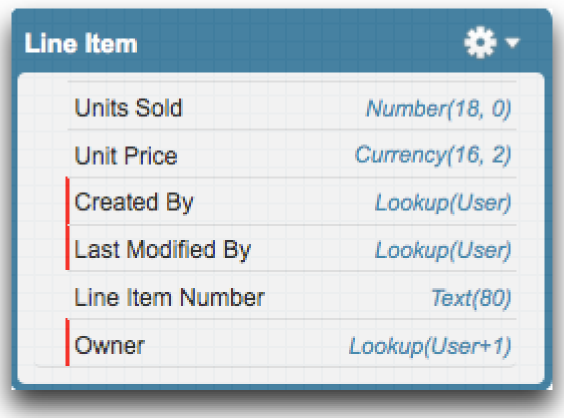
And now for some fields. First a *Unit Price* field.

1. Drag and drop a *Currency* element to the *Line Item* object definition.
2. For *Field Label*, enter `Unit Price`.
3. Hit **Tab** and see `Unit Price` entered for *Field Name*.
4. For *Length*, enter `16`.
5. For *Decimal Places*, enter `2`.
6. Click **Save**.
7. In the *Line Item* object definition, alternate click on *Unit Price*.
8. Select **Manage Field Permissions**.
9. Select the **checkbox** next to *Visible*.
10. Select the **checkbox** next to *Read Only*.

11. Click **Save**.

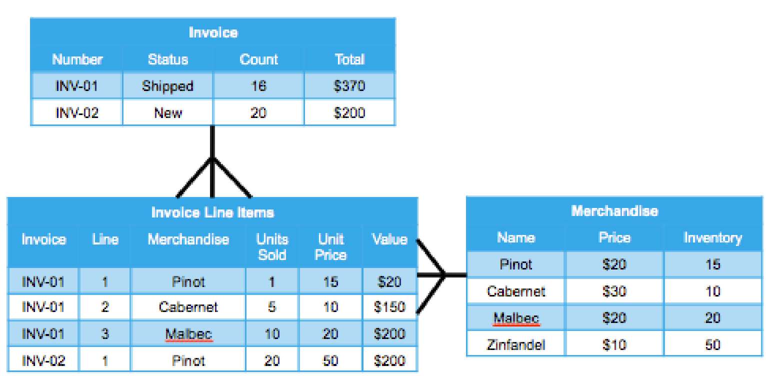
And finally, add a *Units Sold* field.

12. Drag and drop a *Number* element to the Line Item object definition.
13. For *Field Label*, enter `Units Sold`.
14. Hit **Tab** and see `Units Sold` entered for *Field Name*.
15. Click **Save**.
16. In the *Line Item* object definition, alternate click on *Units Sold*.
17. Select **Manage Field Permissions**.
18. Select the **checkbox** next to *Visible*.
19. Click **Save**.



Step 9: Relate the Objects

All of the *Warehouse* app objects are in place, but have no association with each other. The *Line Item* object should relate to the *Merchandise* object so that they can accurately depict what is in your warehouse. The *Line Item* should also relate to the *Invoice Statement* to create intelligent invoices for your customers.



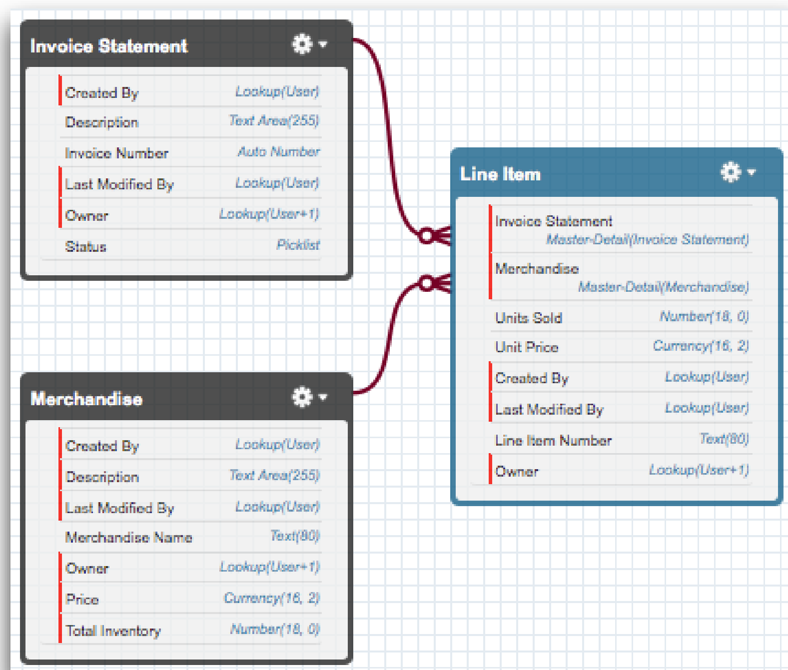
First, relate *Line Item* with *Merchandise*.

1. Drag and drop a *Master-Detail Relationship* element to the *Line Item* object definition.
2. For *Field Label*, enter `Merchandise`.

3. Hit **Tab** and see `Merchandise` entered for *Field Name*.
4. Notice the default values for *Child Relationship Name* and *Related List Label*.
5. For *Related To*, select **Merchandise** from the drop down.
6. Click **Save**.
7. In the *Line Item* object, alternate click on **Merchandise**.
8. Select **Manage Field Permissions**.
9. Select the **checkbox** next to *Visible*.
10. Click **Save**.

Next, relate *Line Item* with *Merchandise*.

1. Drag and drop a *Master-Detail Relationship* element to the *Line Item* object definition.
2. For *Field Label*, enter `Invoice Statement`.
3. Hit **Tab** and see `Invoice_Statement` entered for *Field Name*.
4. Notice the default values for *Child Relationship Name* and *Related List Label*.
5. For *Related To*, select **Invoice Statement** from the drop down.
6. Click **Save**.
7. In the *Line Item* object, alternate click on **Invoice Statement**.
8. Select **Manage Field Permissions**.
9. Select the **checkbox** next to *Visible*.
10. Click **Save**.



Step 10: Try the Warehouse App

Now that you have all the new objects in place, including their relationships, switch back to your other browser tab and refresh the Warehouse app to view what's new.

Wait a minute -- nothing's new! Why not?

That's because, when you add objects to your data model via Schema Builder, some things, such as adding tabs to the app and configuring default page layouts, require a bit of extra work.

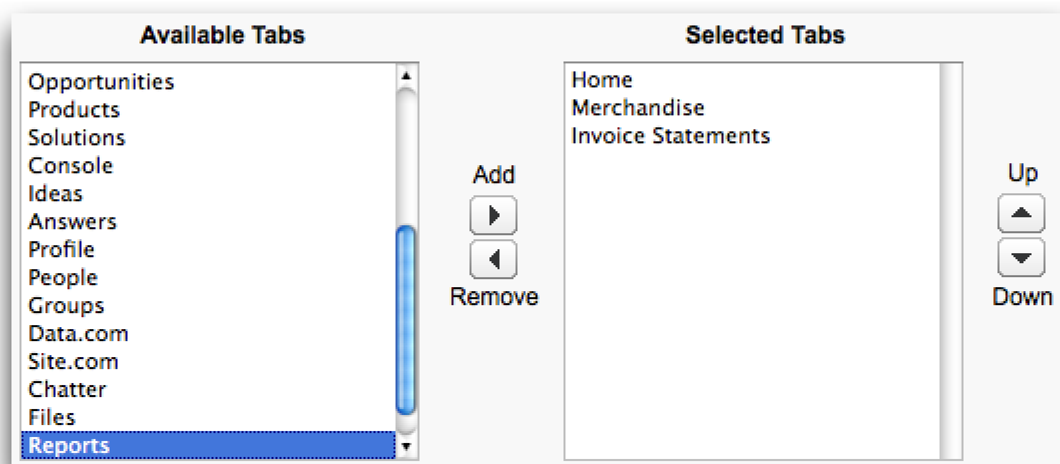
Step 11: Add a New Tab to the App

First, add a new tab for Invoice Statements to the Warehouse app.

1. Switch back to the browser tab with Schema Builder.
2. Click **Close**.
3. Click **Create | Tabs**.
4. Click **New** in *Custom Object Tabs*.
5. For *Object*, select **Invoice Statement**.
6. Pick any *Tab Style*.
7. Click **Next | Next**, include the new tab only on the *Warehouse* app, then click **Save**.

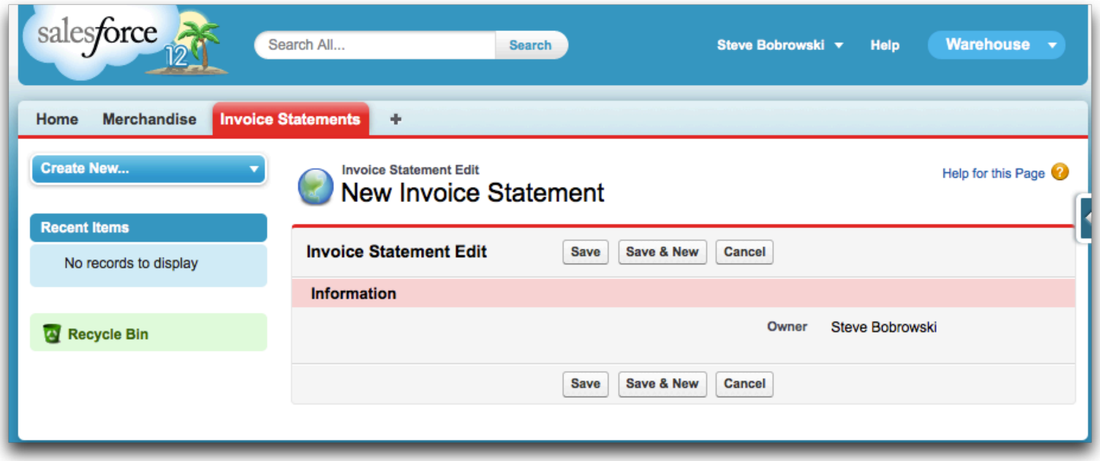
Notice the new tab is out of position, on the far right. To fix this and remove other tabs from the app:

1. Click **Create | Apps**.
2. Click **Edit** next to *Warehouse*.
3. Configure the selected tabs as in the following screen, then click **Save**.



Step 12: Modify the Invoice Statements Page Layout

Now, switch back to the browser tab with the Warehouse app, refresh the page, click the new **Invoice Statements** tab, then click **New**. When you do, notice that the page layout isn't very functional -- it doesn't let you enter any data!



Again, there's a bit of extra work necessary to configure page layouts for new objects that you create with Schema Builder. It's an easy fix.

1. Switch back to the browser tab with App Setup.
2. Click **Create | Objects**.
3. Click **Invoice Statement**.
4. In the *Page Layouts* section, **Edit** the *Invoice Statement Layout*.

Notice that the current page layout does not include any of the custom fields that you created earlier, and it doesn't have any way to access related Line Items.

Modify the page layout as follows:

1. Drag and drop the *Description* and *Status* fields to the *Information* section, right underneath the *Invoice Number* field.
2. Click **Related Lists**, then drag and drop *Line Items* to the form.
3. Click **Save**, then **Yes**.

Now try out the modified page layout for Invoice Statements.

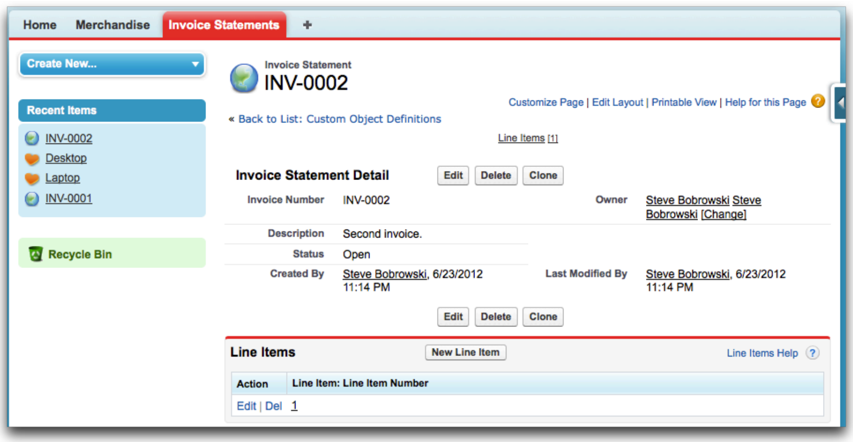
1. Switch back to the browser tab with the Warehouse app.
2. Click **Invoice Statements | New**.
3. Enter a *Description* (e.g., `First Invoice`).
4. Click **Save**.
5. In the *Line Items* section, click **New Line Item**.
6. For *Line Item Number*, enter `1`.
7. For *Merchandise*, enter the first few characters of a valid *Merchandise* object.
8. Click **Save**.

Note: Notice for Line Items, there's no way to enter Units Sold and Unit Price. That's because you need to edit the Line Items page layout, similar to your work with Invoice Statements. To update those layouts, follow these steps:

1. On the Schema Builder layout, right click on the gear next to **Merchandise**.
2. Click **View Page Layouts**.
3. On the newly opened tab or window, click **Edit** next to *Merchandise Layout*.

4. Drag and drop *Description, Price and Total Inventory* to the layout below.
5. Click **Save.**
6. On the Schema Builder layout, right click on the gear next to **Line Item**.
7. Click **View Page Layouts**.
8. On the newly opened tab or window, click **Edit** next to *Line Item Layout*.
9. Drag and drop *Unit Price, Units Sold, Merchandise, Invoice Statement and Value* to the layout below.
10. Click **Save.**

Congratulations, you now have a working app that can track Merchandise and related Invoices. And all without installing and configuring any software or coding.



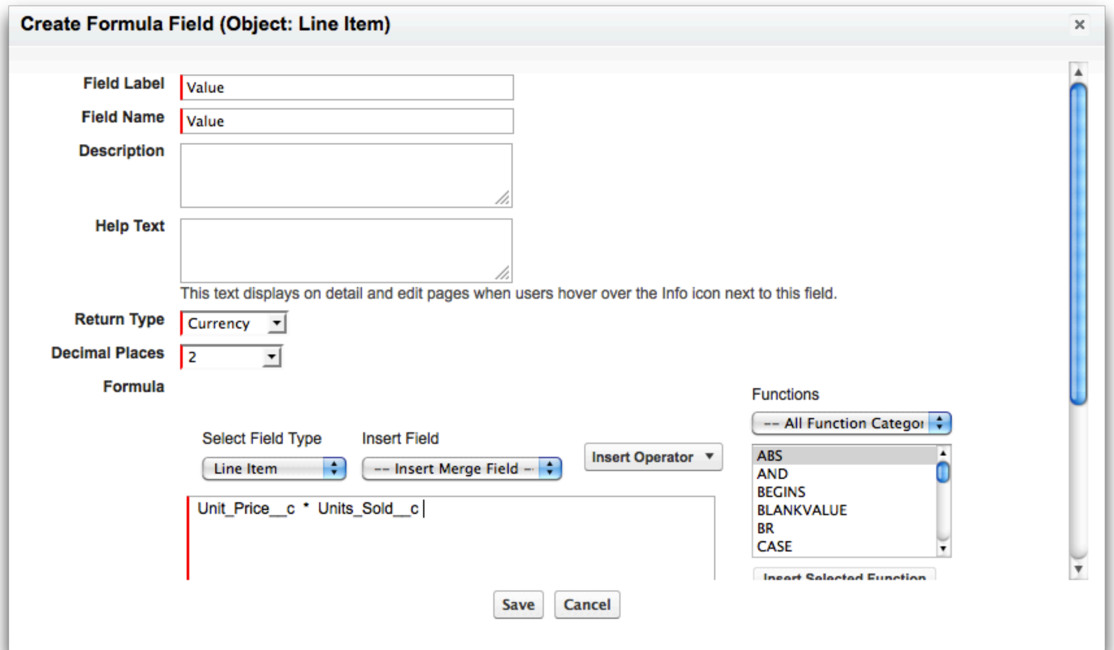
To complete the app, now add some basic business logic.

Step 13: Create a Formula Field

Formula fields allow for real time representations of data based on existing data points. In this use case, add a formula field to calculate the value of a line item by multiplying the price with the amount sold, ensuring that the value is always current even after the data in the fields change.

1. Switch back to the browser tab with Schema Builder.
2. Drag and drop a *Formula* element to the *Line Item* object definition.
3. For *Field Label*, enter `Value`.
4. Hit **Tab** and see `Value` entered for *Field Name*.
5. For *Return Type*, select **Currency** from the drop down.
6. For *Decimal Place*, select `2`.
7. In the *Insert Merge Field* drop down, select **Unit Price**. `Unit_Price__c` should appear in the text box.
8. Click the **Insert Operator** drop down and select **Multiply**.

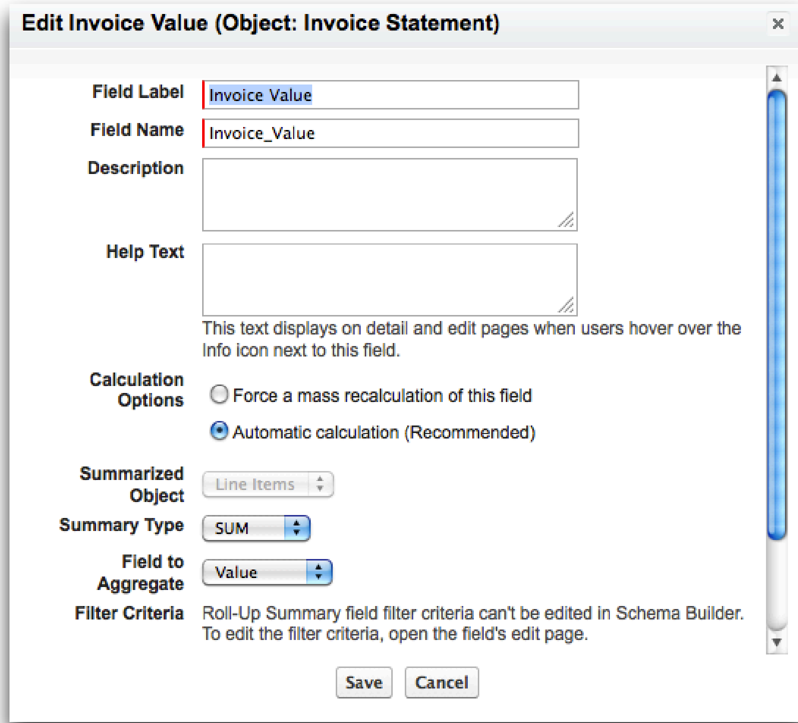
9. In the *Insert Merge Field* drop down, select **Units Sold**.
10. Confirm that the formula is (see below): `Unit_Price__c * Units_Sold__c`
11. Click **Save**.
12. In the *Line Item* object, alternate click on **Value**.
13. Select **Manage Field Permissions**.
14. Select the **checkbox** next to *Visible*.
15. Click **Save**.



Step 14: Create a Roll-Up Summary Field

Master-detail relationships allow for summary fields, which aggregate data from a child relationship. To make your invoice statements accurate in real-time, add a roll up summary field that represents the sum of the Values of the child line items.

1. Drag and drop a *Roll-Up Summary* element to the Invoice Statement object definition.
2. For *Field Label*, enter `Invoice Value`.
3. Hit **Tab** and see `Invoice_Value` entered for *Field Name*.
4. For *Summarized Object*, select **Line Items** from the drop down.
5. For *Summary Type*, select **SUM**.
6. For *Field to Aggregrate*, select **Value**.
7. Click **Save**.
8. In the *Line Item* object, alternate click on **Invoice Value**.
9. Select *Manage Field Permissions*.
10. Select the **checkbox** next to *Visible*.
11. Click **Save**.

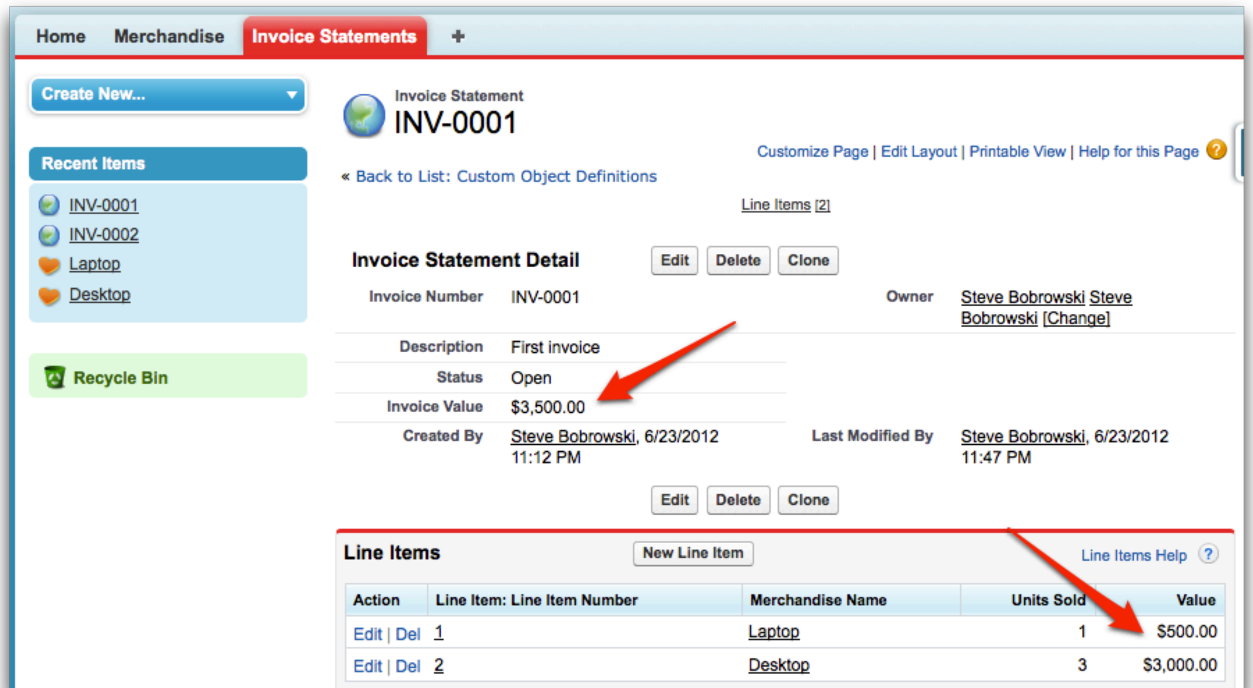

Edit Invoice Value (Object: Invoice Statement)
Field Label
Invoice Value
Field Name
Invoice_Value
Description
Help Text
This text displays on detail and edit pages when users hover over the Info icon next to this field.
Calculation Options
Force a mass recalculation of this field
Automatic calculation (Recommended)
Summarized Object
Line Items
Summary Type
SUM
Field to Aggregate
Value
Filter Criteria
Roll-Up Summary field filter criteria can't be edited in Schema Builder. To edit the filter criteria, open the field's edit page.
Save
Cancel

Step 15: Modify Page Layouts for the New Fields

Before you can view and test out the new formula and roll-up summary fields, you need to do some minor work on the page layouts for Invoice Statements and Merchandise. Here's a quick summary of what you need to do: Return to App Setup, edit the *Invoice Statements* page layout:

- add the new roll-up summary field Invoice Value
- edit the *Line Items* related list to display the new *Value* field (and perhaps others)

Try to figure out the exact steps yourself. When you are done, an Invoice Statement display should look similar to the following screen.



Summary

You have just created a working data model right from your web browser which is ready to start creating your applications. You've adjusted the layouts you'll see from the interface created within the Force.com platform for viewing, creating, editing and deleting records in your browser. You have added some simple business logic to make the data more intelligent.

There is a lot more here that you accomplished that can't see just from creating some records in the browser, however. You have created a fully functional REST API. You have extended the functionality of Apex and Visualforce - the programmatic and interface language of the Force.com platform. You have created a data model which may be access by mobile devices or third party legacy system. You did this because by defining a data model, you are also extending the baseline features of the Force.com platform to understand the kind of data you need for your applications.

Additional Reading

The Force.com workbook (starting with Tutorial #3) includes tutorials on creating validation rules, workflows and approvals. With these, you can further define business logic without programming code. The Force.com workbook can be found on the Workbook home page: http://developer.force.com/workbooks