© Copyright IBM Corporation 2011 All rights reserved

IBM WebSphere® Commerce V7 Feature Pack 2 – Lab exercise

Search customization

© Copyright IBM Corporation 2011. All rights reserved

What this exercise is about

In this tutorial, you will learn how to add a column from the base WebSphere Commerce schema into the Solr index. The sample index does not include every possible field from the base schema and which columns you index will depend on your storefront requirements.

In this sample scenario, you are storing the number of shopper points associated with each product in the custom field CATENTRY.FIELD1. You want to add this field to the Solr index and then add it as a facet in the Madisons storefront. This will allow shoppers to navigate by the number of shopper points they will receive for purchasing products.

This tutorial should take approximately 60 min to complete.

What you should be able to do

After completing this exercise, you should be able to:

- Extend the Solr index
- Make an index field facetable
- Customize the storefront to display a new facet

Introduction

The following naming conventions are used in the exercises:

Reference Variable	Description
<WCDE_INSTALL_DIR>	WebSphere Commerce Developer installation directory
<WC_HOST>	Hostname for WebSphere Commerce. For WebSphere Commerce Developer you can use **localhost**.
<master_catalog_id>	The master catalog for the store to be indexed for search

Requirements

Before beginning this lab, ensure you have:

- Installed WebSphere Commerce V7 Fix Pack 2
- Installed WebSphere Commerce V7 Feature Pack 2
- Completed WebSphere Commerce search configuration

© Copyright IBM Corporation 2011. All rights reserved

Part 1: **Extend the Solr schema**

In this part of the lab, you will add field1 to the Solr schema.

____ 1. Using Windows Explorer, navigate to:

<WCDE>\search\solr\home\MC_<master_catalog_id>\en_US\CatalogEntry\conf.

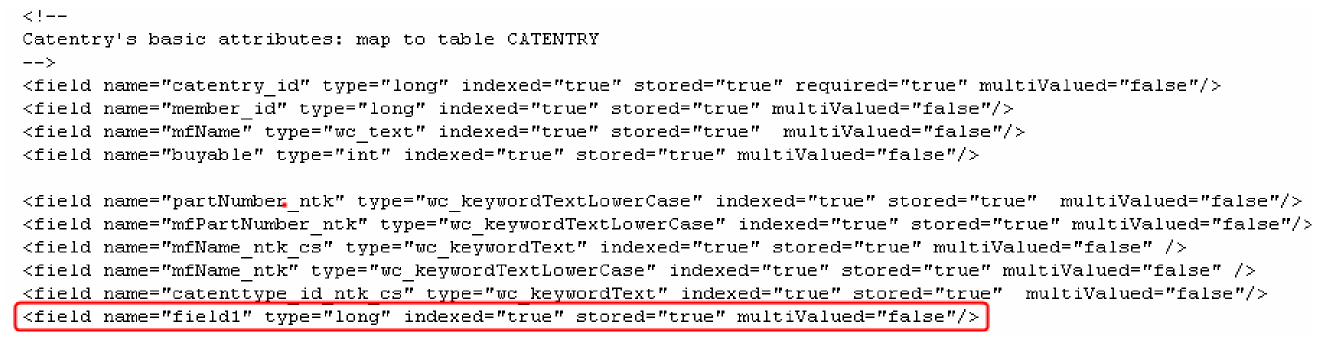
____ 2. Open the file **schema.xml** in a text editor**.**

____ 3. Search for the string "**Catentry's basic attributes**". This will bring you to the section of the file where the index fields are defined.

____ 4. The first block of index fields represent data from the CATENTRY table. Since the field1 value is also stored in the CATENTRY table, you need to add the new index field here.

__ a. At the bottom of the CATENTRY fields add the line:

```
<field name="field1" type="long" indexed="true" stored="true"
multiValued="false"/>
```



```
<!--
Catentry's basic attributes: map to table CATENTRY
-->
<field name="catentry_id" type="long" indexed="true" stored="true" required="true" multiValued="false"/>
<field name="member_id" type="long" indexed="true" stored="true" multiValued="false"/>
<field name="mfName" type="wc_text" indexed="true" stored="true"  multiValued="false"/>
<field name="buyable" type="int" indexed="true" stored="true" multiValued="false"/>

<field name="partNumber_ntk" type="wc_keywordTextLowerCase" indexed="true" stored="true"  multiValued="false"/>
<field name="mfPartNumber_ntk" type="wc_keywordTextLowerCase" indexed="true" stored="true" multiValued="false"/>
<field name="mfName_ntk_cs" type="wc_keywordText" indexed="true" stored="true" multiValued="false" />
<field name="mfName_ntk" type="wc_keywordTextLowerCase" indexed="true" stored="true" multiValued="false" />
<field name="catenttype_id_ntk_cs" type="wc_keywordText" indexed="true" stored="true"  multiValued="false"/>
<field name="field1" type="long" indexed="true" stored="true" multiValued="false"/>
```

____ 5. **Save** and close the file.

© Copyright IBM Corporation 2011. All rights reserved

Part 2: **Extend the queries that populate the schema**

In this part of the lab, you will update the queries that are used to populate the Solr index from the WebSphere Commerce database tables. Each query needs to retrieve the data stored in field1 of the CATENTRY table.

____ 1. If you have closed Windows Explorer, open it again and navigate back to the same directory:

<WCDE>\search\solr\home\MC_<master_catalog_id>\en_US\CatalogEntry\conf.

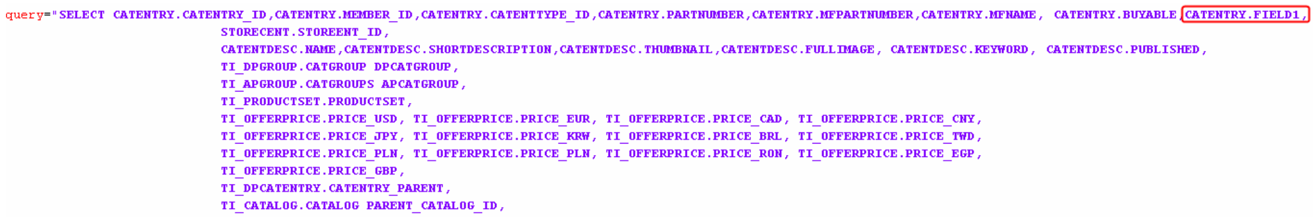
____ 2. Open the file **wc-data-config.xml** in a text editor**.**

____ 3. The wc-data-config.xml file loads catalog entry data in three separate parts. Product data is loaded first. This is followed by bundle data and finally all other catalog entry types. For each of the three sections there is a query for populating the full index and another query for delta updates. For each section (product, bundle and other), you need to update the two queries and then add a new field element to assign the data from the WebSphere Commerce database to the field1 field in the Solr index. All together there are nine updates you need to make to this file.

__ a. Update the product queries and assignment.

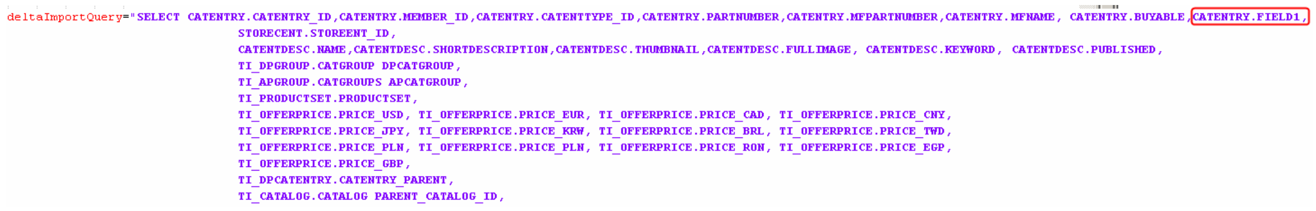
1) Search for the string **buyable** from the top of the file. This is another column in the CATENTRY table that is already included in the query.

2) The first place you find the string is in the main product query. Add **CATENTRY.FIELD1** to the SELECT statement immediately following CATENTRY.BUYABLE. Your updated query should look like the screen capture below.



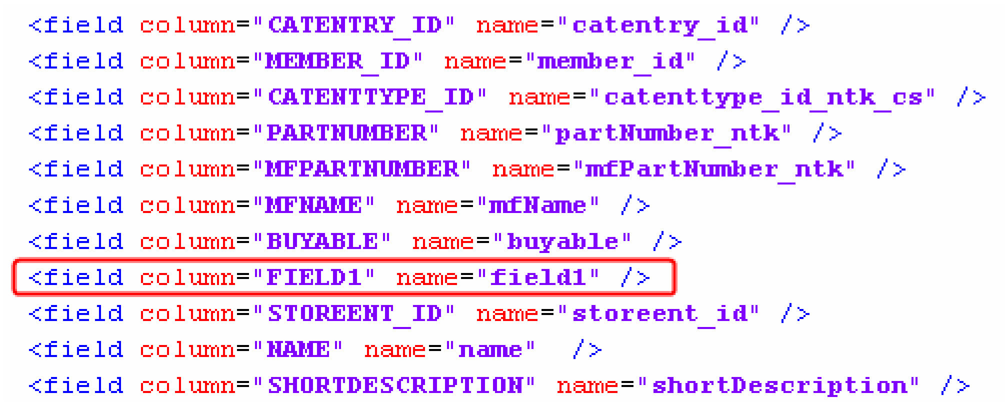
```
query="SELECT CATENTRY.CATENTRY_ID,CATENTRY.MEMBER_ID,CATENTRY.CATENTTYPE_ID,CATENTRY.PARTNUMBER,CATENTRY.MFPARTNUMBER,CATENTRY.MFNAME, CATENTRY.BUYABLE,CATENTRY.FIELD1,
                                STORECENT.STOREENT_ID,
                                CATENTDESC.NAME,CATENTDESC.SHORTDESCRIPTION,CATENTDESC.THUMBNAIL,CATENTDESC.FULLIMAGE, CATENTDESC.KEYWORD, CATENTDESC.PUBLISHED,
                                TI_DPGROUP.CATGROUP DPCATGROUP,
                                TI_APGROUP.CATGROUPS APCATGROUP,
                                TI_PRODUCTSET.PRODUCTSET,
                                TI_OFFERPRICE.PRICE_USD, TI_OFFERPRICE.PRICE_EUR, TI_OFFERPRICE.PRICE_CAD, TI_OFFERPRICE.PRICE_CNY,
                                TI_OFFERPRICE.PRICE_JPY, TI_OFFERPRICE.PRICE_KRW, TI_OFFERPRICE.PRICE_BRL, TI_OFFERPRICE.PRICE_TWD,
                                TI_OFFERPRICE.PRICE_PLN, TI_OFFERPRICE.PRICE_PLN, TI_OFFERPRICE.PRICE_RON, TI_OFFERPRICE.PRICE_EGP,
                                TI_OFFERPRICE.PRICE_GBP,
                                TI_DPCATENTRY.CATENTRY_PARENT,
                                TI_CATALOG.CATALOG PARENT_CATALOG_ID,
```

3) Search for **buyable** again. The second place you find the string is the delta index query. Make the same update to this query as you did to the previous one.



```
deltaImportQuery="SELECT CATENTRY.CATENTRY_ID,CATENTRY.MEMBER_ID,CATENTRY.CATENTTYPE_ID,CATENTRY.PARTNUMBER,CATENTRY.MFPARTNUMBER,CATENTRY.MFNAME, CATENTRY.BUYABLE,CATENTRY.FIELD1,
                                STORECENT.STOREENT_ID,
                                CATENTDESC.NAME,CATENTDESC.SHORTDESCRIPTION,CATENTDESC.THUMBNAIL,CATENTDESC.FULLIMAGE, CATENTDESC.KEYWORD, CATENTDESC.PUBLISHED,
                                TI_DPGROUP.CATGROUP DPCATGROUP,
                                TI_APGROUP.CATGROUPS APCATGROUP,
                                TI_PRODUCTSET.PRODUCTSET,
                                TI_OFFERPRICE.PRICE_USD, TI_OFFERPRICE.PRICE_EUR, TI_OFFERPRICE.PRICE_CAD, TI_OFFERPRICE.PRICE_CNY,
                                TI_OFFERPRICE.PRICE_JPY, TI_OFFERPRICE.PRICE_KRW, TI_OFFERPRICE.PRICE_BRL, TI_OFFERPRICE.PRICE_TWD,
                                TI_OFFERPRICE.PRICE_PLN, TI_OFFERPRICE.PRICE_PLN, TI_OFFERPRICE.PRICE_RON, TI_OFFERPRICE.PRICE_EGP,
                                TI_OFFERPRICE.PRICE_GBP,
                                TI_DPCATENTRY.CATENTRY_PARENT,
                                TI_CATALOG.CATALOG PARENT_CATALOG_ID,
```

4) Search for **buyable** once more. This brings you to the section where the data retrieved from the WebSphere Commerce database is assigned to the Solr index fields. Add an element under buyable to map the FIELD1 database column to the field1 index field. Your updated file should look like the screen capture below.



```
<field column="CATENTRY_ID" name="catentry_id" />
<field column="MEMBER_ID" name="member_id" />
<field column="CATENTTYPE_ID" name="catenttype_id_ntk_cs" />
<field column="PARTNUMBER" name="partNumber_ntk" />
<field column="MFPARTNUMBER" name="mfPartNumber_ntk" />
<field column="MFNAME" name="mfName" />
<field column="BUYABLE" name="buyable" />
<field column="FIELD1" name="field1" />
<field column="STOREENT_ID" name="storeent_id" />
<field column="NAME" name="name"  />
<field column="SHORTDESCRIPTION" name="shortDescription" />
```

__ b. Update the bundle queries and field assignment.

1) Repeat the process in step a. for the bundle section of the file

__ c. Update the other queries and field assignment.

1) Repeat the process in step a. for the final section of the file.

____ 4. **Save** and close the file.

© Copyright IBM Corporation 2011. All rights reserved

Part 3: Make field1 a product facet

In this part of the lab, you will update the configuration table that defines which fields in the index are used as facets in the storefront.

____ 1. Open WebSphere Commerce Developer and start your test server.

____ 2. Open the hints and tips page at **<WCDE>\hintsandtips.html.**

____ 3. Launch the **Database access JSP** using the link provided in the **Useful URLs** section.

____ 4. Update the SRCHATTR table. Each row in this table represents the logical name of a catalog attribute that is defined in the search schema.

__ a. Enter the SQL below to view the contents of the table.

```
select * from srchattr;
```

__ b. Enter the SQL below to add field1 to the table. You can use any available primary key, the value 101 is used here as an example. The value of ‘0’ for INDEXSCOPE indicates that this is a site-wide field.

```
insert into srchattr (srchattr_id, indexscope, indextype, identifier)
values (101, '0', 'CatalogEntry','_cat.Field1');
```

____ 5. Update the SRCHATTRPROP table. This table stores information about which fields are defined as facets.

__ a. Enter the SQL below to view the contents of the table.

```
select * from srchattrprop;
```

__ b. Enter the SQL below to define field1 as a facet. If you used a different primary key in step 3b above, use that same key in place of 101 here.

```
insert into srchattrprop (srchattr_id, propertyname, propertyvalue)
values (101,'facet','field1');
```

© Copyright IBM Corporation 2011. All rights reserved

Part 4: Add the facet to the storefront

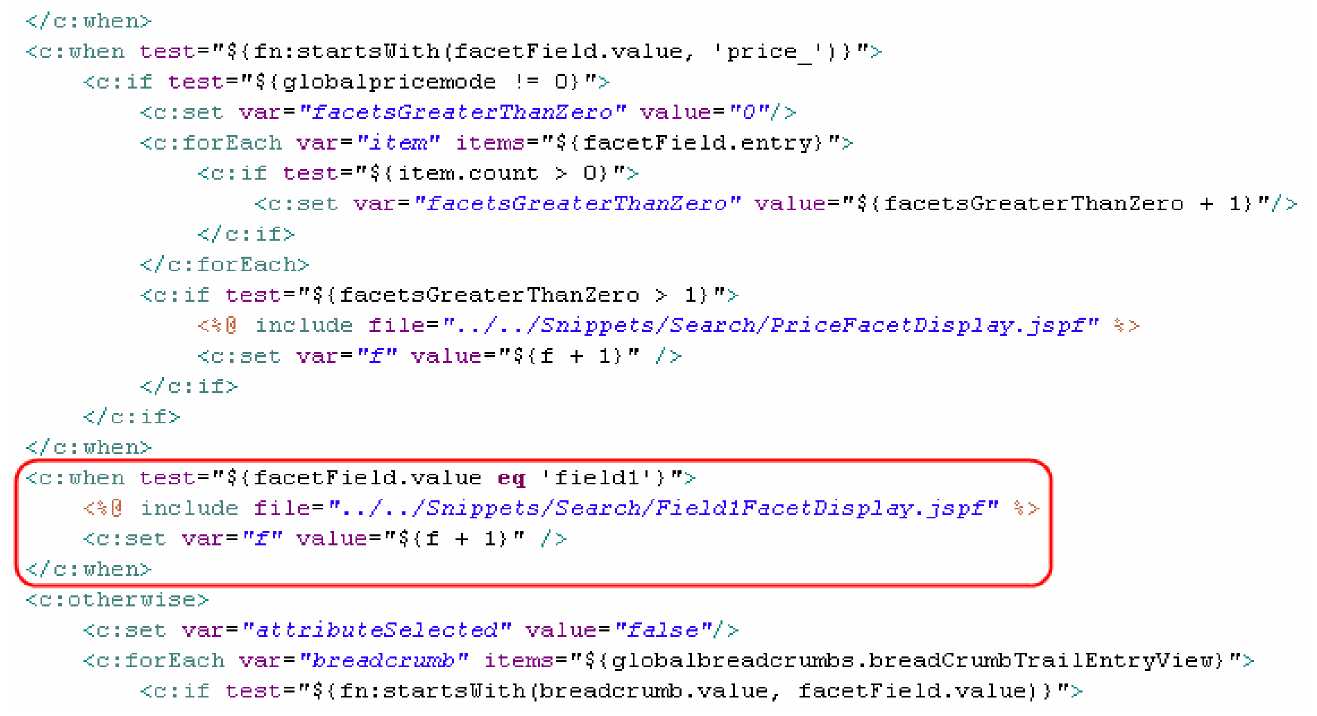
In this part of the lab, you will update the Madisons storefront to display the facet data in the left sidebar.

Note: This lab uses the store name **Madisons** to refer to the starter store where WebSphere Commerce search is configured. Your store name might differ.

____ 1. Open WebSphere Commerce Developer**.**

____ 2. Navigate to **Stores > WebContent > Madisons > Snippets > ReusableObjects** and open the file **SearchFacetsDisplay.jspf**. This JSP fragment loops through the available global facets and includes another JSP fragment to display each facet's data. You will create the new JSP fragment for field1 in a later step.

____ 3. Add a new test into the `<c:choose>` block to look for the field1 facet and include its display JSP fragment. You can copy one of the existing `<c:when>` blocks and update it to match the screen capture below.



```
</c:when>
<c:when test="${fn:startsWith(facetField.value, 'price_')}">
    <c:if test="${globalpricemode != 0}">
        <c:set var="facetsGreaterThanZero" value="0"/>
        <c:forEach var="item" items="${facetField.entry}">
            <c:if test="${item.count > 0}">
                <c:set var="facetsGreaterThanZero" value="${facetsGreaterThanZero + 1}"/>
            </c:if>
        </c:forEach>
        <c:if test="${facetsGreaterThanZero > 1}">
            <%@ include file="../../Snippets/Search/PriceFacetDisplay.jspf" %>
            <c:set var="f" value="${f + 1}" />
        </c:if>
    </c:if>
</c:when>
<c:when test="${facetField.value eq 'field1'}">
    <%@ include file="../../Snippets/Search/Field1FacetDisplay.jspf" %>
    <c:set var="f" value="${f + 1}" />
</c:when>
<c:otherwise>
    <c:set var="attributeSelected" value="false"/>
    <c:forEach var="breadcrumb" items="${globalbreadcrumbs.breadCrumbTrailEntryView}">
        <c:if test="${fn:startsWith(breadcrumb.value, facetField.value)}">
```

____ 4. **Save** and close the file.

____ 5. Navigate to **Stores > WebContent > Madisons > Snippets > Search**. Make a copy of the file **BrandFacetDisplay.jspf** and call the new file **Field1FacetDisplay.jspf.**

____ 6. Open **Field1FacetDisplay.jspf**.

© Copyright IBM Corporation 2011. All rights reserved

____ 7. The first line in this file sets the display name for the facet. Replace the line

```
<span class="left_sidebar_header"><fmt:message key="SEARCH_FACET_MANUFACTURER" bundle="${storeText}"/></span>
```

with the following. Note that the hard-coded value is used to simplify this lab. In practice you should create a new resource string.

```
<span class="left_sidebar_header">Shopper points</span>
```

____ 8. **Save** and close the file.

____ 9. **Restart** and **publish** the WebSphere Commerce Test Server to pick up changes you have made in the first 4 parts of this lab.

© Copyright IBM Corporation 2011. All rights reserved

Part 5: Create sample data

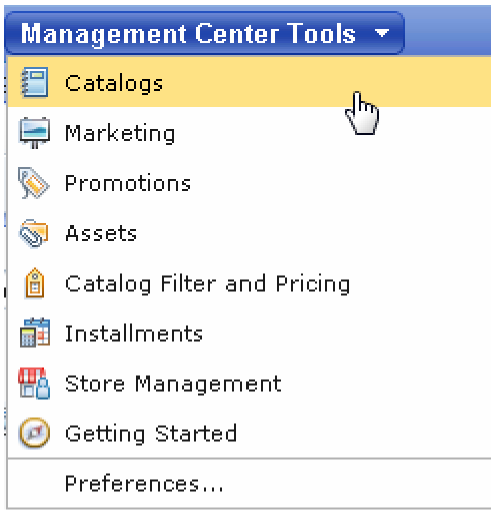
In this part of the lab, you will add sample data into field1. By default, the Madisons catalog does not have anything in field1. You only need to create enough sample data to test the facet in the storefront. Select two or more values to assign as 'shopper points' for the test products. Follow the steps below to populate your test data using the Management Center Catalogs tool.

____ 1. Open WebSphere Commerce Developer. If your test server is already running, **restart** it now to pick up changes from the previous steps.

____ 2. Open the hints and tips page at **<WCDE>\hintsandtips.html.**

____ 3. Launch Management Center using the link provided in the **Useful URLs** section.

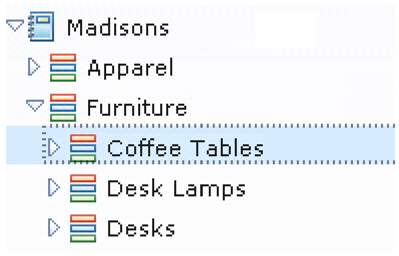
____ 4. Open the **Catalogs** tool.



____ 5. Select the **Madisons** store.

____ 6. Add Field 1 to the catalog entry list view

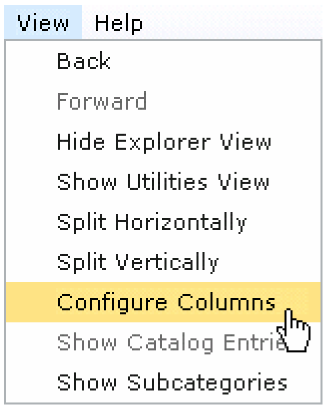
__ a. Expand the master catalog and select the **Coffee Tables** category to display the catalog entries list.



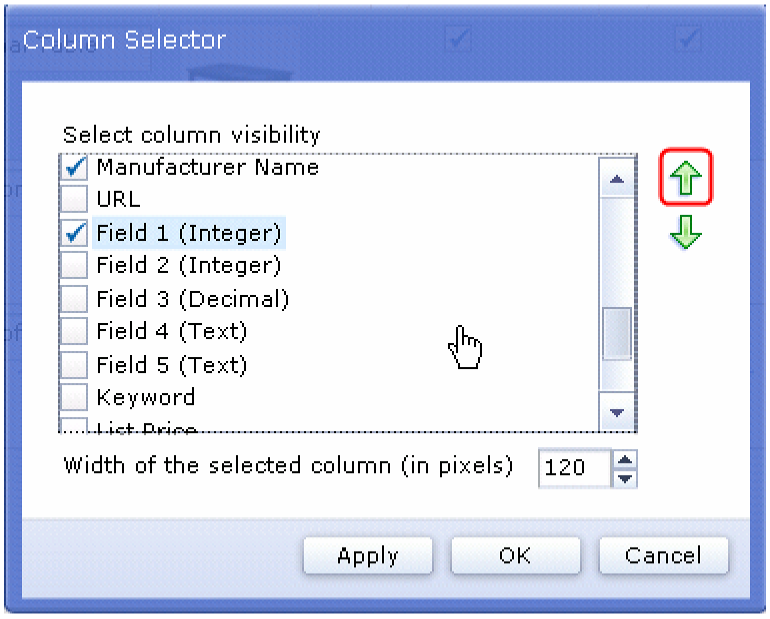
__ b. Select one catalog entry in the list view.

© Copyright IBM Corporation 2011. All rights reserved

__ c. In the menu bar, select **View > Configure Columns**.



__ d. Scroll down and select the check box next to Field 1 to include it in the list display. Click the Field 1 label and use the green up arrow to move Field 1 up to the top of the display order.

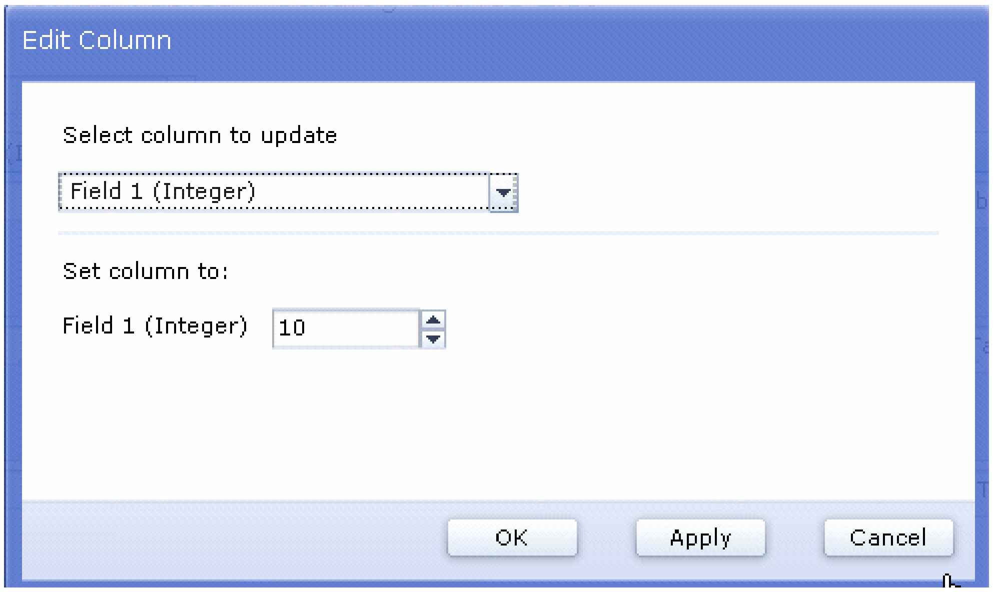


____ 7. Click **OK** to save your changes.

____ 8. Assign values for Field 1.

__ a. Select one or more rows in the catalog entry list view.

__ b. Use the Edit Column popup to assign a value to Field 1 for the number of 'shopper points' associated with the selected catalog entries.


Edit Column

Select column to update

Field 1 (Integer)

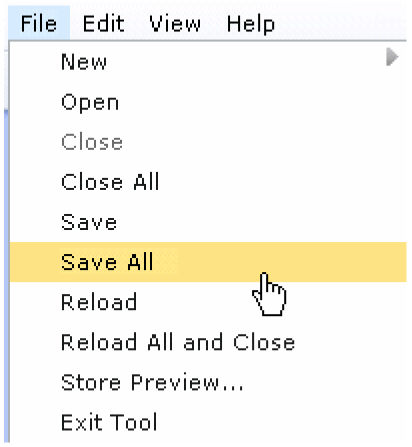
Set column to:

Field 1 (Integer) 10

__ c. Click **OK** to save your changes.

__ d. Repeat steps a - c to assign two or more different 'shopper point' values to various products under the Furniture category.

____ 9. In the menu bar, select **File > Save All** to save all your test data.



© Copyright IBM Corporation 2011. All rights reserved

Part 6: Test your changes

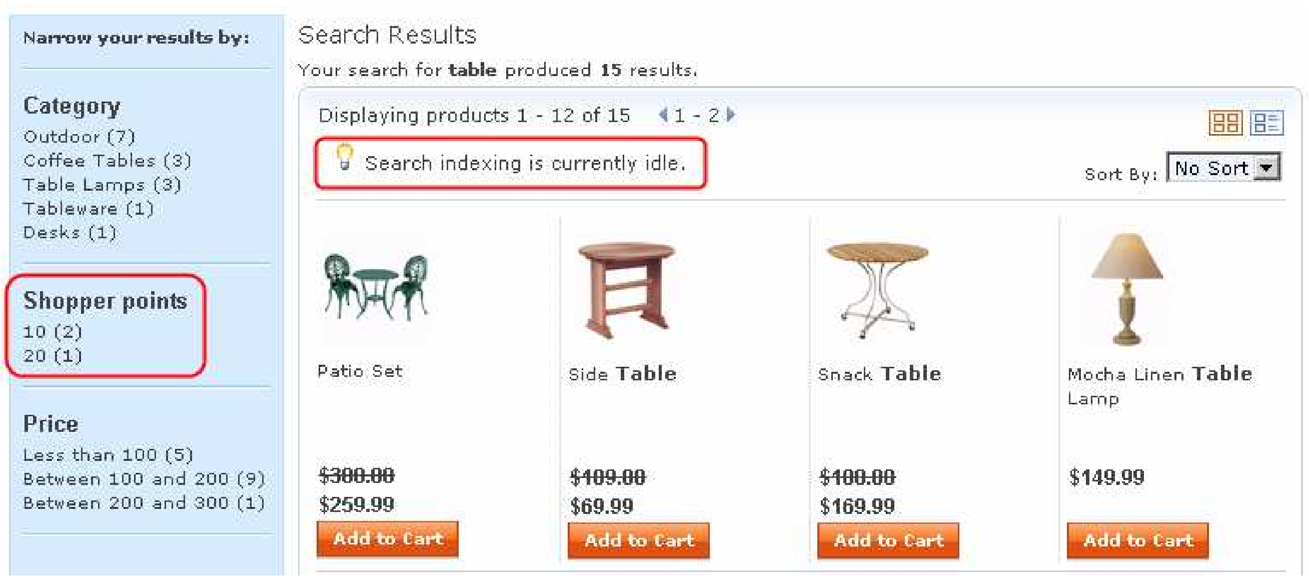
In this part of the lab, you will test your changes in the Madisons storefront. The shopper point data shown in your store is based on the test data you created in Part 5. It might be different than the examples shown below.

____ 1. Preview the Madisons store.

__ a. Select the store preview button from the toolbar.

__ b. The Madisons store information is selected by default. Click **Launch Store Preview** to view the store.

____ 2. Search on the term table. You will see the **Shopper points** facet in the left navigation bar. If you do not see the new facet, the delta index update might not have completed yet. You can see the status of the delta index update on the search results page.



Narrow your results by:

Category
Outdoor (7)
Coffee Tables (3)
Table Lamps (3)
Tableware (1)
Desks (1)

Shopper points
10 (2)
20 (1)

Price
Less than 100 (5)
Between 100 and 200 (9)
Between 200 and 300 (1)

Search Results

Your search for **table** produced **15** results.

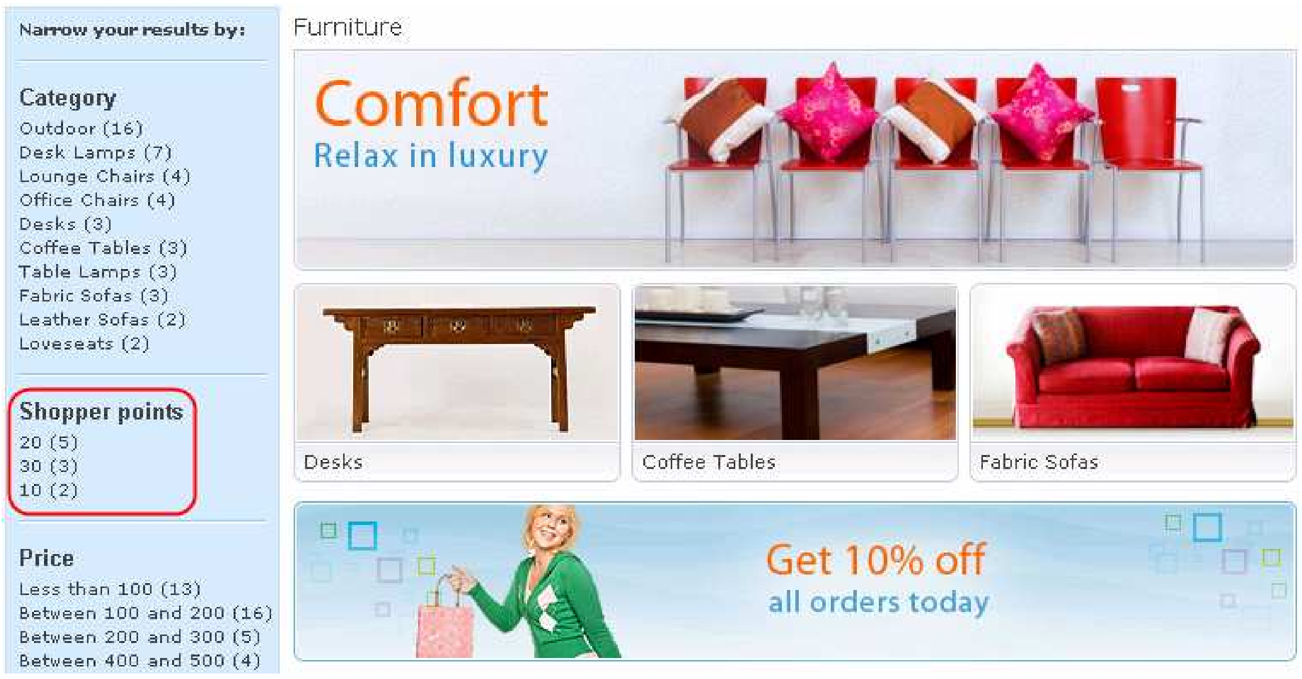
Displaying products 1 - 12 of 15 1 - 2

Search indexing is currently idle.

Sort By: No Sort

Patio Set	Side **Table**	Snack **Table**	Mocha Linen **Table** Lamp
~~$300.00~~ $259.99	~~$109.00~~ $69.99	~~$180.00~~ $169.99	$149.99
Add to Cart	Add to Cart	Add to Cart	Add to Cart

____ 3. Navigate to the **Furniture** category. You will see the **Shopper points** facet in the left navigation bar.



Narrow your results by:

Category
Outdoor (16)
Desk Lamps (7)
Lounge Chairs (4)
Office Chairs (4)
Desks (3)
Coffee Tables (3)
Table Lamps (3)
Fabric Sofas (3)
Leather Sofas (2)
Loveseats (2)

Shopper points
20 (5)
30 (3)
10 (2)

Price
Less than 100 (13)
Between 100 and 200 (16)
Between 200 and 300 (5)
Between 400 and 500 (4)

© Copyright IBM Corporation 2011. All rights reserved

Part 7: What you did in this exercise

In this tutorial you learned how to extend the Solr index to include additional data from the WebSphere Commerce base schema. You also learned how to use an index field as a facet in the storefront.

You should now understand how to complete these tasks:

- Extend the Solr index
- Make an index field facetable
- Customize the storefront to display a new facet