



В настоящее время большую популярность приобрели интернет-магазины. В частности интернет магазин на базе HostCMS.

Я тоже не избежал соблазна. Передо мной стояла задача создать карточку товара по образцу на картинке. В результате карточка товара у меня приобрела такой вид: http://www.albertprim.ru/shop/plazmen_panel/32lc51/

Эта страница с описанием товара насыщена довольно большой функциональностью. Можно оставить отзыв о товаре, можно посмотреть видео о товаре, можно благодаря java-скрипту посмотреть товар в трех видах. Так же на странице выводятся скидки, если они есть, а так же выборки: товар дня, спецпредложения, хиты, сопутствующие товары, "С этим товаром покупают". Об этой последней опции и пойдет речь в этой статье.

Ниже представлена статья из форума, о сопутствующих товарах. Задача: (отображение обратной совместимости товаров) Взято: <http://www.hostcms.ru/forums/22/4090/>

Примечание: Здесь к коду автора я добавил еще код синхронизации данных по сопутствующему товару указанному в админке и содержащемуся в БД в таблице `shop_intermediate_table`. Если данные не совпадают, то происходит соответствующее изменение в таблице, согласно данным в админке.

Сопутствующие товары — очень удобная вещь, но механизм их назначения предполагает, по сути, двойную работу, в тех случаях, когда товары являются взаимно сопутствующими. Например, у нас в магазине есть некий ассортимент устройств, и

набор аксессуаров для них. Было бы удобно, если на странице каждого устройства выводились бы подходящие аксессуары, а на странице каждого аксессуара — перечень устройств к которым он подходит.

Однако, для этого нужно для каждого устройства указать сопутствующими товарами его аксессуары, а потом еще для каждого аксессуара указать сопутствующими товарами те устройства к которым он подходит.

Предлагаемое мною решение позволяет сократить объем работы вдвое. Достаточно для каждого устройства указать сопутствующими товарами его аксессуары, а при отображении страницы аксессуара по тем же связям будут установлены совместимые с ним устройства. Для этого при выводе карточки товара (файл lib_6.php) мы должны написать нижепоказанный код.

```
{codecitation style="brush: xml;"}  
  
$param['xml_show_group_type'] = 'tree';  
  
$param['current_group_id'] = $GLOBALS['shop_item_path']['group'];  
  
$param['xml_show_group_property'] = false;  
  
// запрещаем передачу в XML свойств групп  
  
$param['xml_show_tying_products'] = true;
```

// сопутствующие товары показать

\$xsl_item = 'ВернадекТовар';

////////////////////////////////////

//Синхронизация БД и доп-свойства "Сопутствующие товары"

// Проверяем содержимое доп-свойства "Сопутствующие товары"

\$shop_items_catalog_item_id = \$GLOBALS['shop_item_path']['item'];

// ИД дополнительного свойства

// в списке дополнительных свойств в админке

\$shop_list_of_properties_id = 135;

```
$list = $shop->GetValueItemProperty($shop_items_catalog_item_id,  
$shop_list_of_properties_id);
```

```
$in = $list['shop_properties_items_value'];
```

```
// В $in содержится строка (список идентификаторов),
```

```
// содержащаяся в доп-свойстве "Сопутствующие товары".
```

```
$num_item = $GLOBALS['shop_item_path']['item'];
```

```
// прочесть в таблице строки с данным товаром, сохранить в строку $in_old
```

```
// сравнить с содержимым $in, если не равны, обновить записи в таблице
```

```
$DataBase = & singleton('DataBase');
```

```
$query_select = " SELECT sho_shop_items_catalog_item_id FROM `shop_intermediate_table`
```

```
WHERE shop_items_catalog_item_id = $num_item ";
```

```
// запрос
```

```
$result = $DataBase->select($query_select);
```

```
$items_string = ";
```

```
$prim = true;
```

```
if (mysql_num_rows($result) > 0) {
```

```
while ($row = mysql_fetch_array($result)) {
```

```
if($prim == true){
```

```
    $items_string = $row['sho_shop_items_catalog_item_id'];
```

```
    $prim = false; continue;
```

```
}
```

```
$items_string .= ', '.$row['sho_shop_items_catalog_item_id'];

}

}

// удалим из обоих строк пробелы и сравним их

$items_string = trim($items_string); $in = trim($in);

$mas1 = explode(",", $items_string);

// превратим строку в массив

$mas2 = explode(",", $in);

sort($mas1);

sort($mas2);

$str1 = implode(" ", $mas1);
```

```
// превратим массив в строку
```

```
$str2 = implode(" ", $mas2);
```

```
if(strnatcmp($str1,$str2) != 0){//если строки не равны
```

```
//Удалим все записи с данным товаром:
```

```
$query_delete = "`shop_items_catalog_item_id` = $num_item ";
```

```
$DataBase->Delete('shop_intermediate_table', $query_delete);
```

```
// Превратить строку в массив отдельных слов:
```

```
$mas_Words = explode(' ', $in);
```

```
// если не пуст, пройтись по массиву и вставить новые значения
```

```
// в таблицу `shop_intermediate_table` БД
```

```
if (count($mas_Words) > 0){
```

```
// Проходимся по массиву
```

```
foreach ($mas_Words as $key => $current_value){
```

```
    $DataBase->Insert('shop_intermediate_table', array('shop_items_catalog_item_id' =>  
$num_item, 'sho_shop_items_catalog_item_id' => $current_value, 'users_id' => '19',  
'shop_intermediate_count' => '0'));
```

```
}
```

```
}
```

```
}
```

```
else { //строки равны }
```

```
// конец синхронизации БД и доп-свойства "Сопутствующие товары"
```

```
////////////////////////////////////
```



```
// вывод Главного товара и с ним список сопутствующих
```

```
// и обратно совместимых товаров: "С этим товаром выбирают"
```

```
$site_users_id = 19; // надо указать через глобальные переменные текущего юзера сайта
```

```
// проверим, не является ли этот товар сопутствующим для других
```

```
$query = 'SELECT `shop_items_catalog_item_id`
```

```
FROM `shop_intermediate_table`
```

```
WHERE `sho_shop_items_catalog_item_id` = ' . to_int($GLOBALS['shop_item_path']['item']);
```

```
$DataBase = &singleton('DataBase');
```

```
$tying_hosts = $DataBase->select($query);
```

```
if ($DataBase->get_count_row() > 0 && $tying_hosts) {

$xml_host = "";

$host_param['xml_show_tying_products'] = false;

$host_param['xml_show_modification'] = false;

$host_param['xml_show_comments'] = false;

while ($host_item = mysql_fetch_assoc($tying_hosts)) {

$host_row = $shop->GetItem($host_item['shop_items_catalog_item_id']);

$xml_host .= $shop->GenXml4Item(0, $host_row, $site_users_id, $host_param);

}

$params['external_xml'] = '<tying_hosts>' . $xml_host . '</tying_hosts>';

}
```

/* Вывод конкретного элемента */

```
$shop->ShowItem($GLOBALS['shop_item_path']['item'], $xsl_item, $param,  
$external_propertys);
```

```
{/codecitation}
```

Теперь, в случае если просматриваемый товар является сопутствующим для какого-то другого товара в магазине, в xml будет появляться узел /shop/tying_hosts, содержащий узлы item

Вот кстати табличка, которая отображает: какие товары являются сопутствующими:
Таблица shop_intermediate_table

shop_intermediate_id

shop_items_catalog_item_id

sho_shop_items_catalog_item_id

users_id

shop_intermediate_count

6

158

159

19

0

8

164

168

19

0

9

164

165

19

0

10

168

164

19

0

11

168

165

19

0

12

165

164

19

0

13

165

168

19

0

В табличке четко и наглядно видно, что:

- для товара с ИД=164 (32LC51) сопутствующими являются товары с ИД=168 (37Z3030DR) и ИД=165 (32PFL5322).

- для товара с ИД=165 (32PFL5322) сопутствующими являются товары с ИД=168 (37Z3030DR) и ИД=164 (32LC51).

- для товара с ИД=168 (37Z3030DR) сопутствующими являются товары с ИД=164 (32LC51) и ИД=165 (32PFL5322).

Все в точности как мы определили своими действиями в админке, которые описаны выше!

И действительно, теперь при выводе товара в xml-структуре появляется узел /shop/tying_hosts:

```
<shop id="4" current_group_id="594">
```

```
<tying_hosts>
```

```
<item id="164" group="594">
```

```
<captcha_key>55178</captcha_key>
```

...

```
</tying_hosts>
```

который содержит все сведения обо всех товарах, для которых он является сопутствующим! Это здорово!!!

Теперь, в случае если просматриваемый товар является сопутствующим для какого-то другого товара в магазине,

в xml будет появляться узел /shop/tying_hosts, содержащий узлы item

В xsl это может быть обработано, например, так:

```
<xsl:if test="count(tying/item) > 0">
```

```
<h3>Сопутствующие товары:</h3>
```


<!-- Отображаем сопутствующие товары -->

<div class="clear">

<xsl:apply-templates select="tying/item" mode="tyings"/>

</div>

<p class="clear"> </p>

</xsl:if>

<xsl:if test="count(/shop/tying_hosts/item) > 0">

<h3>Совместимость:</h3>

<!-- Отображаем реверсивно-сопутствующие товары -->

<div class="clear">

```
<xsl:apply-templates select="/shop/tying_hosts/item" mode="tyings"/>
```

```
</div>
```

```
<p class="clear"> </p>
```

```
</xsl:if>
```

При этом, в определении шаблона для отображения списка сопутствующих товаров надо сделать следующее изменение.

Вместо

```
<xsl:template match="tying/item">
```

написать

```
<xsl:template match="item" mode="tyings">
```

Тогда обычные сопутствующие товары, и сопутствующие товары, вычисленные с помощью обратного определения в ТДС,

будут отображаться визуально одинаково! Отлично! Все работает правильно!

Выводы:

Если теперь товар ИД=164 (32LC51) рассматривать как "Устройство", а сопутствующие ему товары

с ИД=168 (37Z3030DR) и ИД=165 (32PFL5322) рассматривать как "Аксессуары",

то при выводе любого аксессуара, можно вывести ИД=164 (32LC51) как совместимое с ним устройство.

Так же и другие совместимые устройства, если таковые есть.

Другими словами:

1. Устройство ИД=164 (32LC51) имеет два аксессуара: ИД=168 (37Z3030DR) и ИД=165

(32PFL5322)

2. При выводе аксессуара ИД=168 (37Z3030DR) как самостоятельного товара,

с ним можно вывести товар ИД=164 (32LC51), как совместимое с ним устройство!

3. Если имеется еще другой товар, допустим товар Z, для которого

ИД=168 (37Z3030DR) и ИД=165 (32PFL5322) являются аксессуарами,

то при выводе ИД=168 (37Z3030DR) как самостоятельного товара,

с ним можно вывести товар ИД=164 (32LC51) и товар Z,

как совместимые с ним устройства!

Что наглядно демонстрирует эта программа!