

## СОЗДАНИЕ САЙТА НА ОСНОВЕ БАЗЫ ДАННЫХ С ПЕРЕМЕННОЙ АТТРИБУТИВНОСТЬЮ

**Автор:** Маслов Владимир, 3 курс.

**Руководитель:** Кульман Татьяна Николаевна, к.т.н., зав. кафедрой Информационных технологий филиала «Протвино»

**Образовательное учреждение:** МУПОЧ «Дубна», филиал «Протвино»

### BUILDING A WEB SITE BASED ON THE DATABASE WITH VARIABLE ATTRIBUTIVENESS Maslov V.

Традиционным подходом к хранению документов разного рода (содержащих различное количество полей) в базах данных (далее – БД) является создание отдельных таблиц для каждого типа документа. Такой подход обладает определенными преимуществами, – в частности, простотой реализации, но, вместе с тем, отличается существенным недостатком – в случае, когда администратор БД решит добавить новый тип продукции, программист будет вынужден создать новую таблицу и изменить программное обеспечение сайта для того, чтобы учесть наличие этого продукта.

Существует иной подход к проектированию, более применимый для задач, в которых рассматривается хранение разнотипной продукции – это база данных с переменной атрибутивностью. В такой БД можно хранить произвольные объекты, содержащие уникальные наборы свойств и значений. Для простого варианта БД такого типа потребуется четыре таблицы (рисунок 1). В первой таблице должны храниться названия документов, их типы и номера (id). Во второй – имена атрибутов, их номера и параметры отображения. В третьей – номера имен атрибутов, значения атрибутов и номера документов, которым они принадлежат. В четвертой – типы документов (один тип – для продуктов, другие типы – для производителя, серии и т.д.).

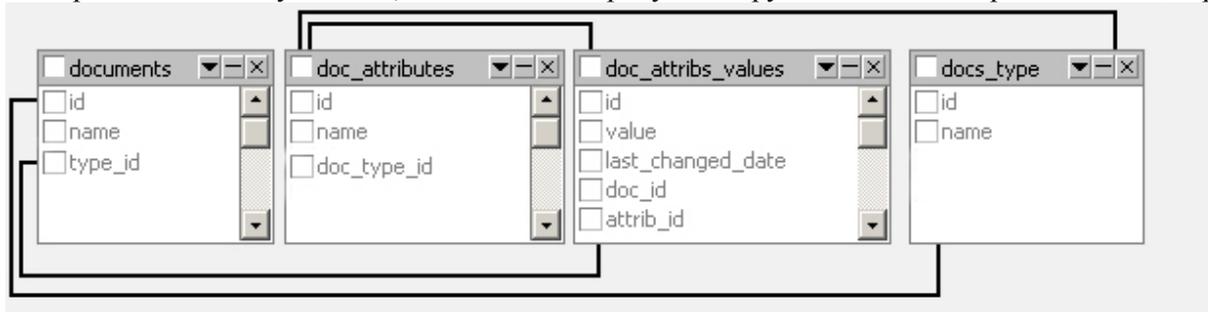


Рисунок 1. Устройство БД

Одной из особенностей данной БД является то, что документы не связаны между собой привычным способом. Механизм наследования продуктов следует делать либо на уровне ПО – построением иерархии по свойствам, либо – созданием отдельной таблицы сопоставления предков и наследников. В базе не хранятся фиктивные сущности, предназначенные исключительно для связывания между собой уровней иерархии, – связи выстраиваются на основе содержимого таблицы свойств при помощи созданного модуля. Другой вариант решения этой задачи предполагает жесткое закрепление иерархии, чего в конкретной задаче следовало избежать.

Идея и структура БД подобного типа была использована для создания конкретной разработки – сайта-каталога товаров. В основу сайта была положена система управления контентом Joomla, для которой необходимо было создать набор скриптов на языке PHP – модуль для навигации по каталогу, главный компонент каталога. При описанном устройстве базы, необходимо было решить такие вопросы, как отображения хранимых данных, их типизация и выстраивание иерархии.

По своей внешней сути, модуль – это построенное дерево иерархии с возможностью выбора нужной категории. Визуальное представление иерархии в виде дерева значительно улучшает юзабилити сайта и становится основным средством навигации по нему. В данном каталоге оно позволяет осуществить выбор производителя, серии для выбранного производителя, или типа продукции для выбранной серии. Эти категории (уровни иерархии) уникальные для каждого продукта.

Принцип работы модуля следующий. Для построения дерева иерархии необходимо создать массив, в котором последовательно хранились бы элементы с их собственными номерами и номерами их элементов-предков, а также со ссылками на страницы каталога, причем ссылки должны генерироваться модулем и передавать в компонент параметры. Компонент должен будет принимать их, находить требуемый продукт (или производителя, или список продуктов) и отображать его содержимое.

Следует создать массив приоритетов, который после перебора таблицы атрибутов будет содержать информацию о том, для каких продуктов иерархия будет строиться, начиная с производителя, а для каких – с другого свойства. Данный функционал реализован при помощи собственных функций сортировок массивов.

В качестве массива для хранения построенной иерархии выступает многомерный массив, используемый как коллекция – для случая двухуровневой иерархии индекс [i][0] – это первый уровень, [i][j] – второй. Цикл выстраивает ветвь иерархии до конца, а затем меняет главный, относительно вложенного цикла, индекс.

На основе полученного массива при помощи PHP модуль формирует рабочий JavaScript-код, который отображает дерево.

Следующим шагом разработки является компонент (рисунок 2). Как известно, любой компонент в Joomla состоит из двух частей: административной (backend) и пользовательской (frontend). Первоочередной задачей являлось создание frontend, в свою очередь, состоящий из контроллера, модели и представления – для разделения функций работы с БД, формирования вывода PHP-кода для браузера и главных управляющих функций, определяющих их соответствие. Очевидное преимущество такой модели – параллельная разработка слабо зависящих друг от друга функций.

Важнейшей решенной задачей является введение типов отображения атрибута. Каждый атрибут обладает полем, в котором хранится сериализованная строка. В этой строке запакован класс, содержащий уровень иерархии для данного свойства (в случае, если его необходимо отобразить в модуле), а также тип атрибута для отображения (например, изображение, схема, статья, и т.д.). В модели компонента функции создают массивы объектов строк-свойств для каждого из типов отображения. Свои типы есть и у продуктов, и у производителей.

Заключительным шагом разработки frontend для компонента стала синхронизация модуля дерева и основного компонента – возможности при открытии продукта определить его положение в дереве и раскрыть нужный пункт списка.



Рисунок 2. Вид frontend компонента.

Следующим этапом разработки стало создание административной части компонента. Был создан универсальный интерфейс добавления продуктов, производителей и серий, который с минимальными усилиями можно приспособить для создания каталога другого типа.

Администратор способен внести продукт с нужным ему количеством свойств, которые могут быть характеристиками или изображениями. Интерфейс, предоставляя широкий функционал, остается удобным для пользователя. Конечной целью было создание интерфейса (рисунок 3), подобного простейшему табличному редактору из офисных пакетов – например, Microsoft Excel.

При помощи библиотеки jQuery были реализованы всплывающие окна (pop-up), добавление и удаление строк таблицы без перезагрузки страницы. Загрузка изображений осуществляется при помощи библиотеки jQuery и технологии Web 2.0 под наименованием Ajax, которая позволяет передавать данные на сервер и получать ответ без полной перезагрузки страницы. Таким образом, администратор даже при загрузке изображения на сервер не покидает основной интерфейс. Затем по сохранению продукта модель компонента считывает введенные данные и заносит их в БД. Функции правки и удаления продуктов также присутствуют в ПО.

## Добавление продукта

Наименование:

ID	Свойство	Значение	Тип
<input type="radio"/> 1	<input type="text" value="Производитель"/>	<input type="text" value="Finder"/>	S
<input type="radio"/> 2	<input type="text" value="Изображение"/>	<input type="text" value="6859321dfoa2dc0c2e3e16026ca450bb122638692"/> ...	IMG
<input checked="" type="radio"/> 3	<input type="text" value="Заголовок"/>	<input type="text"/>	H
<input type="radio"/> 4	<input type="text" value="Параметр 1"/>	<input type="text" value="3"/>	HD
<input type="radio"/> 5	<input type="text" value="Параметр 2"/>	<input type="text" value="3"/>	HD

Добавить

Удалить



Сохранить

Рисунок 3. Вид

backend компонента (интерфейса администратора)

Реализованное решение по удобству неотлично от аналогичных десктопных интерфейсов. Некоторый минимализм интерфейса лишь усиливает простоту его использования, и в результате человек, обладающий минимальными умениями в работе с табличными редакторами, практически моментально его осваивает.

В ходе тестирования созданный сайт показал нормальную работоспособность под нагрузкой в пять тысяч объектов. В настоящий момент ведутся работы по внедрению сайта.

### Литература

1. Рамел, Д. Самоучитель Joomla! : [пер. с англ.] / Д. Рамел. – СПб.: БХВ-Петербург, 2008. – 448 с.
2. Дейт, К. Дж. Введение в системы баз данных, 8-е изд. : [пер. с англ.] / К. Дж. Дейт. – М: «Вильямс», 2005. – 1328 с.