

Overlooked Advantages of Database Driven Websites

[Web Smart Newsletter](#) originally published May 2002 - Updated July 2006. By Eric Holter.

Overlooked Advantages of Database Driven Websites

This month's newsletter reviews the fundamental principles behind database driven web development. While content management is an important aspect of database driven web development, there are other more foundational benefits to this approach that are often overlooked.

How database driven sites work

A database driven approach to web development simply refers to the use of a database for storing a website's content. It involves the separation of a site's content (words and pictures) from its design (the look, feel, and navigation). The design is most typically stored in template files. When a site visitor requests a page by clicking a link, a template draws the appropriate information from the database before it displays the result as an HTML.

By contrast, a static website is comprised of a series of HTML files. These files are self-contained documents that appear the same to all users, regardless of where, how, when, or why they are viewing the document. A single static file holds all of the data for a particular page (text, images, navigational structure, etc) and the design and content are intertwined. When a site visitor requests a page, the HTML file is passed from the server to the visitor's browser. On a static site, visitors are simply viewing documents over the internet, in much the same way they might open files on their own hard drives.

To illustrate the difference between a static and a database driven site, consider a site that contains 25 product pages. Using a static web development model, a programmer would need to create 25 separate HTML pages with the same basic navigational structure, each containing an individual product's specs and information. To add a new product to the site, the programmer would need to create a new copy of the product page file and insert the new product's information into the appropriate places within the file. To make a change to the structure of all 25 product pages, the programmer would need to adjust each of the 25 HTML pages by hand.

Using a database driven approach, the programmer would build a product table in the website's database. This table would contain a series of columns with pertinent information about the product line. For instance, the table might have the following columns: "product identification number," "product name," "product size," "price," "availability," "description," and "image." The programmer would then create a template file to use the information from the product table. This template file would contain the basic structure of a product page, identifying elements like the navigation system, where a product image is placed, where the description and other information goes on the page, and how all the data is to be formatted. When a user visits a particular product page, the template grabs the data associated with that product's identification number from the database, plugs the data into the template and presents the user with a complete product page "on the fly."

Developing Database Driven Websites

How the database approach changes wasteful web development patterns

The typical web development pattern most companies follow is to design and build a static website, maintain that site for 18 - 36 months, and then replace it altogether. Maintaining the site usually means adding and removing press releases and product information, or making minor adjustments to the homepage content. Over this period companies often discover many new possibilities for using their sites, but do not implement them. The reason these new ideas don't get implemented is that making them work would require too many adjustments to the overall navigation or layout of the site. For example, a company might decide that their site would benefit by adding a "customer service" button to their navigation bar that leads to an area with FAQs and comment forms related to customer service issues. This may in

fact be a great idea for the site and could dramatically improve its effectiveness, however, to make the change to the navigation system would require that every single page on the site be manually altered. The cost of doing this can be prohibitive even though it would be a great feature. Instead of adding the new feature, it ends up on a wish list, along with other ideas and possibilities, for when the company is willing and able to do a full "redesign" of the site.

When it is finally time to do a redesign, the site needs so many changes that usually the first site is scrapped and a completely new (static) site is built with all the new features, design and content. This second development project is usually more costly than the first. Unfortunately, because markets and business plans change so quickly, there are soon more new features and content that should be added to the new site. But now the site is even bigger than the first site and to make changes to the navigation or layout is an even bigger, more prohibitive project, so again these ideas and possibilities get stored up for the next rebuild, which will be even more expensive than the first two.

This pattern wastes money, and creates lost opportunities for a website between rebuilds. The inflexible and inefficient static web development approach usually causes companies to feel frustrated with their websites, when they should be enjoying the many benefits and advantages that websites can offer.

The database driven approach to web development changes this frustrating pattern. Because database driven sites separate content (by storing it in a database) and design (by using templates), changes to a site's structure and content can be made incrementally as needed. In contrast to the previous example, a company could add a customer support area to a database driven site by simply adding the new content and adjusting the site templates. This process is far simpler than making global changes to every static page. Additionally, making changes to a database-driven site remains just as easy if the site were to double or triple in size. By contrast, a static site would require two or three times as much effort and expense.

This represents a huge shift in a company's ability to utilize the web effectively. While this basic shift is perhaps the most important difference and advantage in the database approach there are other overlooked advantages as well.

Benefits of Database Driven Redesign

Visual design changes

Most companies and organizations are generally so relieved just to make it through a web development project, that the last thing they want to think about is the next phase or revision of the site. It's easy to overlook the impact that a current development project will have on future work. However, the way that a website is structured, designed, and built, can have an enormous impact on how it will be able to be modified in the future.

One of the primary advantages of a database driven model to website development is the relative ease with which it allows a site to be redesigned. When redesigning the look and feel of a static site with a thousand pages, each of the thousand existing pages must be altered and reformatted to reflect a new design. A database driven approach, by contrast, offers a significant amount of flexibility. A site may have a thousand pages, but use only a few dozen template files. Therefore, to apply a new look and feel to the all of the press releases on the site, only the one template file needs to be altered and the new design will be reflected on every press release page. In this manner, a database approach builds a strong base for a site to grow and change easily over time.

Searchable Database Driven Websites

Advanced searching capabilities

Over the years, we have come into contact with many sites that have great content, but are virtually unusable because their search features don't allow users to find content quickly or accurately. Frequently, the search mechanisms are slow because they must search through hundreds or thousands of static HTML files. Or they are inaccurate because they don't allow users to weigh certain pieces of information more heavily than others. For instance, someone may want to find an articles written by a particular author, not necessarily all the articles that contain that author's name.

Databases are built for searching. One of the primary benefits of a database driven approach to web development is advanced searching.

It is significantly faster to search through a thousand database records than a thousand HTML pages. Additionally, since content is broken up into logical data fields within the database, users can search for very specific content. Advanced queries such as one that would locate, say, all of the articles in a database that have an author named "John," a title containing the words "buy" and "sell," and were published in 1997, are fast and manageable with a database approach. These types of queries would be virtually impossible to facilitate with a static site.

Repurposing Website Content

Leveraging content in multiple locations

With a database driven approach, all site page templates can access any information in the database. This allows companies and organizations to leverage content in a variety of site areas. For instance, you may decide to create a calendar of events on your site. With a typical static website model, the calendar would have to be maintained by hand, removing events from HTML pages as they pass and duplicating the information by hand anywhere else on the site where it is useful.

Using a database driven model, each individual event would be stored as a separate entry in the database. Calendars could then be dynamically generated and displayed based on the current date and time, so that site visitors would see only events from a particular swath of time. Additionally, the information could be displayed in a variety of places on the site, or using an assortment of views, without needing to duplicate the calendar data.

For example, if you wanted to highlight a few upcoming events from the calendar on the homepage, the homepage template could be modified to grab, say, three upcoming events from the database. Or you might have one calendar view that shows weekly information, another that displays a monthly overview, and another that gives detailed descriptions of specific events. Since all of these various templates draw their content from the same events table in the database, the information only ever needs to be entered into the system once. This aspect of a database driven site allows for more effective management of content and reduces the possibility of conflicting information appearing in two places on a site.

Conclusion

The database driven approach is not just a better technical solution. It actually changes the way companies can utilize the web, taking advantage of the true benefits it offers. Having a site's content stored in a database prepares that site to head in any direction in the future. These overlooked advantages of database-driven sites are important to sites of all sizes. Even when a client's site is small, or they don't think they will change much content, at some point they will want to add and upgrade their site. The database approach ensures that past work is not thrown away as a site grows and changes over time.