



Sky Web Works  
6516 Burrows Court  
Plano, Texas 75023  
972.312.1517  
info@skywebworks.com  
www.skywebworks.com

## Database Tutorial - Really it's a page of info...but well worth the read!

Databases are very useful for our ever growing and speed of business. Most databases these days are a "Relational Database Model". Of course in the 80's the "Relational Database Model" became the rage. The Relational Model developed out of the work done by Dr. E. F. Codd at IBM in the late 1960s who was looking for ways to solve the problems with the existing models.

Because he was a mathematician, he naturally built the model on mathematical concepts which he expounded in the famous work called "A Relational Model of Data for Large Shared Databanks".

At the core of the relational model is the concept of a table (also called a relation) in which all data is stored. Each table is made up of records (horizontal rows also known as tuples) and fields (vertical columns also known as attributes).

It is important to note that how or where the tables of data are stored makes no difference. Each table can be identified by a unique name and that name can be used by the database to find the table behind the scenes. As a user, all you need to know is the table name in order to use it. You do not need to worry about the complexities of how the data is stored on the hard drive.

So how do you find data in a relational database if there is no map (like a hierarchy defined by pointers) to follow?

Well, in the relational model, operations that manipulate data do so on the basis of the data values themselves. Thus, if you wish to retrieve a row from a table for example, you do so by comparing the value stored within a particular column for that row to some search criteria.

For example, you might say (not getting into syntax) "Give me all the rows from the 'Directory' table which have 'James' in the 'FIRST\_NAME' column."

The database might return a list which looks essentially like this:

Tami Kowalkowsky	1977 GKI-0001	972-312-1517
James Freedman	1985 GKI-0025	972-312-1517
Dbase Sausage	1965 GKI-5202	972-312-1517

You could then use the data from a retrieved row to query another table. For example, you might say "Okay, I want to know what degrees 'James Sausage' had when teaching 'Beer Games 101'. So I will now use the 'Faculty ID' number from my previous query as the keyword in my next query. I want the row in the 'Beer Games 101 Degrees' table where Faculty ID equals 'GKI-5202'.

This data access methodology makes the relational model a lot different from and better than the earlier database models because it is a much simpler model to understand. This is probably the main reason for the popularity of relational database systems today.



Sky Web Works  
6516 Burrows Court  
Plano, Texas 75023  
972.312.1517  
info@skywebworks.com  
www.skywebworks.com

## Database Tutorial - Really it's a page of info...but well worth the read!

Another benefit of the relational system is that it provides extremely useful tools for database administration. Essentially, tables can not only store actual data but they can also be used as the means for generating meta-data (data about the table and field names which form the database structure, access rights to the database, integrity and data validation rules etc).

Thus everything within the relational model can be stored in tables. This means that many relational systems can use operations recursively in order to provide information about the database. In other words, a user can query information concerning table names, access rights, or some data and the results of these queries would then be presented to the user in the form of a table.

As we said before, most databases that you will come across these days will be relational databases. However, there are many types of relational databases and not all of them will be useful for web applications.

In particular, it will be the client/server databases rather than the stand-alone packages that you will use for the web.

A client/server database works like this: A database server is left running 24 hours a day, and 7 days a week. Thus, the server can handle database requests at any hour. Database requests come in from "clients" who access the database through its command line interface or by connecting to a database socket. Requests are handled as they come in and multiple requests can be handled at one time.

For web applications which must be available for world wide time zone usage, it is essential to build upon a client/server database which can run all the time.

It is most likely that as a company, you will be working with one of the modern relational databases and that you will be able to work in conjunction with an existing database administrator. That is, this tutorial is limited to the "use" of databases rather than to the creation and administration of them. In fact, the creation and administration of databases is a topic well beyond the scope of this tutorial and probably well beyond the scope of your job. After all, database administration is its own entire job description.

However, we have been spending a lot of time going through general database theory because although you may not be designing databases yourself, in order to take the most advantage of them, it is important that you understand how they work. Likewise, it is important that you have a feel for good database design. After all, a database's usefulness is directly proportional to the sleekness and efficiency of its design. And your ability to take advantage of a database is directly proportional to your ability to decipher and internalize that design.

When thinking about good database design, it is important that you keep data retrieval, storage and modification efficiency in mind. It will pay off one thousand fold if or when you have your database built. You will find that certain table structures will provide easier and more intuitive access than others.

Tables should describe only one subject, have distinct fields, contain no redundant data, and have a field with unique values so that the table can be related to others.

You should also keep in mind future expansion of the database and make sure that your design is easily extensible.



Sky Web Works  
6516 Burrows Court  
Plano, Texas 75023  
972.312.1517  
info@skywebworks.com  
www.skywebworks.com

## Database Tutorial - Really it's a page of info...but well worth the read!

Typically, you will go through a requirements phase in which you should simply sit with the problem, interview users, and achieve an intuition about the data and the project.

Next, you should spend time modeling the data, preferably using some standard methodology like ER Diagramming. However, even if you do not model in any traditional way, you can still play with different ideas and think about the pros and cons.

Hopefully, you will also choose a database with full- functionality such as security and concurrency control (making sure that two users cannot simultaneously change a record). There are many excellent choices available in the market today from freeware to commercial products.

At any rate, a mandatory meeting should be held talking between users and company executives to form a conclusion and model for your database project.

Once the database is up and running and populated with data, you will need a way of talking to it. Essentially, there are two ways of doing that: connecting to the database directly using a command shell or by connecting to the database over the network using sockets (if that database allows such interaction).

Command shells are pretty common and come bundled with just about every database. These are usually simple filters which allow you to log on to the database, execute commands and receive output. They can either be very simple command-line shells or nice, fanciful graphical user interfaces. Whichever the case, command shells provide access to the database from the machine on which the database is running.

Socket-based middleware does the exact same thing, but over a network ie. (Internet). Thus, with a socket-based system, I could send commands to my database in Dallas while I am working in Hawaii with my laptop at the beach (lol), by using TCP/IP over the internet to connect to my database.

Web Database follows the client-server database model. A Database Engine sits on some central computer somewhere and serves data to multiple web-based clients (perhaps lots of customers using Netscape Navigator or Internet Explorer).

Because we are dealing with web-based clients however, we must also have a Web Server that handles requests from web-browsers and then forwards them to the Database. Likewise, the web server will wait for the database to respond and then pass on that response to the waiting web browsers. The whole interaction is much simpler as a picture.

Of course, the problem with the above model is that it does not exactly tell the whole story. Actually, though Web Servers are built to talk to Web Browsers, they are not built to talk to Databases. Thus, in order for the Web Server to talk to a Database, it requires a helper (sometimes called Middle Ware).

The most basic type of Middle Ware is a CGI script, and even more advanced would be ASP, JSP, or PHP and is written to translate requests from the Web Server to a format that the Database can understand, and to translate Database responses into something the Web Server can send back out to the Web Browser and that the person using the web browser can understand.



Sky Web Works  
6516 Burrows Court  
Plano, Texas 75023  
972.312.1517  
info@skywebworks.com  
www.skywebworks.com

## Database Tutorial - Really it's a page of info...but well worth the read!

The CGI Script will be responsible for understanding what the Web Server is saying and will also be responsible for knowing how to talk to the Database. This type of CGI script or language is seriously multilingual! This is how the data in the database is produced onto a web page. In these scripts we can do all sorts of tasks, such as multiply, divide, add, subtract, etc. In addition we can have statements that cause something to happen if something exists, which are called if and else statements. This is how the data becomes more interactive.

This tutorial was written to help you better understand the database backend and frontend process. If it was easy to understand, then everyone would be building their own database driven sites. Since this does not hold true, Sky Web Works is here to walk you through this process and at the same time let you be a part of this undertaking. For more information contact our sales department at 972-312-1517 or email sales@skywebworks.

Thankyou,  
Sky Web Works design team