

Database Home

Database Tutorial

[What is a Database?](#)
[Database Management Systems](#)
[Creating a Database](#)
[About Database Tables](#)
[Creating Database Tables](#)
[Adding Data to a Database](#)
[Querying a Database](#)
[Relational Database Design](#)
[Database Driven Website](#)
[Summary](#)

Database Tutorial



This database tutorial is aimed at beginners. Perhaps, you think you need a database but you're not sure. Or maybe you need to create a database driven website but don't know where to start? Or perhaps you already work with databases but you'd like to brush up on the fundamentals?

If this sounds like you, read on!

Table of Contents

This database tutorial consists of the following lessons:

1. [What is a Database?](#)
2. [Database Management Systems](#)
3. [Creating a Database](#)
4. [About Database Tables](#)
5. [Creating Database Tables](#)
6. [Adding Data to a Database](#)
7. [Querying a Database](#)
8. [Relational Database Design](#)
9. [Database Driven Website](#)
10. [Summary](#)

- [Start Tutorial >>](#)

If you are already familiar with databases but would like to learn SQL, check out the [SQL Tutorial](#). If you'd like to learn how to use Microsoft Access, try the [Microsoft Access Tutorial](#).

Web quackit.com

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

[Get your FREE hosting today!](#)

>>

Featured Template:
(Build your websites in minutes!)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

[Create a Website](#) | [HTML](#) | [CSS](#) | [MySpace Codes](#) | [JavaScript](#) | [ColdFusion](#) | [PHP](#) | [VBScript](#) | [Databases](#) | [SQL](#) | [SQL Server](#) | [Microsoft Access](#) | [Web Graphics](#) | [Web Design](#) | [Web Hosting](#) | [Web Servers](#) | [XML](#)

[About Quackit](#) | [Terms of use](#) | [Privacy Policy](#) | [Contact Quackit](#)

© Copyright 2000 - 2008 Quackit.com

Database Home

Database Tutorial

[What is a Database?](#)
[Database Management Systems](#)
[Creating a Database](#)
[About Database Tables](#)
[Creating Database Tables](#)
[Adding Data to a Database](#)
[Querying a Database](#)
[Relational Database Design](#)
[Database Driven Website](#)
[Summary](#)

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

What is a Database?


A database is a collection of data. That may sound overly simplistic but it pretty much sums up what any database is.

A database could be as simple as a text file with a list of names. Or it could be as complex as a large, relational database management system, complete with in-built tools to help you maintain the data.

Before we get into dedicated database management systems, let's start with the basics - let's look at a simple text file example.

Text File

Imagine we have a **text file** called "**Individual.txt**", and that the contents look like this:



```

File Edit Format View Help
IndividualId,FirstName,EmailAddress
1,Homer,homer@quackit.com
2,Barney,barney@quackit.com
3,Ozzy,ozzy@quackit.com
4,Fred,fred@quackit.com

```

We could use this information to do things such as send an email to everyone on our list. We could do this because, due to the way we designed the list, we know that each *row* contains a different individual, and the information on that row is related to that individual. Also, the items in each row are separated by commas. Therefore, we know that the email address next to "Homer" is his email address. We could also call each row a *record*. Therefore, we currently have 4 records in our database.

With a small list like this, a text file may serve our purposes perfectly.

Spreadsheet

Another option would be to store it in a spreadsheet using spreadsheet software (for example, Microsoft Excel). That way, we could do some extra things with our list (such as format it, or sort by first name/surname etc).

Web quackit.com**FREE Hosting!**

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

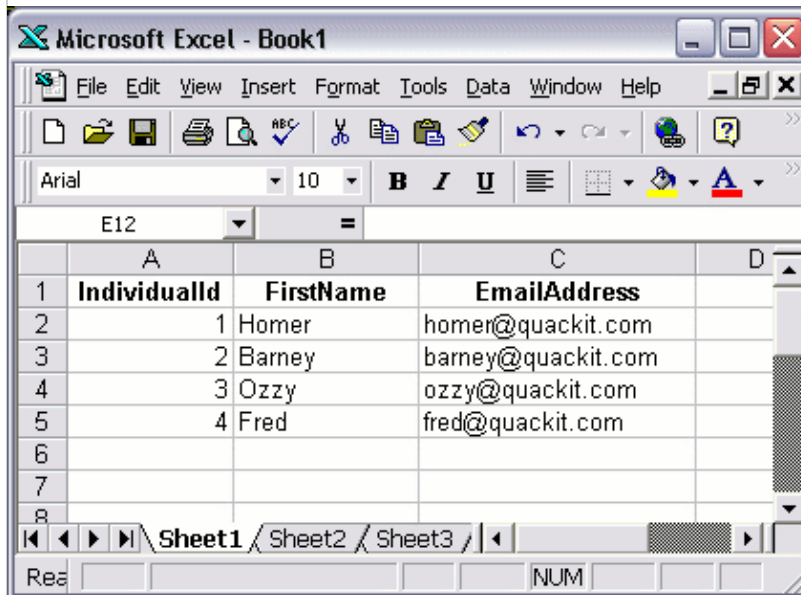
- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

[Get your FREE hosting today!](#)

>>

Featured Template:
(Build your websites in minutes!)

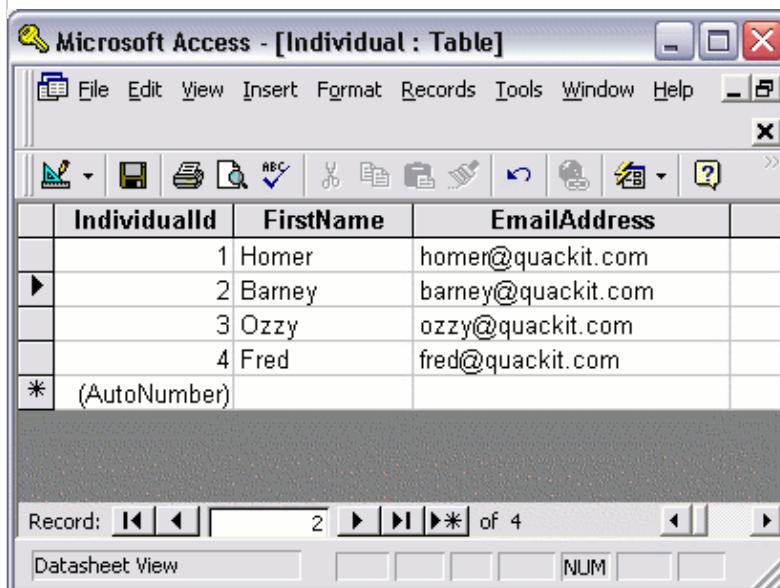
A spreadsheet program like Excel makes these tasks relatively easy to do. Also, programs like Excel organize the data into *rows* and *columns*, making your data easier to comprehend. Something like this:



	A	B	C	D
1	IndividualId	FirstName	EmailAddress	
2	1	Homer	homer@quackit.com	
3	2	Barney	barney@quackit.com	
4	3	Ozzy	ozzy@quackit.com	
5	4	Fred	fred@quackit.com	
6				
7				
8				

Database Software

A better option would be to **store the data in a database table** using specialized database software, such as Microsoft Access. Something like this:



	IndividualId	FirstName	EmailAddress	
	1	Homer	homer@quackit.com	
	2	Barney	barney@quackit.com	
	3	Ozzy	ozzy@quackit.com	
	4	Fred	fred@quackit.com	
*	(AutoNumber)			

Record: 2 of 4

So What's the Difference?

You may be wondering what the difference is between the last two examples (Excel vs Access). After all, both examples have the data organized into rows and columns.

There are many differences between spreadsheet software and database software. The rest of this tutorial will show you why database software is a much better option for creating databases.

- [<< Previous Lesson](#)

- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

Database Home

Database Tutorial

What is a Database?
 Database Management Systems
 Creating a Database
 About Database Tables
 Creating Database Tables
 Adding Data to a Database
 Querying a Database
 Relational Database Design
 Database Driven Website
 Summary

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

Database Management Systems

A **Database Management System (DBMS)** is a software program that enables the creation and management of databases. Generally, these databases will be more complex than the text file/spreadsheet example in the previous lesson. In fact, most of today's database systems are referred to as a **Relational Database Management System (RDBMS)**, because of their ability to **store related data across multiple tables.**

Some of the more popular relational database management systems include:

- Microsoft Access
- Filemaker
- Microsoft **SQL** Server
- **MySQL**
- Oracle

Throughout this tutorial, you will become familiar with some of the key concepts of database management systems. These include:

- Database creation
- Tables
- Adding data to your database
- Querying a database
- Relational database design

What Does a Database Management System Look Like?

Different database management systems look different, but generally, there are a number of common features that you'll usually see across most of them.

Microsoft Access

Web quackit.com

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

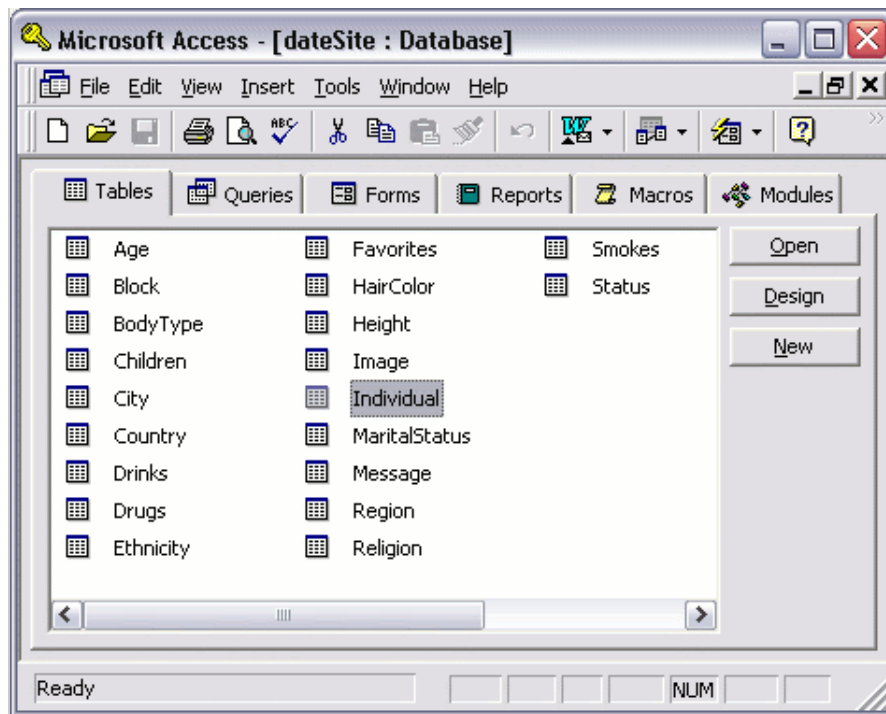
PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

[Get your FREE hosting today!](#)

[>>](#)

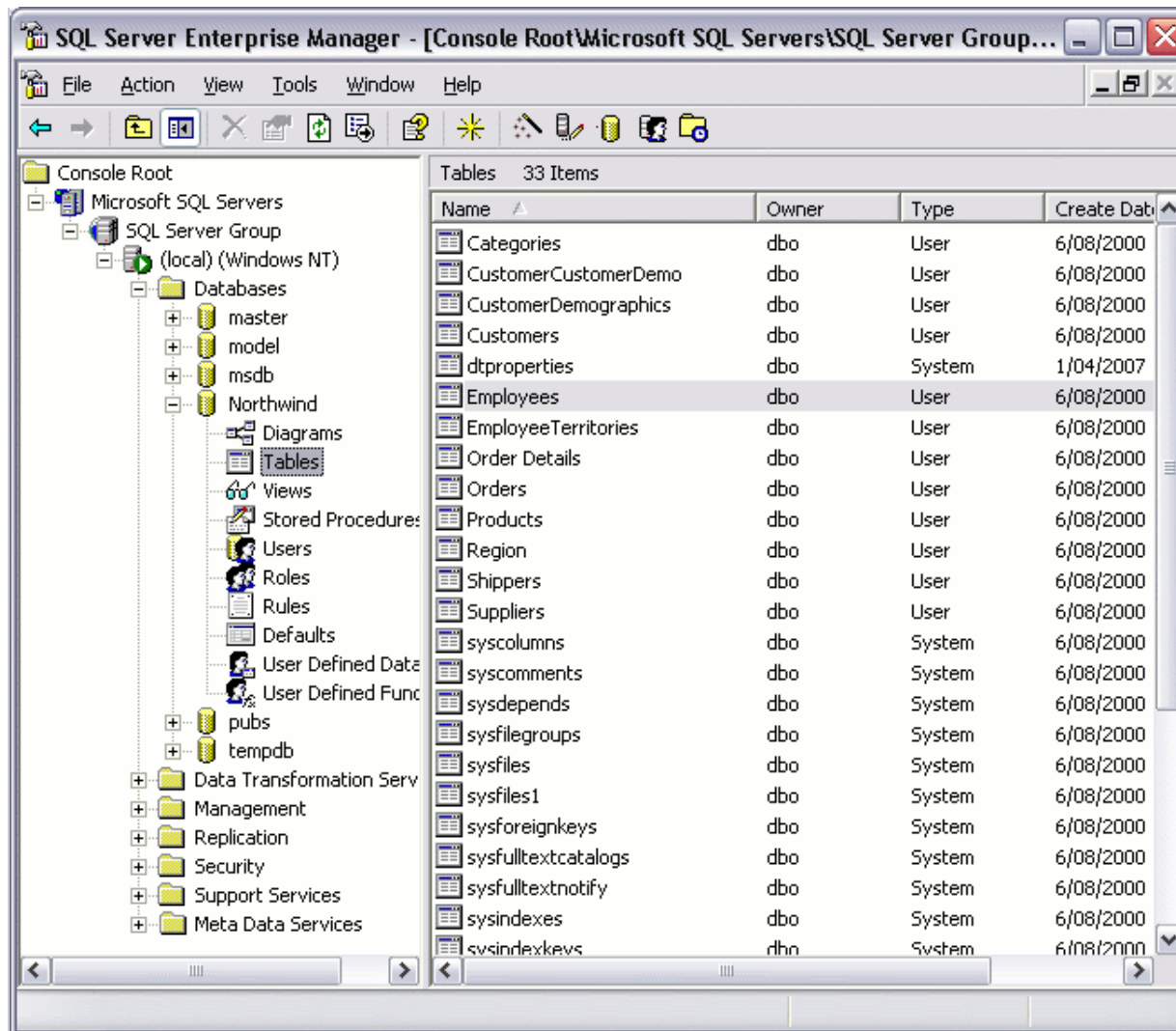
Featured Template:
 (Build your websites in minutes!)



This is the main screen you'll see when opening up Access to view an existing database. The outer part is the database management system and its menu, the middle part is the actual database. In this example, the database is called "dateSite" and has 19 tables. If you were to open a different database, the name of the database would be different and you would see different tables, but the available options would be the same (i.e. Tables, Queries, Forms, Reports, Macros, Modules, Open, Design, New).

Some of these options are common across all database management systems. All database systems allow you to create *tables*, build *queries*, design a new database, and open an existing database.

Microsoft SQL Server



Microsoft SQL Server is a more robust database management system than Access. While Access is better suited to home and small office use, SQL Server is more suited to enterprise applications such as corporate CRMs and websites etc.

The above screen is what you see when you open SQL Server through Enterprise Manager. Enterprise Manager is a built-in tool for managing SQL Server and its databases. In this example, there are 6 databases. Each database is represented down the left pane, and also in the main pane (with a "database" icon).

Which Database System to Use?

If you are using a database for home or small office use, Microsoft Access or Filemaker should be fine. If you need to create a database driven website, then you're better off using a more robust system such as SQL Server, Oracle, or MySQL.

The examples in this tutorial use Microsoft Access. If you don't have Microsoft Access, you should still be able to follow the examples. The tasks we perform are the same tasks you would need to perform regardless of which database management system you use. The key goal with this tutorial is to provide you with an overview of what is involved in creating and maintaining a database.

- [<< Previous Lesson](#)

- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

Database Home

Database Tutorial

[What is a Database?](#)
[Database Management Systems](#)
[Creating a Database](#)
[About Database Tables](#)
[Creating Database Tables](#)
[Adding Data to a Database](#)
[Querying a Database](#)
[Relational Database Design](#)
[Database Driven Website](#)
[Summary](#)

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

Creating a Database

With database management systems, many tasks can be done either via programmatically or a user interface. Creating databases is no exception.

Option 1: Programatically

Many **database administrators (DBAs)** use **Structured Query Language (SQL)** to perform many of their database tasks. To enter SQL, you need to open an interface that allows you to enter your code. For example, if you use SQL Server, you would normally use Query Analyzer.

The following example is the basic code for creating a new database. Parameters can be added to this example if your requirements are more specific.

```
CREATE DATABASE MyDatabase
```

Note: This example assumes you know how to use your database system to run scripts like this. If you don't you, will probably find it easier to use the user interface method (below).

Option 2: User Interface

Most database systems make it very easy to create a database via a user interface. Generally, it's just a matter of selecting an option from a menu, then providing a name for your database.

The following examples demonstrate how to create a database in Microsoft Access.

1. From the "File" menu, click on "New Database":

Web quackit.com

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

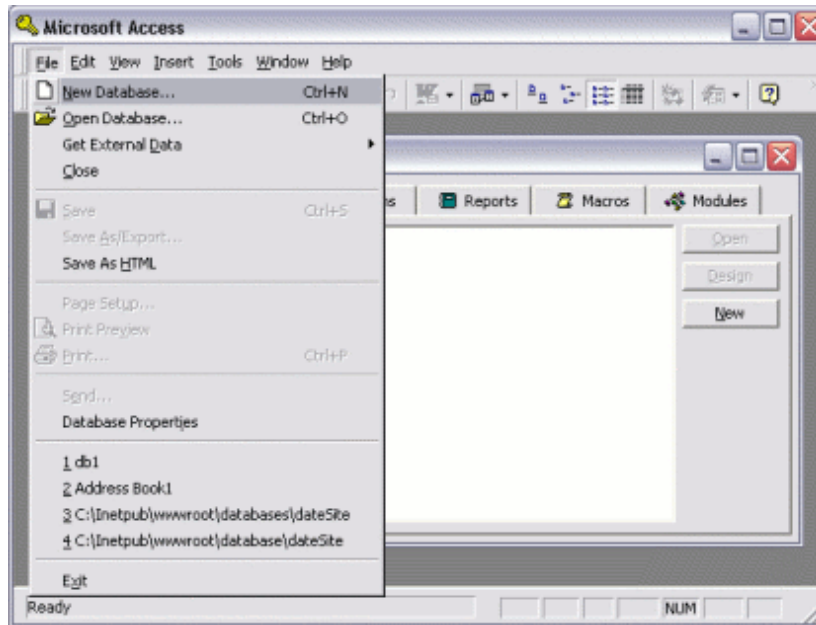
PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

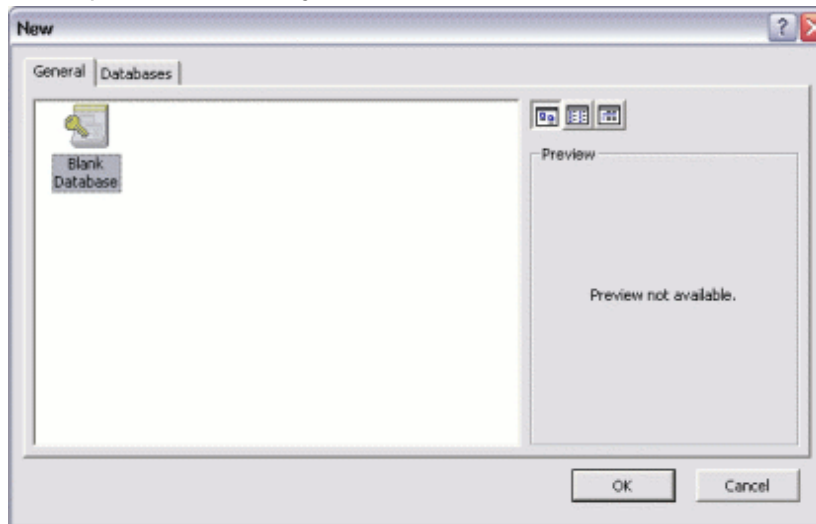
[Get your FREE hosting today!](#)

[>>](#)

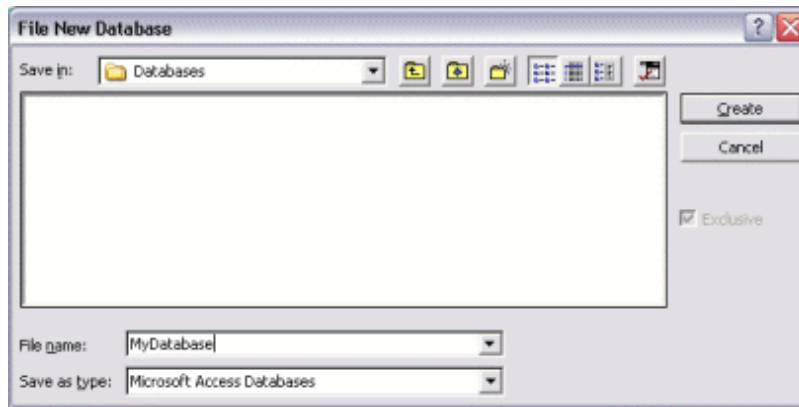
Featured Template:
(Build your websites in minutes!)



2. Choose "Blank Database". (MS Access also gives you the ability to choose from a template, but we'll just use a blank database here):

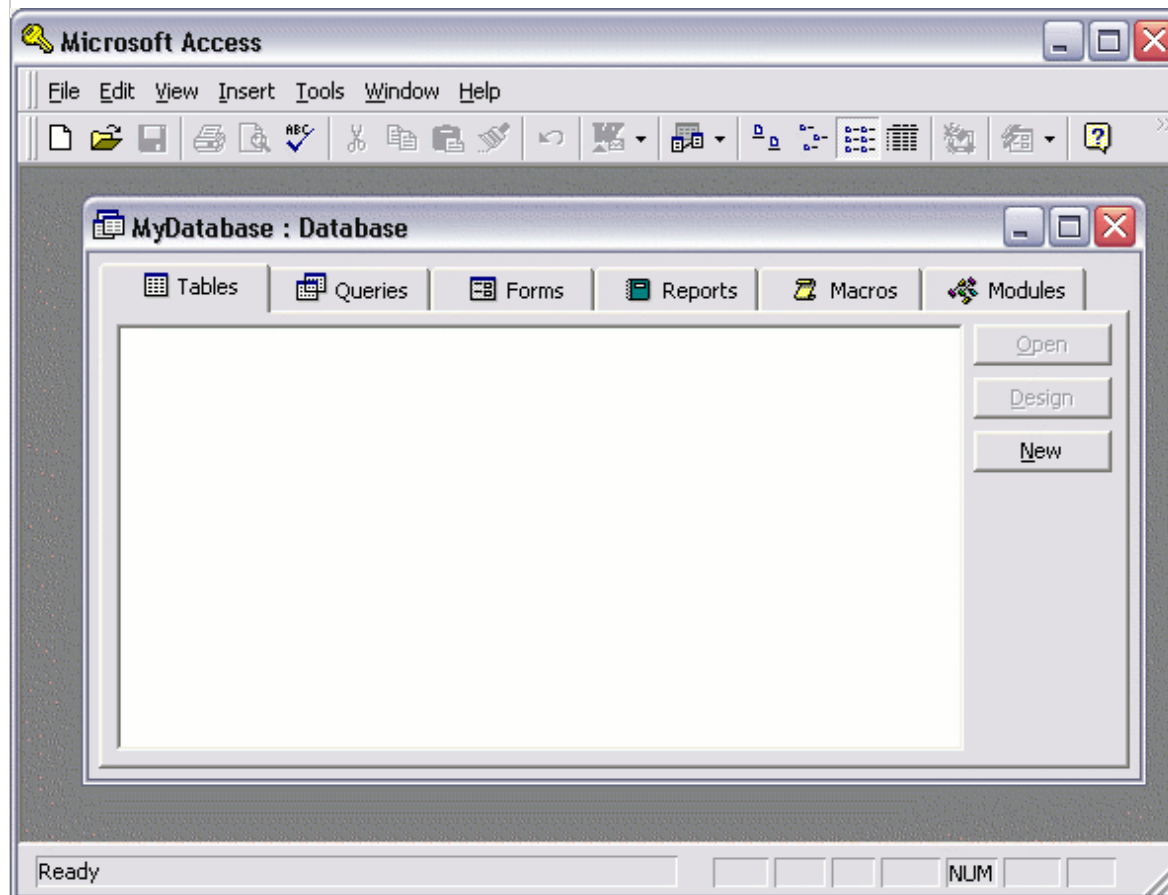


3. Choose a location to save the database:



Your New Database

Once you've completed the above tasks, you should see a blank database, like this:



We know this database is blank because it doesn't have any tables. If it did, you would see these tables in the middle pane of the table tab. Now that we have our blank database, we can start adding some tables.

- [<< Previous Lesson](#)

- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

Database Home

Database Tutorial

[What is a Database?](#)
[Database Management Systems](#)
[Creating a Database](#)
[About Database Tables](#)
[Creating Database Tables](#)
[Adding Data to a Database](#)
[Querying a Database](#)
[Relational Database Design](#)
[Database Driven Website](#)
[Summary](#)

About Database Tables

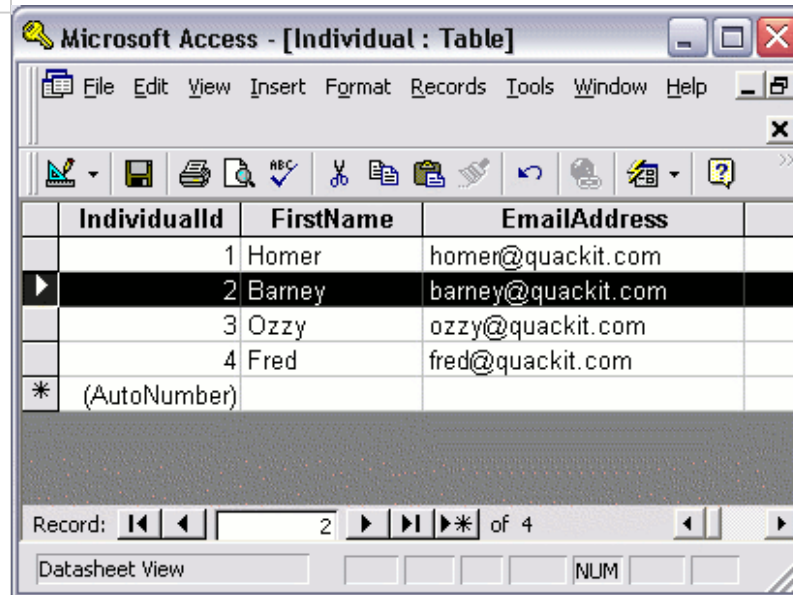


Database tables will most likely be the area you'll become most familiar with after working with databases for a while. Now, before we go ahead and start adding tables to our new database, let's have a look at what a database table actually is.

What is a Table?

In database terms, a **table** is responsible for storing data in the database. Database tables consist of **rows** and **columns**.

In the following example, the second *row* is highlighted in black:



IndividualId	FirstName	EmailAddress
1	Homer	homer@quackit.com
2	Barney	barney@quackit.com
3	Ozzy	ozzy@quackit.com
4	Fred	fred@quackit.com
*	(AutoNumber)	

Record: 2 of 4

Datasheet View

In the next example, the second *column* is highlighted in black. This column has been given a name of "FirstName":

Web quackit.com

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

[Get your FREE hosting today!](#)

>>

Featured Template:
(Build your websites in minutes!)

Microsoft Access - [Individual : Table]

File Edit View Insert Format Records Tools Window Help

IndividualId	FirstName	EmailAddress
1	Homer	homer@quackit.com
2	Barney	barney@quackit.com
3	Ozzy	ozzy@quackit.com
4	Fred	fred@quackit.com
* (AutoNumber)		

Record: 1 of 4

Datasheet View NUM

A row contains each record in the table, and the column is responsible for defining the type of data that goes into each cell. Therefore, if we need to add a new person to our table, we would create a new row with the person's details.

OK, now lets go ahead and [create a table](#).

- [<< Previous Lesson](#)

- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

Database Home

Database Tutorial

What is a Database?

Database Management Systems

Creating a Database

About Database Tables

Creating Database Tables

Adding Data to a Database

Querying a Database

Relational Database Design

Database Driven Website

Summary

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

Creating Database Tables

With database management systems, you need to create your tables before you can enter data. Just as you can create a database programmatically, you can create your tables programmatically too.

Option 1: Programatically

The following is an example of creating a new table. Note that we are specifying the name of the table, the name of each column, and the data type of each column. More parameters can be added to this example if your requirements are more specific.

```
CREATE TABLE Individual
(IndividualId int,
FirstName Varchar(255),
LastName Varchar(255),
DateCreated dateTime
)
```

Option 2: User Interface

Database management systems usually have a "Design View" for creating tables. Design view enables you to create the names of each column, specify the type of data that can go into each column, as well as specifying any other restrictions you'd like to enforce. Restricting the data type for each column is very important and helps maintain *data integrity*. For example, it can prevent us from accidentally entering an email address into a field for storing the current date.

More parameters can be added against each column if you require them.

Web quackit.com

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

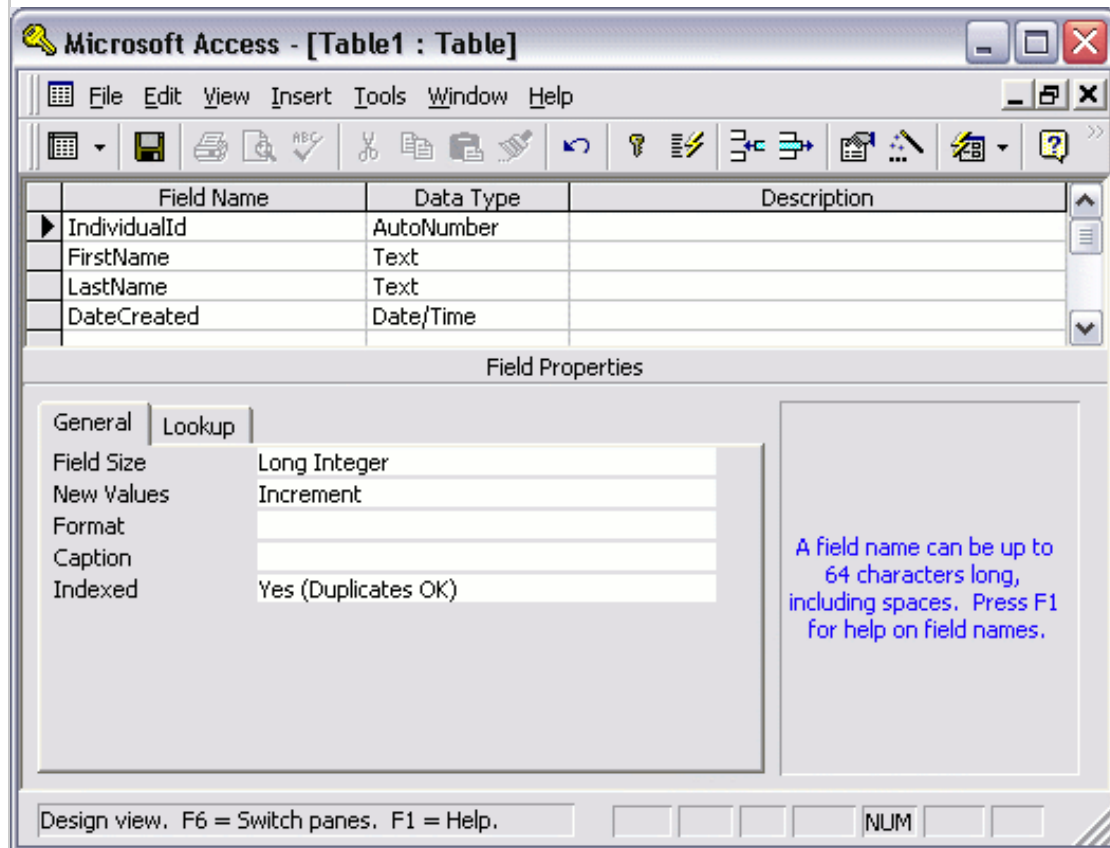
[Get your FREE hosting today!](#)

>>

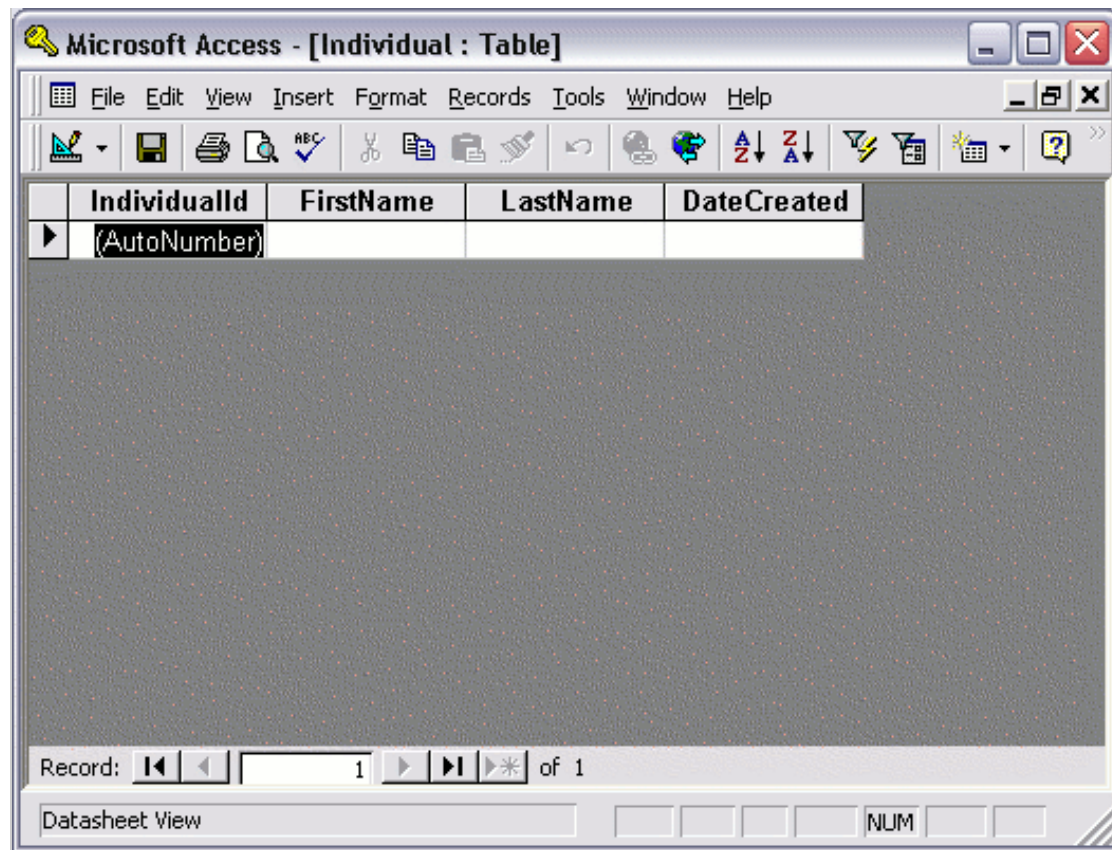
Featured Template:
(Build your websites in minutes!)

For example, you could specify a default value to be used (in case the field has been left blank by the user).

When you create a table via the user interface (or design view), depending on which database system you use, you should see something like this:



Once you've created your table in "design view", you can switch to "datasheet view" to see the resulting table. You should see something like this:



OK, so this is a blank table - it doesn't have any data yet. What we have is a table that contains the columns required before we can enter any data.

So, now that we have a blank table, let's look at how to add data.

- [<< Previous Lesson](#)
- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

[Create a Website](#) | [HTML](#) | [CSS](#) | [MySpace Codes](#) | [JavaScript](#) | [ColdFusion](#) | [PHP](#) | [VBScript](#) | [Databases](#) | [SQL](#) | [SQL Server](#) | [Microsoft Access](#) | [Web Graphics](#) | [Web Design](#) | [Web Hosting](#) | [Web Servers](#) | [XML](#)

[About Quackit](#) | [Terms of use](#) | [Privacy Policy](#) | [Contact Quackit](#)

© Copyright 2000 - 2008 Quackit.com

Database Home

Database Tutorial

What is a Database?

Database Management Systems

Creating a Database

About Database Tables

Creating Database Tables

Adding Data to a Database

Querying a Database

Relational Database Design

Database Driven Website

Summary

Adding Data to a Database

There are a number of ways you can enter data into a database table. The method you choose will largely depend on your context.

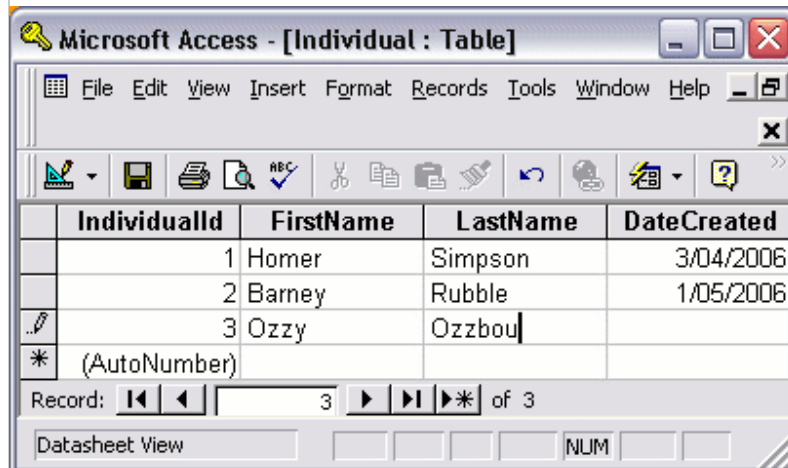
You will need to choose from the following methods:

- Direct entry
- Form
- Structured Query Language (SQL)
- Website or other application

Here's an explanation of those methods.

Direct entry

You can type directly into the table while it's in Data Sheet view. Initially, this may seem like the quickest and easiest method, but it's not suitable if you have lots of data, and/or if non-technical users need to enter data.



Form

If you use a desktop database program (such as MS Access), you can set up a form, so that non-technical users can enter data into the



Web quackit.com

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

[Get your FREE hosting today!](#)
>>

Featured Template:
(Build your websites in minutes!)

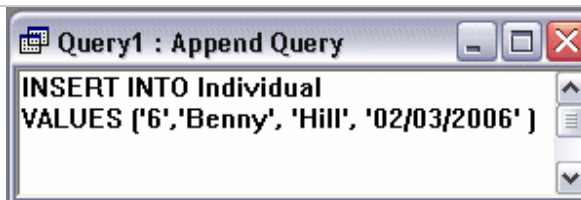
form. Once they submit the form, the data is automatically inserted into our table. The form could insert data into multiple tables too - saving you from having to open up each table to manually insert the data.

Most enterprise database systems don't have the ability for setting up a form. This is probably because they're designed for larger scale applications with hundreds, thousands, or even millions of users. In this environment, a form would be created through other means (for example, using HTML).

Applications such as Access provide a form wizard, which steps you through the process to building a form.

Structured Query Language (SQL)

You can use a programming language called SQL to



insert the data (we could also have used SQL to create the database and tables if we'd wanted to). One advantage of this is that you can save your SQL script for re-use. This could be handy if you need to insert the data into multiple databases.

It's also useful to create scripts that insert "lookup" data - this is generally a base set of data that never changes (such as Countries, Cities, etc). If you ever need to rebuild your database, you can simply run your ready made script against it (which saves you from manually re-entering the data).

Website or other application

You could build a program that uses the database to store and retrieve data. The person entering the data doesn't need to have direct access to the database. They don't even need to have database software. By using SQL, your database could be part of a larger application - such as a website.

This is probably the most common method of adding data to a database. If you've ever registered with a website, your details would have been inserted into a database using this method.

- [<< Previous Lesson](#)

- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

[Create a Website](#) | [HTML](#) | [CSS](#) | [MySpace Codes](#) | [JavaScript](#) | [ColdFusion](#) | [PHP](#) | [VBScript](#) | [Databases](#) | [SQL](#) | [SQL Server](#) | [Microsoft Access](#) | [Web Graphics](#) | [Web Design](#) | [Web Hosting](#) | [Web Servers](#) | [XML](#)

[About Quackit](#) | [Terms of use](#) | [Privacy Policy](#) | [Contact Quackit](#)

© Copyright 2000 - 2008 Quackit.com

Database Home

Database Tutorial

[What is a Database?](#)
[Database Management Systems](#)
[Creating a Database](#)
[About Database Tables](#)
[Creating Database Tables](#)
[Adding Data to a Database](#)
[Querying a Database](#)
[Relational Database Design](#)
[Database Driven Website](#)
[Summary](#)

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

Querying a Database

Queries are one of the things that make databases so powerful. A "query" refers to the action of retrieving data from your database. Usually, you will be selective with how much data you want returned. If you have a lot of data in your database, you probably don't want to see everything. More likely, you'll only want to see data that fits a certain criteria.

For example, you might only want to see how many individuals in your database live in a given city. Or you might only want to see which individuals have registered with your database within a given time period.

As with many other tasks, you can query a database either programatically or via a user interface.

Option 1: Programatically

The way to retrieve data from your database with SQL is to use the "SELECT" statement.

Using the SELECT statement, you can retrieve all records...

```
SELECT * FROM Individual
```

...or just some of the records:

```
SELECT * FROM Individual
WHERE FirstName = 'Homer'
```

The 2nd query only returns records where the value in the "FirstName" column equals "Homer". Therefore, if only one individual in our database had the name

Web quackit.com

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

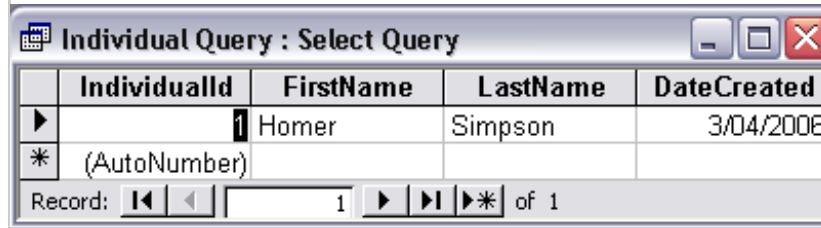
[Get your FREE hosting today!](#)

>>

Featured Template:
(Build your websites in minutes!)

"Homer", that person's record would be shown.

Something like this:



IndividualId	FirstName	LastName	DateCreated
(AutoNumber)	Homer	Simpson	3/04/2006

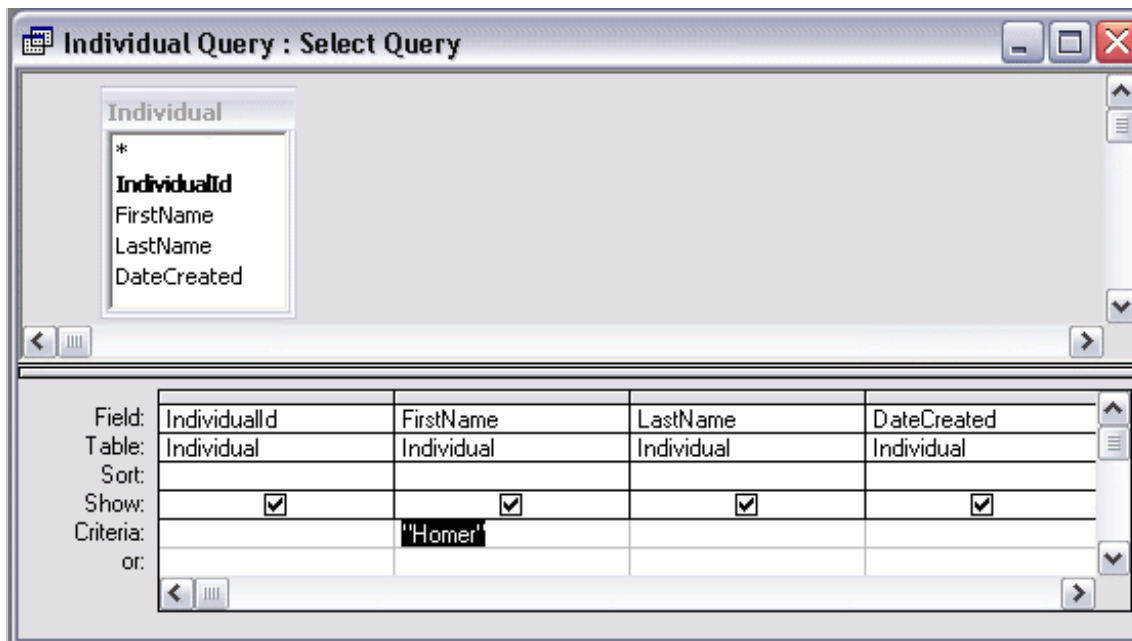
SQL is a powerful language and the above statement is very simple. You can use SQL to choose which columns you want to display, you could add further criteria, and you can even query multiple tables at the same time. If you're interested in learning more about SQL, be sure to check out our SQL tutorial after you've finished this one!

Option 2: User Interface

You might find the user interface easier to generate your queries, especially if they are complex.

Database management systems usually offer a "design view" for your queries. Design view enables you to pick and choose which columns you want to display and what criteria you'd like to use to filter the data.

Here's an example of design view in Microsoft Access:



When using design view, the database system actually uses SQL (behind the scenes) to generate the query.

- [<< Previous Lesson](#)
- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

[Create a Website](#) | [HTML](#) | [CSS](#) | [MySpace Codes](#) | [JavaScript](#) | [ColdFusion](#) | [PHP](#) | [VBScript](#) | [Databases](#) | [SQL](#) | [SQL Server](#) | [Microsoft Access](#) | [Web Graphics](#) | [Web Design](#) | [Web Hosting](#) | [Web Servers](#) | [XML](#)

[About Quackit](#) | [Terms of use](#) | [Privacy Policy](#) | [Contact Quackit](#)

© Copyright 2000 - 2008 Quackit.com

Database Home

Database Tutorial

What is a Database?

Database Management Systems

Creating a Database

About Database Tables

Creating Database Tables

Adding Data to a Database

Querying a Database

Relational Database Design

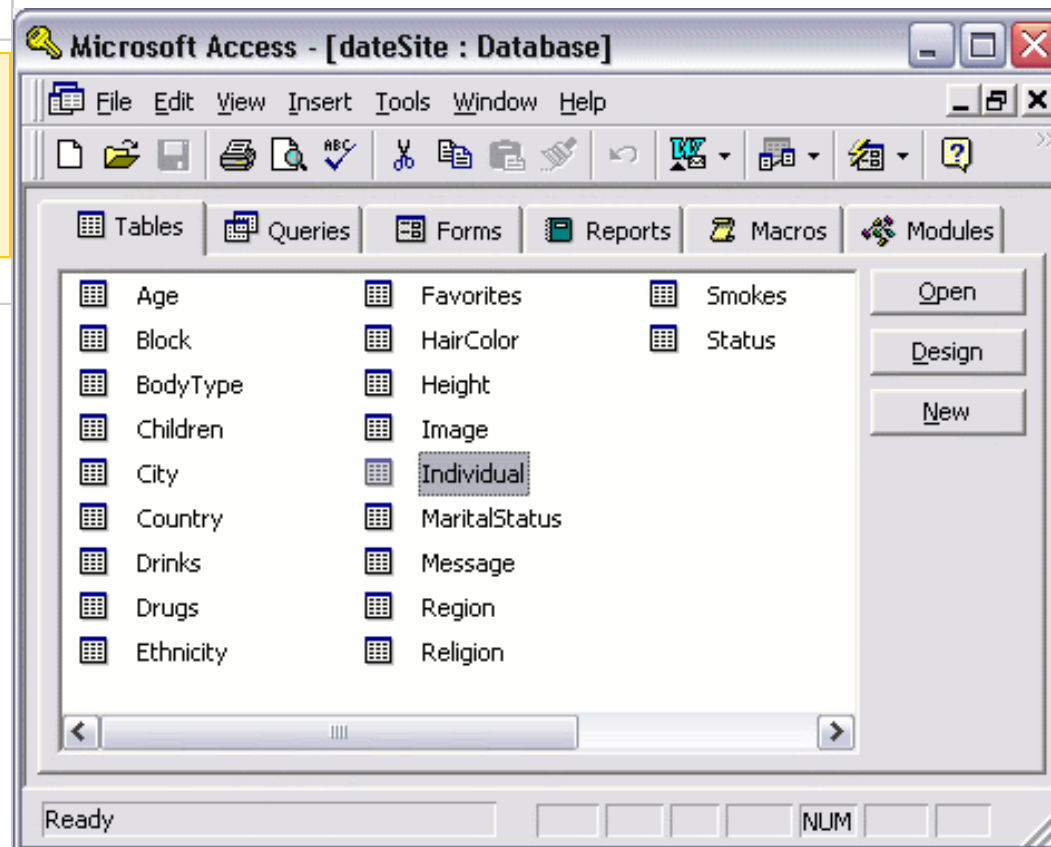
Database Driven Website

Summary

Relational Database Design

Most popular database management systems are *relational systems*, and are usually referred to as **Relational Database Management Systems (RDBMS)**. What this means is that their databases can contain **multiple tables**, some (or all) of which are related to each other.

For example, consider the following screenshot:



In this example, **the database has 19 tables.** Each table serves a

Web quackit.com

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

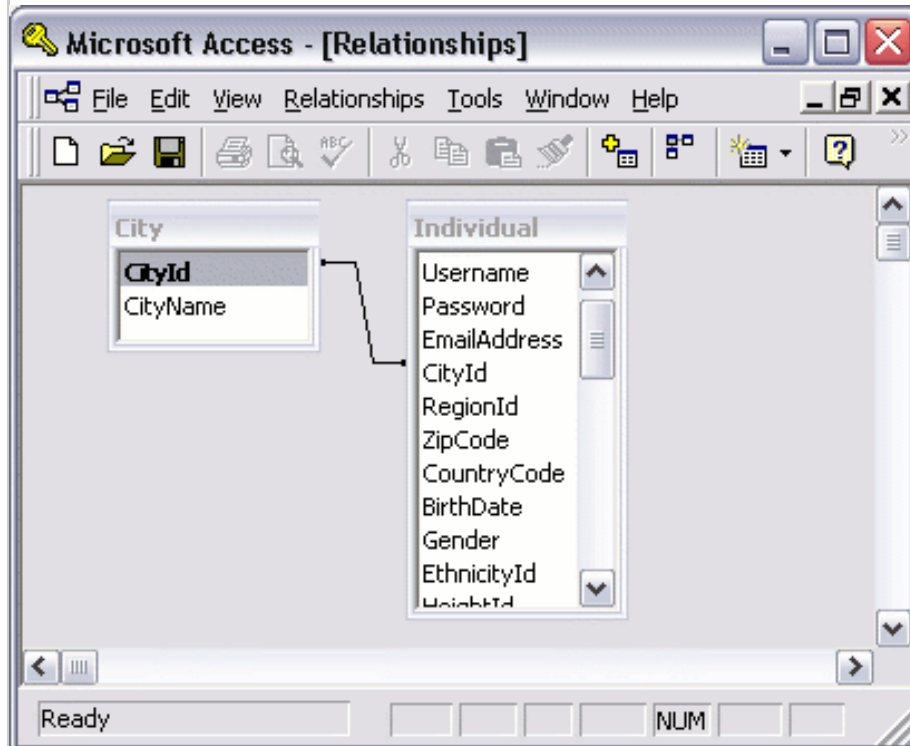
[Get your FREE hosting today!](#)

[>>](#)

Featured Template:
(Build your websites in minutes!)

specific purpose. This enables us to organize our data much better. It can also help us with the *integrity* of our data.

Using the example above, the Individual table can hold data that is strictly about the individual. The City table can hold a list of all cities. If we want to know which city each individual lives, we could store a "pointer" in the Individual table to that city in the City table.



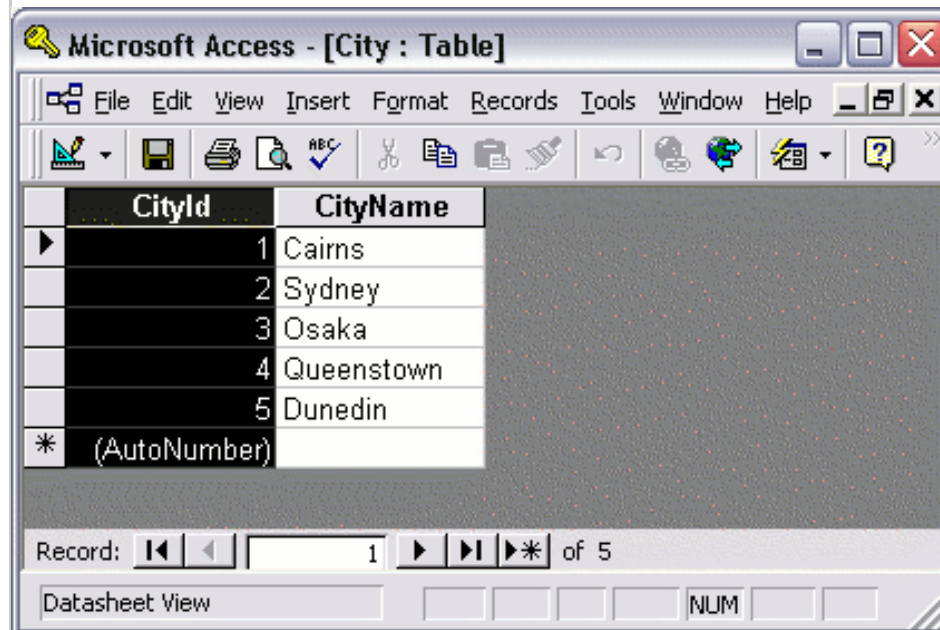
The above example demonstrates the relationship between the Individual table and the City table. The individuals in the "Individual" table live in cities that are defined in the "City" table. Therefore, we can cross-reference each "Individual" record with a "City" record.

How Does This Work?

Firstly, in the **City table, each record has a unique identifier.** A unique identifier is a value that is unique to each record. This identifier can be as simple as an incrementing number. So, in our City table, the **first record has a number of 1, the second record has a number of 2, and so on.**

Secondly, when entering each individual into the Individual table, **instead of writing out the full city name in that table, we only need to add the city's unique identifier.** In this case, the unique identifier is a number, so we enter this number into the **"CityId" column** of the "Individual" table.

The following screenshots demonstrate this:



The screenshot shows a Microsoft Access window titled "Microsoft Access - [City : Table]". The window displays a table with two columns: "CityId" and "CityName". The "CityId" column is highlighted as an AutoNumber field. The table contains five records:

CityId	CityName
1	Cairns
2	Sydney
3	Osaka
4	Queenstown
5	Dunedin

The "CityId" column is highlighted as an AutoNumber field. The record navigation bar shows "Record: 1 of 5". The "Datasheet View" button is visible at the bottom left, and the "NUM" button is visible at the bottom right.

IndividualId	FirstName	EmailAddress	CityId
1	Homer	homer@quackit.com	2
2	Barney	barney@quackit.com	1
3	Ozzy	ozzy@quackit.com	3
4	Fred	fred@quackit.com	3
*	(AutoNumber)		0

So, by looking at both tables, we can determine that Homer lives in Sydney, Barney lives in Cairns, and both Ozzy and Fred live in Osaka. At this stage, nobody in our database lives in Queenstown or Dunedin.

Primary Keys and Foreign Keys

Primary keys and foreign keys are terms that you will become very familiar with when designing databases. These terms describe what role each of the columns play in their relationship with each other.

The column that contains the unique identifier is referred to as the *Primary Key*. So, in our City table, the primary key is the CityId column.

A *foreign key* is the column in the other table that points to the primary key. Therefore, the CityId column in the Individual table is a foreign key to the CityId column in the City table.

Lesson

• [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

[Create a Website](#) | [HTML](#) | [CSS](#) | [MySpace Codes](#) | [JavaScript](#) | [ColdFusion](#) | [PHP](#) | [VBScript](#) | [Databases](#) | [SQL](#) | [SQL Server](#) | [Microsoft Access](#) | [Web Graphics](#) | [Web Design](#) | [Web Hosting](#) | [Web Servers](#) | [XML](#)

[About Quackit](#) | [Terms of use](#) | [Privacy Policy](#) | [Contact Quackit](#)

© Copyright 2000 - 2008 Quackit.com

Database Home

Database Tutorial

[What is a Database?](#)
[Database Management Systems](#)
[Creating a Database](#)
[About Database Tables](#)
[Creating Database Tables](#)
[Adding Data to a Database](#)
[Querying a Database](#)
[Relational Database Design](#)
[Database Driven Website](#)
[Summary](#)

Database Driven Website



A database driven website is a website that has most of its webpage content in a database. Therefore, the website content isn't actually sitting in files on the server, it is sitting in tables and columns in a database.

A website with its content stored on the file system is often referred to as a *static* website, whereas a database driven website is often referred to as a *dynamic*.

Content Management Systems

A website with dynamic content usually has a **CMS (Content Management System)** to assist the content providers in updating the website.

A CMS is usually provided in the form of an administration area where content providers need to log in before they can add content. Once logged in, they can create, update and delete articles. They may be able to upload files such as Word documents, PDF files etc. They might be able to upload images too.

All of this content can be stored in the database. Some may be stored on the file system too though. For example, although documents and images can be stored in the database, there are sometimes reasons to store them on the file system. Performance is often a key reason. Database size is another.

Discussion Forums and Blogs

Discussion forums and blogs have become a popular feature for many websites. Most, if not all, forums and blogs are database driven.

Web quackit.com

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

[Get your FREE hosting today!](#)

>>

Featured Template:
(Build your websites in minutes!)

Users can register their details, then add content. When the user clicks the "Submit" button, their details/content is inserted into the database. Then when someone decides to view this content, it is read from the database using SQL (Structured Query Language).

Combination of Static and Dynamic

Some websites have a combination of static content and dynamic content. There could be any number of reasons for this. Often, smaller websites will be static. There's little need to configure a database just to store a handful of webpages - much easier and cheaper to keep them as files on the server. Even websites like this might contain some added functionality such as a discussion forum, or a blog. In this case, the discussion forum or blog will need its content stored in a database.

Benefits of a Database Driven Website

Database driven websites can provide much more functionality than a static site can.

Extended functionality could include:

- Enabling many (potentially non-technical) users to provide content for the website. Users can publish articles on the website without needing to FTP them to a web server.
- Shopping cart
- You can provide advanced search functionality that enables users to filter the results based on a given field. They can then sort those results by a field - say "Price" or "Date".
- Customized homepage
- You can allow your users to perform tasks such as registering for a newsletter, post questions to your forums, provide comments on a blog, update their profile,

etc.

- Integration with corporate applications such as CRM systems, HR systems etc
- Much more

Creating a Database Driven Website

The most common tasks for database driven websites is inserting, updating, and deleting data. Some of these are the same tasks that you learned in this tutorial, however when using a database driven website, you need to use a different method to do these tasks. You need to use a programming language called SQL (Structured Query Language) to insert, update, and delete your data.

Don't worry, this is not as scary as it may sound. **SQL is a very easy language** to learn and, once you start using it, you will be thankful you took the time to learn it. In fact, you've already learned some basic SQL statements in previous lessons.

To create a database driven website, you need the following skills:

- You need to be able to build a static website HTML, and preferably CSS and JavaScript
- You need to be able to write basic code using a server side scripting language such as PHP, ColdFusion etc
- You need to know how to write basic SQL
- You need to be able to design/build a database. If you need to learn about databases but missed the start of this tutorial, [here it is](#) :)

If you're familiar with PHP or ColdFusion, check out how to connect to and query a database in each of these technologies.

- [Querying a database using ColdFusion](#)
- [Querying a database using PHP](#)

- [<< Previous Lesson](#)

- [Next Lesson >>](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

Database Home

Database Tutorial

[What is a Database?](#)
[Database Management Systems](#)
[Creating a Database](#)
[About Database Tables](#)
[Creating Database Tables](#)
[Adding Data to a Database](#)
[Querying a Database](#)
[Relational Database Design](#)
[Database Driven Website](#)
[Summary](#)

Database Summary



So, you've made it to the last page of this database tutorial... Well done!

You should now have a general understanding about databases and how they're used. This tutorial was intended for beginners trying to gain an understanding of databases. Databases are not like most other files and require a little bit of thought in order for you to understand the concept.

If you have your own database management system installed, you should have a better understanding of where to start and what the various options mean.

What Next?

Most of the examples in this tutorial used Microsoft Access. If you'd like to learn how to use Microsoft Access, try the [Microsoft Access Tutorial](#).

Also, you'll have noticed SQL coming up throughout this tutorial. SQL is a very powerful language, but is also very easy to learn. You can achieve a lot even by learning just a little SQL. If you're interested in learning more about SQL, check out the [SQL tutorial](#). Once you complete this tutorial, you will be able to do things such as:

- Select only the columns you want from a query
- Query multiple tables
- Create databases programatically
- Create tables programatically
- Query multiple tables

Web quackit.com

Got a MySpace Page?

Get "www.yourname.com" for your MySpace page. [Learn how >>](#).

FREE Hosting!

With every domain name you register with [ZappyHost](#), you get FREE (ad supported) hosting.

PLUS you get:

- FREE Website Builder
- FREE Blog
- FREE Starter Web Page
- FREE Email Account
- & much more! (Total value \$123 for FREE!)

[Get your FREE hosting today!](#)

>>

Featured Template:
(Build your websites in minutes!)

- Use built-in functions
- Create an index
- And more...

Sounds good? [Learn SQL now!](#)

• << [Previous Lesson](#)

Enjoy this website?

1. Link to this page (copy/paste into your own website or blog):

2. Add this page to your favorite social bookmarks sites:



3. Add this page to your [Favorites](#)

Oh, and thank you for supporting Quackit!

[Create a Website](#) | [HTML](#) | [CSS](#) | [MySpace Codes](#) | [JavaScript](#) | [ColdFusion](#) | [PHP](#) | [VBScript](#) | [Databases](#) | [SQL](#) | [SQL Server](#) | [Microsoft Access](#) | [Web Graphics](#) | [Web Design](#) | [Web Hosting](#) | [Web Servers](#) | [XML](#)

[About Quackit](#) | [Terms of use](#) | [Privacy Policy](#) | [Contact Quackit](#)

© Copyright 2000 - 2008 Quackit.com