Chapter I: Databases - In Theory and Everyday Life

What is a Database?

The word *database* is not one that you hear in everyday conversation. It's one of those technical terms that's used by business and I.T. people that might evoke images of computers and long reports of names and numbers or indecipherable data. Some people might think of marketing or mailing lists. In fact, *a database is simply any collection of data that's organized so that it can be retrieved and used as needed*. Usually, it refers specifically to data that has been stored within a computer system so that it can be quickly manipulated into reports. Databases take many forms but any time a computer needs to present information of any kind, whether it be a store's customer data or patient data at your doctor's office, there's usually a database involved.

Everyday Database Examples

Our everyday lives are constantly influenced by electronic data from the web pages and e-mails on our computers to the uplink at a credit card terminal. In June 2011, the EMC corporation, a worldwide I.T. consulting firm, stated that the world's collection of electronic data was doubling every two years. They also forecasted that during 2011, 1.8 zettabytes of data would be created or copied. A zettabyte is 1 *billion* gigabytes of information. To show you what that figure means, the average new home computer in 2012 might have included 500 gigabytes of storage on the hard drive. That means the amount of data estimated to be generated in 2011 would fill 3.6 million home computers or more than 212 million DVDs. That's a lot of data to store and the amount of data being generated is increasing every year.

A large portion of this data is stored in separate files such as Microsoft Word documents and image files but much of it needs to be stored in a way that can be quickly accessed and searched by record. For some examples of this, let's look at the different ways you might access databases throughout the day.

- You get up in the morning and, if you're like me, you check your e-mail. That means your computer or phone connects to whatever service you use and requests a list of new e-mails that have come in since you checked last. Your program retrieves at least the essential parts of the e-mail including the sender's address and subject line. All of this information is stored within a database on your e-mail service.
- Many of your morning e-mails are probably spam which means that your e-mail address is stored in the database of an online marketer somewhere, probably several.
- After you get the incoming e-mail, you decide to send an e-mail to your friend, Bob, asking if he wants to get together after work for a movie. In the "Send To" field, you type in the first characters of Bob's name and the program accesses your electronic address book to get Bob's e-mail address. Address books and Rolodexes have always been a type of database with fields for the name, phone number, address, etc.. Now they're often *electronic* databases that can be quickly searched and updated.
- On your way into work, you stop at the local coffee shop to get your coffee and danish. When the cashier swipes your card, the software requests your record from the credit card company's database to verify that the card number you just provided is valid. If you stop for gas, the pump

might verify your ZIP code when you swipe your card. The ZIP code that it verifies against is also part of the database record from your credit card company.

- If the building you work in has a moderate amount of security, you might have to punch in a code, swipe a security badge or even scan your fingerprint to get in the door before you can get to your desk or work site. The system then consults a security database in which you are, hopefully, listed as a current and authorized employee and decides whether to admit you or make your morning more challenging. Depending on your company's policies, the card swipe and its result might be recorded back to the security database for reference.
- If you use a computer at work, the programs you work with access data from company databases to provide the customer, order, shipping, employee or financial information that you use in your job. The computer itself, your printer and the phone you use might be listed in one or more inventory and administrative databases on your company's network.
- One or several times that day, you probably read and post comments to Facebook, Twitter and other social networks. All of these comments and their attachments such as photos and videos are stored in databases that power the various websites.
- Finally, on your way home for the night, you stop by the pharmacy and the grocery store. The pharmacist uses a database, likely more than one, to verify the details of your prescription order and check for any interactions or side effects you need to know about. The clerk at the store scans the barcodes on your items which are simply numbers that the register uses to look up your items in a database to get the right price and adjust inventory.
- You probably used your credit card again at the grocery and pharmacy. These transactions are stored in a database on your credit card company's system along with the ones for the the morning coffee, your lunch that day at a restaurant, the tank of gas and the movie tickets. All the items can then be included on your statement at the end of the month.

In most of these examples, I talked about single databases being used to store data but since companies and government agencies love to keep information, the data can be linked to, duplicated, manipulated and transferred into other databases many times over. As I've *repeatedly* warned people on a certain social network, once you put your information online, you lose all control of where it will go and who might use it. Essentially, you pay for the convenience of having your data available to the services you use by potentially sharing it with anyone else who might find it useful.

My real point here is that, in this electronic age, more and more of your daily activities generate or rely on information stored in databases of one kind or another. Unless you are completely off the grid, your daily life follows and leaves an electronic trail. It is in your best interests to learn how databases work so that you will be aware of how information is stored and shared between systems.

Types of Databases

As I mentioned earlier, even your address book qualifies as a database. It's a list of records with a set of fields for names, addresses and other bits of contact information. It has a structure which you can use to quickly record and retrieve the information you need. Electronic databases can be as simple as a text file on a local computer where the information is stored in rows of text. They can also be extremely complex and hosted on servers where they're accessed by multiple programs from around the world. It all depends on the requirements of the people using the data. The following are a few types of databases that you might see in everyday life.

Text files - Data Exchange and Basic Storage

The simplest form of electronic database can be stored on a local computer or network in a plain text file. While providing the fewest features in terms of sorting and indexing, plain text formats are the most compatible with different software systems and are also readable by humans. Three prominent formats

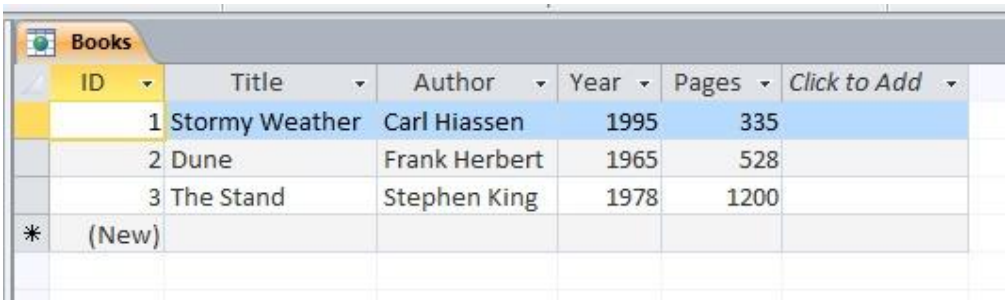
of text-based databases right now are CSV, XML and JSON as I'll explain below.

CSV

CSV stands for *Comma Separated Values*. It's the simplest database format in which the data is stored in a consistent set of fields and separated by commas or other separators such as spaces and tabs. If you were to store a book collection in this format with the title, author, release year and pages, it might look like this:

```
"Stormy Weather","Carl Hiassen",1995,335
"Dune","Frank Herbert",1965,528
"The Stand","Stephen King",1978,1200
```

Other names for this format include *comma delimited data* and *tabular data*. It's a format that goes back decades to a time before the personal computer. CSV can still be used to transfer data between programs. For example, you could save the data from your favorite spreadsheet program to CSV and import it into another analysis program that did not read the original format but could import CSV data.



Books

ID	Title	Author	Year	Pages	Click to Add
1	Stormy Weather	Carl Hiassen	1995	335	
2	Dune	Frank Herbert	1965	528	
3	The Stand	Stephen King	1978	1200	
*	(New)				

Figure 1.1 - CSV is commonly used for the transfer of data between programs but can also be used to store and work with small amounts of data.

XML

A newer standard of text data is the *XML* format. XML stands for *Extensible Markup Language* and was developed in the 1990s mainly for data exchange over the Internet. Figure 1.2 shows a sample of XML based on the book list in Figure 1.1.

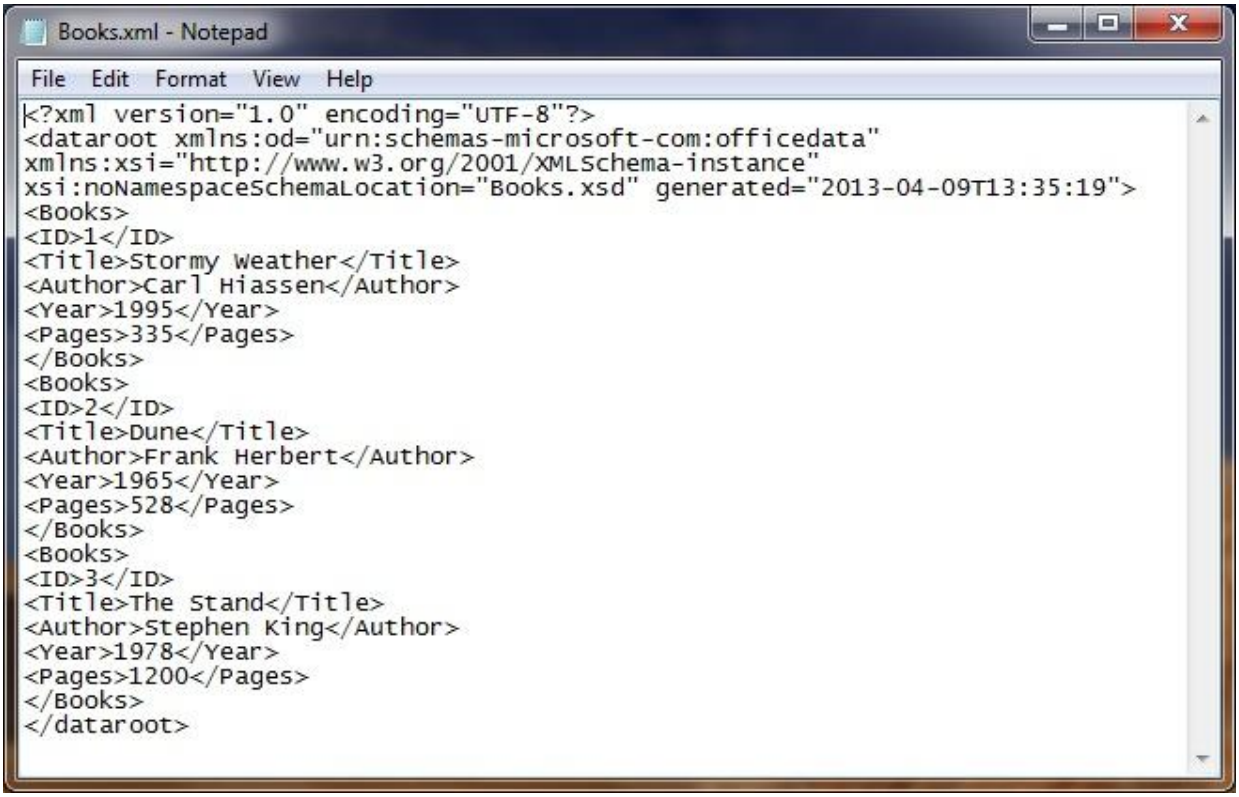


Figure 1.2 - XML can be used to store and transfer data in a structured format.

XML gets its name from the fact that the various fields are marked with the tags that you see indicating the field names such as Title and Author. The word "extensible" means that there is no limitation on what tags can be used. Unlike HTML, the markup language used to create web pages, the tags in an XML file are determined by the user based on the needs of the data. Extra documentation might be included in an accompanying XSD file to provide more information on the structure of the data.

XML has a couple of advantages over CSV. First, it can handle more complex data such as categories and sub-categories of report data. For example, if you had a list of employees and each employee had a list of absentee days, that data could be exported to one XML file and then imported to another program. The link between each employee and his or her specific absences would remain intact. Unlike some other data formats, XML is also readable by both human and machine although it can look a little confusing to a person who is not familiar with the initial data. The final advantage is that when transmitting data over the Internet, XML can go through security precautions such as firewalls and e-mail filters without posing any risk to the system receiving it because it's simply a text file. Other *binary* formats like spreadsheet and word processing files can carry viruses and are often blocked by filtering systems.

XML really shines when it comes to transmitting continuous record-based data that needs to be regularly updated and synchronized. Years ago, I worked with an automated system that maintained an inventory of the computer workstations in the company. Every time a user turned on their computer, an XML file would be generated by a small client program on their machine with their computer name and current specifications. The file would be sent to the main program on the network and the program would update its database of workstations with this record. When we found that the program wasn't doing such a great job of updating the inventory, it was easy for me to design another database program that would import the XML files and maintain a better inventory.

JSON

The *JSON* format is the newest of the three text formats, having been developed starting in 2001 by Douglas Crockford at his company, State Software, Inc.. It was developed as a data interchange format which would be easily readable and writable by both humans and computers and is used by some

applications as an alternative to XML. The name stands for *JavaScript Object Notation* as the format is based on the JavaScript web programming language. Unlike XML, JSON recognizes actual data types such as numbers, strings and True / False values. The format can specify individual named values as well as arrays of values and it enables highly structured data to be written to a plain text format. Figure 1.3 shows a sample of JSON based on the previous list of books.

```
[
        {
                "Title": "Stormy Weather",
                "Author": "Carl Hiassen",
                "Year": "1995",
                "Pages": "335"
        },
        {
                "Title": "Dune",
                "Author": "Frank Herbert",
                "Year": "1965",
                "Pages": "528"
        },
        {
                "Title": "The Stand",
                "Author": "Stephen King",
                "Year": "1978",
                "Pages": "1200"
        }
]
```

Figure 1.3 - A sample of JSON used to store book information

As database formats, all three of the above formats are limited by the fact that they are stored within text files. The program generally has to import the entire file to work with the data. There is no security on a text file meaning that it can be read by anyone and if it's deleted or corrupted, the data is lost. For these reasons, most programs that work with more than small amounts of data use other formats.

Mobile Databases - Smartphones, Tablets and the Web

Most sophisticated computer programs, including those on your smartphone or tablet, need something more than a text file to store their data. At a certain point, the data needs to conform to a structure and a set of rules. Relationships between different categories of data must be enforced, the data must be easily searched and the program must be able to access the right information without having to sift through an entire collection. These requirements are common enough to different programs that they can be delegated to a separate software which will manage the database and provide access to the data. This software is often written independently from the programs that you use on a daily basis. If the program you're using is well designed, you should never even be aware of the existence of the database manager itself.

There are many database software titles available. Here are just a couple with examples of where they're being used. See the links section at the end of the chapter for the websites associated with the software titles listed here.

SQLite

SQLite (pronounced SEE-kwel LITE) is a public domain database software promoted for its small size, speed and reliability. These are definitely considerations for programmers who want to design small, efficient programs and make their users happy. Since it's public domain, programmers can design their programs around an SQLite database without having to pay any licensing fees. It's used by such well known programs as Mozilla Firefox and McAfee anti-virus.

Microsoft SQL Server Compact and Express Editions

Microsoft's SQL Server is one of the big names in the database world with various editions of the software being used for everything from small mobile apps to giant business systems. The Compact and Express editions of the software are designed specifically for smaller applications. The Compact edition focuses on mobile apps and the Express edition is used for desktop and website applications. These editions of the software are also free for programmers to use and are sometimes bundled with other programming tools. If you are developing websites on a hosting service or network powered by Microsoft Windows, there's a good chance that you're also using SQL Server Express for your data needs.

Oracle Database Express Edition

Oracle is another heavyweight in the database field. The Express edition of their self-titled database software can be used to create software and for software testing by programmers. Oracle Express also has limitations on the amount of data it can store and the amount of computer memory it can access but, for the average developer or user with a single machine installation, it's enough to work with.

MySQL

MySQL is one of the most popular software titles for use with web applications. It's an *open source* software which means that the source code is available for independent programmers to modify as needed and also very often means that the software is free. MySQL is cross-platform meaning that it runs on different operating systems including Windows, Linux and OS X. In my experience, MySQL and Microsoft's SQL Server Express are the two primary database systems for use with web applications. If you build your own website and sign up for a website hosting account, you'll often see one or both of these offered for use in storing whatever data your site needs to access. WordPress, the popular blogging tool and content management system, uses a MySQL database in order to store the data for the site including posts and other content. MySQL also does not have the limitations on database size that SQL Server Express and Oracle Express do and the databases can run into the terabytes (*trillions* of bytes).

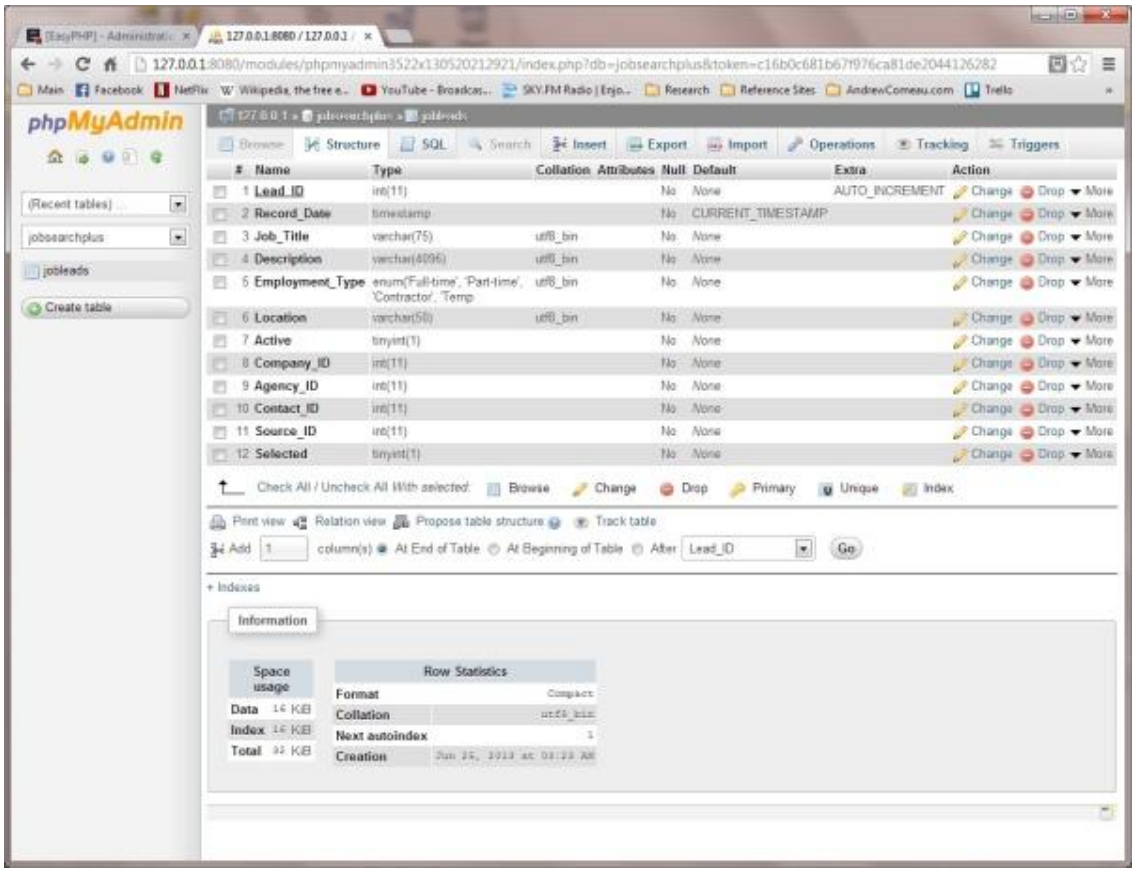


Figure 1.4 - MySQL is a popular database for use with web applications and is often administered in a web browser with the phpMyAdmin web application.

Desktop Database Software - Local Analysis

Another type of database software enables individual, non-technical users to create databases and analyze data. Several software titles offer powerful data analysis and reporting features that can be run straight from the desktop. MySQL, SQL Server Express and Oracle Express might qualify for the advanced user as they can be installed locally and used to create databases but others are designed for the average business user or even the home user and are often packaged as part of a larger suite of applications. This includes programs like Microsoft Access and OpenOffice Base. These programs enable the user to create databases in one or more files on the user's computer, analyze the data in various ways and create reports. The software can even link to other spreadsheet and analysis programs to pull additional data into the analysis. Some desktop databases include advanced features such as embedded programming languages and the ability to package user-designed databases to be run on remote machines and in Web applications.

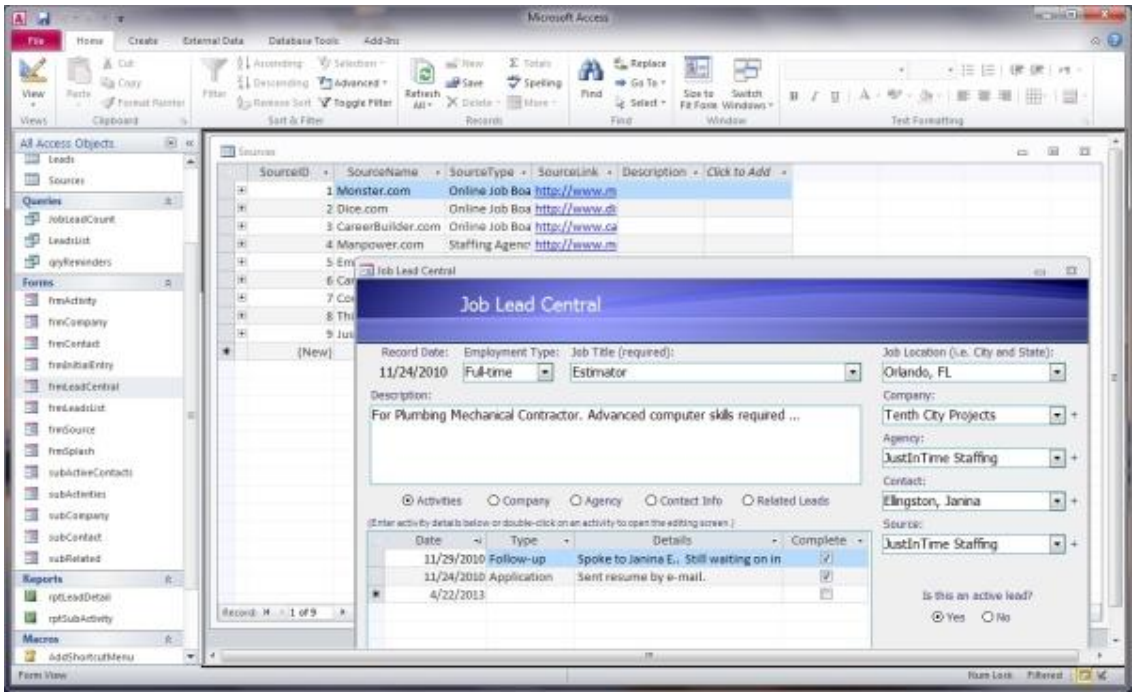


Figure 1.5 - Microsoft Access and other desktop database software can be used to create sophisticated applications for analyzing data.

Early in my programming experience, I used desktop database software including Borland Paradox and Microsoft Access to design database applications for the company where I was working. These applications worked with a variety of data including manufacturing statistics, employee and payroll information and customer communications. By using these software tools, I was able to save the company hundreds and maybe even thousands of dollars in software purchases and training hours while providing the other employees with quick access to the information they needed.

Desktop databases often represent a *single-tier* design. This means that whatever forms, reports and other interface features the user creates around the database are stored within the same database file or files as the data itself. An installation of the management software is usually required in order to work with the database. Some desktop database systems can allow for a basic *multi-tier* arrangement by splitting the forms and reports into a separate file called a *front-end* that is linked to the tables in another file called the *back-end*. This way, the design of the presentation and business programming is separated from the data and changes can be made to one without affecting the other. Multiple copies of the front-end which link to the same set of database tables can also be distributed to users to enable the database to be more efficiently accessed by multiple people.

Most of the databases I've mentioned so far are great for moderate amounts of data, even up to a few gigabytes. They provide a structure for the data and maybe even security to determine who can access it. In the right hands, they can be used to produce very sophisticated applications. At a certain point, however, it's time to go to the next level. Mobile and desktop databases typically support a limited number of users and are also limited in the amount of data that can be stored, the security features available and other management tools that help to maintain the kind of data collections you find in companies, universities and other organizations.

Server Databases - Organizational Data and Enterprise Applications

Just like the other types of database software, server database software can be used for small databases but is actually designed to support immense database systems containing up to hundreds of *petabytes* of data (1 petabyte = 1,000,000 gigabytes). These systems can contain multiple databases on powerful networked machines and provide data access to thousands of users. These massive systems can be worlds unto themselves that even contain sizeable amounts of programming code used to run automatic operations on the data within the tables and provide data to the reports. Server

databases represent *multi-tier* systems and are accessed by multiple pieces of software, websites and devices by users across a company or university campus. Very often, the databases themselves will be maintained by one or more full-time database administrators (DBAs) who will monitor the security and performance of the systems, respond to any access problems and either design new databases as needed or provide expertise for the software designers.

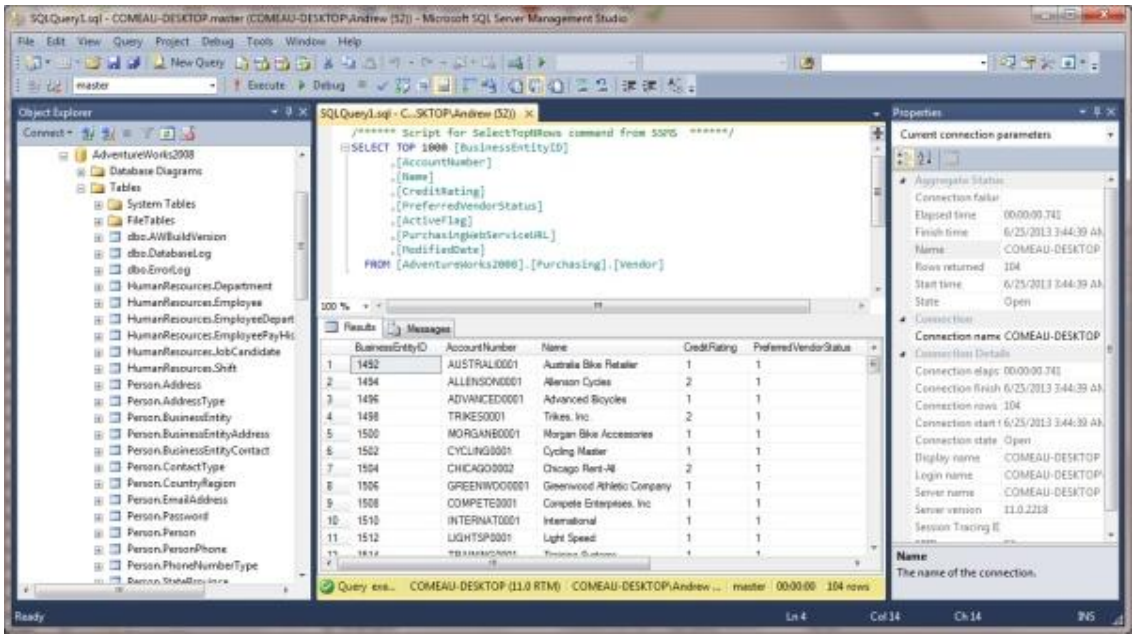


Figure 1.6 - Server databases can host immense database systems that serve many hundreds or even thousands of users.
This screenshot shows a view of a database maintained in SQL Server 2012.

At the server level, the database software is no longer just a standalone program or a series of files but might actually be integrated into the operating system of the network machine on which it runs. It could also be running as a collection of services that can be started, stopped and managed by the administrator. For example, Microsoft SQL Server can use Windows security to verify that a user is authorized to access specific databases. When a user logs into their Windows desktop machine and runs any program that accesses the database server, SQL Server will permit them access only to the databases for which their user account has been authorized by the person responsible for the system. This means that the user only has to enter a password once, at Windows startup, and the DBA can grant or revoke their access to specific resources with a couple clicks of the mouse. Another benefit is that database servers are often listed as network resources so that a person with enough knowledge can find the right database no matter which machine they're using on the network and without needing to know what machine it's stored on. Programmers can design applications to call a database over a company network or even the Internet so the average user can continue to be completely unaware of the database itself so long as everything is working correctly.

The two big players in the server database market are Oracle and Microsoft. As I've mentioned, Oracle has both their self-named Oracle database and MySQL while Microsoft has Microsoft SQL Server. IBM has its own database titles, DB2 and Informix. Organizations might use more than one title depending on their needs. I personally have Microsoft SQL Server Developer Edition and MySQL installed on my local machines for development and testing while the hosting services that I use for my websites offer access to both SQL Server Express and MySQL for the creation of databases to support the sites. Sometimes, tools like WordPress use specific database titles but it can also come down to the preference of the individual database developer or programmer. Cost can also be a factor. Depending on the needs of the organization and the applications being developed, database software might be a free download or it might cost an organization thousands of dollars in license and support fees.

Cloud Databases - Outsourced Data Storage

The newest category of database seeing wide use is the Cloud Database. Cloud computing is a computing model which enables companies and individuals to lease computing, data storage and database services from outside providers, paying only for the services they use. Applications and databases are stored and run on the provider's servers and accessed over the Internet from any computer in the world with an Internet connection. It's similar to the way in which a website designer leases space and resources from a web hosting company but with more types of services available and more flexibility in configuring them. Cloud computing enables an organization to add resources for a project and obtain the resources as a paid service rather than a large equipment investment and the service can be easily managed, increasing or decreasing the CPU, memory and other resources as the needs of the project change. Software licensing costs are included as part of the service which further reduces the cost for the subscriber. Cloud computing providers might also offer service and security guarantees which can lift some of the burdens from local I.T. departments.

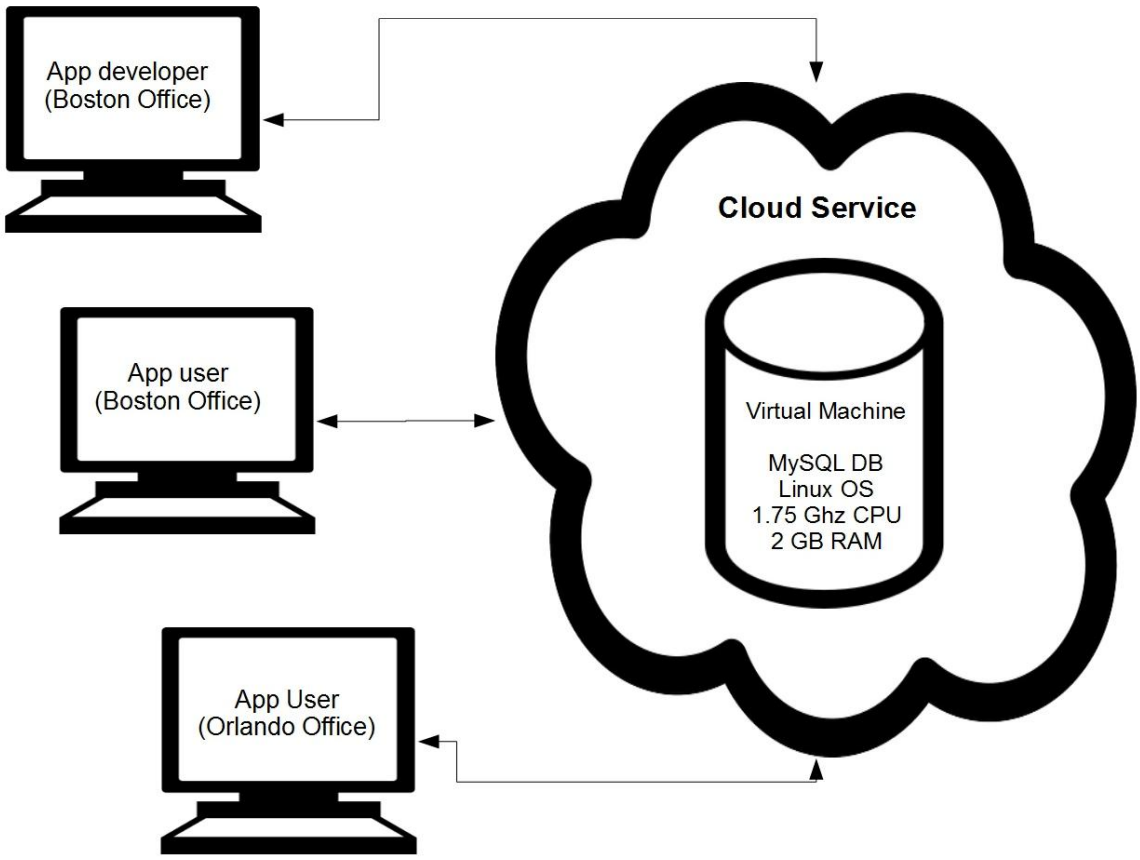


Figure 1.7 - Cloud databases are run remotely through a cloud computing service and accessed through any computer or device.

Cloud databases are one of the many services offered by a cloud computing service provider. A subscriber creates an account with the provider and would then have access to a web-based control panel, much like the one on a web hosting account. This control panel enables the user to create as many computing environments, databases and other remote resources as are needed. The user specifies the resources needed for the database, often including the CPU, server memory (RAM), hard disk space as well as the database and operating system software to be used. Cloud computing services might offer both Windows and Linux configurations with both MySQL and SQL Server available. These resources are then allocated to the database project by the service, the database is created and the subscriber is provided with the information needed to manage the database over the Internet. The subscriber would then connect to the database as needed through the database management program of their choice or through a custom application that they or their company

designs to use the data. The resources such as bandwidth or operational hours used by the project are billed to the subscriber monthly. If the application is not performing well because it lacks the necessary resources such as CPU or memory, the subscriber can login to the service's control panel and request that additional resources be allocated.

With this type of model, a single subscriber can create and run as many different applications and databases in different operating systems and configurations as their budget will allow. An application developer can quickly create test applications and environments and then dispose of them when they're no longer needed. A company can lease database resources on a monthly basis and treat it as an operating expense rather than buying and maintaining the actual servers. From a security standpoint, there is the issue that the subscriber's data is being stored on another company's servers and this might be a concern depending on the application but this is just one of the questions to consider when designing any application.

Currently, a number of large and well-known companies including Google, Amazon, Microsoft and Oracle are offering various selections of cloud services and provide free trials of the services for potential subscribers to try. There are also smaller cloud computing companies offering specific services such as ClearDB.com, a cloud database provider used by Microsoft's Azure service to provide support for MySQL databases. You might see a service like this referred to as 'Database as a Service' (DaaS) with other types of software delivered in this way being referred to as 'Software as a Service' (SaaS).

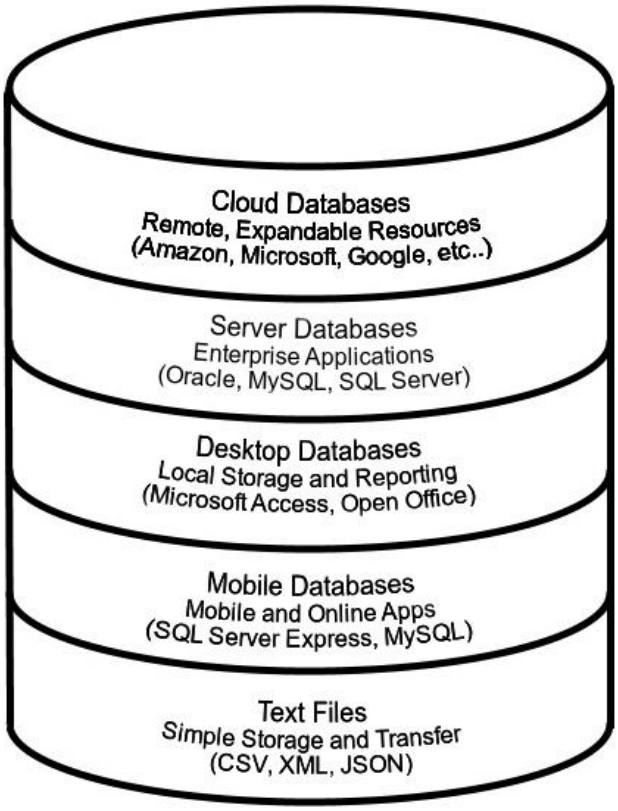


Figure 1.8 - There are many types of databases to suit the need of any application. The type of technology needed is one of the first considerations when designing a new application.

Figure 1.8 shows a summary of the database types that we've looked at in this chapter. Each level of technology has its advantages and appropriate uses and it's important to have a basic understanding of these technologies in order to make the best decisions when organizing a collection of data or building an application. A sophisticated database solution that uses too many resources for the environment where it will be used causes just as many problems as a solution that is not adequate to the needs of the data being stored.

Chapter Summary

The flow of information has always been important. In today's electronic world, it's easier to store and manage than ever. Our everyday activities including purchases, communications, work and entertainment leave a trail of electronic data that is stored in various types of computerized databases. These databases can take many forms depending on the needs of the data and the organization managing it and might range from small databases on personal smartphones to massive networked databases owned by corporations and government agencies. Many software titles and services are available to assist in the cataloging of data, each having their own features, limitations and appropriate use. Understanding how databases work can provide you with a better perspective of how information is stored and used and is essential in the pursuit of both technical and non-technical careers.

For Further Study

Review Questions

1. What is the basic definition of a database?
2. How does information stored in a Microsoft Word file or JPG image file differ from information stored in a database?
3. Other than data storage, what is a popular use for CSV and XML data?
4. What is an advantage of the XML format over the CSV format?
5. If you were transmitting continuous information over the Internet that needed to be collected and stored on the other end, what data format would you use?
6. How does a desktop database like OpenOffice Base differ from a server database like Oracle's MySQL.
7. If you are designing a website, which two database software titles are you likely to be using?
8. What are the advantages of a multi-tier design in a database application?
9. What are some of the advantages of cloud computing for a business over the traditional purchase of equipment? What is one potential concern?
10. Name three resources that can be specified and allocated for a database on a cloud computing platform?

Exercises

1. Examine your own daily routine for examples of information that needs to be stored somewhere and ask yourself how this might be done. How long does the data need to be kept? How much data actually needs to be stored and how might the collection of data grow over time? How confidential is the data? What would the consequences be if the data is lost or mishandled? Consider the examples of databases in everyday life from the first part of this chapter as well.
2. Using some of the links provided, check out the websites for some of the software titles mentioned in this chapter. Search for information on the abilities and limitations of each software.
3. Review the cloud computing services mentioned and compare the services they offer and costs involved.

Terms to Remember

Back-end - In software development, the database is often referred to as the back-end. The use of this term implies that the business and presentation programming is designed separately from the database and the design of each can be changed without affecting the other. See *Multi-tier Design*.

Cloud Computing - A computing model in which computer and data services are leased by an individual or organization from a remote service provider. This enables computing resources to be purchased on an as-needed basis rather than as an large equipment investment. Many services including document storage, software and database resources can be purchased under this model.

CSV - Comma Separated Values, a plain-text file format in which information is broken down into fields and stored in order to transfer the data between programs or simply store small to moderate amounts of information on disk. Also known as delimited or tabular data.

Database - A collection of information, usually in electronic form, stored in a specific format for easy retrieval and use in reports and other applications.

Database-as-a-Service (DaaS) - A service under the cloud computing model in which subscribers can design databases on the server of a remote service provider, often specifying the database software to be used along with the server space, CPU speed and server memory. The subscriber can then use local software to access the database over the Internet from any computer in the world and pay only for the server resources that they actually use. The amount of resources allocated to a database application can be changed as needed.

Front-end - In software development, the front-end represents any logic and presentation code such as user forms and interfaces, reports and business logic that exists separately from the database. See *Multi-tier Design*.

Gigabyte (GB) - One billion characters of information. More precisely, 1024^3 bytes (1024 x 1024 x 1024 or 1,073,741,824 bytes).

JSON - JavaScript Object Notation, a text file format that uses a format based on the JavaScript language to store complex data.

Multi-tier Design - Software design in which the program code is developed and exists independently from the database and the two communicate with each other, enabling multiple users at different locations to use the same application to access the same data.

Open Source - A software distribution model under which source code is made freely available for study and modification by independent developers. This means that the software is often free as well although more advanced versions and product support might be provided as a commercial product.

Petabyte (PB) - One million gigabytes of information (1024^5 bytes).

XML - Extensible Markup Language, a plain-text format which uses user-defined markup tags to store structured data for transfer between systems or over the Internet.

Zettabyte (ZB) - One billion gigabytes of information (1024^6 bytes).

Links

- JSON text format official site - http://www.json.org
- MySQL Database official site - http://www.mysql.com