

Database to Desktop: Turning Data into Information

Abstract

Increasing the value of the data contained in databases has been an elusive goal. Dedicated applications cannot keep up with the changing needs of the information consumer. The growth of the Internet, the maturing of database products, and the ever-increasing power of desktop systems has given us the power to convert our data into information. This paper discusses the technological implementation of flexible database to desktop solutions for two DOE databases.

Introduction

Database technology has evolved and matured at a phenomenal pace. In just two decades, we have seen it's birth with the realization that data was bigger than the applications that created and cared for it. We found that sharing it made good business sense, that it was nearly worthless if people could not access it easily, and less than worthless if it were wrong. During the 1980's, we saw the development of competing database technologies, with no clear winner emerging until the end of the decade with the rise of the relational database. Relational theory provided a mathematical foundation upon which modern databases could be built. The exploding growth in the power of the computer during these years made performance issues more bearable. Structured Query Language (SQL) gave a common tongue with which to "speak database." Data Analysis and Normalization gave us the One True Path towards building great databases.

At the same time, application software was struggling with the "vertical system" syndrome, wherein systems were built to serve highly specific business purposes. Human resources, accounting, and payroll systems each served obvious needs. Each system had it's own data, which meant the system was self-reliant, and paychecks wouldn't be held up because the accounting system crashed. However, each system was maintaining separate data about common business entities, such as employees. The price of this approach was paid in obscure ways. If an employee moved, it could take months before all of the places the employee's address was kept came to light. There was no "one place" to go to maintain company information about you, an employee. Additionally, changing information consumer requirements took a painfully long time to implement. Sometimes the need for new features was gone by the time they were implemented!

The 1990's have brought us completely into the database era. Every application seems to be associated with one. Ever increasing power has put relational database applications on the desktop. Some of the largest software companies in the world are best known for their database products. SQL has evolved into an international standard. The World Wide Web is a vast reservoir of data, and no shortage of Internet applications exist to collect, categorize, and commit to memory huge quantities of it. The Golden Age of Information has arrived. We are on the brink of turning our vast stores of data into information.

This paper will describe how two of the DOE computer systems, Computerized Accident/Incident Reporting System (CAIRS) and the Occurrence Reporting and Processing System (ORPS), have been positioned to take advantage of this brave new world. It will give an introduction into the usage of ODBC data sources, illustrating the setup on a Windows-based system and the actual importation of data into a spreadsheet.

Where We Are Today

As Data Gatherers, we are superb. There is a system to collect data about anything and everything relevant to the complex, if not several. The previous CAIRS and ORPS systems were typical applications of their time. They were (and still are) vertically oriented, and make use of database management software

to contain the data. Centrally maintained reporting software converted the data into information, ostensibly for the purpose of enabling management to recognize trends and take action to increase safety throughout the complex. The database management systems that contain the ORPS and CAIRS data are modern relational systems, but the data structures are essentially the same. The user interface has been modernized to utilize web browser technology.

These modernization efforts have yielded several benefits. First, modernized relational database technology provides a more reliable service. Additionally, for those who understand and can use SQL, a ready-made query interface, albeit somewhat cumbersome, is available. The web browser technology has provided a user interface that many are already familiar with. Additionally, a less obvious, but very important feature of browser technology is the integration of strong encryption and site authentication with the application, without the need to build security into the application. The Internet can be used to transport application data because privacy issues are minimized.

One of the biggest benefits is due to the acceptance of the Open Database Connectivity (ODBC) standard for accessing database management systems. Nearly all vendors now provide ODBC “drivers” to enable application client software to access their database server software in a standard and portable way. There are benefits for the application developer: by making ODBC calls to the data source, their code is now relatively insulated from the effects of a change in database vendors. Simply plug in the new vendor ODBC driver and the application should run as before. This assumes (a rather large assumption) that standards have been adhered to in developing the application software, otherwise some changes will be necessary. Even with the potential need for such changes the application developer is better off. There is the same benefit for the vendors: it is now easier to convince customers that changing databases is a legitimate possibility. The sword cuts both ways, of course, because it’s also easier to lose customers. The big winner, however, is the information consumer. The explosion of desktop tools that incorporate ODBC access to data sources provides ways to analyze data in endless ways. This frees the information consumer from having to express their needs to an information technologist, waiting for the implementation, and hoping that nothing was lost in the translation.

Putting ODBC to Work

The process involved in setting up an ODBC data source is not a trivial one. A number of potential pitfalls await you, including local network and firewall issues and the potential for a botched installation of the software. You may require the services of a technician versed in configuring the desktop to access data sources.

To make use of an ODBC enabled database, the information consumer needs to have a bit of background information and an ODBC enabled tool to make use of the data source. The first item of business to identify the data source itself. This identification must include the network address or name of the server and the specific database software in use at that address. For purposes of illustration we will use the CAIRS database server, which is located at *cairs.tis.eh.doe.gov*. The database software in use there is Microsoft SQL Server version 6.5. We will also assume that the desktop system is using a Microsoft Windows operating system. This means that in order to access the database, we will need Microsoft drivers for SQL Server installed on our system. If the necessary drivers are not already installed on the desktop as a result of the installation of other products, they may be obtained from the Microsoft web site.

The next step is to build the system data set name (DSN) that Windows will use to refer to the data source. This is accomplished using the ODBC control in the control panel. This control allows you to see the drivers installed on your system, and is the tool used to add, remove, and modify data sources on your system. You will need to build a system DSN to access the SQL Server database at *cairs.tis.eh.doe.gov*. It is also important to make sure that the Multiprotocol drivers for SQL Server are part of whatever driver installation you have. These drivers are required because they perform the encryption services, and the CAIRS server will not respond to requests for data unless it is via these drivers.

Once the DSN is successfully created, a functioning connection to the database is assured, as the process of creating the DSN accesses the database and provides a connection test to the data source. This is the hard part – once the DSN is created, any number of ODBC enabled clients can now be used to access the data source. As a first example, assume that you want to incorporate some information from the CAIRS database into a spreadsheet application. In Excel, all that would be required would be to open the spreadsheet, select Get External Data from the Data menu, which brings up an application called Microsoft Query to obtain the data from the data source. The [presentation slides](#), which accompany this paper, illustrate the process from creation of the data source to the importation of data from the CAIRS database.

Beyond Spreadsheets

The spreadsheet is the classic desktop analysis tool. Its usefulness and power are unquestioned. But there are other benefits to ODBC connections to data sources. Certainly one such application is the integration of data from published sources within the DOE complex with local database applications. Many contractors within the complex have developed their own applications that largely parallel the information contained in the CAIRS and ORPS systems. It is now possible for those applications to “go to the source” of the data for information they are required to report, such as codified information, and ensure that valid codes are being reported. Since nearly the entire database is published for each of these two systems, once access has been granted to the data there is little need for assistance from the publishers of the data. Information consumers can immediately fill a need for any data themselves. Local database reference tables can be kept synchronized with the official data source.

Perhaps the greatest benefits lie ahead, in the not too distant future. On Line Analytical Processing (OLAP) tools are becoming more widely available, and as the technologies mature, will become easier to use. As re-engineering efforts continue with these and other applications databases, data warehousing concepts will be applied where appropriate, which in turn will facilitate more advanced data extraction techniques, which are collectively known as Data Mining tools. Data Mining is the next generation of information processing, extending the usefulness of information by attempting to build *predictive* rather than *retrospective* views of the data. Highly competitive industries, such as retail, financial, and insurance, already employ these techniques. To remain competitive, they must use them. The retail business that recognizes trends in consumer preferences first has a major advantage over its competitors.

The implications for just these two systems are important. If we can, for example, use the safety information in these databases to begin to identify trends, then perhaps we can take steps to correct the real problems, which may not be at all obvious at a casual glance. The easier we can make it to put information in the hands of subject matter experts, the better they will be able analyze the data for such trends. Data mining tools will make it easier to visualize data, which can help analysts discover patterns within the data that perhaps a tabular charts may not be able to do.

Summary

Modernization of the CAIRS and ORPS databases has yielded many benefits. Relational database technologies provide an industry standard way of organizing and accessing the data they manage. Both ORACLE (ORPS) and SQL Server (CAIRS) provide SQL interfaces to access the data. They also provide industry standard ODBC drivers that enable application software to “plug into” the database and access it’s contents without requiring extensive changes to the application. The ODBC drivers also support integrated security features, such as data encryption, which enable the transmittal of data over public networks with no loss of privacy. The user interface to these systems is industry standard browser technology, freeing the development staff from the necessity of building their own. Powerful data analysis tools and techniques are coming of age, and the use of modern data access methods has positioned both systems to take advantage of their benefits as they arrive on the scene.