Published: October 1998

Introduction

Microsoft® Access 2000 is a critical part of Office 2000 and helps deliver on the areas listed above. Microsoft Access is one of the best-selling desktop database of all time; users love the broad range of tools Microsoft Access provides to enter, analyze, and present data. Microsoft Access 2000 brings not only the traditional broad range of tools but also adds more productivity features, database Web document creation, increased integration with SQL Server™, and a tight office programmability model.

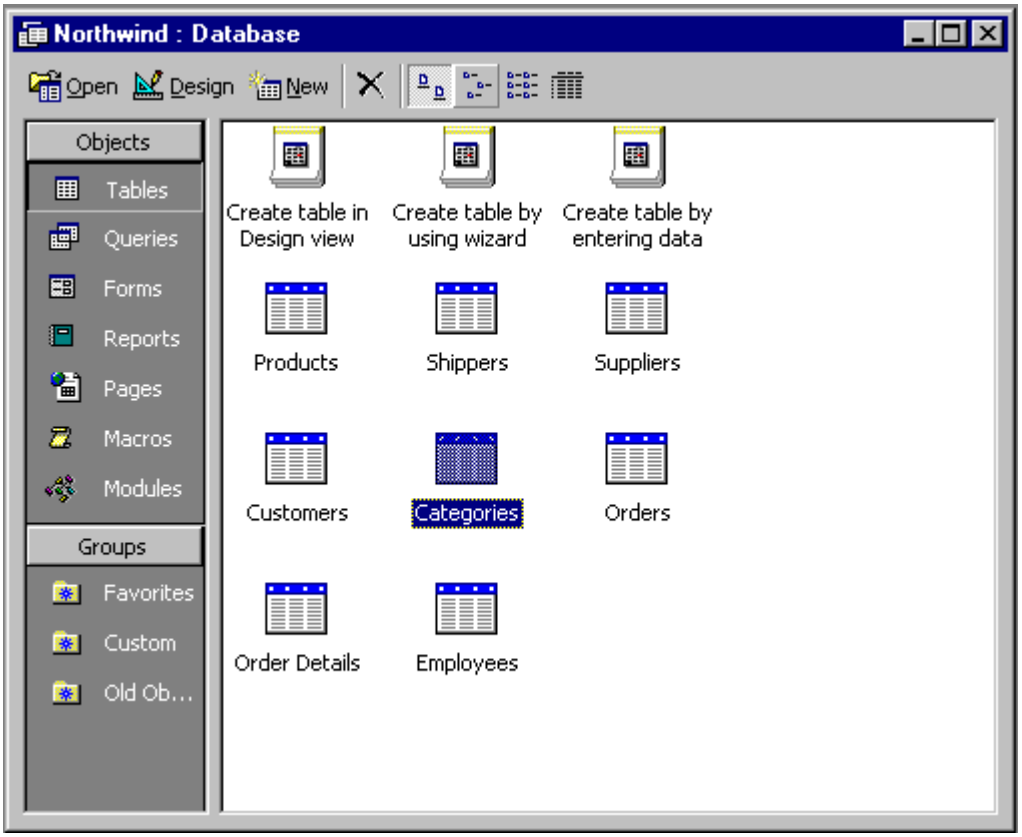
Benefits and Features

Personal Productivity

Microsoft Access was originally built to provide users with powerful database tools, yet be easy enough for any user. Access 2000 provides several new productivity enhancements to help end users, power users, and developers to be more productive. The most obvious change is the new Database Window.

Access Database Window

The database window was changed to accommodate the new objects exposed in Access 2000, to make it more user friendly, and to be consistent with the new user interface metaphor used throughout Office 2000.



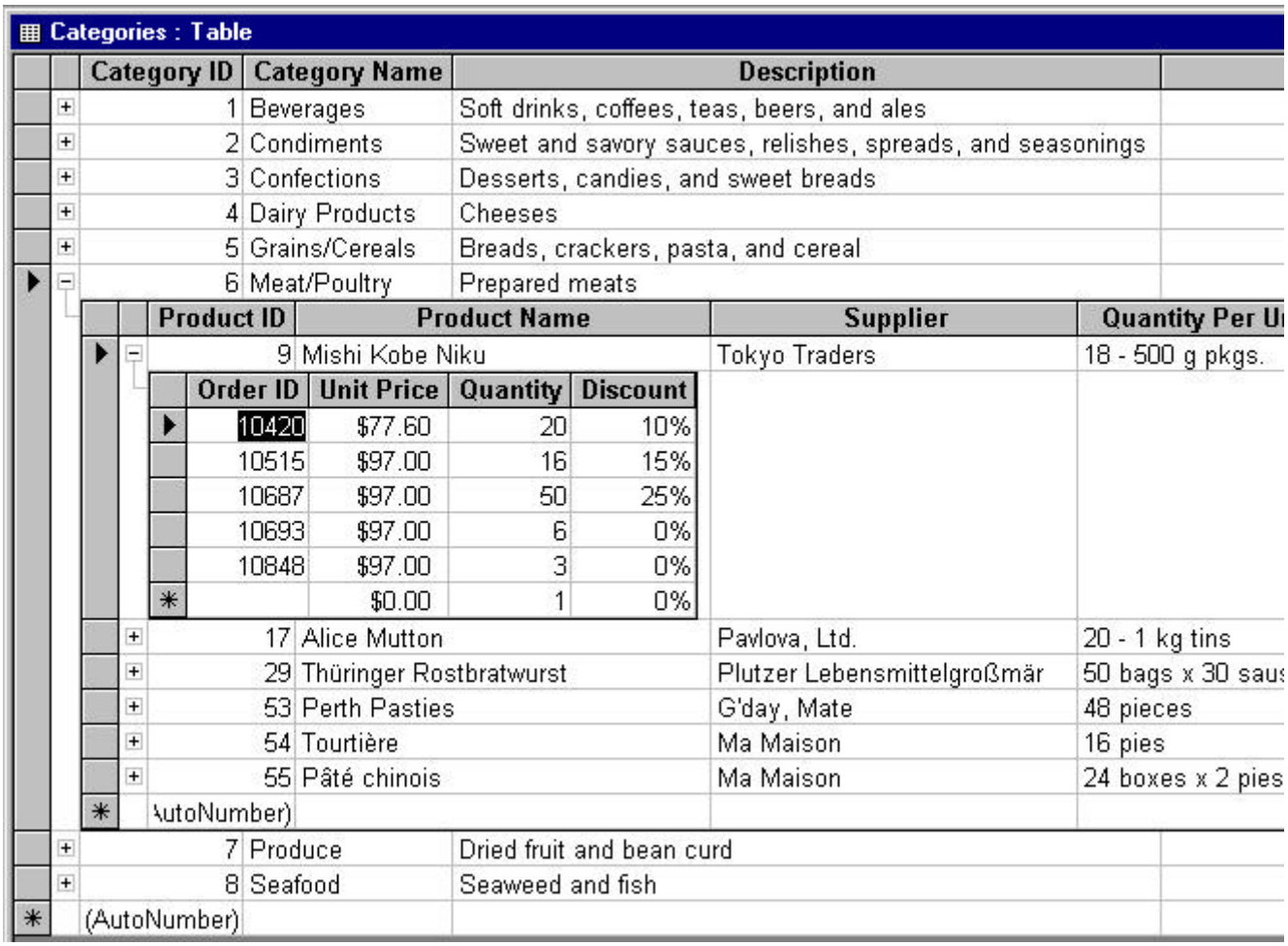
The changes in the database window include:

- A new tab for Data Access Pages

- New tabs for views, stored procedures and database diagrams (Microsoft Access projects only)
- User interface designed like Outlook bar
- Microsoft® Internet Explorer software 4.0 Listview control for the list of objects
- "New <object>" items in the list of objects for easy access to wizards and designers
- The ability to create defined groups containing any type of Access object

Subdatasheets

Microsoft Access 2000 allows the user to browse hierarchical data in datasheet view. The user can view subdatasheets with table, query, form and subform datasheets. The subdatasheet can be bound to a table, query or form. In Access 2000, instead of seeing a single table or recordsource in the datasheet, the user can insert subdatasheets to view related data. For example, a user is viewing the Categories table in the Northwind database and the Categories table has a one-to-many relationship with the Products table. Instead of being able to see only the data in the Categories table, the user can see the products for each category in a subdatasheet under each category row. The user will be able to drill down whenever relationships between tables exist.



Name AutoCorrect

Microsoft Access 2000 automatically fixes common side effects that occur when a user makes name changes via the Access user interface. To accomplish this, Access will store a unique identifier with each object created and name mapping information that allows Access

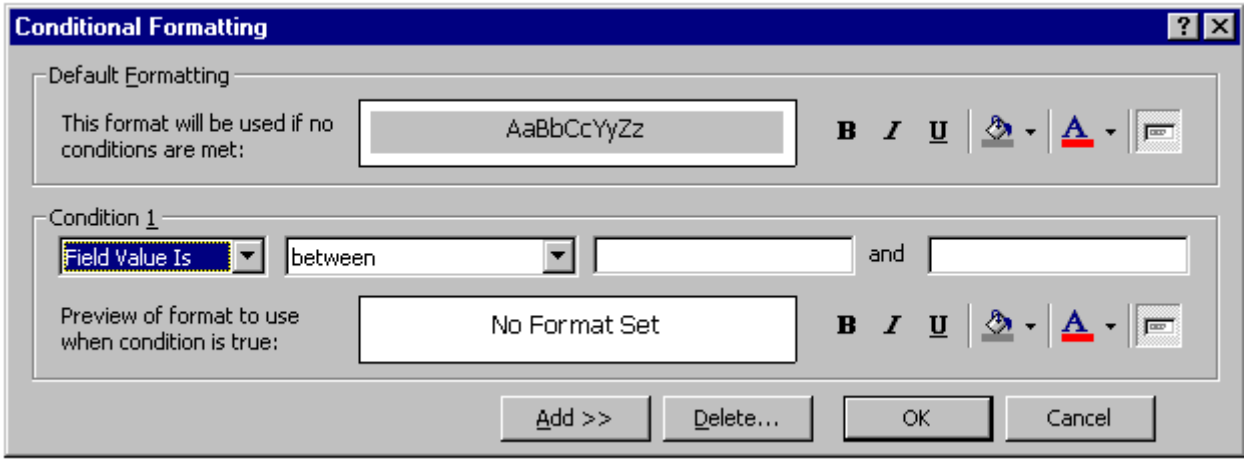
to correct binding errors when they occur. When Access detects that an object has been changed since the last Name AutoCorrect, it will perform a complete name fix-up on all items for that object when the first binding error occurs.

A typical example of this is when a user changes the name of a field in a table and that field is also being used in queries, forms, and reports. In previous versions of Access, the queries, forms, and reports would be broken when a field name was changed. In Access 2000, the field name is automatically updated throughout Access objects and the user is not required to do any work to keep using the application.

Conditional Formatting

Microsoft Access users have requested for years the ability to quickly and easily format fields based on the value of data. Today there is limited facility for this. Although it is available for decimals, and for positive and negative values, it is highly undiscoverable and difficult (the format uses the cryptic Format strings and looks something like this: "[BLACK] $#,###.00; [RED]($#,###.00)".

To address this problem, Access 2000 introduces conditional formatting with the following UI:



When the user first opens the Format | Conditional Formatting . . . dialog, it will be populated with the default formatting showing the current format. This will serve as a bridge between the formats applied elsewhere in the product (through the property sheet and other dialogs) and this dialog, where the formatting is all brought together. The user will be able to control the formatting of fields on both forms and reports based on the values of the fields or on user-defined functions. Users will be able to specify multiple conditions for each field.

Corporate Reporting

Wouldn't it be great if you could build Access forms and reports that ran perfectly in the browser? Microsoft Access 97 made progress towards this goal by letting you "publish" your Access objects by converting them to Web formats, but the conversion often fell short and users sometimes didn't get what they were hoping to get. Data Access Pages help solve this problem because they are targeted for the browser. Think of Data Access Pages as Access forms and reports for the Web. There's no conversion process: they are HTML files that you design in Access and run in the browser. Data Access Pages bring the ease of use of Access forms and reports to data bound HTML files.

Data Access Pages

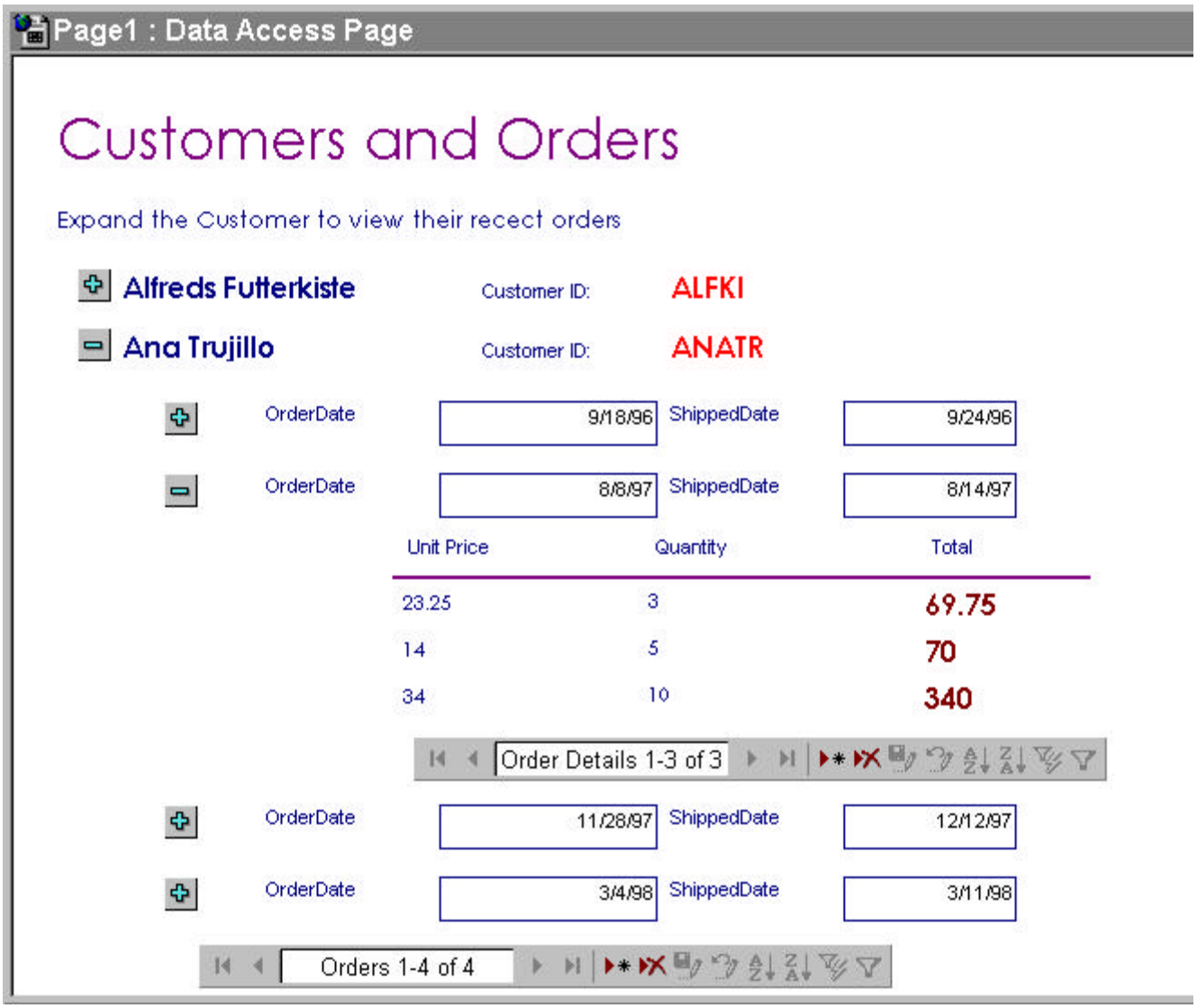
Data Access Pages are essentially HTML pages with data binding capabilities. The pages can be run within the Access shell or in the Microsoft Internet Explorer 5.0 browser software. Unlike forms and reports, Data Access Pages are stored outside the MDB as an HTML file

so users without Access on their machine can browse them. Users can also mail Data Access Pages to each other, allowing other users to view data-bound Data Access Pages in Outlook 2000.

Access can open any existing HTML file in the data access page designer. Once this page is opened in Access, users can add data-bound fields to the page. To build a data access page, users will work with the new data access page designer. It uses Internet Explorer for its design surface and has familiar tools like a property sheet, a field list, a toolbox, and wizards. The controls that you place on Data Access Pages are HTML intrinsic and ActiveX® controls and are similar to the controls that are used when building Microsoft Access forms. Data Access Pages also have a rich object model and support writing script in VB Script or JScript® development software, so you can program in the language of your choice using the easy, powerful, integrated development environment (IDE) that Access customers have come to expect.

Data Access Pages will be more than a simple forms package for the Internet. Grouped Data Access Pages provide users with a totally new way to interact with data, giving them the opportunity to drill in and work with hierarchical data in an interactive manner, which has not been possible before. Bringing the richness of hierarchical data to HTML pages is one of the key reasons we have chosen to build Data Access Pages. In addition, Access 2000 users will be able to include Office 2000 Web Components (a Spreadsheet Component, a PivotTable®Component, and a PivotChart™ Component) in their Data Access Pages, opening up huge possibilities for data analysis and reporting solutions that can be built using Access 2000 and Office 2000.

On the following page, you see a grouped data access page. The page shows a list of customers in the Northwind database. The user can drill into a list of orders for a particular customer by clicking the expand/collapse button next to the customer name. The user is given information on the order, such as order date and ship date. Some items to note about this picture are the rich controls and rich 2-D layout. Note that there is a 2nd record navigation control tied to both customers and orders. This will scroll through employees in this region. The user can further drill down to view the details of each order, including computed fields, by using the expand/collapse button. The design of these grouped Data Access Pages is simple, allowing Access 2000 users to build data-bound HTML pages like this one quickly and easily.

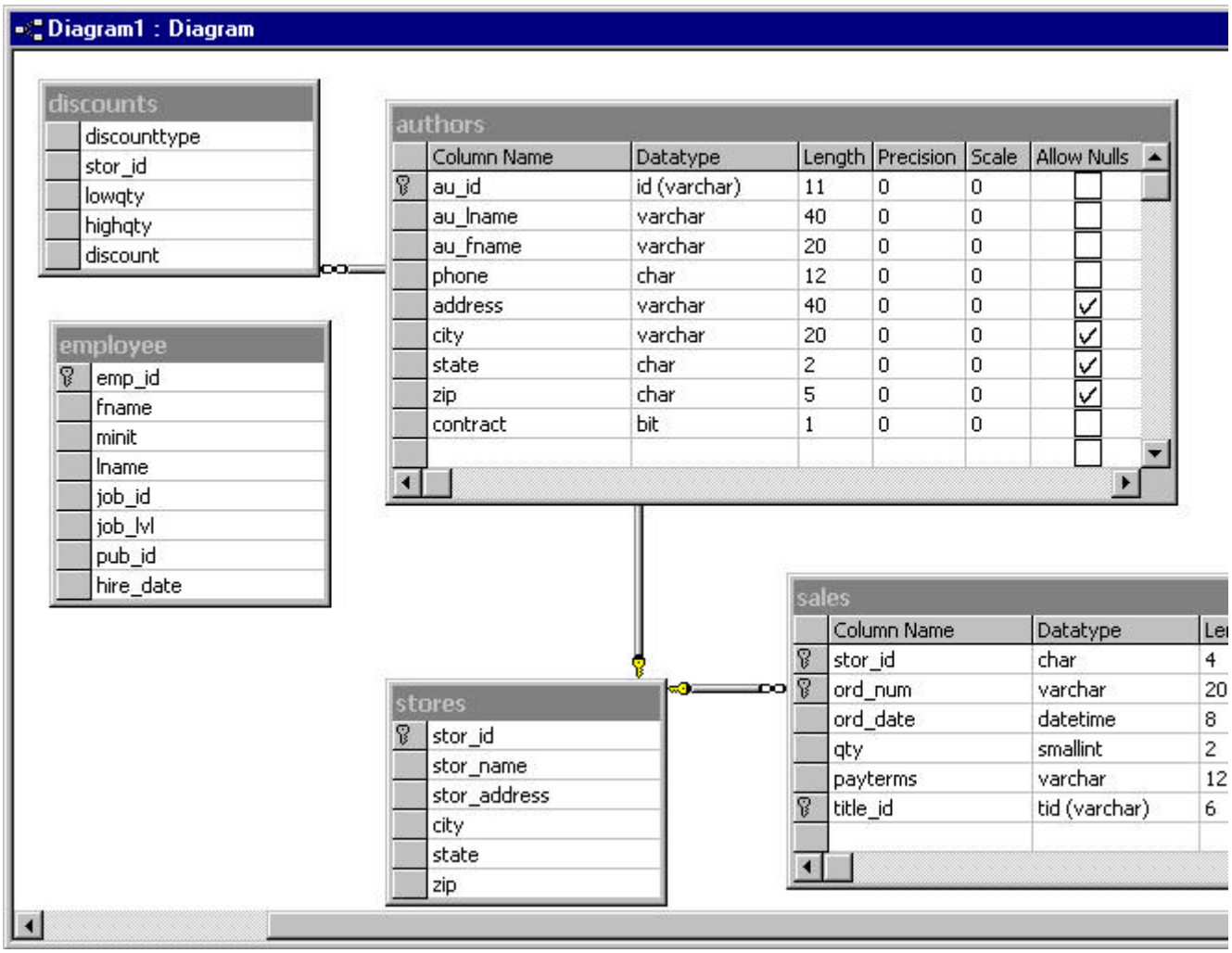


Connectivity to Enterprise Data

Microsoft Access 2000 will allow connection to SQL Server directly via OLEDB instead of going through the Jet database engine (via linked tables). When a user chooses this approach, Access will create a Microsoft Access project (an .ADP file instead of an .MDB file) which will store forms, reports, macros, and modules. The data-related objects (tables, views, relationships and stored procedures) will be stored on the SQL Server. This will allow power users and developers to build Access 2000 applications that work directly against a SQL Server-based back end. To provide Access users with the tools they need to develop databases easily, Access 2000 integrates Microsoft® DaVinci visual database design tools.

SQL Server Integration

The SQL Server integration features that have been built into Access 2000 (Microsoft Access projects, DaVinci integration, etc.) are targeted towards the mid- to high-end Access user who is knowledgeable about SQL Server and generic client/server issues and techniques. The user interface changes are therefore tailored to this user, striving not for pure ease-of-use but for simple, clean functionality.



Supported Data Sources

When using an Access project file, the user can connect directly to either SQL Server 6.5 (with Service Pack 3), SQL Server 7.0, or MSDE (Microsoft Data Engine). In all configurations, Access supports the creation and manipulation of SQL objects, including tables, views, stored procedures, triggers, and database diagrams, (which represents a functional superset of the Access Relationships window). These objects – tables, views, stored procedures (with support for parameters), and SQL strings – are all valid data sources for Access forms and reports, and Data Access Pages.

Client / Server Design

There are four key items users need to consider when choosing whether they will use Jet or SQL Server for their database engine. In priority order, the items are:

- Simplicity
- Data integrity
- Number of simultaneous users
- Amount of data
    - *Simplicity*. Jet is far easier to use and administrate than SQL Server. This makes Jet a good starting place for relatively simple database applications. Users should only

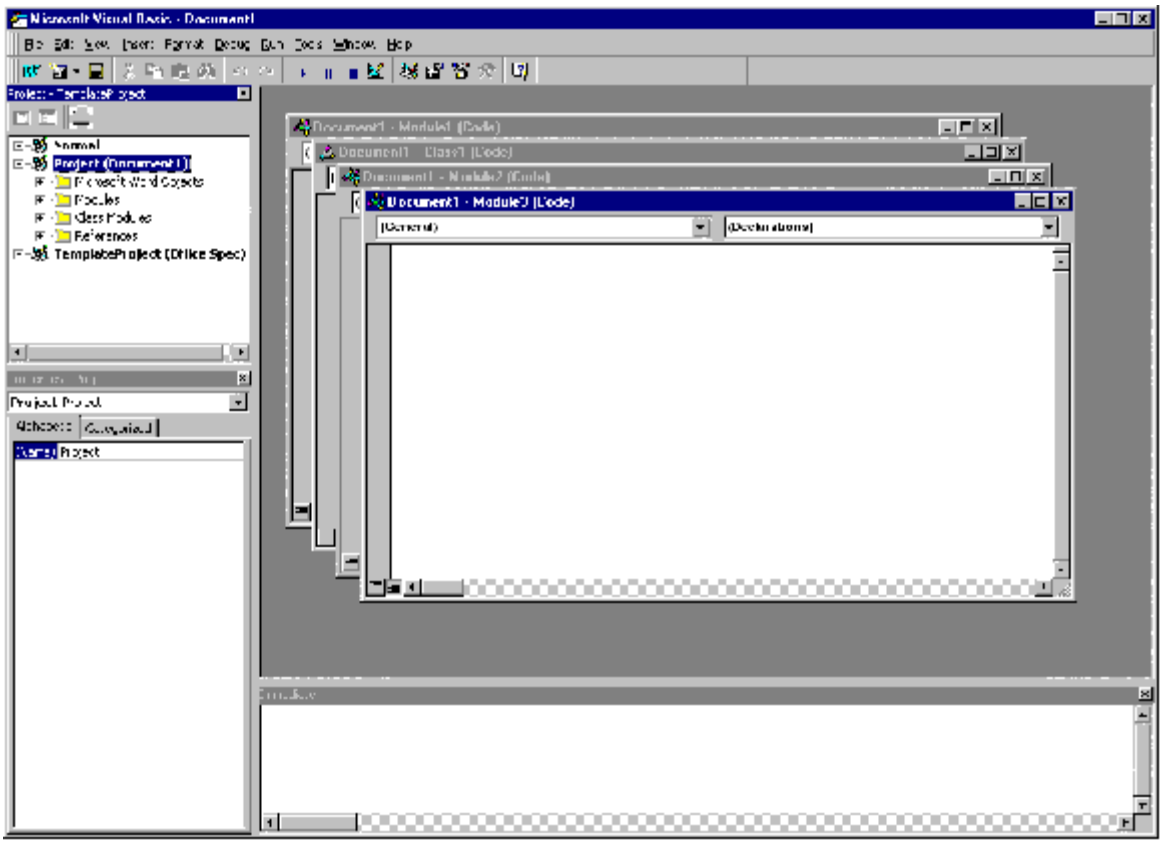
move to SQL Server when they need the advantages offered by SQL Server, such as scalability.

- *Data Integrity*. SQL Server is a client/server database. Jet is a file/server database. The big advantage of SQL Server is its ability to run queries and log transactions. If anything goes wrong during a write to the database (disk error, network failure, power failure, etc.), SQL Server can recover because it logs the transactions. When the system comes back up, SQL Server will revert back to the last consistent state. Since Jet cannot log transactions, the database may be corrupt. You will need to revert to your last backup copy.
- SQL Server is the right choice for systems that involve very important transactions (e.g. financial) or store mission critical (24 hour a day, 7 day a week systems) data. When deciding between Jet and SQL Server, developers should ask themselves: If this database goes down for a couple of hours, will this be acceptable or a huge problem? The more important the database is, the more SQL Server should be the database of choice.
- *Number of Simultaneous Users (Performance)*. SQL Server can handle more simultaneous users than Jet. The limit in Jet depends on what the users are doing. Reading data is relatively low impact and Jet can handle up to 255 simultaneous users. The practical limit is between 50 and 250 users, depending on what your application is designed to do. SQL Server can scale to a much higher number of users.
- *Amount of Data*. Jet can handle up to 1 gigabyte of data per MDB file in Access 97 and up to 2 gigabytes of data in Access 2000. You can create larger databases by using linked tables to several different MDB files. SQL Server has a much higher limit. SQL Server also has a performance advantage over Jet for large sets of data and many simultaneous users. Because Jet is a file-server system, the query processing must happen on the client. This involves moving a lot of data over the network for large databases. SQL Server runs the query on the server. This loads the server more than Jet, but can reduce the network traffic substantially (especially if the users are selecting a small subset of the data).

Programmability

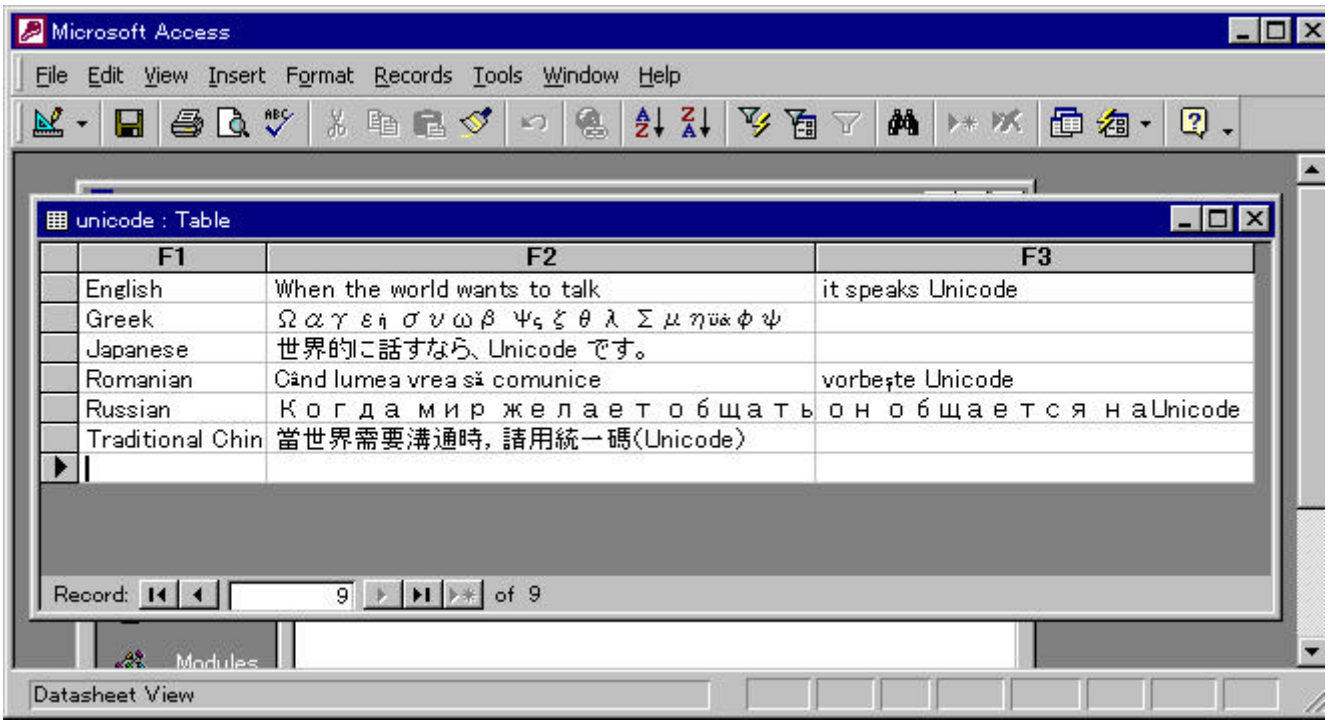
Access 97 natively hosted the Visual Basic for Applications programming language. For consistency with the other Office 2000 applications, Access 2000 integrates the Visual Basic programming environment and, as a result, users will see the same Visual Basic Environment that users see in Microsoft Word 2000, Excel 2000, and PowerPoint 2000.

Access 2000 will allow users to develop scripts for Data Access Pages using MSE, a new shared scripting IDE.



Worldwide Support

In Access 2000, we have added Unicode support, which will enable users to store and display multilingual text in many languages. For example, if you created an application that contained address information for international clients, you would be able to see a Japanese name in your table next to a Greek name. This will allow international users much greater flexibility when they are creating databases. This feature will also allow multiple-language support in forms and reports.



With the addition of Unicode support to Access 2000, users will have the ability to store all character sets within one database. Some characters require more storage space than others. For example, a database containing Chinese characters will be larger in size than a database containing only alphanumeric characters. Access will automatically compress data contained in fields to minimize the size of the database.

Access 2000, along with all other Office applications, will support the Global Interface feature. With the Microsoft Office 2000 Language Pack, users will be able to select the language to be used in the Access user interface.

New Jet Features

Access 2000 contains several major improvements to the Jet database engine. New Jet 4.0 features include:

- Full Unicode support as described previously
- Row-level locking
- An enhanced conflict/error model
- Enhanced counters [Autonumber fields now support seed and increment (code only)]
- A native OLEDB provider which provides a native interface directly to Jet
- New SQL syntax (optional ANSI compliant SQL queries)