



## Microsoft Access or Microsoft SQL Server: What's Right in Your Organization?

SQL Server Technical Article

Writers: Luke Chung

Technical Reviewer: Matt Nunn

Published: December 2004

Applies To: SQL Server 2000 Service Pack 3a (SP3a)

**Summary:** This paper explains how Microsoft® Access is used within an organization. It also explains when to use Access and when to use Microsoft SQL Server™.

# Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2004 Microsoft Corporation. All rights reserved.

Microsoft, Visual Basic, Visual Studio, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

---

# Contents

<b>Executive Summary .....</b>	<b>1</b>
<b>Strategic Mission and Vision.....</b>	<b>2</b>
Matching the Correct Technology to the Solution to Maximize Returns .....	2
Using Multiple Tools for Success.....	2
<b>Database Needs in an Organization.....</b>	<b>2</b>
Enterprise Level.....	2
Department Level .....	2
Workgroup or Team Level .....	2
Individual and Small Groups.....	3
<b>Database Pyramid.....</b>	<b>3</b>
<b>Database Evolution .....</b>	<b>4</b>
Simple Databases Evolve into Sophisticated Ones .....	4
Few Databases Evolve to the Next Level.....	4
Hardware Evolves .....	4
Evolution Is Unpredictable.....	4
Anticipate Evolution.....	4
<b>Database Challenges in an Organization.....</b>	<b>5</b>
Maximizing Return on Investment .....	5
Managing Human Resources .....	5
Rapid Deployment.....	5
Flexibility and Maintainability .....	5
Scalability (Secondary) .....	6
<b>Access versus Excel .....</b>	<b>6</b>
Role of Excel .....	6
<b>Access Fills a Large and Important Segment .....</b>	<b>6</b>
<b>Database Solution Costs.....</b>	<b>6</b>
Quantity of Database Solutions .....	7
Quantity versus Cost .....	7
<b>Advantages of Access.....</b>	<b>8</b>
Return on Investment.....	8
Rapid Application Development .....	8
Integrates with Microsoft Office.....	8
Easy to Deploy .....	8
Windows versus Web Data Entry .....	8
Interfaces with Numerous Database Formats .....	9
Excellent Report Generator.....	9
Approachable Development Environment.....	9
Solutions with Less Code.....	9

Ideal for Network Solutions .....	9
Handles Disconnected Situations .....	9
<b>Limitations of Access .....</b>	<b>9</b>
Not for Web Solutions.....	9
Security.....	10
Data Integrity.....	10
Backups.....	10
Transaction Logs and Rollbacks .....	10
Network Bandwidth .....	10
Scalability .....	11
Deployment Issues.....	11
Limited User Interface .....	11
<b>Why Access Is Important .....</b>	<b>11</b>
Best Choice for Specific Segment .....	11
Many Database Problems Solved by Access.....	12
ROI: Access Solutions Cannot be Cost Justified on Other Platforms .....	12
Access Competitive Advantage.....	12
<b>Exploring the Myths of Access Limitations.....</b>	<b>12</b>
Most Database Problems Are Small .....	12
Few Database Problems Exceed the Capabilities of Access.....	12
SQL Server Eliminates the Scalability Issue.....	12
Hybrid Solutions .....	13
<b>Using Access Strategically.....</b>	<b>13</b>
Reasons to Use Access.....	13
When to Use Access .....	13
Migrating Access Applications.....	13
<b>What to Do with Access 97 Databases .....</b>	<b>14</b>
<b>Using SQL Server Strategically .....</b>	<b>14</b>
<b>Conclusion .....</b>	<b>15</b>

## Executive Summary

Organizations have a wide range of database needs and resources. From simple, short-term needs to long-term mission critical requirements, organizations create and support a variety of databases. Choosing the right technology is an ongoing challenge. Many organizations struggle with deciding whether or when they should be using Microsoft® Access and Microsoft SQL Server™. Both are powerful and established database alternatives with their unique strengths and weaknesses.

Microsoft Access is composed of two parts: the application layer and the data layer (Jet database engine). SQL Server only provides the data layer. The Access application layer can interact with SQL Server, either through linked tables in an .mdb database or directly through an Access Database Project (ADP). Other products like Visual Studio® .NET, Visual Basic®, and C++ can also interact with SQL Server.

Both databases have an important role in most organizations, because each is optimized for different segments of database solutions. The strength of Access is its ease of use, rapid application development environment, and simplistic distribution (assuming the recipient also has the correct version of Access installed). It can even support multiple user shared deployments. The strength of SQL Server is its more robust data integrity, scalability, security, and manageability.

Depending on the situation, the strength of Access may outweigh its deficiencies. Overall, if the database problem is targeted for Microsoft Windows®-based operating systems only and can be solved with Access, the need for SQL Server is minimized. This is particularly true if information workers who are not programmers can use Access to solve their own problems. Not only are immediate costs lower, Access users may create a solution that would be difficult for someone else to understand and create for them as quickly.

For more challenging situations, an information technology (IT) professional creating a SQL Server solution is more appropriate. Whether the front end is in Access or not, a SQL Server database offers many features that a file server Access database cannot. Whether you need triggers, stored procedures, transaction logs, Web application support, or security, SQL Server is a solid choice for critical needs that Access (using the Jet database engine) cannot match.

In most cases, it's easy to determine whether a new application should be using Access (with the Jet database engine) or SQL Server. The challenge for most organizations is how to anticipate and manage the small fraction of Access applications that need to migrate to SQL Server each year.

Few (we estimate less than 2 percent) Access databases need to migrate every year, yet some organizations want to ban Access completely. This is often because IT professionals are included when an Access application is breaking down. They fail to recognize the ability of Access to solve 98 percent of database situations that never require their involvement. When an Access application is created initially, the features needed in the future cannot be anticipated, nor can the budget be justified. It's a case of database evolution, and now it's time to evolve from Access and the Jet database engine. You have many ways to migrate an Access application without losing the existing investment, and an organization that manages this well attains a significant competitive advantage.

## Strategic Mission and Vision

### Matching the Correct Technology to the Solution to Maximize Returns

Different levels of an organization have different database needs. Choosing the right technology and approach for each level affects its ability to perform long term, and the returns it generates.

### Using Multiple Tools for Success

An organization has a variety of database challenges. No tool solves every issue. Many tools and approaches are available, each with their own strengths and weaknesses. Some manage large amounts of data in a structured and secure manner. Other tools manage a relatively small amount of data in an unstructured, minimally secure, yet highly flexible manner. Depending on the objectives, one tool may be superior to the other.

## Database Needs in an Organization

Some databases are critical to the survival of an organization, while others are quick and simple systems for specific analysis. No matter how large or small the organization, databases are used at all levels for a variety of reasons:

### Enterprise Level

Enterprise level applications are critical applications that the entire organization requires for its survival. Examples include accounting systems, customer transaction tracking, high-volume data processing, and other critical systems vital to the organization's ability to complete its mission. In large organizations, this is often considered the function of the data center. Critical issues include processing large amounts of data, maintaining historical data and legacy systems, accuracy, security, encryption, and administrative depth (such as backups, disaster recovery, and transaction logs).

### Department Level

Applications built for departments are less critical for the survival of the entire organization. Although these may include important data center applications, other applications may be managed in the individual department. Department level applications are usually created by professional developers and maintained by dedicated personnel. They often use or pass data into the data center repositories.

### Workgroup or Team Level

Workgroup applications focus on the needs of a smaller group of people working together. These applications can often change rapidly, to meet the needs and challenges of the workgroup, from either internal or external market forces. Workgroup applications tend to be PC-based and are often controlled by the line of business using it. These applications may involve professional developers, but many applications are created by power users and others who are not developers. These applications may retrieve data from data center systems, but do not commonly return data to these

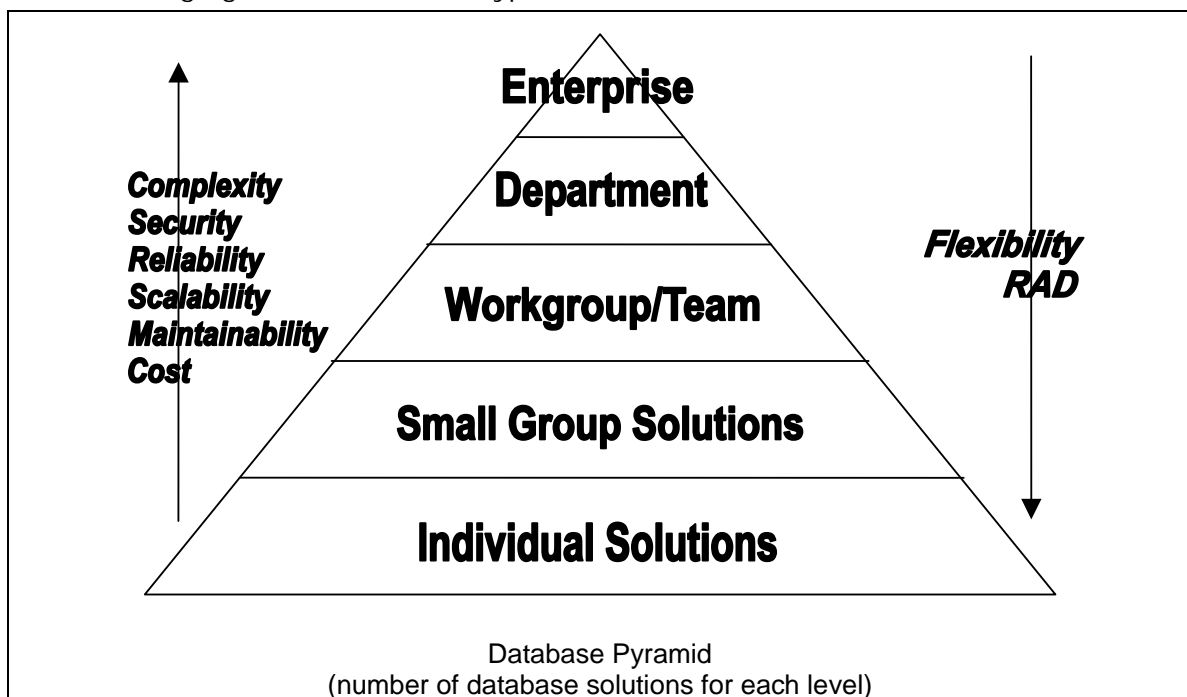
systems. Examples include data analysis, report generation, and managing the needs of the workgroup to perform its functions.

## Individual and Small Groups

On individual computers, many people create their own databases in Excel and Access. These tend to be single user applications that have relatively short life spans. Their purpose is to simplify the work of the individual or small group of people who created it. Most of these applications are created by people whose primary job function is not programming.

## Database Pyramid

The following figure shows various types of databases.



The majority of database solutions are simple. As systems handle larger problems, the number of applications an organization has or can afford decreases.

At the low end, flexible and rapid application development (RAD) solutions are used. Life cycles are short, bureaucracy and structure are limited, and mistakes are not life threatening to the organization. Cost per solution is relatively low.

Moving up the pyramid, the solutions become more sophisticated and critical. As the number of users increases, security and reliability become more important, and solutions need to scale. Maintainability is more important, because systems are built by many people and continue beyond their participation. More time is spent designing systems, because more people and issues are touched. When changes are made, the complexity and critical nature of the system requires longer implementation, testing, and documentation. All this increases costs, because mistakes become more expensive, and the organization's survival is more dependent on them.

## Database Evolution

### Simple Databases Evolve into Sophisticated Ones

Many database applications start at the bottom of the pyramid. Someone creates a spreadsheet or small database, finds it useful, and shares it with a few people. They like it, and more features are added. More people rely on the system, and over time, the simple solution that someone created for their personal use becomes critical for the department or enterprise.

### Few Databases Evolve to the Next Level

It's important to remember that this is the exception and not the rule. For every application that successfully evolves from one level to the next, hundreds if not thousands are created and never evolve. Many are discarded because they weren't useful, were implemented poorly, or the environment (business) changed. Others remain usable, never needing to migrate.

### Hardware Evolves

The types of business (not database) problems remain fairly static over time, compared to hardware gains. Problems that required mainframe solutions two decades ago now run on portable computers. When it comes to performance, the solutions at the bottom of the pyramid become more important over time. More database challenges are solved by that segment, while the top of the pyramid handles problems that were previously beyond the reach of computing or budgets.

### Evolution Is Unpredictable

It would be more efficient and less expensive to develop the critical applications of tomorrow correctly today, but that's usually not possible. It's difficult to predict which of the multitude of small databases today will become critical applications years from now. What's created or envisioned today for those databases may not be what's needed or critical in the future. An organization's requirements and infrastructure evolve over time. It's the evolution of the databases that makes them critical, and not the original vision of the author.

### Anticipate Evolution

Successful databases evolve over time. A successful IT strategy embraces this natural trend. Anticipating the transition is part of a successful database strategy. This means preparing for times when applications need to migrate to new platforms or be completely rewritten.

When this occurs, do not blame the existing platform, but rather appreciate the success of the organization and the system that takes it to the next level. The existing system should be considered a prototype for the next system, because the business needs are well defined and users are supportive. This significantly reduces the risks of the new system. These risks include expensive systems that are never delivered or built, or only fulfill a fraction of their original intent. Take what works and migrate that.

The transition is also an ideal opportunity to add new features and clean up the system, because after years of enhancements, many original assumptions may be wrong.

## Database Challenges in an Organization

Every organization has to overcome database challenges to fulfill their mission. These challenges include:

- Maximizing return on investment
- Managing human resources
- Rapid deployment
- Flexibility and maintainability
- Scalability (secondary)

## Maximizing Return on Investment

Maximizing return on investment is more critical than ever. Management demands tangible results for the expensive investments in database application development. Many database development efforts fail to yield the results they promise. Choosing the right technology and approach for each level in an organization is critical to maximizing return on investment. This means choosing the best total return, which doesn't mean choosing the least expensive initial solution. This is often the most important decision a chief information officer (CIO) or chief technology officer (CTO) makes.

## Managing Human Resources

Managing people to customize technology is challenging. The more complex the technology or application, the fewer people are qualified to handle it, and the more expensive they are to hire. Turnover is always an issue, and having the right standards is critical to successfully supporting legacy applications. Training and keeping up with technology are also challenging.

## Rapid Deployment

Creating database applications quickly is important, not only for reducing costs, but for responding to internal or customer demands. The ability to create applications quickly provides a significant competitive advantage.

The IT manager is responsible for offering alternatives and making tradeoffs to support the business needs of the organization. By using different technologies, you can offer business decision makers choices, such as a 60 percent solution in three months, a 90 percent solution in twelve months, or a 99 percent solution in twenty-four months. (Instead of months, it could be dollars.) Sometimes, time to market is most critical, other times it may be cost, and other times features or security are most important. Requirements change quickly and are unpredictable. We live in a "good enough" rather than a perfect world, so knowing how to deliver "good enough" solutions quickly gives you and your organization a competitive edge.

## Flexibility and Maintainability

Even with the best system design, by the time multiple month development efforts are completed, needs change. Versions follow versions, and a system that's designed to be

flexible and able to accommodate change can mean the difference between success and failure for the users' careers.

## Scalability (Secondary)

Systems should be designed to manage the expected data and more. But many systems are never completed, are discarded soon, or change so much over time that the initial assessments are wrong. Scalability is important, but often less important than a quick solution. If the application successfully supports growth, scalability can be added later when it's financially justified.

## Access versus Excel

### Role of Excel

Even though Excel is not a database, in many organizations, people store more data in spreadsheets than any other platform. This is not the preferred method for IT professionals, but it works. Decision makers need to analyze data, and they know Excel. This is one of the greatest benefits of client computing.

Although Excel is not a relational database, it solves many simple database problems. That's because many database problems can be solved with simple database solutions. Only a small percentage of Excel spreadsheets reach the limits of Excel, but when they do, many should and do migrate to Access.

## Access Fills a Large and Important Segment

The success of Access as the most popular database in the world is a testament to its capabilities and the need for database solutions by productivity workers. Access is the first choice when it comes to relational databases, because of its ability to quickly create useful database solutions.

It may not have all the features, scalability, performance, reliability, and security of more sophisticated solutions like SQL Server, but for many situations, those features are irrelevant or secondary to what Access offers. Access offers an excellent solution for database challenges for individuals, small teams, and workgroups across a network.

The number of database challenges within an organization that can be solved by Access is much larger than solutions solved by more complex and expensive solutions. Over time, with the reduction in hardware prices and increase in performance, more database situations are handled by Access.

## Database Solution Costs

Different database problems require different solutions. If an organization's only database response is a \$200,000 or more solution, it cannot profitably pursue opportunities worth less than that. That may not be a problem today, but it gives competitors an opportunity if they have less expensive solutions. Over time, some of those small opportunities grow into big ones.

The solutions and the costs vary significantly based on the platform selected. The following table shows approximate numbers.

Platform	Average Cost
Excel	\$500
Access individual	\$3,000
Access simple multiple user	\$10,000
Access workgroup or department	\$50,000
Visual Basic or Jet	\$200,000
Visual Basic, Visual Studio .NET, Java, or SQL Server	\$500,000
Oracle or IBM DB2	\$2,000,000
SAP, Tandem, and others	\$10,000,000 or more

Although there are \$1,000,000 Access applications and \$20,000 .NET applications, the numbers in the preceding table show order of magnitude for a large organization, and what they generally spend for solutions on those platforms.

Solutions created for the first three platforms (Excel and simple Access applications) are often created by those who are not IT professionals. Managers, analysts, and administrators create these solutions as part of their jobs, without IT budgets or guidance. Most of these solutions would rarely make economic sense if IT staff fulfilled them, nor would they be able to create them in a timely manner. Many applications created by those who are not IT professionals are not maintainable and are poorly designed.

With workgroup applications and beyond, defined budgets, design processes, and more structured development efforts occur, and people specializing in application development are involved. However, costs vary widely, based on the platform selected.

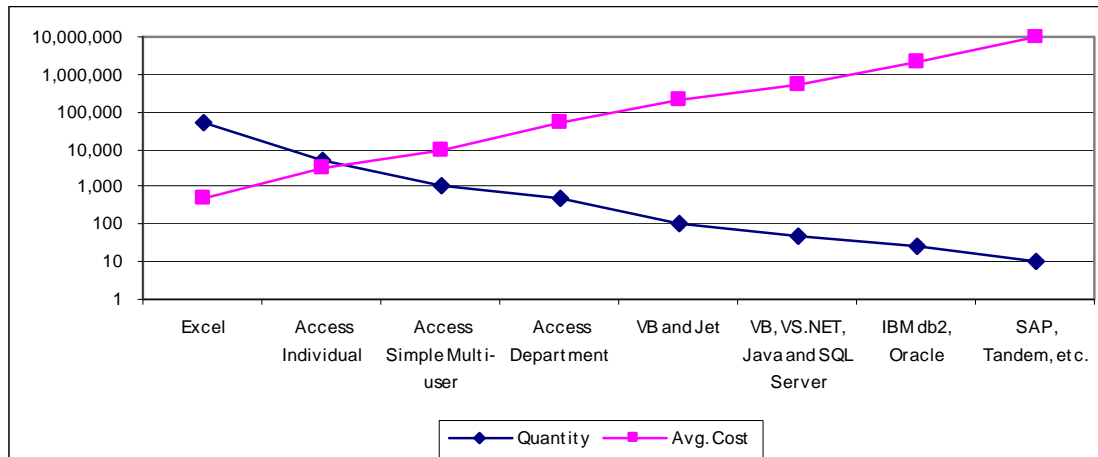
## Quantity of Database Solutions

As illustrated in the database pyramid earlier in this paper, there are more small databases than large ones. The following table shows an estimate of the relative number of database solutions by platform in a large organization.

Platform	Quantity
Excel	50,000
Access individual	5,000
Access simple multiple user	1,000
Access workgroup or department	500
Visual Basic or Jet	100
Visual Basic, Visual Studio .NET, Java, or SQL Server	50
Oracle or IBM DB2	25
SAP, Tandem, and others	10

## Quantity versus Cost

When you compare quantity and cost, there's an exponential relationship between the number of solutions and average cost. The following figure shows the comparison on a logarithmic scale.



Not surprisingly, as the cost of each implementation increases, the number of solutions decreases. It's the CIO or CTO responsibility to survey the entire spectrum of database challenges and deploy the appropriate technology, given limited resources and time.

## Advantages of Access

Access is the most popular database program because those who are not IT professionals can cost-effectively solve a wide range of database problems, and professional developers can create sophisticated multiple user solutions.

## Return on Investment

If it can be solved in Access, it's probably less expensive than alternative solutions, which maximizes return on investment.

## Rapid Application Development

The Access development environment lets you create results quickly. Access solutions often require significantly less code than alternatives. It's an effective platform for prototyping.

## Integrates with Microsoft Office

Access is part of Office, and integrates with other popular Office programs. Enabling users to view data and export (or paste) into Excel or Word is powerful to knowledge workers.

## Easy to Deploy

Access databases are easy to deploy if the recipient has the same version of Access. From placing it in a shared directory to sending it as an e-mail attachment, an Access database can be distributed by end users, similar to the way they distribute Word or Excel files.

## Windows versus Web Data Entry

Web users are trained to accept behavior that would not be acceptable in Windows-based applications. Consider the example of changing a quantity and clicking **Update** to

refresh total sales. Access easily (and inexpensively) supports this and other features, such as copying and pasting records, displaying multiple one-to-many relationships, and basic features, such as doing spelling checks. This provides a friendlier and richer data entry experience than Web solutions.

## Interfaces with Numerous Database Formats

Access links to many data sources, including SQL Server.

## Excellent Report Generator

The Access report generator is excellent. Subreports are useful for showing multitable relationships. Additionally, Access can link to many data sources. Many client database applications have significant report generation features.

Web reports have less functionality and often do not print properly, even with more effort.

## Approachable Development Environment

The Visual Basic for Applications integrated development environment (IDE) is similar to Visual Basic, and offers a productive development environment. Unlike .NET or Java, you can edit code while debugging, which is a timesaver.

## Solutions with Less Code

The less code required for a solution, the better. It's easier to create and easier to maintain. N-tier solutions are definitely not RAD, and are not beneficial if you don't need to share your data.

## Ideal for Network Solutions

Access is designed for file server solutions on local area networks.

## Handles Disconnected Situations

Access supports portable computers and disconnected solutions that can't be handled by Web applications. Access databases can also be easily sent using e-mail to others. In limited low data collision situations, Access replication is appropriate for remote database sharing.

## Limitations of Access

Although it has many advantages, Access has limitations that prevent its use in some cases.

## Not for Web Solutions

Access isn't designed to create Web sites. The Data Access Pages are of limited use in intranets, but not on the Internet. The underlying Jet engine is also not useful, except when the number of simultaneous users is low. Access is optimized for Windows, and not for the Web.

## Security

Although Access databases (using the Jet engine) can be password protected and encrypted, these databases do not have the same level of security as SQL Server or mainframe database systems. If data security is critical, a SQL Server solution is the better choice. SQL Server offers 128-bit encryption and storage in a remote location from the user and application. Combined with Web services, SQL Server allows distributed data in a controlled and highly secure manner.

## Data Integrity

Similarly, data integrity and recovery is not as robust on file-based databases using Jet, compared to SQL Server with its triggers, transaction logs, and repair processes.

File server databases using Jet may become corrupt and require regular maintenance to maintain optimal results. Even with maintenance, the chance of failure is much higher than with SQL Server.

The Total Visual Agent (<http://www.fmsinc.com/products/agent/>) product addresses the administrative needs of daily database maintenance (compacts and backups), but it's not the same as the built-in features of SQL Server.

## Backups

Access databases (using the Jet engine) are easy to back up (copy the .mdb file). However, these databases require user initiative, unless an automated process or tool, like Total Visual Agent, is in place.

If the Access database is open and the data is changing, it cannot be backed up while users are in it. This is a major problem, if the database is critical and used 24 hours a day, seven days a week.

The administrative tools for SQL Server have more features, compared to Access. Backups can run anytime, even if users are active. This is done through the use of transactions. Access supports transactions, but it is limited compared to SQL Server.

## Transaction Logs and Rollbacks

If you need to know who modified what data, and undo changes, SQL Server's built-in features and triggers support this.

An Access application can try to replicate the tracking of changes by managing user interaction with the data. However, it would require programming and could not be managed at the core data level. Mistakes in the application or other applications in contact with the Access data could cause data changes that are not documented. There are also no rollbacks in Access after a transaction is committed.

## Network Bandwidth

A well designed application using SQL Server can significantly reduce the amount of data moving across the network, because only the requested records are passed from the database to the application.

File server databases pass the whole table (or at least the index) across the network. This may not be significant for small files, but performance suffers as the data grows.

## Scalability

One Access database (using the Jet engine) is limited to 2 gigabytes (GB). If a database exceeds that, the solution can't be entirely solved by Access. Access databases also have problems with too many simultaneous users. The number depends on what they are doing.

If there's a lot of data, SQL Server is the better choice. SQL Server also supports more users and traffic, not only through its limited bandwidth traffic, but also with the ability to improve performance, by investing more in hardware through more memory, more CPUs, and more computers. This option is not available for file server databases like Access.

Related to scalability is the option of consolidating multiple Access databases into one large SQL Server database. Although you may combine several Access databases into one, if the size, security, or other requirements exceed Access capabilities, SQL Server is ideal. Having one large repository has the potential for using Business Intelligence (BI) tools and Web publishing with significant business and operational opportunities.

## Deployment Issues

If users don't already have Access installed, Access requires deploying the database and installing Access. Access is large, and different versions of Access or Office can cause problems.

Updating Access databases when updates are released is also challenging. FMS offers a program, Total Access Startup (<http://www.fmsinc.com/products/startup/>), which addresses both the Access version and database deployment issues, but it's not a built-in feature of Access.

An advantage of Web applications is centralization. No deployment is required, assuming everyone has a Web browser. Updates to the application are made in one place only and are immediately available to all users. Any Web application that has more than a modest amount of traffic should use SQL Server instead of Access (with the Jet database engine) for its data storage.

## Limited User Interface

Applications built in Access, unlike Visual Basic, are limited in appearance. Multiple document interface (MDI) applications cannot be built in Access. In general, users know if an application is written in Access. For some situations, programs such as Visual Basic or WinForms in Visual Studio .NET provide a more desirable user experience on Windows.

## Why Access Is Important

### Best Choice for Specific Segment

Access is the best solution for the segment between Excel spreadsheets and more sophisticated database solutions. In the pyramid, this is the area of individual to workgroup solutions. Access is the most popular database in the world, because it services this segment extremely well.

## Many Database Problems Solved by Access

Access does the job well and for many situations, a more sophisticated solution would offer little beyond what Access delivers.

## ROI: Access Solutions Cannot be Cost Justified on Other Platforms

Access is a RAD tool. Solutions created in Access often require less code than other platforms, and can be created by people who cost less than developers of other platforms. Some databases are not worth much. A \$40,000 business opportunity may support a \$10,000 Access solution. But if the IT department can only offer \$50,000 solutions, the choice is simple, because the higher cost solution cannot be justified.

The problem is that over time, some of those small opportunities turn into larger, unforeseen opportunities. Not trying them on a small scale means you may lose a future business opportunity. If your competitors can justify more investment based on the business they generate, the original \$40,000 opportunity could turn into a multi-million-dollar business unit.

## Access Competitive Advantage

By being low cost, Access offers the opportunity to pursue business that would otherwise be left to competitors. A fraction of those seemingly small opportunities may become significant in the future. Being able to profitably participate in such engagements is strategically important for an organization.

## Exploring the Myths of Access Limitations

Access is frequently criticized for its scalability and migration limitations, but this is often unfair. The following sections explain why.

### Most Database Problems Are Small

Most database problems manage relatively small amounts of data, usually well under 100 MB. This is well within the capabilities of Access, and using a product like SQL Server would be unnecessary for such small amounts of data. (although SQL Server does offer features that might be important beyond database size.) Today, this amount of information comfortably fits in RAM.

### Few Database Problems Exceed the Capabilities of Access

Access databases (using the Jet engine) can support up to 2 GB of data. Access applications can link to multiple databases, so even using Jet databases, Access applications can manage a large amount of data. Few database problems involve this much data.

### SQL Server Eliminates the Scalability Issue

Microsoft has designed Access to be scalable. Access applications can eliminate Jet and use SQL Server as its data repository. Access interfaces (.mdb files) can link to SQL

Server data, and ADPs work directly against SQL Server. SQL Server eliminates the scalability issue for data size and number of users.

When people focus on the limitations of Access scalability, it's important to note that the issue is the Jet database engine, and not Access as the front end to SQL Server. It is extra work to migrate to SQL Server or convert an .mdb file to an ADP, but a significant portion of the development investment is preserved.

## Hybrid Solutions

If an application exceeds the capabilities of Access, a hybrid solution with Access and other interfaces against SQL Server is often appropriate. You can create Visual Studio .NET applications for Web solutions against SQL Server, with a role for Access inside the organization for administrative functions and reports. Using Access where it's appropriate maximizes return on investment.

## Using Access Strategically

Now that we have discussed the advantages and disadvantages of Access, how should it be used?

## Reasons to Use Access

The following are some reasons to use Access:

- An organization has a wide range of database challenges, and those challenges evolve over time.
- Access solves the largest segment of the database challenges.
- Database solutions solved with Access offer excellent return on investment.
- Many solutions are not cost-effective with more expensive alternatives.

## When to Use Access

The following are some suggestions for when to use Access:

- **Windows-based, single and multiple user database solutions.** The number of simultaneous users Jet can support depends on what's being done. We generally consider up to 25 simultaneous users to be a reasonable number. Replacing Jet with SQL Server eliminates this limitation.
- **Prototyping.** Often, the prototype is sufficient or "good enough."
- **Cost and concept justifying solutions.** Develop these solutions before starting larger and more expensive solutions.

## Migrating Access Applications

Using Access, like any other database, also means preparing for alternatives when its limitations are encountered. Only a fraction of Access solutions ever need to migrate to the next level. Options include:

- Optimizing and fixing problems in the Access application to keep it in Access. Make sure skilled Access developers are available to support important Access applications.
- Migrating the data from Jet to SQL Server.
- Converting the Access .mdb file to an ADP against SQL Server.

- Converting the Access application to something else, such as Visual Studio .NET  
The choice depends on the enhancements required and the amount of investment that can be justified.

## What to Do with Access 97 Databases

With the prevalence of Access 97 databases and the challenges organizations have with upgrading to newer versions of Office and Access while managing these legacy databases, the question of "What should I do?" often arises. A quick inventory of client computers and file servers reveals a large number of Access databases across an organization. The immediate reaction is that the problem is too large to solve. We believe it is challenging but manageable, because:

- For many Access 97 databases, you don't have to do anything. Like orphaned Excel or Word files, these Access files can be archived because they haven't been used in years and the author is no longer involved.
- Simple individual Access databases and applications (not linked to other databases) can be converted from Access 97 to Access 2000 format with a menu selection in Access. They will then run under a newer version of Access. The Access 2000 database format works with Access 2000, Access 2002, and Access 2003. A limited amount of testing should be conducted to confirm the conversion worked successfully.
- For entrenched, multiple user, and complex Access applications, the migration process could be more expensive and not worth the investment. In these cases, it would be easier to leave the application and continue running it under Access 97. This works if users have Access 97 installed on their computer. There is no problem with Access 97 installed on a computer with other versions of Access. The challenge is making sure it's launched whenever the Access 97 database is opened. This can be done manually with shortcut links that specify the Access 97 .exe file, or by using Total Access Startup from FMS to manage and require this.
- There are Access applications that need to be migrated beyond the database format conversion. This could range from needing the features of SQL Server to fixing bugs or adding new features. This would be like any other system development project where the value of the new functionality must be compared to the cost. In the worst case scenario, the entire application is discarded and a new application started. Even if that occurs, the Access application would serve as a prototype for what users and the organization want from the application.

Moving forward, if Access databases are still used, regular administrative tasks to compact and back up these databases need to be performed to optimize performance and data integrity. Whether you do it yourself or use a third-party product like Total Visual Agent, this is necessary if the data is important. We would also recommend using Total Access Analyzer, a database documentation and analysis tool, to detect if there are any problems, suggestions, or performance opportunities by changing the Access application.

## Using SQL Server Strategically

The following are some suggestions for when SQL Server should be used:

- **Performance with a large number of users.** SQL Server databases scale well across multiple users and are appropriate for high volume situations such as Web

sites. Situations with 25 or more simultaneous users are candidates for SQL Server instead of Jet.

- **Different types of clients.** If multiple applications need to share data (for example, applications written in Access, Visual Basic and .NET), SQL Server is superior to Jet.
- **Security.** SQL Server databases are far more secure than data in file server databases.
- **Database maintenance and integrity.** SQL Server is superior, because its built-in features of automated repair, transaction logs, triggers, and stored procedures ensure the database remains healthy, and changes are properly processed and audited.
- **Professional developers and database administrators.** These people should be trained to manage and configure SQL Server, and optimally develop applications based on SQL Server.

## Conclusion

It's all about database evolution. The database needs of an organization are unpredictable and change over time. Microsoft Access solves many database problems, but not all, and neither do other tools. What Access offers is the best solution for its range of capabilities. As the most popular database product in the world, Access dominates one of the most important database segments.

Access cannot and was not designed to solve every database problem. What it does offer is a cost-effective and quick solution for a wide range of common database challenges in Windows. In many cases, it's "good enough" or was "good enough" for its original mission.

SQL Server is available for more complex solutions. SQL Server offers the critical back-end database that allows organizations to scale their important databases securely.

Anticipate and welcome the natural evolution of databases, and you will find an important role for Access in the overall database strategy of your organization. Compared to alternatives, Access offers tremendous return on investment opportunities.

Organizations that effectively manage and deploy Microsoft Access and SQL Server take advantage of the strengths of both of these industry-leading products, and give their organization a competitive edge financially and in response to user, market, and customer conditions.

### For more information:

<http://www.microsoft.com/sql/>