



153 High St., Box 850  
Sutton, Ontario, L0E1R0  
info@liveddm.com  
www.liveddm.com  
1.877.671.LIVE (5483)  
905.722.5625 (fax)

**Question:** What is the advantage of Microsoft SQL Server over Access?

**Answer:** The following information was compiled from a database professional organization and Microsoft. It clearly reflects the advantages of SQL over Access.

### Introduction

Access utilizes a File Server design while SQL Server employs a Client/Server approach. This is a fundamentally different architecture which has many, many implications.

Note - When we refer to Access in this discussion we are really referring to JET, the standard database engine that is traditionally used with Access.

### Reliability

With Access each client reads and writes directly to the raw data tables. If a client machine crashes while writing data this will usually cause the back-end database to also crash and become corrupt. The same thing will occur if the network fails, has a glitch or temporarily becomes overloaded. This problem becomes more apparent as the amount of data or the number of users increases.

With SQL Server the clients do not talk directly with the tables but with an intelligent data manager on the server. This in turn reads and writes data from and to the tables. If a client machine crashes, or the network hiccups, this will not affect the underlying tables; instead the data manager realises that the transaction has not been completed and does not commit the partially transmitted data to the database. The database therefore continues to run without problem.

The client/server system also maintains an automatic 'transaction log'. If a backup has to be restored the transaction log can be run and should restore all completed transactions up to the time of the crash.

The client/server software itself is designed for mission critical systems and is orders of magnitude more reliable than a file server system. On one system that we support the client used to experience around one to two crashes per year (admittedly their network was not exactly state of the art!) when running with an Access database. After we converted it to SQL Server two years ago the system has not experienced a single crash.

### In Microsoft's own words....

The following comes from Microsoft article Q300216.

"Microsoft Jet is a file-sharing database system. A file-sharing database is one in which all the processing of the file takes place at the client. When a file-sharing database, such as Microsoft Jet, is used in a multiuser environment, multiple client processes are using file read, write, and locking operations on the same shared file across a network. If, for any reason, a process cannot be completed, the file can be left in an incomplete or a corrupted state. Two examples of when a process may not be completed is when a client is terminated unexpectedly or when a network connection to a server is dropped.

Microsoft Jet is not intended to be used with high-stress, high-concurrency, 24x7 server applications, such as Web, commerce, transactional, and messaging servers. For these type of applications, the best solution is to switch to a true client/server-based database system such as Microsoft Data Engine (MSDE) or Microsoft SQL Server. When you use Microsoft Jet in high-stress applications such as Microsoft Internet Information Server (IIS), customers have reported database corruption, stability issues such as IIS crashing or locking up, and also a sudden and persistent failure of the driver to connect to a valid database that requires re-starting the IIS service."

### Data Integrity

Data integrity in SQL Server is enhanced by the use of 'triggers' which can be applied whenever a record is added, updated or deleted. This occurs at the table level and cannot thus be forgotten about, ignored or bypassed by the client machine. For example audit processes cannot be avoided (accidentally or deliberately) with this scenario.

## Performance

With Access all tables involved in a form, report or a query are copied across the network from the server to the client's machine. The tables are then processed and filtered to generate the required recordset. For example if looking up details for one particular order from an orders table containing, say, 50,000 records then the whole table (all 50,000 records) is dragged over the network and then 49,999 of these records are thrown away (this is an over-simplification since indexing can be used to mitigate this to some extent). Contrast this with SQL Server where the filtering takes place on the server (if designed properly) and only 1 record is transmitted over the network.

This can affect performance in two ways. Firstly SQL Server is highly optimised and can usually perform the required filtering much more quickly than the client machine and secondly the amount of data sent across the network link is vastly reduced. For most databases the main performance bottleneck is data transmission over the network hence reducing this can give a really dramatic improvement in performance.

Predicting likely performance improvements is very difficult but an average overall speed improvement of 3 to 5 times, and possibly much more, would not be unexpected.

## Network Traffic

As can be seen from the previous section, network traffic is greatly reduced in a client/server scenario, often by many orders of magnitude. This both improves network reliability (by reducing collisions, etc.) and also improves the performance of the network for other software (as there is less traffic on the network). Where there is a slow connection, such as over a telephone dial-up, Access is usually so slow as to be all but unusable (obviously this does depend upon the amount of data) whereas a SQL Server application, if designed for this environment, can still be perfectly useable.

## Low Bandwidth

This occurs when you are accessing your database over a connection that only supports low data speeds, which, for all practical situations, means anything other than a LAN. In all low bandwidth situations Access/JET usually performs so slowly as to be unusable whilst a correctly designed SQL Server system can be similar in speed to running it over a LAN. The main low bandwidth situations are:

- **Dial-up.** Allowing remote salesmen, off-site workers, home workers, out of hours users and the like to dial into the network over the normal telephone lines. Most file server databases are completely unusable over dial-up unless some additional technology, such as Terminal Server, is used (and this brings it's own complications).
- **WAN.** If you want to link more than one site to a database then typically you would use a WAN (Wide Area Network). Irrespective of the communications technology used (which would usually be leased line, VPN {Virtual Private Network} or ISDN), WANs tend to have a low bandwidth compared to LANs and in addition are often heavily loaded with traffic. Traditional file server databases do not work well over a WAN and will often have both performance problems and reliability problems (owing to the less than perfect connections that most WANs provide).
- **Internet.** A database that is being run over the Internet needs to be stable, scalable, able to handle heavy loads and capable of coping with failed connections; none of which are usually associated with file server database architectures. Small scale, non-critical databases can be run over the Internet but in most situations you should migrate to a client/server design.
- **Wireless LAN.** These are increasingly popular and are usually fine for accessing a spreadsheet or Word document where a wired solution is inconvenient or is just not practical. However file/server databases do not usually work well over most wireless links due to the low bandwidth that they offer (even a 10Mhz wireless link will usually operate at only half of that speed or less).

### **Scalability**

A file server system such as Access is designed for small workgroups and is scalable to perhaps 10 concurrent clients. Above this level performance starts to degrade rapidly as more users are added. With the SQL Server client/server architecture many hundreds, or even thousands (with the appropriate infrastructure), of concurrent users can be supported without significant performance degradation.

### **Drawbacks**

SQL Server is a (much) bigger and more complex beast than is Access. Although it is now easier to manage than in the past it is less suitable for a company with no IT support staff (in-house or outsourced) than is the simpler Access. It also costs more to implement than does Access (you need to buy SQL Server licenses whereas the Access run-time version is royalty free) and the development of a SQL Server system will usually take longer and cost more than an equivalent Access system.

### **Summary**

The main benefits of SQL Server over Access are improved reliability, better performance, reduced network traffic and increased scalability. Drawbacks are a slightly increased deployment cost and a more complex support environment. For small workgroups of up to ten users on a Local Area Network with modest data requirements (no more than 50,000 total records ie. Individual pieces of information) and without high reliability requirements then Access can be a satisfactory option . Outside of these parameters you should probably look to a client/server solution such as SQL Server.