Pervasive.SQL vs Microsoft SQL

So why do Exchequer Software use Pervasive SQL instead of Microsoft SQL or Oracle?

Exchequer has identified Pervasive.SQL as the best tool for developing Enterprise with in terms of reliability, performance, robustness, scalability and cost of ownership, some of the many reasons are;

- Pervasive.SQL supports NT, Novell, Sun, and Linux operating systems.
- Pervasive.SQL does not require a database administrator to look after it.
- Exchequer keeps winning awards from its rivals who are based on MS SQL or Oracle. Why? Because accounting systems are not solely about the database or language they are written in, but about the features and functionality they provide.
- Enterprise is an impressive system *because* of the technological foundations upon which it is built.
- Many of the advanced features like drill down, OLE server, open period design are a reflection of what the underlying technology can provide. Take either one of those ingredients away and you have a different recipe.

But wouldn t it be easier to integrate Enterprise with other applications if it were written in MS SQL or Oracle?

For the most part it makes little difference which database sits behind Enterprise for extracting data, because of the tools Exchequer make available to automate these tasks for you.

For extracting data *from* Enterprise there are numerous ways to achieve this: -

- Report Writer
- OLE COM Server for Excel
- ODBC
- COM Toolkit
- DLL Toolkit

(The main technology used to connect to MS SQL is also ODBC)

But what about inserting data into Enterprise?

One would assume that sharing the same database between separate applications would make things very simple. The reality is that no serious accounting system allows direct access to its files for insertion – as this would be extremely dangerous to the integrity of the data.

Again for the most part it makes little difference which database sits behind Enterprise to insert data from, the reason for this is very straight forward;

When an accounting function is performed, numerous business rules happen at the same time causing simultaneous updates of associated tables. With the best will in the world an external application has very little hope of emulating those business rules accurately, especially when you consider upgrades. This would therefore compromise the integrity of the accounting data making it impossible to prove the systems audit.

For example lets step through adding an invoice into Enterprise. As the invoice is stored, the following business rules are executed: -

- The header and footer information is stored within the database.
- The customers balance is updated
- Each line of the invoice is stored as a separate record within the database.

As each line is stored, the following business rules for each line are executed: -

- The stock levels are adjusted.
- Stock valuation tables (like FIFO) are adjusted.
- Multi location stock balances are updated.
- Customer stock on screen analysis is updated.
- Job costing values are updated
- The cost center/departmental analysis is updated.
- General Ledger balances are updated.
- Commitment balances are updated.
- The Tax control totals are updated.
- The local as well as transaction currency totals are updated.
- And so on for at least another 50+ operations

Just imagine having to do all that every time you write/edit a record in the database!

Pervasive.SQL vs Microsoft SQL

So what's the solution?

Imagine having to replicate that sequence – impossible. This is why we provide a developers toolkit as a DLL or COM technology to hide away the complexity for the business rules and allow you to simply add in an invoice without having to worry about the business rules.

Microsoft's DNA & .NET initiatives all describe the need to separate the business logic away from the data, making the underlying database a simple data tool as it should be.

Virtually all accounting systems will need to protect the integrity of their data from direct access, so the underlying database technology really is irrelevant providing you have comprehensive access to and from the data. It really makes no difference if an accounting system is written using a particular flavour of SQL.

What is the client server architecture of Exchequer Enterprise?

Exchequer Enterprise supports a two tier model for its client server technology, meaning that;

- The client server database engine on the server performs all data access.
- Each workstation sends requests via a communications protocol like TCPIP to and from the server requesting data or instructing certain operations to take place.
- The workstations do not open the tables themselves; in fact only one set of tables is opened by the server.
- Record locks and transaction control are all synchronised by the server.

The result is a very fast system with minimal network traffic across it because most of the communications between the server and workstation does not contain any physical data.

The workstations do perform some local processing to prevent the server being overwhelmed with work, for example when compiling reports the workstations will keep running totals rather then rely on the server to provide them. This provides a very balanced optimized system making efficient use of the processing resources available on a typical network.

Why does the Client/Server version of Enterprise need a Mapped Drive, and Access Rights to the data directories?

The access rights required to the Enterprise directories are not related to the client server technology in any way whatsoever. Enterprise is most definitely client server, in that;

- The data is stored and accessed exclusively via a database manager (Pervasive.SQL), which resides on the server, using networking communications protocols.
- Only one set of files is opened by the server Pervasive engine, all users access the data via TCPIP or IPX connections and are not hard connected to the data tables at any time, but are controlled by indirect calls to the database manager.
- Enterprise requires a mapped drive as a convenient means of resolving the Pervasive SQL server IP, or IPX address, thus ensuring compatibility across NT, Novell, Linux and Sun Solaris operating systems, which Pervasive support.
- The only reason Enterprise requires full access rights to the directories is due to the fact Exchequer have chosen to store the system in a central place and because during the production of reports Enterprise generates temporary files which it then deletes after the report has finished (these files generally end in .SWP).

This fact has no bearing on the client server data connection whatsoever, and in no way lessens the data integrity of the system, it is purely an organisational decision that we as the software developers have taken.

In v4.31 these temporary files have been moved to a SWAP directory so that only this sub directory requires full rights, and the remaining directories can simply be setup with read/write access instead. We therefore confirm that Enterprise is based on a true client server technology.

Pervasive.SQL vs Microsoft SQL

Pervasive	**v**	**Microsoft**
Dyson	**v**	**Hoover**
Technically Better	**v**	**Market Brand Leader**

What is more important?

a.) The brand behind the database
b.) The functionality of the accounting system

What is more important?

a.) Keep the IT staff busy
b.) Keep costs down, run fast, easy-to-use and be maintenance free

The decision process behind buying an accounting system should be in the following order:

1. MD / FD
2. Accountants
3. Users / Operators
4. IT Manager

Unfortunately, the selection of accounting software is often delegated in the early stages to an IT Manager, whose priorities are often quite different to the direct users of the software.

IT Managers will often dictate a choice of database first, and accounting system features second. This is not the best way to assess a tool destined to run an organisations finances for a considerable length of time.

The database argument will begin fading into the background as Microsoft's .NET initiative separates the database technology from the business rules.

Ironically it is not possible to bench mark against Microsoft SQL because Microsoft's license agreement prevents you from publishing any benchmark results without their permission. This has stifled any objective comparisons.

- Pervasive.SQL is used by more financial based applications than any other database.
- Great Plains (authors of Dynamics) have over 85,000 sites running on it in the US.
- Peachtree (owned by Sage) have even more sites, running into millions of users running on Pervasive.SQL.
- Maximizer contact management products have over 25,000 sites running on it, as to Macola (US) with around 11,000 sites.
- System's Union, authors of SUN have most of their current users running on Pervasive.
- One of the largest hospitals in the US has a single 2,000 user network running the hospitals administration on Pervasive.SQL.
- In terms of volume, Exchequer have an Enterprise site in the UK (H J Heinz) who import over 1 million records every day from their mainframe system into Enterprise using Pervasive.SQL. This system has been running under NT for three years now. The data files run at around 5-7 gigabytes.

Microsoft SQL should not be considered a rival to Pervasive SQL

- Pervasive will run along side MS.SQL on the same server without conflict.
- The tools used to access both databases are very similar
- In the near future the underlying database will become transparent to the systems using them under Microsoft's .NET initiative, since they will all have to conform to a common gateway.

Pervasive.SQL vs Microsoft SQL

Five-Year Cost of Ownership Comparison
Pervasive.SQL v Microsoft SQL 7.0 - 10 Clients

Cost Category	Pervasive	Microsoft
DBMS One Server plus Clients License	$995	$1,999
DBMS Development Tools	$295	$549
Installation	$900	$3,600
DBA Cost (Year 1)	$450	$11,000
DBA Cost (Years 2 to 5)	$1,800	$44,000
Training Developers	$1,500	$1,275
DBAs and Server Administrators	$3,625	$3,400
Documentation	$250	$100
Two Upgrades Over Five Years	$990	$2,874
Developer/DBA Support Cost (Year 1)	$1,700	$3,390
Developer/DBA Support Cost (Years 2 to 5)	$3,400	$6,780
Server Hardware	$7,307	$7,307
Operating System and System Software	$1,129	$1,129
Total	**$24,341**	**$87,403**

© Aberdeen Research Inc.