



Classy chassis

Mark Whitehorn turns his attention to “professional” quality hardware for client server databases and what you can expect for your money. Plus, tips and tricks for Access.

Last month I was trying to “sell” you a very expensive piece of kit. I hoped to convince you that standard PC chassis were inappropriate as servers for mission-critical databases and that you ought to buy something expensive from a reputable manufacturer.

A fair question here is, “What do I get for my money?” In other words, a “professional” server is likely to cost twice as much as a PC of the same specification, so where does the money go? The following list is not exhaustive but it gives some idea of the extra hardware features you can expect from a “professional” server:

- RAID disk arrays.
- Fault monitoring and prediction on hard drives, processors and memory (anyone who remembers HAL at work in 2001 — A Space Odyssey will appreciate this).
- Pre-failure warranty, which allows the components identified as liable to failure, to be replaced before they do so.
- Dual peer PCI buses for high I/O bandwidth.
- Array controllers which allow RAID storage to be added while the system is in operation.
- Tape backup systems which provide rapid backup (40Gb per hour).
- Redundant power supply and UPS options.
- Redundant controllers on standby in case of failure.

Most of these are self explanatory. However, a brief word about RAID (Redundant Arrays of Inexpensive Disks) may be helpful. “Brief” is the operative word, as there are several flavours and the area is relatively complex. A RAID allows you to store your data on several disks, which holds a number of advantages as far

as database servers are concerned. For a start, a RAID can be set up in such a way that if even one of the disks fails, your data remains safe because each piece is stored redundantly on more than one disk.

Clearly, this has the disadvantage that although you might have five 2Gb disks and therefore 10Gb of disk space, you won’t be able to store 10Gb of data. The data is more secure, though. Additionally, there is a rather elegant spin-off from the redundant nature of RAID which is that the data can often be retrieved more rapidly.

Furthermore, RAID can be hot-swappable, so if a disk fails you can extract the duff one, slot in a good one and the data which was on the failed disk will be reconstructed from the others. The database can continue to perform, albeit more slowly, until the new disk is fully online.

It doesn’t take a genius to work out that the money is going into two main areas: speed, and keeping the server up and running. You may never need the RAID array; that redundant controller may spend all its time idling. Nevertheless, it would make your day having arrived one morning to discover that even though one of the disks has failed, your system is still running.

Remember it well

A database server needs RAM, and then it needs more RAM. A database server will use RAM to hold the RDBMS engine and to cache data, transactions and indices. The more RAM you give it (within reason) the faster it will work. Think in terms of 64Mb as a minimum and 100Mb as a more reasonable starting point. Make sure the server you buy has free memory slots so the

Time, gentlemen, please...

Following reader Gareth Wade’s query about adding time lengths to give a running total, (*Quickies*, PCW April) I have since received a couple of solutions from others.

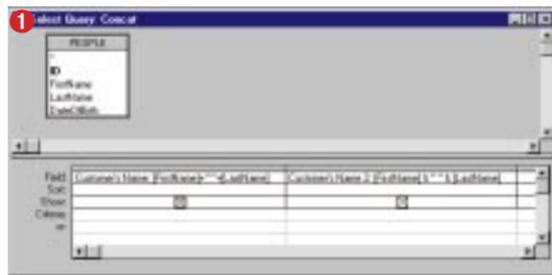
■ MA Roberts writes: “Regarding the ‘Database Quickie’ from Gareth Wade (PCW April) I have had a similar problem to Gareth’s when adding together time values. The short time format field in Access does not work satisfactorily for adding values that will exceed 23:59. One workaround I am using is to convert the time field into a number type. When adding time records together in a number-type field, Access converts the time to a fraction of the 24-hour day. It is therefore necessary to multiply this field by 24 to get a value for hours and decimal minutes. This field may then be used to obtain a value for time (in hours and decimal minutes) multiplied by an hourly rate such as might be used in a time-sheet style application.”

■ And Vidar Eggen writes from Oslo: “I had the same problem when wanting to report on the added flying time of pilots as well as aircraft at my flying club. What we really need in Access is the [tt:mm] format of Excel. Until then, I use this basic formula:

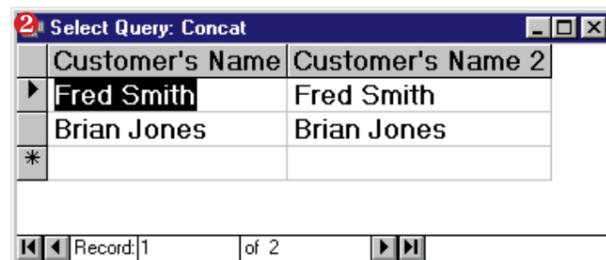
```
=Str$(Sum(Hour([Time]))+Sum(Minute([Time]))\60) & ":" &
Format$(Str$(Sum(Minute([Time])) Mod 60); "00")
```

“The output of this formula will be ‘123:45’ in text format. The reason for converting the sums to a string is to make sure that five minutes is printed as ‘05’, not ‘5’. If quoting from this, please rephrase my language into readable English!”

(Vidar’s English seems fine to me and is considerably better than my Norwegian! — MW)



machine and turned it loose on a large database. Rather to my surprise Access used all the four processors on the machine and was able to query 1,000,000 indexed records in a couple of seconds. Interestingly,



Excel used only one of the available processors and its performance was startlingly unimpressive.

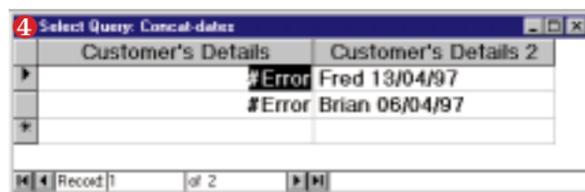
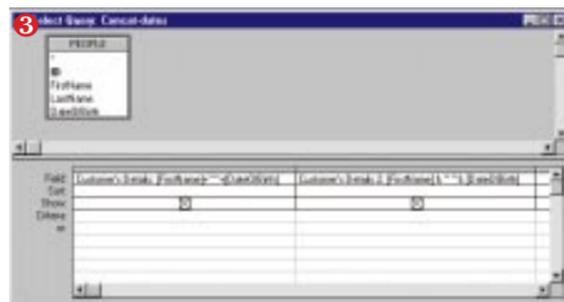
So the bottom line is, what do I use as my test database server? I am

memory can be expanded later without replacing all the RAM it currently holds.

Purchasing processors

In general, buy several processors. Simple operating systems can only use a single processor, but complex ones like UNIX, VMS and NT4 Server can use multiple processors. Mind you, just because an operating system can make use of several CPUs doesn't mean that the programs which run on top of it can too. But all the respectable RDBMSs can and do, and this isn't restricted to client-server RDBMSs.

I recently mounted Access 97 on an NT4



currently running a Compaq ProLiant 5000 with four 166MHz Pentiums, 384Mb of RAM and 9Gb of disk space. I look upon it not so much as a server, more as a non-intrusive sleeping aid. Okay, it is more expensive than the tablets, but it has fewer side effects.

Listing 1

```
"Report generated " & Now() & " for " & [Customer]
```

Listing 2

```
Between Date() - 7  
And Date()
```

Listing 3

```
Between #1/1/1981# And Date()
```

Tips and tricks for Access

By popular demand, I am introducing tips and tricks this month. It is aimed at Access users and will include information which ranges from simple to complex and should provide something for everyone.

1. **Queries**, by default, use the field name as the column name in the answer table. If you ever want to alter this, insert the text you would prefer to use, followed by a colon, before the field name. Thus:

```
Customer's Name: [FirstName]
```

will replace "FirstName" with "Customer's Name" in the answer table. Which leads us neatly to the second tip.

2. The **"+" operator** can be used to concatenate text strings, as many people know. Thus:

```
[FirstName]+[LastName]
```

in a query will join the two strings together. But this unhelpfully yields names like "BrianJones". So, simply add in a literal space like this:

```
[FirstName] + " " + [LastName]
```

which then gives: "Brian Jones".

However, you can also consider using the lesser known "&" operator. Amazingly,

```
[FirstName] & " " & [LastName]
```

gives exactly the same result: "Brian Jones" (Figs 1 & 2). You're thinking, "Hello, he's flipped. If it gives the same result, why bother to use it?" Well, the advantage of getting into the habit of using "&" is that it automatically converts all data types into text. This has no effect if the two fields are already text (as in the case above) but it does wonders with expressions like

```
[FirstName] & " " & [DateOfBirth]
```

which will produce "Brian 06/04/1967" rather than the "#Error" produced by the + operator when it tries to concatenate data from two disparate data types (Figs 3 & 4). In other words, if you habitually use "&" instead of "+" you won't have to use type conversion functions. You could even use it in a report like Listing 1.

3. **When adding fields to a query grid**, you can drag and drop them as usual. However, you can also select multiple contiguous fields using shift-click on the first and shift-click on the last, and then dragging and dropping as usual. You can also add non-contiguous fields by CTRL-clicking on them.

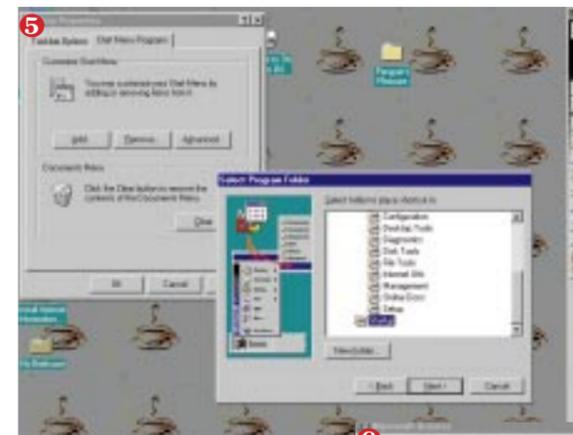
4. **When looking at a table of data**, you can resize any field to fit the widest data in the field by double-clicking on the right-hand column border (this has to be done at the top of the table). You can hide the column by dragging the right-hand column border across to the left-hand border. You can open this up again with the mouse but it takes practice, as you often simply widen the adjacent column instead. It is often easier to use the menu system Format

Listing 4

```
Set button OnClick event to "=Clicked([Screen]. [ActiveControl]. [Caption])"
```

Listing 5

```
Function Clicked(capStr as string)  
    Debug.Print capStr  
    ' Or assign it to the parameter for the query  
End Function
```



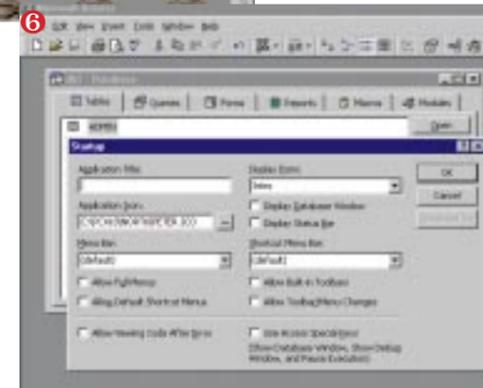
delay would be seven days, so the default could remain as Date() and the validation rule could be set as Listing 2. The "Between... And" operator is a useful one to get to know. You can use it with fixed dates like Listing 3 or for data types other than dates: Between 1 And 45

Show Columns, or Format Unhide Columns, in Access 97.

5. **Record selector**. Most of us have discovered the delights of the "Find Specified Text" button (the binoculars) which will search a table for information. By default, this will search the current field and the dialog box allows you to specify the entire record. However, it is often easier to click the record selector (at the left of the record) before starting the search and this automatically sets the search to run against all fields. This works in form views as well as in datasheet views.

6. **"Between... And"**. As regular readers will know, I am obsessed with the idea that the data in our databases should be as "clean" as possible. Take something as simple as a DateOfBirth field in a maternity database. The default (for the child, not the mother) might be the function Date().

But perhaps not every child's record might be entered on the day (staffing levels being what they are) so the simple answer is to allow any entry. But that opens the door to really whacky entries. Suppose we know that all entries, no matter how delayed, are always brought up to date on a Sunday, say. This means that the longest



Gaining automatic Access

A reader has written in for help: "I'm struggling with Access 97 for NT, trying to build a database for other people to use. I have seen other databases where a form opens up automatically, which is what I want but how do I do it?"

"Additionally, since many of my users are not used to databases, is there any way of limiting the options which are available to them when Access is running?"

"Finally, is there a way of getting Access to start as soon as NT fires up? Can I set up the machine so that whenever the user switches on the PC, it goes into NT, lets them log in, and then goes into Access?"

The simple answer is yes. Let's start from the outside and work inwards. To fire

up a copy of Access, you can add it to NT's StartUp menu. Click on Start and choose Settings, TaskBar and the Start Menu Programs tab. Click Add and then browse in the normal way to find Access. Then click on Next, select the StartUp folder and finish. This seems more complex than it used to be in Windows 3.1 — weren't operating systems supposed to be getting easier to use? (Fig 5)

Access 97 has a new startup form which allows you to control many options when it kicks into life. You can get to this by right-clicking the border of the Database window and choosing StartUp, or go via the menu with Tools, StartUp. Fig 6 shows the sort of options that are available.

The subject of passing captions from buttons has exercised the brains of several readers. For instance: "I was intrigued by the buttons and captions capture question. Your method in the February issue appears to be the way to go and can be speeded up using [Listing 4] (using Access 7) which calls [Listing 5]." Kelvin

Malcolm Fraser, too, sent in a solution and it's on our cover-mounted CD, this month, in a file called CAPTIONS.MDB. Grateful thanks are also due to Steve Foster, Richard Todd, Malcolm Bacchus, Wilf Davies, Alan Berry and others who provided elegant solutions.

Beyond the pale™

I discovered the following rumour on the web site of Bloor Research, at www.bloor.co.uk. It appears that the word Metadata™ has been trademarked™ to a pharmaceutical™ company in the US. At first sight this sounds (if you'll forgive the mixed metaphor) unbelievable, but I suppose those people who issue trademarks in the US may not be as familiar with planet database as those of us who live there. To make matters worse, the company that was awarded the trademark™ has started sending "Cease and Desist" letters to anybody it finds using, er, well, that word™. I'm currently trying to trademark the word "pharmaceutical™" in the hopes of wreaking some small revenge™.

PCW Contact

Mark Whitehorn welcomes readers' correspondence and ideas for the Databases column, at database@pcw.co.uk.