Fresh fields

Mark Whitehorn comments on areas of Microsoft's internet strategy relating to the future of DBMS. Plus, the TechNet troubleshooter CD-ROMs.

I have just returned from Microsoft's Tech Ed conference where Microsoft's internet strategy was much discussed. There are two areas of said strategy which are likely to be of major interest to DBMS fans.

The first is the rocketing growth in tools which provide web access to databases. Tools for Microsoft's Access and SQL server have already been released and the next six months will see rafts of things appearing. What makes all this doubly exciting is that being a new field, there are no standards or yardsticks for comparison. So, for a while, we are all going to live in interesting times.

The second issue concerns the way in which all the data is controlled on the Internet. At Tech Ed I discussed this with Microsoft's Brad Silverberg (senior vice president, Internet Platform and Tools division), and he used Microsoft's TechNet (*see overleaf*) as an example. The data in TechNet isn't organised as a classical relational database; instead, it is mostly text and is organised more like a huge hypertext help system. This is currently distributed as a set of three CD-ROMs which are sent to each subscriber on a regular basis.

He said that this type and quantity of data should be on the Internet. We might envisage a situation where the data was originally distributed on CD-ROMs to each subscriber, transferred to writeable media and then subsequently updated from a web site. The problem, as Brad pointed out, is that this huge volume of data is not only expanded month by month, but additionally the existing data is edited and updated. Currently there are no standards on the internet for flagging changes to data with time, so working out a system for downloading just the changes is a nightmare.

Well, there may be no standards on the Internet, but this problem has been well understood in the database world for years. In fact, it is a small and relatively simple subset of the problems which can occur when data is replicated across two

or more sites. All of which suggests to me that we can expect to see two distinct changes over the next few years.

Currently, web sites store essentially static data. As tools become available to tie RDBMSs to web servers, we will rapidly move to a situation where web sites display mainly static data with some embedded dynamic data which will come from an RDBMS. Some sites already do this. However, I don't think we'll stop there.

The web is going to become the repository for mind-bogglingly huge amounts of data. Unless that data is properly organised, it will rapidly become unmanageable. I believe that we will shortly see tools arriving which combine RDBMS and web server functionality. This combination will be conceptually very different from linking because the RDBMS will take over the management of *all* the data that the server presents to the world. With a tool like that, management of the information in TechNet will be simple.

The good news is that it looks like our databasing skills will be needed for many years yet, solving all of these new database problems.

In-flight databasing

I was both interested and exasperated to find myself beset by another database problem on my return journey. It was the old story: breakfast in Nice, lunch in Edinburgh, luggage in limbo. I didn't see it again until the following day. The enforced separation from my dirty socks for 24 hours didn't worry me; what concerned me was the cavalier manner in which airlines use, or do not use, the information they hold.

Airlines manage massive amounts of information about discrete items of data: passenger details, seat allocations and luggage details. For years they have managed to control the passenger and seat allocations reasonably well (failure to do so creates fist fights in the aisles, which are bad for PR). The control of luggage was traditionally less well implemented until the arrival of terrorist activities. It is now, shall we say, "politically incorrect" for an airline to be uncertain of the whereabouts of a given piece of luggage.

Many of us have been held inside a stuffy aircraft while the airline tries to find that last elusive passenger and tells us that, unless he or she is found, the entire luggage hold will have to be eviscerated. I have never complained, because it tells me that the airline knows which pieces of luggage are on the aeroplane and I'm duly grateful: by extrapolation we can assume that it also knows which luggage *isn't* on board.

So the database problem is easy. You know which passengers are in which seats, and which are unlucky enough to be parting from their luggage at 500 knots: it should only be a trivial task to send the aircraft, in flight, a list of those hapless passengers so that they can be informed before arrival.

What actually happens is that you, rather than the airline, do the database work. You hang around querying the carousel until it returns a null; in my case, this took about half an hour. Then you submit a query to the baggage reclaim server, only to discover that there is a FIFO (First In First Out) queue and you are L (Last). When your query reaches the head of the queue, you discover that the server performed a replication with London some

---

On my bookshelf

The SQL Server Handbook — A Guide To Microsoft Database Computing, *by Ken England and Nigel Stanley.*
Covers SQL Server version 6.0, and although 6.5 has just arrived, a large proportion of the book is still relevant. Database servers are substantially different from PC RDBMSs, so for the many people who are currently moving up, such a book has to be well worth considering.

The authors avow that it is "definitely not intended to be a re-hash of the documentation set", which is certainly true; instead the book concentrates on the working principles behind SQL Server. Thus, instead of simply telling you *how* to create a device, it explains *what* a device is, and why you need them. The subject areas covered range from devices, through data

integrity and database concurrence, to database administration and integration with Access and VB.

In general, the book is well laid out and readable. I had only two real criticisms. Firstly, it tends to favour the command line. For example, we are told that a device can be created using TRANSACT-SQL (after first using "sp_helpdevice" to identify an unused device number). The syntax is:

```
disk init
name = 'authors'
physname = 'c:\sql\data\aut.dat'
vdevno= 2
etc…
```

After all that, there is a brief note that using the GUI tool SQL Enterprise Manager will, among other virtues, locate an unused device number for you and specify it
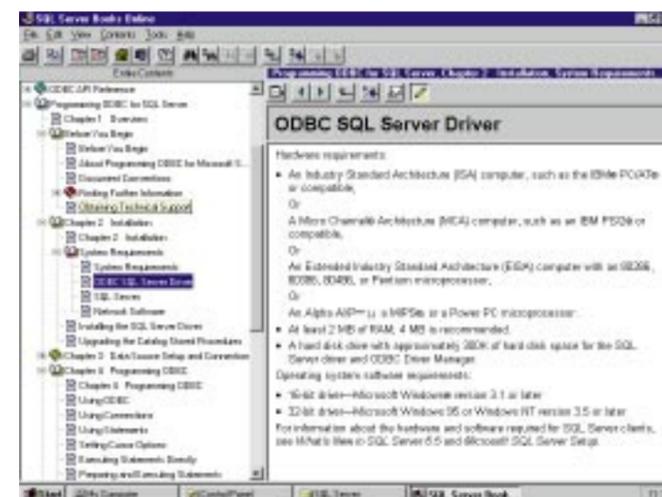
automatically. I don't know about you, but I'd choose a GUI tool any day.

Secondly, there isn't enough text devoted to the mechanics of driving a server. For example, in the above section, we are told that we "need to obtain a device number using the system procedure 'sp_helpdevice'". What the novice user *isn't* told is how or where to issue that command.

Nevertheless, this book offers an invaluable background. Considering the amount that you will be spending on the server, the expense must be worthwhile.

● *The SQL Server Handbook — A Guide To Microsoft Database Computing* by Ken England and Nigel Stanley. Digital Press, ISBN 1 - 781555 - 581527. Available from ICS Solutions (01256 469460). Price £29.99.

On my bookshelf

The online help in SQL Server is as good as online help ever seems to be, which is why a text-based book is still desirable

2 6 8
PERSONAL COMPUTER WORLD
SEPTEMBER 1996

2 6 9
PERSONAL COMPUTER WORLD
SEPTEMBER 1996

time ago and already knows about your luggage… ARGHHH! Let's use the technology properly, guys.

TechNet troubleshooters

For those who haven't come across it yet, Microsoft has compiled an excellent source of information onto a series of three CD-ROMs entitled TechNet. They're a mine of troubleshooting information about a range of Microsoft products. Be warned, though; it takes a pretty hefty outlay for access to this resource *(see PCW Contacts, page 273)*.

The extracts which follow give a flavour of the kind of information you'll find:

Q. *When I add two tables to my query that do not have a defined relationship, Microsoft Access automatically joins them. Can I prevent this from happening?*
A. Microsoft Access 2.0 automatically joins two tables in a query if the tables meet the following criteria:
● There is no relationship defined between the tables.
● Each table contains at least one field whose name and data type matches the name and data type of a field in the other table.
● One of the tables has a primary key defined on the matching field.

Only one AutoJoin is automatically created between two tables. Even if there is more than one join possible between the two tables, a join is created only between the first fields that meet the above criteria. If you add three tables that meet the above criteria to a query, three joins are created; one for each table pair. You cannot turn off this functionality: you must either delete the join line after it has been created, or manually define a relationship between the two tables.

Q. *Why can I update more fields in my query than I could in Microsoft Access version 1.x?*
A. In Microsoft Access 2.0, when a query includes fields from more than one related table, you can update data on both sides of the join. This means that in a query which combines data from two tables, you can update data from both of those tables.
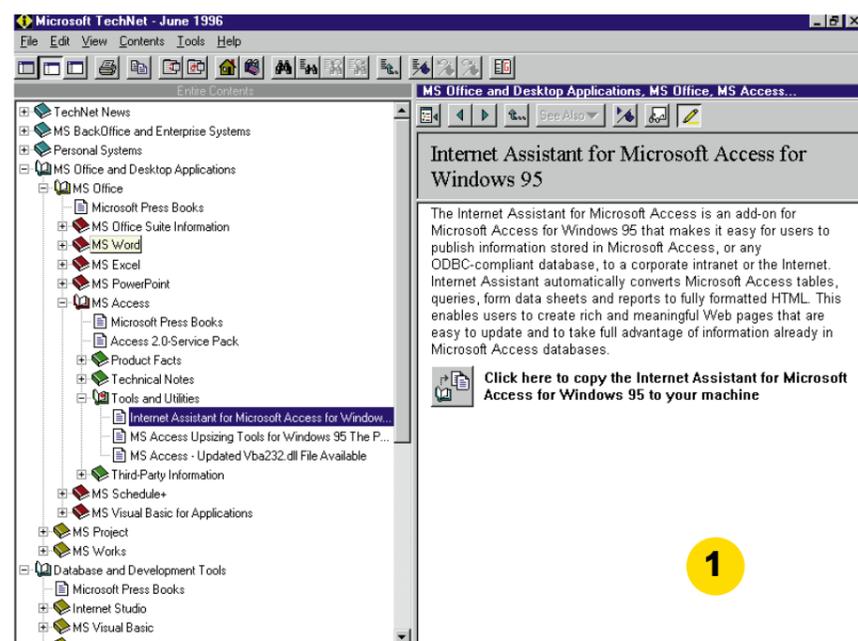
If you want to prevent users from updating fields in a multiple-table query, create a form based on the query and then set the Locked property for the fields you do not want users to update.

Q. *Why do I see a number instead of "(counter)" for my counter field?*
A. Microsoft Access 2.0 enters a counter value when you start to edit a new record. In Microsoft Access 1.x, this value was entered after you saved the record. Since this value is now provided earlier, if you start editing a new record and then cancel it, the counter value is still used even though no record is stored with the value. For example, when you add a new record to a table containing two records, the counter value is three. If you cancel this new record and later add another new record, the counter value is four for the new record.

Counter values are not re-used when you delete records. For example, if in a table of 15 records you delete the last three records and then add a new one, the counter value for this new record is 16. To reset the next available counter value, compact the database. After the database has been compacted, the next available counter is set to one higher than the last counter value in the table.

Q. *Why is the data sorted in my query but not in my report?*
A. Reports create their own internal queries to present the data. If you want to present the data in your report in a particular order, you must set the sort order explicitly in the Sorting And Grouping

dialogue box. To do this, open the report in Design view and then choose Sorting And Grouping from the View menu.

Q. *How can I keep a group of records together in a report?*
A. The new KeepTogether property for groups in Microsoft Access version 2.0 gives you the ability to keep groups of like information together. This property is available in the Sorting And Grouping dialogue box for reports. Using this property, you can keep an entire group together (including the group header, all records and the group footer), or keep the group header with the first record.

Q. *Why is every other page of my report blank, and how can I correct this problem?*
A. The problem occurs when the total width of your report exceeds the width of the paper specified in the Print Setup dialogue box. For example, blank pages print if your report form is 8ins wide and your left and right margins are 1in wide for a total width of 10ins, and if the paper size specified in the Print Setup dialogue box is only 8.5ins wide.

Using this example, if controls (such as text boxes) extend beyond 8.5ins, the controls are printed on a second page. Otherwise, you receive a warning message stating that some pages may be blank. Blank pages generated after the warning are not counted in the total pages of your report.

Windows

The following information applies to

Micro-soft Access versions 1.0, 1.1, 2.0 and 7.0:
Symptoms: *Using a make-table (SQL Select...Into) or an append (SQL Insert...Into) query with criteria that have no matching records, causes an empty table to be created. In Microsoft Access versions 1.x and 7.0, if the empty table contains a counter field, the first record added to the table will have a counter value of one. In Microsoft Access version 2.0, the first record added will have a counter value of two.*

Cause: Microsoft Access version 2.0 provides the first value for auto-increment fields internally. Because it provided 1 internally for the previous auto-increment column, the next value is 2, which shows in the new table.

Resolution: In version 2.0, create the table manually, instead of using a make-table query or append query, to start the counter at 1.

Status: This behaviour is by design.

Steps to Reproduce Behaviour:
1. Open the sample database North-wind.mdb (or NWIND.MDB versions 1.x and 2.0).
2. Create a new query based on the Employees table.
3. On the Query menu, click Make Table. Enter "Empty Table" (without the quotation marks) in the Table Name box, and then click OK.
4. Drag the EmployeeID and LastName fields to the QBE grid.
(Note: In versions 1.x and 2.0, there is a space in the field names Employee ID and Last Name.)
5. In the Criteria row for the EmployeeID column, enter "<1" (without the quotation marks).
6. Run the query. Note that a new, empty table called Empty Table is created.
7. Open the Empty Table table and enter a name in the LastName field. Note that the counter starts at two instead of one in version 2.0. In versions 7.0 and 1.x, the counter starts at one, as expected. ■

Fig 1
In full swing: you can use this to download new and exciting add-ons which are supplied on the TechNet CD
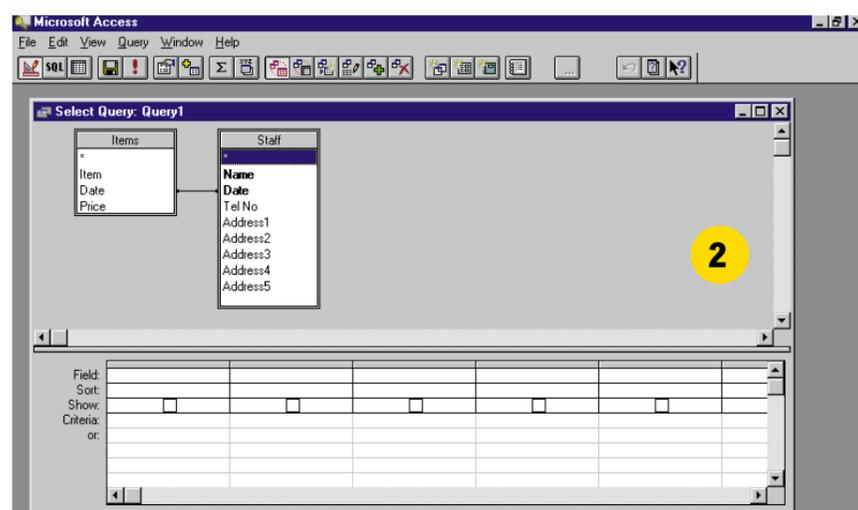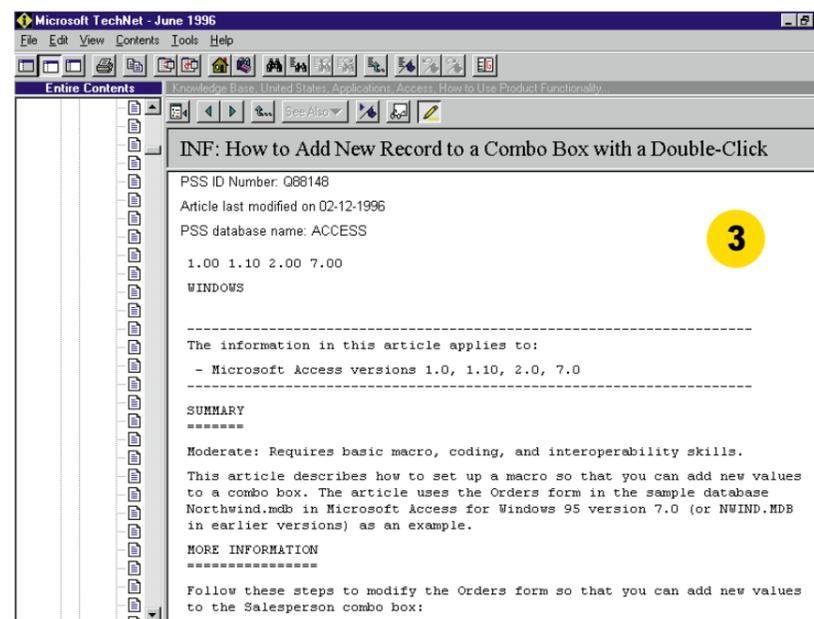
Fig 2
TechNet is also a source of riotously interesting information

Fig 3
The behaviour described in the first extract from TechNet — Microsoft Access, making joins between tables in a query. In some cases (such as this) the assumptions it makes may not be exactly what you had in mind

PCW Contacts

Mark Whitehorn welcomes readers' correspondence and ideas for the Databases column. He's at **m.whitehorn@dundee.ac.uk**

MS TechNet costs £249 per year for a single user; £550 per year for the server-based version with unlimited access. (Both prices excl. VAT). **0800 281221**