

Join the gang

Mark Whitehorn finds another variant of the gang-screen, struggles with Access, and covers more of Ted Codd's rules for running a tight database.

Is there no end to the gang-screen wars? Having reported last month how to find the gang-screen in Paradox for Windows, and some time ago the gang-screen in dBase 5 for Windows, I came across a sub-variant of the dBase one. If you open up the Help-About dBase 5.0 for Windows screen, and press ALT-28 (hold down the left-hand ALT key, and then press the 2 and 8 keys, above the normal keyboard), the number 28 strolls across the screen. This was apparently inserted into the code by a former Borland employee called Kevin Brown as a tribute to racing driver Davey Allison, who died in 1993. Presumably Allison drove car 28?

No Access for groups

Having installed Access 2.0 as a networked version a couple of weeks ago, I was able to run Access from a variety of different workstations, create tables and run Wizards, and foolishly assumed all was well. I discovered I was wrong when trying to teach Access to a class of ten people. I demonstrated the table Wizard and invited the participants to have a go. Only one managed to run the Wizard, while the others received the error message: "Another user or instance of Microsoft Access is currently using the Table Wizard or the Field Builder."

I found this impossible to contradict; after all, I had just told ten people to use it at the same time. But I also suspected that an attempt by multiple users to use the same software simultaneously shouldn't have come as a major surprise to an application installed for use on a network. I assumed that I had fluffed the installation in some way, perhaps failing to set the correct access rights to the *.MDA files which contain the Wizards. While the participants

were enjoying lunch, I went hungry, logged in as Supervisor and gave every user their own copies of the *.MDA files. After altering the [Libraries] section of their MSACC20.INI to point to the new sub-dir, I was able to continue the course.

At the end of the day, I contacted Microsoft. It transpired that I had followed the installation instructions perfectly, but they happened to be wrong. Three of the *.MDA files (WZLIB, WZTABLE and WZBLDR) contain Wizards which modify the library database they reside in and can't run concurrently. The solution Microsoft has found for this problem is the same as the one I used; namely to give each user their own copy of these files and alter their MSACC20.INI to point accordingly. You can even get a document (REF Q121662) from Microsoft which details the changes.

Now, I accept that software is likely to have bugs, but going by the date on the documentation of the bug fix, Microsoft has known about this bug since 13th October 1994 at the latest, and has neither fixed it nor amended the documentation. Instead, all over the world, individual network supervisors are expected to install the software incorrectly (as per instructions) and then suffer an intermittent fault as different users on different parts of the network occasionally clash in their attempts to use the same

Fig 1 Paradox treats nulls as equal during a join, not something I would like to explain to the relatives (or next of kin) of some of these patients

PATNO	BOTTLENO	TYPE	TYPE
1	1.00	A+	2.00 A+
2	2.00	O-	3.00 O-
3	3.00	A-	1.00 A-
4	3.00	A-	8.00 A-
5	5.00		5.00
6	5.00		6.00
7	5.00		7.00
8	8.00		5.00
9	5.00		5.00
10	6.00		7.00
11	7.00	O-	3.00 O-
12	8.00		5.00
13	8.00		6.00
14	8.00		7.00

Wizard simultaneously. I was lucky: the fault was obvious to me because I was teaching a class who were working in sync, but this is the sort of problem that could plague you for months.

How many person-hours have been spent attempting to sort this problem, for which a fix (albeit an untidy one) has been known for nearly a year? How many Microsoft-hours would it have taken to fix the problem, the install routine, or to slip a note into the manuals? Does Microsoft actually care one way or the other?

Microsoft's answer to my complaint is as follows: firstly, very few people actually install the software on a network, so it isn't a major problem. Secondly, two of the .MDA files have been fixed and are up on CompuServe (LIB 15 - Wiznet.EXE) so people can download them.

The first seems an odd response from a company which professes so much interest in networking. The second is really no answer because everyone still has to discover the problem first (which is where the time is wasted), and since WZTABLE.MDA hasn't been fixed, you still have to mess around with sub-directories, .INI files and the like.

Input masks

I used to like input masks, but now I'm not so sure. Having thrashed the phone number version to death two months ago, I thought I was safe, but no. As David Probett has pointed out, the input mask for Access which I suggested (>LLOa\OLL) is unsuitable for postcodes like B1 2BC. Arghhhh! In this electronic age, why don't the committees which design entities like postcodes ever think about databases? David kindly suggested (>LA9a\OLL;0_) which will allow all of the correct forms, but allows mutants like D23 4HN as well.

This isn't David's fault; the problem is the variability of the codes themselves. I still think the only real cure is to use Access Basic.

BOTTLENO	BLOODBAN.TYPE	PATNO	PATIENTS.TYPE
1	A+	2	A+
2	O-	3	O-
3	A-	8	A-
3	A-	1	A-
7	O-	3	O-

Fig 2 Access
treats nulls with the respect they deserve, and doesn't treat them as equal in a join

patients are new, so we haven't yet discovered their blood types. In other words, some of the information is unknown, so quite correctly it is represented in

Does anyone out there have any code they would like to share?

Codding about

Following on from last month's list, here are some more suggested rules for modern, PC-based RDBMSs:

3. *The RDBMS must ensure that any field declared as a primary key, or part of a primary key, is not allowed to contain null values.*

The main point here is that a null value means: "We don't know the value which should be inserted here." A null is not the same as a zero or space character. Since the value in the primary key field is used to provide a unique identifier for the record, we have to be sure what value it has, so null values are highly inappropriate in a primary key field.

4. *Every piece of information in a table must be accessible by using a combination of the table name, field name and primary key value.*

This excellent rule is essentially Codd's Guaranteed Access Rule (see PCW May '95).

5. *Null values must not be treated as equal in joins.*

Some RDBMSs, like Paradox, treat nulls as if they were equal in joins, which, as Mr. Spock would have said, is illogical.

Values which are known can be said to match; so 2 equals 2 and 3 equals 3. However, an unknown value cannot be said with any certainty to match another unknown; so one null value cannot be said to equal another null value.

What happens if they are allowed to "match"? Have a look at Figs 1 and 2, showing the results of a query on two joined tables in Paradox and Access. We are matching blood in a blood bank, to patients. Some of the blood in the bank has yet to be typed, and some of the

the tables with null values. Given this example, do you think nulls should be treated as matching? If so, you get the job of explaining to the relatives of patient 5 why he was given un-typed blood.

6. *Joins on non-identical field types must not be allowed.*

Lotus' (sorry, IBM's) Approach breaks this particular rule and serves as an excellent example of what happens.

I was told by a Lotus person that it is a "feature" which makes life easier for the users. You see, some users are unable to distinguish between text and numeric fields because they can put numbers into both types. Thus a user may have a text field in one table and a numeric field in another, both of which contain numbers. When they try to join these fields, they become disgruntled because the RDBMS refuses to sanction the join. The solution, according to Lotus, is simple: remove the restriction and allow the RDBMS to join them.

However, let's have a look at why the restriction needs to be in place. The two tables in Fig 3 are clearly meant to be joined by their respective ID fields, but one is a text field, the other is numeric. The problems which arise if the RDBMS allows you to join them is clear from Fig 4. Approach is unable to distinguish between the person with the ID of 3 and the person with an ID of 3A. The result is that both of them end up responsible for the orders attributable to person 3. This will delight Sarah Jones and infuriate Jerry Ferish.

7. *When joins are performed on tables containing existing data, the referential integrity of that existing data must be checked and the join must fail if the data violates the proposed join.*

If an RDBMS imports data, we cannot expect it to be responsible for the prior integrity of that data. However, as soon

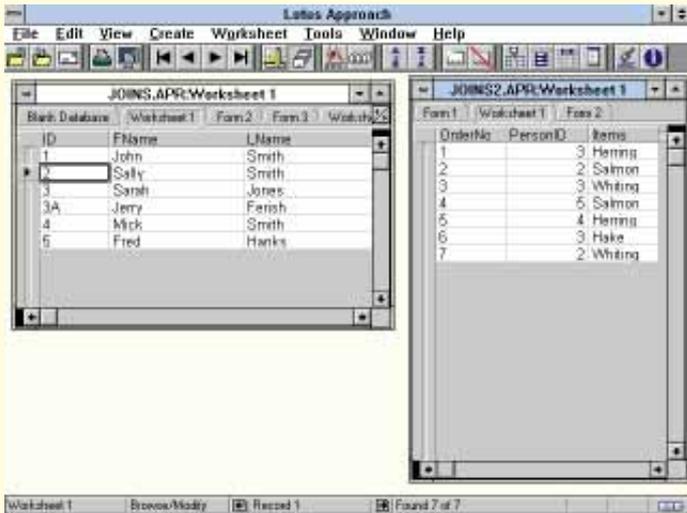


Fig 3 Here are two tables in Approach which have been joined by the Identity Field (ID in JOINS.APR and PersonID in JOINS2.APR). However, ID is a text field while PersonID is a numerical field

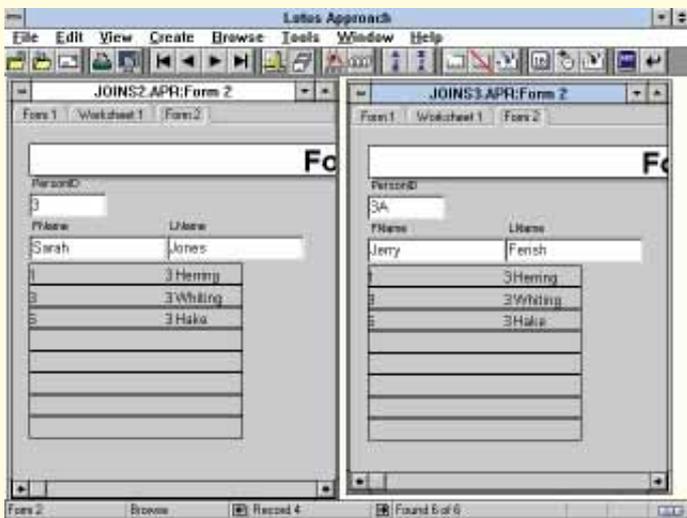


Fig 4 Both Jones and Ferish end up responsible for orders 3 and 6 because Approach seems unable to distinguish between 3 and 3A in a join. I've had to fiddle this screenshot to get both records onto the screen together, but the effect is just as shown

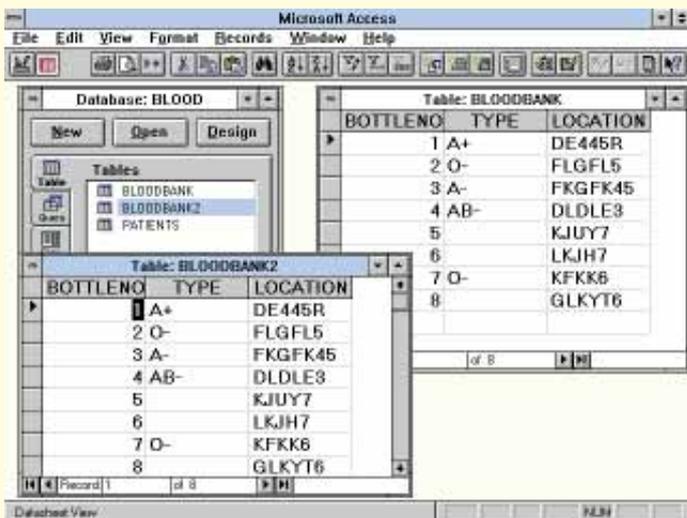


Fig 5 The result of pasting a table, complete with data, in Access. Note that the two tables must have different filenames

as we use that RDBMS to, say, establish referential integrity between a pair of tables, we have every right to expect the RDBMS to check that the existing data conforms. Many RDBMSs make this check, but here again Approach falls short of perfection; it doesn't check.

"Why might this be?" you ask. According to Kevin Harvey, the designer of the

product, very few people import existing tables of data. Odd that, especially as we are also told that Approach used the dBase file format just to be compatible with existing datafiles. However, as long as you are sure you will always create data from within Approach, and never import it, you need not worry about this problem.

Tips and tricks

Forms in Access

In Access, I often construct several forms which are similar, but subtly different. For example, I might have two forms: one for entering data and the other for viewing the same data. Both contain the same fields but they have different labels and colours, and the latter would be read only. Clearly it is a waste of time to build both forms from scratch, so from within the database window Access allows you to select a form and treat it just like any other object. Thus you can cut or copy it to the clipboard and then paste it back.

Access is quite smart about this. If you paste a table, for example, it will ask if you want to paste the structure, the structure and data, or append the data to an existing table. The same trick can be performed with Queries, Forms and Reports. With some imagination, this trick can be used for more than just copying objects. For example, suppose that you have a relatively complex query which works well, but needs improvement. If you take a copy before you start the improvement process, should anything go wrong and you end up destroying the original rather than improving it, you have an instant backup. This is a really useful technique.

Paradox for DOS

"I have a MASTER table of Users like this:

User ID	A1
Address	12 The Cottages
etc...	

User ID is the key field.

Periodically, an ASCII delimited text file is created which includes all the users. I import this into a TEMP table and then I need to add only the new users (i.e. those for whom the User ID is not in the MASTER table) into the MASTER table. What I have been doing is performing a delete query to remove those that are in the first table and then inserting those that remain. Is there a way to perform this sort of query in one step? I have tried various combinations of sets, NOT and NO with no luck."

You just need to perform an ADD of the TEMP table to the MASTER table. All the "key violations" should drop out into a KEYVIOL table which you can then ditch.

The steps are: TOOLS{More}{Add} type in "TEMP" <Enter> type in "MASTER" <Enter> SELECT {New Entries}.

PCW Contacts

Mark Whitehorn welcomes readers' correspondence and ideas for the Databases column. He's on penguin@cix.compulink.co.uk