Guessing game

Mark Whitehorn teaches his database to play charades and look for names that sound alike.

I presume that at some stage in your life you have been foolish enough to play charades. You have stood in front of your peers desperately trying to act out the name of a play, book or film that someone else has supplied.

Databases are not normally expected to play charades. However, given that users aren't always the best at spelling, it would be really useful if they could search a database for words that sound like the word they enter. For example, you need to find the record for that guy err... Whithorne, Whitehourn, Whitworn... whatever.

The answer is to write a function that is capable of 'deciding' whether two words sound similar. Like most problems, the more you look into this, the more complex it becomes. It is reasonably clear that William sounds like Wiliam; but does it sound like Willian or, indeed Billiam? The good news is that someone has already worked out a way of answering this question. The algorithm is widely published and is known as Soundex. The someone in question is rumoured to be a Victorian, which sounds reasonable given their obsession with words, accents and pronunciation. True, they didn't have computers, but then that simply illustrates the difference between algorithms (which are simply ways of solving a problem) and implementation.

In fact, there are multiple variations of this algorithm around. This is perfectly reasonable, since whether two words sound alike will always be a matter of opinion and people will tweak the mechanism to suit their own ears/beliefs. In addition, even I, as a non-linguist, appreciate that the algorithm will have to be altered for different languages and/or regional accents.

Soundex works by turning each of the two words that you want to compare into coded strings. You then compare the strings and, if they are identical, the words sound alike. For example, the word 'Penguin' codes to P525, as does the word 'Pingoin', so they sound alike. So, what is an example of the coding algorithm?

Soundex assumes that the first letter of the word is vital and has to be correct, so that letter forms the first part of the code. Thereafter, letters that sound alike are given identical codes, along these lines:

```
'B','P','F','V':= '1'
'C','S','K','G','J','Q', ↙
'X','Z':='2'
'D','T':= '3'
'L':= '4'
'M','N':= '5'
'R':= '6'
'A','E','I','U','O','Y':= '7'
'H','W':= '8'
```

(*Key: ↙ code string continues*)

Penguin therefore translates as P752775.

However there are three more rules that need to be applied:

- If two or more identical numbers straddle an 8 (an H or W), then all of the straddling letters (except for the first

Lost the default?

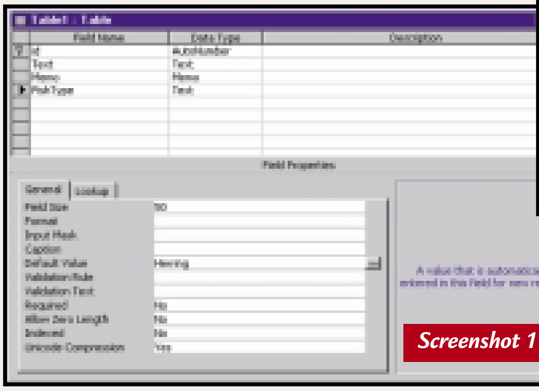


Table1 : Table

Field Name	Data Type	Description
id	AutoNumber	
Text	Text	
Memo	Memo	
FishType	Text	

Field Properties

Field Size	50
Format	
Input Mask	
Caption	
Default Value	Herring
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	No
Unicode Compression	Yes

A value that is automatically entered in this field for new records.

Screenshot 1

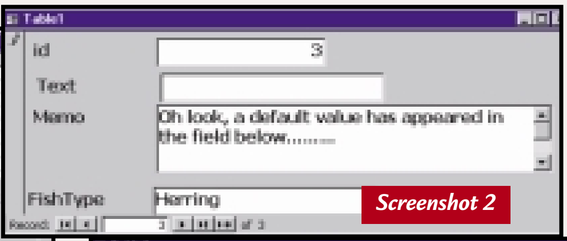


Table1

id: 3

Text:

Memo: Oh look, a default value has appeared in the field below........

FishType: Herring

Screenshot 2

Ctrl & Alt & Space brings it all back

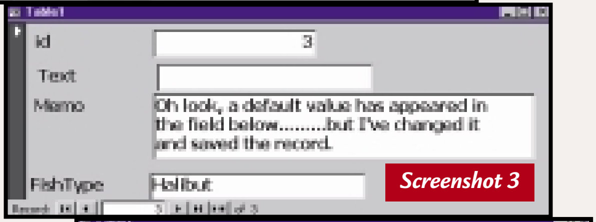


Table1

id: 3

Text:

Memo: Oh look, a default value has appeared in the field below.........but I've changed it and saved the record.

FishType: Halibut

Screenshot 3

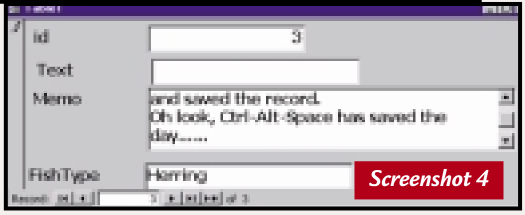


Table1

id: 3

Text:

Memo: and saved the record. Oh look, Ctrl-Alt-Space has saved the day......

FishType: Herring

Screenshot 4

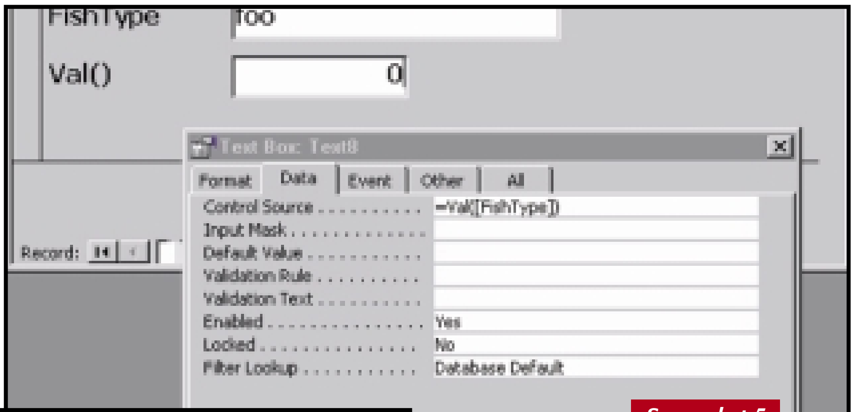
Access allows you to add a default value to a field in a table (screenshot 1), and then that value appears in any forms based on that table (screenshot 2). You are free, of course, to alter that value (screenshot 3). But what if you want to revert to the default at some later stage? Simple answer, just go to the field and press Ctrl & Alt & Space (the obvious key combination) whereupon the default magically re-appears (screenshot 4).

one) is turned into a seven. So P43686634 would become P43687734.

● If two or more identical numbers sit together, then all except the first is turned into a seven. So P43333534 would become P43777534.

● Thereafter, the first three numbers in the string are used, excluding 7 and 8. So P768773482 would become P634.

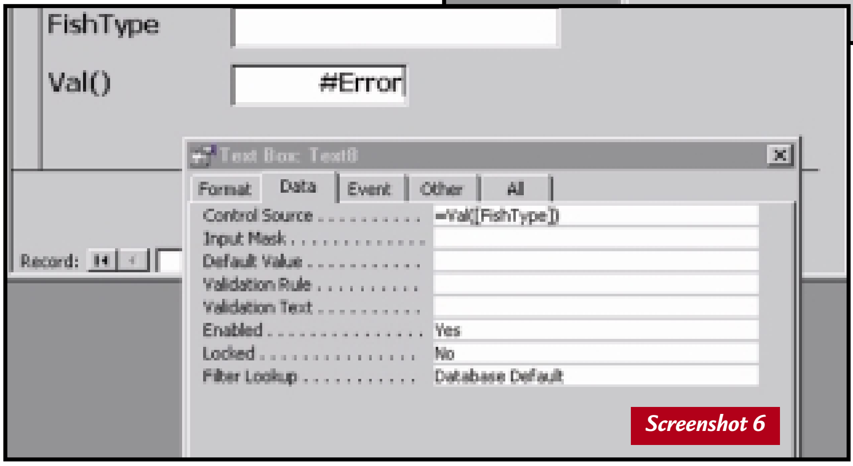
I was going to supply a working example in Access, but I know that, out there, you readers are just champing at the bit to send in your variants. I invite you to do just that and we'll put the best ones on the CD-ROM. Just to help out those people who want to give it a try, I've also include a block of marginally documented code on the CD-ROM (as a text file) that I wrote in the mid-1980s to implement Soundex in Pascal.


FishType: foo
Val(): 0
Text Box: Text8

Control Source: =Val([FishType])
Input Mask:
Default Value:
Validation Rule:
Validation Text:
Enabled: Yes
Locked: No
Filter Lookup: Database Default

Screenshot 5

Val() doesn't like nulls


FishType:
Val(): #Error
Text Box: Text8

Control Source: =Val([FishType])
Input Mask:
Default Value:
Validation Rule:
Validation Text:
Enabled: Yes
Locked: No
Filter Lookup: Database Default

Screenshot 6

What may also help (and is therefore also included in the text file) is the original email which prompted me to look at Soundex again in the first place. This is from Robin Claw (robin@claw2000.freeserve.co.uk) who wants to use Soundex in a hospital system. Why would his request for help be of use to you? Because he includes a partial implementation in QBasic, written originally by Mark Myatt. The algorithm used is a variant of the one I outlined above, but none the worse for that, and well worth examining because QBasic is closer to VB than Pascal.

Some database engines already provide help in matching sounds. For example, SQL Server implements not only a Soundex() function, but also a Difference() function that will tell you how close the sounds of two words are. Difference() essentially counts the number of characters that match in the strings returned by Soundex. So the command:

```
SELECT SOUNDEX('Blotchet-↙
Halls'),SOUNDEX('Greene'), ↙
DIFFERENCE('Blotchet-Halls', ↙
'Greene')
```

returns the answer: B432 G650 0. In other words, there are zero matching characters. Which is fair enough, Blotchet-Hall sounds pretty unlike Greene no matter what your accent.

Oracle provides a Soundex() function based on an algorithm published in *The Art of Computer Programming Volume 3: Sorting and Searching* by Donald Knuth; which is simpler than the one above.

Font of all knowledge

Incidentally, I realise that we *Hands On* people must give the impression that we carry all of this stuff around in our brains, and I'm sure that some do. I don't. Some of it I do know, some I have to ferret out. One of the best resources I have found is Microsoft's TechNet. I was talking to a developer in Seattle who said that a recent survey carried out by the big M suggested that 70 per cent of the answers to questions which come into the helplines are to be found on TechNet. This gem is available on CD-ROM at something like £170 per annum. Eminently worth it if your job is problem solving, but too expensive for the casual user. However, TechNet is also available for free on the Internet. To give it a try, go to www.microsoft.com/technet.

It is a mine of information. For example, I didn't know that: if you place the insertion point in a field with the data type OLE Object, choose Insert Object from the Edit menu, and then click the cancel button in the Insert Object dialog box, the record will still be 'dirty' (a dirty record in Microsoft parlance is one that has been edited!).

If the field is part of an existing record, the previous contents of the field are not deleted or modified, but the record is rewritten when you leave it.

Happily this applies only to Access versions 1.0, 1.1 and 2.0.

What is the VALue of a null?

Val() is a function in Access which returns the numeric value of a string, which can be useful when you need to convert, well, numbers that happen to be in a text field into numerical data (screenshot 5).

The only problem is that Val() returns an error if you feed it a null value. This isn't a problem when the error appears in a form (screenshot 6) but it's a pain if the error appears in a block of code that you have written. The answer is to error trap; check for a null value before using Val().

CONTACTS

Mark Whitehorn welcomes your comments on the Databases column. Contact him via the *PCW* editorial office, or email: **database@pcw.co.uk**