

RE: How do I add a new selection to a drop down list?

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.gettingstarted/2009-02/msg00088.html>

- *From:* Ken Sheridan <KenSheridan@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 5 Feb 2009 09:28:36 -0800
-

As the other respondent has said you'd normally add a new row to the table from which the combo box (which is what this type of control is called). The list can also be a 'value list' as a property of the control itself rather than being derived from a table, but this is only suitable for fixed lists such as days of the week or months of the year, not for variable sets of items to which you might want to add new ones, as its storing data as a property of the control, whereas in a relational database a fundamental principle (the 'information principle') is that data should only be stored as values at column positions in rows in tables, and in no other way

Say for instance you had a combo box with a list of cities which gets its list from a table named Cities then you'd insert a new row into this Cities table to add a new item to the combo box's list. You can automate this by simply typing a new city name into the combo box, but this requires writing some VBA code in the combo box's NotInList event procedure, so may be a little beyond your current level of experience, but don't be put off; its not nearly as scary as you might think. You might not want to try this until you've got your teeth around Access a little more, but this is how its done:

Firstly to add code to an event procedure you'd do as follows:

Select the control in form design view and open its properties sheet if its not already open. Then select the relevant event property in the properties sheet. Click on the 'build' button; that's the one on the right with 3 dots. Select 'Code Builder' in the dialogue, and click OK. The VBA window will open at the event procedure with the first and last lines already in place. Enter the lines of code between these two existing lines.

In this case you'd select the Not In List event property and enter the following code:

```
Dim cmd As ADODB.Command
Dim ctrl As Control
Dim strSQL As String, strMessage As String

Set ctrl = Me.ActiveControl
strMessage = "Add " & NewData & " to list?"
```

RE: How do I add a new selection to a drop down list?

```
strSQL = "INSERT INTO Cities(City) VALUES('"' & _  
NewData & "'")"
```

```
Set cmd = New ADODB.Command  
cmd.ActiveConnection = CurrentProject.Connection  
cmd.CommandType = adCmdText
```

```
If MsgBox(strMessage, vbYesNo + vbQuestion) = vbYes Then  
cmd.CommandText = strSQL  
cmd.Execute  
Response = acDataErrAdded  
Else  
Response = acDataErrContinue  
ctrl.Undo  
End If
```

What this code does is in effect construct and execute what you may be familiar with as an 'append' query, but in this case does it entirely in code rather than by actually creating such a query with the query designer. The INSERT INTO & . line is an SQL statement which is the equivalent of an 'append' query. SQL stands for 'Structured Query Language'. The following lines execute this 'query' if the user has answered yes to the prompt in a message box which is popped up by the MsgBox function. You'll see that this is followed by a line 'Response = acDataErrAdded'. This tells Access to 'requery' the combo box so it shows the new city in its list. If the user answers no then the city is not added and the 'Response = acDataErrContinue' line tells Access to carry on without popping up an error message to tell you the city isn't in the list. The next line 'ctrl.Undo' then undoes the combo box, removing the city name you type in.

Don't worry if that seems confusing to you; it did to us all at one time. As you gain experience you'll in time find that it's actually very straightforward.

As far as learning Access in general is concerned, then the first stage in this process is not really about Access as such, but about the relational database model itself. Access is a relational database system, like many other database products, and it's very important that you have a good understanding of the principles of the relational model as such before beginning to develop database applications with Access. You'll find plenty of good stuff online if you Google around a bit, but to give you a taste, here are my 'stock' replies on (a) the relational model and (b) normalization, which is what the process of ensuring that a database is made up of the right tables, with the right columns in each and the correct relationships between them. The two summaries below contain some duplication, as the two topics are closely linked:

(a) The database relational model was originally proposed by E F Codd in a paper in 1970 in the journal 'Communications of the Association for Computing Machinery'. Since then there has been a vast amount of further theoretical

RE: How do I add a new selection to a drop down list?

RE: How do I add a new selection to a drop down list?

work, and the relational model has shown itself to be a robust one. Without going too deeply into the theoretical basis, which can be quite abstract, a relational database in essence models a part of the real world in terms of its entity types and the relationship types between them. Note the inclusion of the word 'type' in both cases here. While its almost always used in the former case, its often omitted in the latter case. This is a little bit sloppy but not too important. When one talks about a 'relationship' it really refers to a relationship value. As an example 'marriage' is a relationship type, but my being married to my wife Fiona is a relationship value, represented by our names on the marriage certificate, which is the physical equivalent of a row in a Marriages table with columns Husband and Wife, each referencing the primary key of a table People. This is a many-to-many relationship type (I've been married twice so would be in two rows, my first wife would also be in two rows as she remarried too). It is resolved into two one-to-many relationship types, People to Marriages in each case, in one case via the Husband column in the other via the Wife column.

In a relational database tables model Entity Types. In the above example People is an entity type, modelled by the People table. Marriage is also an entity type, modelled by the Marriages table. As we've seen its also a relationship type. In fact a relationship type is just a special kind of entity type.

Each column in a table represents an attribute type of each entity type, so attribute types of People might be FirstName, LastName, DateOfBirth etc. This table would also have a PersonID numeric column (usually an autonumber) as its primary key as names are not unique. Each row in a table represents one instance of the entity type, and the attributes of each instance are represented by values at column positions in the row. This is the only way that data can be legitimately stored in a relational database.

Its important that there is no redundancy in the information content of the database. This is achieved by the process of 'normalization'. Normalization is based on a set of 'normal form's ranging from First Normal Form (1NF) to Fifth Normal Form (5NF). There is also one called Boyce/Codd Normal Form (BCNF) which was inserted when it was found that the original Third Normal Form was deficient; it didn't cater satisfactorily for tables with two or more candidate keys where the keys were composite and overlapped, i.e. had a column in common. I won't go into the details of normalization here; you'll find it written up in plenty of places.

To see an example of redundancy and therefore a table which is not properly normalized take a look at the Customers table in the sample Northwind database which comes with Access. You'll see that it includes City, Region and Country columns. If you look at its data you'll se that we are redundantly told that São Paulo is in province SP which is in country Brazil 4 times. This is not just inefficient, it is dangerous as it leaves the table open to inconsistent data being entered. There is nothing to stop somebody putting São Paulo in the UK, USA or in each in separate rows in the table for instance. To normalize the table it should be decomposed into Customers, Cities, Regions and Countries tables, each of the first three with

RE: How do I add a new selection to a drop down list?

a foreign key referencing the primary key of the next table up in the hierarchy.

(b) Normalization is the process of eliminating redundancy from a database, and involves decomposing a table into several related tables. In a relational database each table represents an entity type, e.g. Contacts, Companies, Cities, States etc. and each column in a table represents an attribute type of the entity type, e.g. ContactID, FirstName and LastName might be attribute types of Contacts and hence columns of a Contacts table. Its important that each attribute type must be specific to the entity type, so that each 'fact' is stored once only. In the jargon its said that the attribute type is 'functionally dependent' solely on the whole of the primary key of a table.

To relate tables a 'referencing' table will have a foreign key column which makes the link to the 'referenced' table, e.g. a Contacts table might have a CompanyID column as a foreign key, while a Companies table has a CompanyID column as its primary key. Consequently no data other than the CompanyID needs to be stored in a row in the Contacts table for all the company information for that contact to be known; its available via the relationship and can be returned in a query joining the two tables on the CompanyID columns.

Similarly the Companies table might have a CityID column and the Cities table a StateID column. If its an international database the States (or more generically Regions) table would have a CountryID referencing the primary key of a Countries table. So via the relationships, simply by entering (in reality this would be selected from a list of Companies in a combo box, not typed in) a CompanyID in the Contacts table the location of the contact's company is also known. Redundancy, and therefore repetitive data entry is avoided.

To see how a database is made up of related tables take a look at the sample Northwind database. Its not completely normalized in fact (deliberately so for the sake of simplicity) but illustrates the main principles of how tables representing entity types relate to each other. An example of its lack of proper normalization can be found in its Customers table. You'll see that this has City, Region and Country columns so we are told numerous times that São Paulo is in SP region (as is Resende) and that SP region is in Brazil. Not only does this require repetitive data entry, but more importantly it opens up the risk of inconsistent data, e.g. it would be perfectly possible to put São Paulo in California in one row and California in Ireland! Proper normalization as I described above would prevent this as the fact that São Paulo is in SP region would be stored only once in the database as would the fact that SP region is in Brazil and California is in the USA.

An example of what at first sight might seem to be redundancy, but in fact is not, can also be found in Northwind. The Products table and the OrderDetails table both have UnitPrice columns. It might be thought that the unit price of a product could always be looked up from the Products table, so

RE: How do I add a new selection to a drop down list?

RE: How do I add a new selection to a drop down list?

its unnecessary in Order Details. However, the unit price of a product will change over time, but each order needs to retain the price in force at the time the order was created. Consequently a UnitPrice column is needed in both tables; that in products holds the current price and is used to get the value for that in Order Details (code in the ProductID control's AfterUpdate event procedure in the Order Details Subform does this), which then remains static when the current price (in products) changes. In each case UnitPrice is functionally dependent on the key of the table, so there is no redundancy.

Ken Sheridan
Stafford, England

"Ama Anika" wrote:

I am new to Access, therefore do not know the terms. I can work with database that already exists. I now face the problem of addition new selections to a drop down list. How do I proceed?

Also, I would need assistance to learn Access to become a real professional user

RE: How do I add a new selection to a drop down list?