Learn Access Advanced 2019

Simon Sez IT®

Your source for software training.

Table of Contents

Section 1 – Introduction

Section 2 – Primary Keys and Indexes

Section 3 – Using Datasheet View

Section 4 – Linking to Other Access Databases

Section 5 – Using ODBC and SQL Server

Section 6 – Splitting an Access Database

Section 7 – Macros

Section 8 – The Ribbon

Section 9 – Visual Basic of Applications

Section 10 – Access Events Model

Section 11 – Forms

Section 1 – Introduction

Video: Course Overview and Content

Deb: Hello everyone and welcome to this course on Access 2019 Advanced. My name's Deb and I'm your instructor on this course.

Now in this introductory section we're going to be taking a look at assumptions, we're going to talk about programming, I'm going to give you some information about the sample databases I'll be using on the course, and I will explain the exercise and the course files. Finally we'll talk about the changing features of Access 2019.

Now I'm sure you're eager to get started with your course but it is important that I explain these things to you starting with the assumptions I'm going to make.

So the first assumption is that you have Access 2019 installed on your PC. We're not going to cover anything related to the installation of Access.

The next assumption I'm going to make is that you have a good basic working knowledge of Access 2019. And what do I mean by that? Well, if you look at the course content for our Microsoft Access 2019 beginner's course I'm going to assume you're familiar with all of these topics. And when I say familiar I don't just mean knowing the terminology. I mean that you could explain to me with a reasonable level of confidence what each of these tools and utilities are. So it's definitely worth having a scroll through the contents and having a look at the different topics covered in the beginner's course. If you do find that you have some gaps in knowledge then I would say it's definitely worth going back and reviewing that course before coming to the advanced course.

Now if your knowledge of Access is in an earlier version, if it's a pre-2007 version then 2019 is really quite different. The new version has so many changes from the basics, such as the ribbon structure, to things like new reporting facilities. So again if that applies to you I think you should probably work your way through the beginner's course first.

From 2007 through to 2010, 2013 and 2016 a good working knowledge of these versions will give you a good basis. So for example, if you're fairly confident with Access 2013 or 2016 you should be okay with moving straight into Access 2019.

The next thing I'd like to talk about is programming. Now some users work with Access without ever writing any program code themselves. And you can develop some good Access applications without getting to grips with programming. However those people are missing out quite a bit because you can do a lot more with Access if you learn about programming. So we're going to spend quite a lot of time looking at macro code and VBA code. So if you do have any reservations it is definitely worth giving it a try as it will help you get more out of your Access database. And the way that I explain it, I try to do it so that it makes sense to someone who doesn’t really come from a programming background.

Now let’s look at sample databases. In the beginner’s course we developed a travel company database called Esprit de Tour and we also as part of the exercises developed a movies database. Now if you haven’t worked through the beginner’s course you might want to familiarize yourself with those databases.

In the advanced course we won’t only be focusing on one sample database because there is such a broad range of features, but we will be looking at one or two sample Microsoft databases as well. So it is advantageous of you to become familiar with the two databases from the beginner’s course.

And apart from the sample databases it is a really good idea if you can work on a database of your own alongside the advanced course, which will more accurately reflect your own interests and the type of work that you do.

Now as part of this course I’m going to be setting you exercises. And the exercise files are all in the exercise folder. I will also be using some sample files and you will find those in the course files folder. So make sure you download those and keep copies somewhere safe.

If you’re coming to Access 2019 from 2016 or 2013 then you can get a good overview of all of the new features from the Help page, What’s new in Access 2019. Now a lot of these are fairly minor but it is important to make yourselves familiar with them. So I would definitely recommend having a read through the Help page before you proceed.

So that’s enough from me by way of an introduction to this course. It’s now time to do some work and we’ll do that in the next section. So I will see you over there.

Section 2 – Primary Keys and Indexes

Video: Primary Keys and Indexing Rules and Guidelines

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this section we’re going to look at primary keys and indexes. And as part of this I’m going to introduce or reintroduce you to one of the databases we created in the beginner’s course, the Night Movies database. Now if you haven’t covered the beginner’s course then there is a copy of this database in the course files folder. And we’re going to be looking at primary keys, indexes, and rules and guidelines.

So the first thing to do is to open up the Night Movies database and we’re going to start by looking at the Navigation pane. Now the shortcut for bringing that up is F11. And we’re going to go into the Design of the actor table. Now because the Night Movies database references the actor table we need to close it first before opening the actor table. So now we can right click and open it up in Design view.

So in the actor table we have eight fields and one of the fields, the ID field is marked as the primary key. And you can see that little key symbol in the first column there. Now if you remember the primary key uniquely identifies each record in a table. So the ID in this case is the primary key field.

Now you don’t have to have a primary key on a table but almost all tables do have one. So in the case of this table where the ID is sort of a dummy number, it’s the number assigned and it’s auto numbered so it doesn’t really relate to the name of any of the property of the actor. It just identifies that actor.

Now as we’ve seen the ID field is the primary key but other fields are indexed as well. So if I click in some of these other field names, so for example, if I click in Actor Name, if you case your eyes down to the field properties and look at Indexed you can see it says Indexed No. However if I was to click in the Family Name field and look at the field properties it says Indexed Yes, Duplicates OK. If I click in Gender I can see that that is also indexed. And if I click in Birth Country I can see that that is also indexed as well. So let’s now look at the purpose of indexes.

So I’ve opened the actor table and let’s look at some of the data in it. Now the ID values have no relation to the actor themselves. They are auto numbered and assigned when actors are added to the table. And actors in general are always added to the end of the table and they essentially get the next auto number in the sequence. And it’s worth remembering that auto numbers are not reused. So for example, if I deleted someone their ID would not become available to someone else.

So let’s suppose that there are no other indexes available on this table and let’s suppose then that I am doing a search on actor information and I’m looking for the information on Henry Fonda. Now without any other indexes available the only way Access is going to be able to find that information on Henry Fonda is to go through each actor in the table. So it will go to the first record. It will say, Is this Henry Fonda? No. It will go to the second record. Is this Henry Fonda? No. So on and so forth until it gets all the way down to record number 17 just here where it says, Is this Henry Fonda and then we say Yes. So this process reflects the fact that primarily the reason for indexing records is to make it quicker to find them. So one of the options is to make the actor’s whole name an indexed field. So if someone searches for Henry Fonda what Access will have done is to maintain a lookup index. So it won’t have to go through each of these records, each of these numbers. It will go to its actor name index. It will go to the entry Henry Fonda and it will find that I want the record in the actor table with the ID 17. So essentially I can go directly to record 17 without searching through all of the other records.

So having an index on the actor name would be a good thing, as it can help users find what they’re looking for much more quickly. However, not all queries are as simple as that. For instance, a different query might be find me all of the actors born in the U.S.A. before 1930 whose first name is Henry. So in that case you might find yourself stepping through all of the actors and instead of about 30 actors which is essentially what we have you might have a table that has 30,000 records and stepping through each of those records can be a very lengthy process. However, if we had an index on birth country that would certainly help us out a lot as we could just go through records whose birth country was the U.S.A. And if we had an index on given name we could look up the given name index. So essentially we could go to all of the Henry’s and that would give us a subset of all the actor records to begin with.

However there is a downside to this approach and that is that with each additional index there is a maintenance load on the system. So whatever records are changed whenever an actor is added to the database, modified, deleted, etcetera then the indexes also need to be maintained as well. And we find that the cost of improved querying is a deterioration in the performance of updates.

So with all that in mind how do you achieve a balance? Well that will totally depend on the nature of the database and also the use of the database. So for example, if your database is relatively stable but there are many, many people querying the database then having good indexing is going to be advantageous. However if you have a database where there are many, many updates but not too many queries then essentially the opposite is true. So the nature and use of a database will be one of the primary drivers in decisions about indexing.

Now the other thing to point out regarding the cost of maintaining indexes, indexes also take up a lot of space. So the more indexes you have that’s when space can become a bit of an issue if you’re dealing with indexes with very large volumes.

Now there are some rules and guidelines that I'd like to run through next. First of all the primary key is always indexed. And you can have up to 32 indexes. And there are certain types of fields that you can't index. So for example, you can't index long text fields and you can't index OLE type fields either. And although this isn't a hard and fast rule but more of a general guideline where you have a field that has a very limited number of possible values. So something like Gender, for example, where we only have the male or female value. Generally speaking indexing something like gender will not really help when it comes to querying performance. Having said that, in more modern databases where more than two genders are recognized I think there's something like 20 recognized gender classifications now, you may well need to introduce a gender index.

Now when you're indexing something like birth country where you might have 50, 100, 200 entries then it's also good to consider indexing in that scenario.

Now a couple of other things to point out, clearly with something like country duplicates will occur. So we could certainly have multiple actors for the same country. So when you index the birth country field you would specify that duplicates are okay. They are allowed. The same would be true for given names.

Now when it comes to actor name it's a bit of a different story. In theory two actors might have the same name, although the way that the rules in the acting profession are set up currently the convention is that two people never have the exact same name. however if you don't allow duplicates and you do find two actors with the same name you'll need to find some way of dealing with that.

Now one final point to mention is that decisions about indexes are not final. So if you decide you need to add or remove an index it is something that you can change later on. So if you find that some aspect of the performance of your database is causing you a problem you may well need to change the indexing arrangements later on.

Now in the next section we're going to turn our attention to some more detailed work on indexes, including the use of composite indexes, auto indexing and a brief discussion on the impact of compact and repair on indexes.

That's it for this module. I will see you in the next one.

Video: More About Indexes

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this section we're going to look more deeply into the use of indexes. And first of all we're going to look at composite indexes, sometimes referred to as multifield indexes. We're going to look at auto indexes. And finally I'm going to describe the effect of compact and repair on indexes in Access 2019.

In this version of the actor table in the Night Movies database the actor name is an indexed field. And if you wanted to search on the full actor name, I might for instance specify the actor name of Henry Fonda. Now there is an alternative to this approach. If you consider that the actor name is essentially made up of two other fields. It's made up of the given name and also a family name. And you can treat the given name as an indexed field and the family name as an indexed field. And the two together as what we would call a composite index field, essentially made up of two fields. Now if I wanted to do a search on Henry Fonda instead of searching the actor name field what Access will be doing is looking up an index that has two parts. So the first part says Henry and the second part says Fonda.

Now in order to set up a composite index you need to go up to the Design tab and you'll see you have a button there called Indexes. And if we hover over we get our screen tip that says It displays the list of fields on which the table is indexed. So let's click on Indexes.

Now I'm actually going to remove the gender field for the reasons I explained previously. So I'm going to highlight it and just press the Delete key on my keyboard and I'm going to create a composite index. So I'm going to call it FirstLast and I'm going to start with the given names field. And now in the next row of this dialogue I'm going to leave the index name blank and I'm going to put family name again in ascending. And the fact that the index name is blank for this row means that Access will treat this as a composite index.

Now one important point here is that you can include several fields in a composite index, up to ten. And the first field, so in this case Given Names, doesn't then need its own separate simple index or one that only has a single field. If you were to just search on Given Name Access would use this composite index but only the first part of it. Whereas for Family Name, the second field, if you want to be able to search on family name you would leave the simple index for family name that's there already. So essentially family name is there twice as a simple index and also as a composite. So I'm now going to close this dialog box.

Now I've opened the dialog again as I want to talk about some other important points related to indexes. So let's return to our primary keys. Now as we've mentioned before primary keys must be unique. Now a primary key can actually be a composite index. And although that is the case, although the key overall must be unique the individual component field values might not be. However no field used in the primary key can have a null value.

Now for each index, so let's take this one here, Family Name, and if you look down at the properties in the bottom corner of this dialog box you can see it says Primary No, Unique No, Ignore nulls No. But what exactly does Ignore Nulls mean? Well when Access is building its index, in this case the family name index, what happens if family name is null? If Ignore nulls is set to No then there is still an entry placed in the index for family name for this record, but the index value there is null. Of course you might expect in a large table where ignore nulls is set to no you may have a number of records where this particular index has a null value. But if you say Ignore nulls Yes then any records for which this particular index is null will not have an entry in that table. So generally speaking Ignore nulls is set to no. But if you want to include null values from an index you would set that to Yes.

Now it's worth noting for a composite index the composite index value will only be null if every individual field value in that composite index is null as well. So if any of the individual field values are not null then the composite index value will not be null as well.

Now the next topic I want to look at is Auto Indexing. So let's suppose I want to add a new field to this table and we'll just add a field called Text. And for the data type we'll leave it on short text. Now having added that field if you look down at the properties it says Indexed No. Now if instead of adding that field I'd added a field called Numb_Test and we'll stick with that short text data type, if you look down in the properties it says Indexed Yes, Duplicates OK. And also if I'd added a field called Test_ID and again we'll keep that on short text. Again you can see in the properties it says Indexed Yes, Duplicates OK. So why is that? Well to get an explanation we're going to jump into our Access Options. So let's go up to File, we'll go all the way down to Options and we're going to click in our Object Designers page. And you can see here in this top section in Table Design view the fourth option down Auto Index on import/create. And there are four items by default in this field. So we have ID, key, code, and numb. So if you create a table or import a table where the name of a field begins or ends with any of those four then that field will automatically be indexed.

Now you can change this. So if I wanted to not index anything that begins with Numb I could delete that out and all I would need to do would be to replace it with something else. So I could say semicolon and maybe something like Index Now. So if I was to add a field which started or ended with Index Now then that field will automatically be indexed. Now I don't want to do that so I'm going to cancel out of here to cancel that change that I've made. And I'm also going to delete these three fields that I just added.

Now the last thing I'd like to cover here briefly is compact and repair. And as part of your normal maintenance function on your database you should be running this on a regular basis. So this is important when you have a lot of indexes as it rebuilds all of the indexes. So if your database has been running for a long time it will tidy up all of the tables and rebuild the indexes. And that will normally mean that from that point onwards your database will run with maximum efficiency, at least for a period of time. So I strongly recommend running compact and repair on your database. And you can find that option if you go into the backstage area, clicking on File, you have that option there in the Info tab, Compact & Repair database.

So that is it for this module. I will see you in the next one.

Section 3 – Using Datasheet View

Video: Setting up Northwind Database

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this and the next couple of sections we're going to be taking a look at Datasheet view. Now Datasheet view is something that you should be reasonably familiar with but in this section we're going to concentrate on some of the more advanced features.

Now to do this we're going to use a different sample database. We're going to use the standard sample database from Microsoft, the Northwind database. So in this module I'm going to show you how to set it up. Now if you have already got this database set up, as I said it is a fairly common one to use from Microsoft, then you can skip this module.

Now in older versions of Access in order to set up the Northwind database you had to download it, which wasn't too much of a problem but it's now been made a lot simpler. All you need to do is jump into New and in the Search bar just here if you just type in Northwind and press Enter it will pull back that Northwind database. So really very simple to find it. I'm going to click on the Northwind database. Again you can see here it's provided by Microsoft and this is a sample database template which demonstrates how Access can manage small business, customer's orders, inventory, purchasing supplies, so on and so forth.

Now you can also see here the download size, which is just about 2 mg, which really isn't a huge size. But even so when you click Create it may take a few seconds just to download. And there we go. There is our sample template.

Now as always you'll probably get this standard security warning. So I know that this template has come from Microsoft which is a trusted source so I'm just going to say Enable Content. Now remember that you could move the database into a trusted location so that that doesn't pop up but for now I'm fairly happy just clicking Enable Content.

And then what you'll see is that you're given the option to log in to the Northwind Database. And you'll see it comes up with a default user of Andrew Cencini. And things like this Login dialog we're going to cover a little bit later on in the course. So I'm just going to click Login and there we have the home page for the Northwind database.

Now the last thing I'm going to do is I'm going to save this database into the course files folder. So that might be of use for any of you that can't download the template at this moment in time. So you can use mine or you can do it this way and save your own copy.

So I'm going to go to File and I'm going to go down to Save As. I'm going to save my database as an Access database and click the Save As button. I'm going to browse to my location which is the course files folder and I'm going to give it a file name of. So I've given this a file name of

SimonSezIT Northwind Traders A01.accdb and I'm going to click on Save. And because I've saved it as a new file name I need to enable content again and I also need to log in again. So let me just do that. And there we go. We're now ready to go.

So in the next section I'm going to explain more fully the use of this database and then we'll get into using Datasheet view. So I will see you then.

Video: Working in Datasheet View

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we’re going to be taking a further look at working in Datasheet view and we’re going to build on some of the basics that we learnt in the previous course and look at some more advanced topics. So for this you’re going to need to have the Northwind database open. And in this module we’re going to first start to look at a necessary update query, we’re then going to look at navigation and keyboard shortcuts in Datasheet view, we’re going to look at finding records, how you can add and change records, and finally how you can delete multiple records.

Now one thing I’d like you to know is the heading there on the title bar. So it says Northwind Traders A01. So hopefully you can remember how to change that up there and that will allow you to keep track of the versions.

Now some of the tables in this database, including the Orders table, have dates in them. Now if this is the first time that you’ve downloaded the Northwind database from Microsoft what you’ll probably notice is that your dates are going to be different to the ones that I have here. So basically what I’ve done is I’ve updated mine to reflect a more accurate date. So for example, I’m recording this course in November 2019 and so all of my dates, including all of the dates in the Orders table, I’ve updated to reflect my current year. And you’ll see if I scroll across we have more dates in the Paid Date column as well and you can see that I’ve updated those to 2019.

Now depending on when you’re coming to this course, so it might be this year, it might be next year or maybe even a few years in advance, this might be something that you want to do before you begin. You may want to update all of the dates to accurately reflect the year that you’re coming to this course. Now this isn’t something that you have to do. It will just make it a little bit more relevant to you.

And to help you out with that I’ve actually written a small query which will go through and update these dates for you. So that really gets around the need to go in and manually change all of these records, which can be a little bit of a pain. If you would prefer to do it manually then you can do it by selecting the column and going to Replace and replacing whatever date you have in here or whatever year you have in here with your current year. That’s a way around doing that. However alternatively in the course files folder you will find a file, it’s an SQL query, and you’ll see it has a .sql file extension. It is essentially just a text file. And it’s the one I used to take the dates in the Orders and change them into the dates you can see there. So please take a moment to review those dates before you get underway with following through in this part of the course.

Now it’s also important to recognize Datasheet View is not just a tool for somebody with a very simple database. It’s very often the case that users will use Datasheet view to update data. And it can be expensive to create forms for all kinds of maintenance on a database so sometimes users do prefer this approach to maintaining their data.

And it's important to understand some fundamental features of Datasheet view so we're going to start out by looking at keyboard shortcuts.

So I want to start by directing your attention as always to the Help file. And you can see there I've just searched for Keyboard Shortcuts. And as we scroll down you can see one of the topics is Work in a Datasheet View. And this will give you all of those useful keyboard shortcuts for working in Datasheet view, moving around Datasheet view. So you can see things in this table like move to the next field you can press the Tab key or the right arrow key, move to the last field in the current record press the End key, move to the previous field Shift-Tab or left arrow. And there's a number of these in here and these can really improve your efficiency when you're moving around or working in Datasheet view. So I would definitely recommend even if you're not somebody who particularly uses a lot of keyboard shortcuts I would definitely make yourself familiar with some of these because it really is going to improve the way that you work within Access.

Now if we scroll down a bit further you'll see a section here which says Work with Subdatasheets. Now you may not have come across subdatasheets before and we are going to be looking a lot at subdatasheets in the next module. But it's also worth noting that there are keyboard shortcuts for working within subdatasheets as well.

Next let's look at some features of Find in Datasheet view.

Now one thing you may not have noticed is that you can determine if the Find begins with just looking in the current field or in all fields by your initial selection. So for example, if I put my cursor in the Order field and then bring up Find by pressing Control-F by default I search in the current field. Generally speaking Access will remember the last search term that you used. But if I just have that single field selected it will default to looking in the current field. So I could now enter a search term of say Nancy and click Find Next and it's not going to find it because it's just searching in that first column essentially.

Now what if I cancel out of here, if I was to highlight the entire row and press Control-F to bring up my Find you can see it now defaults to Current Document. So if I now type in Nancy and click Find Next it's found her.

Now another page in Help that can be very useful when using Find is Wild Card Characters. So you might be familiar with some of them, so things like asterisks, question marks, square brackets. And I am going to demonstrate that last one in a moment and I'll leave you to work through the others. So here you have a list of all of the different wild card characters that you can possibly use and I'm going to demonstrate the square brackets now.

So let’s close out of Help and in my Order table I’m going to make sure that I have a whole record selected and we’re going to do Control-F to bring up Find. And this time I’m going to search for Ann, square bracket, A and E, close my square bracket (Ann[ae]). So essentially what I’m doing there is I’m asking Access to find anyone called Anna or anyone called Anne and I’m going to say Find Next. And there we go. It’s found Anna Bedecs down here. If I do Find Next again it’s found Anne Hellung-Larson. So it’s matching both of the characters that appear within those square brackets.

Now when working in Datasheet view you’ll need to be able to add, change and delete records. Now when you want to add a new record sometimes I see people scrolling down to the bottom of the datasheet but there are various ways of doing you without having to do new. So if you jump up to your ribbons, your Home ribbon in the Records group we have a New button just there. So if click New it takes you down to the first column in a new record.

Now in this particular table the Order ID is an automated field. So I can’t actually type anything into this field here. So if I Tab across, so in this field I’m going to click on the dropdown and I’m going to select Andrew Cencini and you’ll see that as soon as it gets to the point where it’s allocated that Order ID auto number the record I’m working on is now a record that’s being edited. And you can see there is a new line at the bottom for the next new row. So essentially when you’ve added enough information for Access to recognize that you’re currently working on a new record it will give you that new record again at the bottom of your table.

Now when I’ve finished entering my record I’m going to need to save. And again up in that Records group you can see that you have a Save button. Alternatively you can use the keyboard shortcut Shift and Enter. Now if when you save Access recognizes that data is missing you’ll see a message pop up accordingly, which means it just requires a little bit more information added into the record before you can save. So let’s give save a go and see what happens. Yep, I don’t get an error message so all looks good with that record.

Now I could move on to a new row and start adding another one but the final thing I want to look at is deletion and I’m going to delete this record that we’ve just added in.

So if I wanted to delete it’s very simple. I can just select the record, I can press the Delete key on my keyboard, I could go up to the Records group and press Delete up there, or alternatively I could right click and in my contextual menu you can see I have a Delete Record option. Now you’ll always get this warning just letting you know you’re about to delete one record. And one important thing to note here is that you’re not going to be able to undo this. So if you delete something and then think, “Oh, I really wish I hadn’t have done that” Undo will have no effect. So deleting is essentially permanent in this case.

Another thing you can do is delete multiple records. So if I select this record just here, hold down my Shift key and select the record that’s two below it’s going to highlight all of those records.

And again I could go in and I could delete this record in any of the ways that I’ve just shown you. Now I don’t actually want to do that so I’m just going to click away from there.

So I’ve just given you a run through of some of the basics for working with Datasheet view and maybe extended your knowledge in some of those areas. In the next module we’re going to look at how to control the display in Datasheet view and also some of the sorting and filtering options, so I will see you over there.

Video: Customizing Datasheet View and Advanced Sorting and Filtering

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to continue looking at Datasheet view and we're going to follow on from the idea from the previous section where there are many situations where you might use Datasheet view for data maintenance purposes. And even things like querying a database.

Now there are many ways in which Access can help you and in this module I want to look at different ways you can arrange Datasheet view to make it more convenient to use. So we're going to start out by looking at how you can format Datasheet view, we're going to talk about reorganizing columns, we're going to start exploring sorting options and also filtering options, and finally we will look at how you can add a totals row to do calculations.

Now for this section I'm going to use the Orders table in the Northwind database and we're going to first look at formatting.

Now when you look at a table in Datasheet view the formatting is primarily dictated by a set of options you can change within the Access Options. So we're going to jump into the backstage area and go to Options. So we're going to select File and straight down to Options. And we're going to stay on this Datasheet page. And you can see at the top here we have gridlines and cell effects. So do I want horizontal and vertical gridlines in my table? Now I don't really want that so I'm going to keep those unchecked. I can have a default cell effect, which is currently set to Flat. So I'm going to change that Raised just so you can see what this looks like. And I'm fairly happy with my font size and font weight. Now I'm going to click on OK and what you'll see is that not much change is noticeable and that's because in order to see any changes you've made to the formatting through Options you need to close the table and reopen it again. And there you can see now I have that kind of raised look to the borders of my cells or of my records I should say.

Now it's also worth noting that if I was to double click to open another table, so Order Summary, you will see that those same default settings now apply. I'm going to close Order Summary down.

Now I can also make local changes to the formatting that only apply to the currently selected Datasheet view. So if I go to the Home ribbon and I go over to my Text Formatting group and maybe I change the font from Calibri to let's say this one just here, Berlin Sans FB. And now if I open Order Summary again you'll see that those changes haven't actually taken effect. So you can use that Home ribbon to essentially make local changes which only apply to the current datasheet. And I'm going to close down Order Summary again.

Now when you make a change to something like font size, etcetera it's very often you will need to be able to change column widths and row heights in order to be able to see all the data. If you select a font that's slightly bigger than the one you were using you might not have enough room in your columns to accommodate the new size of that font.

Now it might be that be that you need to adjust the width on one or more of your columns and that's a very simple thing to do. So for example, if I take the employee column just here, if I right click you'll see that we have an option in our contextual menu called Field Width. And I could come in here and manually type in an exact width or alternatively I could select this Best Fit option. And if I click that it will essentially widen that column to the length of the longest item in that column. So it's going to accommodate everything in that column very nicely.

Now when it comes to row height each row must be the same height. So if I was to go to a row, so let's say this one for example, and if I right click my mouse and go to Row Height and I'm going to change this to lets say 30, I'm going to click on OK, and you can see every row has been changed to the same row height.

Now one thing I'm going to do here is I'm going to change this font back to Calibri. So the point I'm really trying to get across here is that some formatting you do applies to the whole sheet and some to individual columns. So for instance let's suppose I took the Ship City column and changed it to bold. So I'm going to go up to Text Formatting and click on Bold. You'll notice that all of the text in the datasheet changes to bold and not just in the column. I'm going to un-bold that. Now with the same column selected I'm going to center align the contents. And you'll see that this only applies to the selected column. So some changes apply to the whole sheet and some to the selected column.

Now when you're using Datasheet view the order of the columns or fields in the table is the same as the order in the table design, and that might be something that you want to change. So for example it you're doing a lot of sorting and filtering you might want to move the columns that you're sorting and filtering next to each other, even if it's just temporarily.

And it's very simple to change the order of the columns. So for example, I want to move this Ship City column over to the right of the Customer column. So all I need to do is make sure I have it selected, click on it, and as I drag you can see I get that big black line. So wherever that black line is when I let go of my mouse it's going to move that column to that location. And I'm actually going to move that column back to where it was.

I could also if I wanted to move multiple columns I could highlight those columns, I can click and I can drag and it will move all three of those columns. So again I'm just going to move those back.

Something else you can do to make this a little bit easier is that you can hide or unhide columns. So for example, if I never needed to change any data that appears in this Ship Via column I could right click it and you'll see that I have a Hide Fields option. And that kind of just hides it away. So that field hasn't been deleted, it's still there, it's just not currently in my view.

If you want to unhide any columns that you've currently got hidden if you right click your mouse and go to Unhide Fields anything that doesn't have a tick next to it is currently hidden. So I can just click Ship Via and Close and that column is now back.

Now something else you may not know with Access is that it's fairly similar to Excel with regards to freezing fields. So if you are an Excel user you may be familiar with the functionality of Freeze Panes. Well Access has something very similar. So say for example I wanted to always keep Order ID and the Employee column visible even as I was scrolling across all of this data. Well I can just freeze the two columns. So I can select them, right click and you'll see we have a Freeze Fields option. So now when I scroll across you can see that Order ID and Employee stay where they are and I can compare them against every column in my datasheet. If you want to unfreeze these panes I'm sure you can guess where we're going with this. Right click your mouse and you have Unfreeze All Fields.

So next we're going to look at sorting. Now sorting on individual fields is simple enough. So if I wanted to sort on customer I can select that column, right click and I have a Sort A to Z or a Sort Z to A. Now remember what you see in this contextual menu is somewhat determined by the type of data that you have in the column. So for example, if I was to click on this Shipped Date column where we have dates in here as opposed to text when I right click I now get Sort Oldest to Newest or Sort Newest to Oldest. So this contextual menu will change depending on what column you're sorting on.

Now in Access you're not restricted to sorting on just a single field. You can sort on multiple fields. And if you cast your eyes up onto the Home ribbon into this Sort and Filter group you can see we have an Advanced dropdown and we have an option for Advanced Filter and Sort.

Now when we click this it will take you into a screen which I'm hoping will look reasonably familiar to you, particularly if you've worked through the Access beginner's course. And what we can do here is we can take our fields from our Orders table. So in this top half of the screen we have a graphical representation of all of the fields in our Orders table. And the bottom half we have a grid structure. So we can use this grid, which is basically the standard query grid, and use it to sort the order of the table. Now I'm not going to go into how you do this right now but we will be looking at this a bit later on in the course. But you could build a pretty complex sort using this approach.

Now you can sort on multiple fields just by making sure they are in the correct order from left to right across the Datasheet view. So if I wanted to sort by Ship Via and Ship City, so all I would really need to do here is make sure that the Ship Via column is to the left of the Ship City column, which it is, and I'm actually just going to move this one next to Ship Via just to make it a little bit easier. So whichever one is further to the left Access will assume that that's the first one that you want to sort on. So if I now select Ship Via and right click and do Sort A to Z you'll see that it sorts it. So I have Shipping Company A and then I have Shipping Company B, so on and so forth. But it hasn't sorted Ship City. However, if I was to select both of these columns, right click and do Sort A to Z, so what you'll now see is that the sort is based on both of those fields. So for Shipping Company A we have all the ship cities in alphabetical order, so on and so forth down the list.

Now I'm actually going to remove both of these sorts and again if you glance up to the Home ribbon in the Sort and Filter group you can see you have an option to Remove Sort. And that essentially resets everything back to how it was.

Now I've just gone in and reset some of those formatting options back to how they were originally just so everything is a little bit easier to see because the last thing I want to talk to you about is filtering.

Now one of the reasons this is important if you're using Datasheet view for maintenance is if you have a very large number of records you can filter just to find the records you need to work on. And I'm going to assume that you are somewhat familiar with filtering.

Now we're going to do a filter on Ship Name and you'll see if we click the dropdown just there we do have a number of different filters we could use. We have some text filters and we could also make our various different selections depending on what we want to see in our filter. But before we do this I want to point out a couple of things.

Down here in the Navigation bar right at the bottom there is a Filter On/Off toggle button. It currently says No Filter. And you can use this button to toggle on or off a filter or specific filters and you can use this as an alternative to the filter we've just shown. And what you probably didn't know is that you can select the value you want to filter on before you apply the filter. So for instance, if I click in the Ship Name, if I click in where it says Christina Lee and then go up to my Sort and Filter group and click this Selection dropdown you will see I have a number of text filters that are already based on that selection Christina Lee. So if I wanted to filter for records that are equal Christina Lee in the Ship Name field I can apply that now and it pulls back those five records. Now notice that that filter toggle in the Navigation bar is now showing as Filtered. And if I click on this button it's going to change it to Unfiltered and my whole list is back again.

Now even though that I've unfiltered this set of records notice that it remembers that there is a filter in existence. So if I click this button once again it's going to rerun that last filter I applied for Christina Lee on the Ship Name column. So now let me apply a second filter. So we're going to say Ship Via and in this case I'm just going to select Shipping Company C. So in this case I have a composite filter. I have the ship name of Christina Lee and I have a ship via Shipping Company C. And once again if I use this toggle at the bottom I can go from my filtered records to my unfiltered records and back again.

Now just a couple of final points with regards to filtering. If you go back up to that Advanced dropdown you'll see that the second option down is Filter by Form. So the currently applied filter is in force. We have a specific value for ship name and a specific value for Shipping Company C, but I could go in and apply a specific date and build up a complex filter in that way.

Also if we jump back up to Advanced if I go into Advanced Filter and Sort it will take me back into a grid where I can build up a complex sort filter using the querying approach that we're going to look at later on in this course. So I'm just going to jump in and I'm going to clear all filters to put my data back to how it was originally.

And the very last thing I want to show you is that along with Microsoft Excel you can have a totals row in Datasheet view in Access. So where you have any number of records and some of them hold numeric data if you want to do things like determine totals of columns, maybe the maximum, the minimum or maybe even the average of columns you can do those by applying a totals row.

So again in the Records group you see we have an option for totals. And it's as simple as that. If we scroll down to the bottom you can see we have our totals row. Now there's currently nothing in that totals row because we need to tell Access what kind of calculation we want to do. So for example, if I go to something like Shipping Fee maybe I want to see the total amount of all of the shipping fees. So if I click the dropdown I'm going to choose a Sum calculation in here. If I wanted to do something like a Min of a Max, if I go to Order Date and click the dropdown maybe I want to find the earliest order date. So that would mean doing a minimum in this column. And I can see the earliest date is February the 15th 2019. So that totals row is very straightforward.

If you want to turn it off just go back to Records and click on Totals again and that will remove that row.

So we've covered quite a lot in this module. That's the end of this section. I will see you in the next one.

Video: Subdatasheets

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to be taking a look at Subdatasheets.

Now when you're working in Datasheet view and if you need to update and modify data it may seem as though you are restricted to working on one table at a time, but it is possible to show more than one table in Datasheet view and also show tables that are linked together in one to one or one to many relationships.

So in this module we're going to have an introduction to subdatasheets, I'm going to show you how you can add a subdatasheet, we're going to learn how to remove or hide a subdatasheet, I'll show you how you can open and close subdatasheets, and finally I will show you how you to add an update data in Datasheet view using subdatasheets.

So we're going to start with our Esprit de Tour database. And I really want to look at the relationships within this database.

Now if you haven't done our beginner's course you may not be as familiar with relationships but I'm currently in tblTrip and I've opened up the Relationships window and you can see there is tblTrip table in the top corner. And for a trip there are a number of tours which are specific instances of a trip. So we also have trips going to many countries and also a trip can have many types. Now the relationship I'm most interested in is the trip to tour relationship.

Now if I go back to the Trip table in Datasheet view you can see along the left some plus signs. And if there is a subdatasheet for this datasheet if I click on the plus sign I will see entries in the subdatasheet. So I'm going to click on the plus to the left of Iconic Italy and what comes up is a dialog. Now I'm going to come back to this dialog box a little bit later on. So for now I'm just going to click on Cancel.

So if there is a subdatasheet you will access the records in the subdatasheet using those plus signs. And when you have used it it turns into a minus sign. But more on that later.

Now let's click on the plus again and bring up that dialog. So this is telling me that there isn't currently a subdatasheet added to this datasheet. So I'm going to Cancel again and I'm going to add one. So I'm going to make sure my record is selected, I'm going to go to the Records group, I'm going to click on this More dropdown, and you can see I have an option in the middle there for Subdatasheet. Now you can add a subdatasheet to correspond to another table or query and we're going to concentrate on tables. And the subdatasheet we're going to add is the Tour subdatasheet. So I'm going to select tblTour.

Now when I select that it will display the two fields at the bottom, the linked fields in these two tables and they're the same two tables you will see linked in the relationships diagram for this database.

So the child field, that is the field in the child record and that's the tour record, is Trip Code. And the field it's linked to in the master record, and that is in the trip record, is the Code in the trip record. Now normally when you do this Access will correctly identify the linked child fields and linked master fields. So I'm going to click on OK to add.

Now still at this stage there is still no real evidence that a subdatasheet exists so let's click on the plus next to Iconic Italy again. And now you can see the list of tours for that trip. So these are all of the tour records for the trip which has the code ITI. And in fact if you look through the tblTour table and identify all of the tours for which the trip code is ITI you will see that all of these are the tours.

Now one thing to make clear, you can't have multiple subdatasheets. So in this case from the top level trip records. So you couldn't show subdatasheets for tours and countries at the same time but you can nest them. So having set up at the top level the trip record in the datasheet and then the subdatasheet showing the tours if you have a relationship between tour records and some other records in your database that had a one to one or one to many relationship with tour records then under each of these tour records you could put a lower level subdatasheet linked to each of these.

So for instance, you might want to show all of the bookings for each of the individual tours. And maybe for bookings you could show all of the names or the individual people that have booked onto each tour. And in fact you can go down to a nesting level of eight. So using this method it can give you a really sophisticated structure.

Now there may come a point in time where you want to remove or hide a subdatasheet. So first of all you need to make sure that you have a record selected, then we're going to go up to our Records group and click on More, go down to Subdatasheet and you'll see you have an option there for Remove. So if I remove this subdatasheet you'll see now that that's gone. And if I click on the plus now next to Iconic Italy it takes me back to that Insert Subdatasheet dialog box.

Now let me just put that tblTour back on again and it's worth noting that one other way that you can do that is to use the properties of the table because the current table because the current subdatasheet is a property. So let me go into the design of the trip table and I'm going to bring up the property sheet and you can see on the property sheet that one option we have there is Subdatasheet name. And what I can do is click the dropdown and change that to None and then when I go back to my trip table and click on the plus next to Iconic Italy you will see that that subdatasheet has been removed again. So that's an alternative way of removing subdatasheets. Now once again I'm going to put subdatasheet back again.

Now one other thing is that if I've got the subdatasheet open for a particular record I can close it again by clicking on the minus. And again if you jump up to that More dropdown and go to Subdatasheet you'll see that you have an Expand All and a Collapse All. So if I click Expand All it's going to expand out all of those subdatasheets and I could effectively do the reverse and collapse up all of them.

Now finally in this section we're going to look at data maintenance. Now one of the reasons for using subdatasheets is that people use them for data maintenance. For straightforward situations they are particularly convenient but as a general rule I avoid using subdatasheets if other people are going to be updating this data. I would generally consider it to be safer to give people forms to update data with. But if your users are safe and pretty skilled at working with datasheets then you might want to give them access to do that.

So let's suppose in this case I'm going to maintain the tour information about the Iconic Italy trip. And one of the things I'm going to do is increase the price for the last trip on the list. So I'm going to increase this from $2,320 to $2,520. And we're also going to add a new tour. Now note that the child price here is zero and that's because this is an adults only trip.

So the maintenance of the data in Datasheet view works pretty well. I can delete all tours, although I would say I probably wouldn't want to do that. I tend to like to hang on to old tours for historical or reporting reasons, but you can certainly let people update data in Datasheet view using subdatasheets.

Now a couple of things to note here. By default when you add a subdatasheet you get all of the fields in the relevant record. So in this case start date, price adult, and price child. You can of course hide those fields if you don't need them. So if I right click on Price Child I can hide that field. And when you add a record using the subdatasheet approach in the underlying tour record the trip code ITI is auto added on the basis that you are looking at the subdatasheet for ITI.

So I'm going to now save this as version 22 and I will see you in the next section.

Video: Exercise 01

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module I'm going to be setting you Exercise 01. And Exercise 01 is not quite as straightforward as it might first seem. However, if you are pretty good with queries you should be able to figure it out without too many issues.

Now all I want you to do is to add a subdatasheet to the Genre table. I'm just going to open up the Genre table and all I want you to do is make sure that we can just see the names of the biopic movies. So I want it to look like this, just the names. So there is a bit of a challenge for you.

My solution for this exercise is in the exercise folder and it corresponds to version A22 of the Night Movies database in the course files folder. So that's it for Exercise 01. I will see you in the next section.

Section 4 – Linking to Other Access Databases

Video: Linking Two Tables to Other Access Databases

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to be taking a look at how you can link tables to other Access databases.

So we're going to start out with some background to the features, I'm going to talk through with the difference between importing and linking, I'm going to show you how you can establish relationships between linked tables, I'm going to show you how to use the Linked Table Manager, and finally I'm going to show you how you can remove a link. So let's look at the background to the requirements we're going to consider in this module.

Now Access has been around for a long time and traditionally it's been based on a single file. It was a desktop-based database system. And one file would contain all of the data. So that's all of the tables, forms, queries, reports, etcetera. And as time went on this restriction of putting everything in one file became more and more difficult, particularly when you get a few different people wanting to share a single database. So given them all access to one file and essentially keeping them all in step and making sure one didn't delete things, etcetera, those considerations became more and more important.

Another factor is that as Access databases became more significant they very often contain data that was useful to other databases or other applications. So it became more and more important to be able to access data in Access databases from other databases. So that is essentially where the requirement to link to other databases has come from.

Now if you think about the Northwind Traders database it contains a lot of information about invoice orders, forms and reports but it also contains information about customers. And information about customers tends to be used widely in a business context. And in order to demonstrate linking to tables in other databases what I'm going to do is to create a tiny demo database. So just the beginnings of a CRM or a Customer Relationships Database Management system. And in doing that I'm going to access some of the customer information that is already in the Northwind Traders database.

Now as a point of review let's open up the customers table and you'll see here that each customer has a unique ID. And a customer is basically a company in this case. Now bearing that in mind we're going to create a new database. So let's close this one and I'm going to create a new blank database. So I'm going to save this in my course files folder with a name that includes CRM, Customer Relationship Management database, just so I can easily identify it. So there is my first version of my CRM database.

Now the first thing I'm going to do is I'm going to add a table which will record each contact I have with my customers and it's going to be a very simple table. It's just going to contain the customer ID, the type of contact and a couple of other pieces of basic information. So I'm going to go away and create that table now and I'll come back to you in a few seconds.

So here is the design. Now each customer contact record has a unique ID, that's the first field there. And note that that is also the primary key field. The second field is customer ID so that must correspond to the ID of an existing customer. And at the moment I only have this table so I have no customer details but what I'm going to do is access the customer IDs from the Northwind Traders database. So in this database I am going to say you only put in customer contact details for customers that exist in the Northwind Traders database.

Now the other three fields we have there are contact type, so that might be something like email or phone. We have contact date which will include the time. And finally notes about this contact. Now at the moment all I'm interested in here is that second field, the customer ID.

Now given that I want to access data in another database I'm going to close down this table, I'm going to go up to the External Data tab and within this Import and Link group if you click the dropdown underneath New Data Source and hover your mouse over the From Database option you'll see that one of the options we have is From Access. And in this dialog or this wizard you're really just required to make two choices. And the first one is where to locate the data that you want to get. So I'm going to browse to the course files folder and select the Northwind database.

Now it's worth pointing out here that once you've located where you're getting data from if you move that database to another location, another folder it's likely to cause you a few problems. So you really need to understand how Access deals with that situation when you move a database and we're going to look at that a little bit later on.

Now the second choice that you have is if you want to import or if you want to link. Now in this case we could import the table into this CRM database. I could say I want to take a copy of the Northwind Traders database customer table and once I've taken the copy and imported it in effect the connection would be lost. So it would be that I've taken this as my starting point from now on and I'm going to use that data but it might change in the customer table which wouldn't reflect because I'm just doing essentially a download or an import of that particular table or database at that particular time. Now if you what you really want to do is not import it but instead leave it in the Northwind Traders database so it will be maintained then you would select to link rather than import. So essentially what we're going to do here is we're going to create a linked table and click on OK. And you'll be given a list of available tables. Now the one I'm interested in here is the customers table, and click on OK. And what you'll see now is that I have my two tables in my CRM database. And one of them, the customers table, has a little blue arrow that indicates it's a linked table.

Now just some important points on this. Being able to do this will depend on the security on the Northwind Traders database. And if you're using that that's fine but if you're using this one of your own databases just bear in mind that some databases may not have enough access rights to be able to do this.

Now the second point on this is that I sometimes rename these linked tables to remind me which database I'm getting this data from. So I might rename this customers table to be NWT for Northwind Traders, underscore, Customers. And remember that when you're changing the name here it doesn't essentially change it in the Northwind Traders database.

And if I wanted to look at that data I can double click to open it up and it seems to be absolutely fine. I can see all of that information. Now I'm just going to close that down again.

Now one other thing to note is that if I wanted to go into Design view to make any changes the chances are that I'm going to get a warning. And it tells me the table is a linked table whose design can't be modified. If I wanted to change the design I would need to do it in the source. Now I can choose to open it anyway. So if I want to look at the design I can but I'm not going to be able to make any changes to that particular design.

So let's look at one or two of these fields. I'm going to click in that company field. Now notice that it says in the bottom righthand corner, This property cannot be modified in linked tables. So again I can look but I can't change anything.

Now let's close this design and go up into Database Tools because I want to look at relationships next. And I'm going to add both of these tables. So I've held down my Control key, selected them both, and I'm going to click on Add.

Now what I want to do is I want set up a relationship between the ID in the Northwind Traders database and the customer ID in the customer contact table in the CRM database. So we know how to do this. We've done it before. We can click on ID, we drag and drop it on to customer ID. So I'm just going to quickly check that that all looks okay. So we have ID in the NWT_Customers table linking to customer ID in the tblCustomerContact table and that all looks good.

Now one thing that's worth pointing out is that I am unable to enforce referential integrity. You can see that option is grayed out. And that's because this option you just can't do with linked tables. So I'm going to click Create to create that relationship.

Now you can use the relationship for different useful purposes. So for example when you are adding a record to the customer contacts table you can perhaps bring up a list of the available customers in the Northwind Traders database customers table and select one of those. But if someone who is working in the Northwind Traders database tries to delete a customer that you've used in the CRM database they won't be prevented from doing so.

Now I mentioned earlier that if you move a database that contains one or more linked tables it can cause you some problems. And there is a tool in the External Data ribbon that can help you identify and resolve some of these problems and it's called the Linked Table Manager.

Now there's only one linked table and if at any stage I wanted to check it's okay if I select it and click on OK Access will check that all the linked tables can be successfully accessed. And I can see there that mine is okay and I'm going to close that window.

So now what I'm going to do is I'm going to move that database and I'm going to put it temporarily into a different location.

Now having done that let me try and open the customers table. And you can see here I'm getting a message that it cannot find the file. Let's go to the Linked Table Manager, select our table and click on OK. Now instead of telling me that everything is okay it opens up File Explorer so that I can browse to find where that database has now gone. So all I would need to do is browse to where I've got it stored off and there we go. We're back to saying that everything is okay and just to double check let's now try and open the customers database. And there we go.

Now one other thing that's worth bearing in mind, if you are dealing with a very large table, so let's suppose this link table has thousands and thousands of huge records in it and there are only a few records you're going to use you can link queries. So essentially you could create a query that just gets the fields you want from that table and link the query to the linking database. Now I'm just going to close out of this and I'm going to go away and move that table back to where it should be.

And finally if you want to unlink a table that's very straightforward. Just right click and you can delete the link. So let's click on Delete. And it's worth noting that this doesn't delete the table in the Northwind Traders database. It just deletes the link. And you can see that that is explained in this reassuring warning message before we actually proceed. So I'm going to say Yes and my linked table is gone.

That's it in this section. I will see you in the next one.

Section 5 – Using ODBC and SQL Server

Video: Acquiring and Installing SQL Server Express

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to look at ODBC and SQL Server in relation to Access 2019.

Now as we've mentioned before Access used to be a single file database system. And that single file contained the data and the user interface components, such as forms and reports. And also that as time has gone on this simple model has needed to expand and change. Now we've already looked at accessing Access tables in one database from a different Access database and now we're going to take this one step further. So we're going to say that in Access you can use data that isn't held in Access databases but in other database systems. So Access can use data that's in any ODBC compliant databases. And fortunately one ODBC compliant database is Microsoft's very own SQL Server. So in this section we're going to explain a little more about open database connectivity, I'm going to show you how you can acquire SQL Server Express, I'm going to show you how you can download it and also install it, how you can set up some sample data in SQL Server Express, and then in the next section I will show you how to access that data using Access 2019. But first let's talk a little bit about ODBC.

So ODBC is a standard whereby providing a database system is ODBC compliant then the data management side of it is interchangeable with other ODBC compliant database systems. So if we're using the example from the preceding section and I need some customer data from my new CRM database provided it's an ODBC complaint database then Access should be able to handle it.

However, in terms of this course and showing you how all of this works we do have a couple of challenges. So one of them is that you may well not have access to an ODBC compliant database system. If you do then you're going to be fine and you can work through this with me. If you don't then one option you have is installing and setting up an ODBC compliant database yourself on your PC. And that is essentially what I'm going to go through in this section.

Now if you don't have the permission, the time or even the space on your PC to do this then that's absolutely fine but it is still worth you working through this module with me to gain a greater understanding of how it all works.

Now the ODBC compliant system we're going to install is SQL Server Express, and that is free from Microsoft so you won't need to spend any money in order to work through this.

Now I mentioned that we have a couple of challenges. The second challenge that we have is that when installing this system there are many, many options. So it might be that you're doing this on Windows 7 or maybe Windows 8 or maybe like me on Windows 10. So essentially I can't demonstrate this as though it were a lesson because my system might be different to the one that you're using. So this really is more of a demonstration of how you would do it.

Now if you already have an ODBC compliant database set up and available to you then most of this section you won't need. However I will say at the end of this video I will be showing you how to acquire some data that we're going to use in the next section because in the next section I'll be showing you how to access the data from Access 2019.

Now SQL Server is Microsoft's ODBC compliant database system and it comes in a number of different editions. Now the page we're looking at here are Microsoft.com will probably have changed by the time you come to this page. They do tend to change their information and the way their pages look quite a bit, but the general principles will remain the same. So what I'm essentially saying here is that if when you come to this course the page you go to looks slightly different to the one I'm using the information is still going to be on there, you just might have to hunt around for it a little bit.

Now from here I'm just going to click on this Try Now button and you can see that I have two different versions that I could download if I wanted to. So I have SQL Server 2019 on Premise and SQL Server 2019 on Azure. Now if you scroll down a little bit it says or download a free specialized edition and we have a developer edition and an express edition. And it's this express edition that we're going to be looking at.

Now before we download there is one thing that you need to do. You really need to find out if you are running a 64 bit or a 32 bit version of Windows. And if you're not sure about that the way you can do it is to go to Control Panel and then if you click on System and remember I'm using Windows 10 here so the options you see might be slightly different but you'll normally have something called System or System Type. And underneath here it will say System Type and you can see here mine says 64 bit operating system. So I'm just going to make a mental note of that.

So let's now download SQL Server 2019 Express. And as usual by default everything that you download will go into your Downloads folder. And I can see my bottom bar here it's a .exe file. And this download here is essentially a management tool and that will manage the download of SQL Server Express itself.

So I've clicked on it to run it and I'm going to say Yes and now what we have is the Express edition dialog box. And it's asking us to choose an installation type and we're just going to do a basic installation. I'm going to accept those license terms and then I need to check my install location. So currently it's going to say to C:\Program Files\Microsoft SQL Server. And I can change that location if I want to. Now I'm happy with that so I'm going to say Install. Now you can see at this stage it hasn't yet asked me if I have a 64 or 32 bit system. And depending on the choices you've made you may or may not actually need that information but it is useful to know just in case you are asked because once you're in the process of installing it's quite hard to go back to Control Panel and look that information up.

So you can see what it's doing is two things here. So it's downloading that SQL Express Server Manager and it's also installing the software. And this can sometimes take a few minutes so you're going to need to be a little bit patient.

Now once the installation is complete you should see a message saying that it's been successful. And it's important to note the information on this page. So you're going to need to make a note of the instance name. So in this case SQL Express. Then underneath that we have SQL Administrators and that's just linking to my laptop and my username, so the account that I'm logged in as. We've then got some information about the features installed and then finally we have the version. So take a picture of that just to keep it safe somewhere, or write it down because you're going to need this information later on.

And the final thing you need to look at here is that connection string at the top. And you can see that there's a button to the right. And if you hover over it it becomes active. And if we click it that's going to copy that connection string to your clipboard. And what I'm going to do is I'm just going paste that temporarily into Notepad. I'm going to minimize that down and we're going to use it a bit later on.

Now the other thing I suggest you do is to install SSMS, which is the SQL Server Management system. And this will make it a lot easier for you to see, update or enter data into SQL Server Express. So that's what we're going to do next. We're going to install SSMS.

And that will take you to this Downloads page. So you can see I have Download SSMS selected in the left hand panel and I have a link here. So let's just go ahead and download that. And again that's going to go to your default folder of Downloads unless you set downloads to go to a different location. Again this can take a few minutes so again you need to be patient and wait for this download to finish.

So I can see my download is now finished. So I want to run it. It is an .exe file so I'm going to click on it and here we have the Installer. So again we've got the location it's going to install into and I am happy with that. I could change that if I wanted to. And you can see underneath it says By clicking the Install button I acknowledge and I accept the licensed terms and privacy statement. So we're going to click on Install.

Now when this setup is completed you'll see this dialog. And the only option we have there is to close. Now I will say that sometimes you are required to restart your machine. It didn't ask me to do that in this case, but if you do get that message then you will need to restart your laptop or your PC or whatever it is that you're using.

Now the last thing we will need is some sample data. And if you've used an earlier version you will be aware of a database called the Adventure Works database. And this has been used as sample data for SQL Server for a long time. And particularly this Wide World Importers sample database. And we're going to need some data for this in order to use it in the next section.

So to find this data you have two options. There is a version of it in the course files folder which you could use. Or you could jump onto this website and get the latest version from the web. And you can see here it just says SQL Server samples. We have the Wide World Importer sample database. And the version that we want to download here is the Wide World Importers DW-Full.bak. So that is the full sample database. And you want the one that says OLAP. So you can click on this link to download it and I'm going to save this in the course files folder. So it's entirely up to you if you want to use the one in the course files folder or come on here to get the latest version.

So pretty much have everything set up for the next module so I will see you over there.

Video: Setting Up SQL Server Database

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this section we're going to continue using Access with ODBC and with Microsoft SQL Server in particular. So in the previous section we downloaded and installed SQL Server Express. We also downloaded the SQL Server Management Studio and we downloaded a backup of the standard Microsoft sample SQL Server database, the Wide World Importers database. And we're going to take all of this and we're going to put it all together.

So we're going to start by restoring the Wide World Importers databases from backup into the SQL Server, then we're going to browse the database in SSMS, we're then going to create something called a DSN, and then we're going to use that DSN to make some of the data in the Wide World Importers database available to us in Access 2019.

So we have a backup of the Wide World Importers database or perhaps you grabbed it from the course files folder. And what we're going to do now is to restore that backup as though it were our own into our local instance of SQL Server. And to do that I'm going to use SSMS and we're going to need to run this as administrator.

So the first thing I'm going to do is find the software and I can see there it is under my Recently Added list. I'm going to right click my mouse, I'm going to go to More and select Run as Administrator.

Now one of the things I need to explain here is that I will be explaining enough so that you know what's happening but just remember that this isn't an SQL Server or an SSMS training session.

Now the first thing I need to explain is that with SSMS you can connect to a number of database servers. And database servers can be different physical devices, maybe devices that are on a network. They might be your own computer, the one that you're running SSMS on for example, and they may not actually be different servers. They might be different database servers on the same device. So it's extremely flexible but it can also be a little bit confusing.

Now for the purpose of this section you only need to bear in mind that we are trying to connect to that instance of SQL Server Express that we downloaded and installed in the preceding section. So in this Connect dialog box I need to put in a server name. Now this is the information that you should have made a note of earlier when we were installing.

So what I have here in server name is essentially the name of my PC followed by the server instance, the instance name. And that is what you should have made a note of earlier. And it's already populated mine for me, SQL Express, which if I check my notes is exactly the one that I had written down. I need to make sure that I have Windows authentication selected. And the username you can see has been populated and that in this case is the administrator which is me.

So again I have my laptop listed there followed by my username. And I’m going to click on Connect.

So I’m now connected to my instance of SQL Server. And if you look on the left in the Object Explorer you’ll see that I have my name at the top. So let’s just drag this out, make it a bit bigger. So we have my name at the top and then underneath that I have a tree view control and the very first entry is Database. Now currently I only have system databases and those are the ones involved in managing the SQL Server itself, but I also have these database snapshots which currently are empty. I don’t have anything in there. So now what I need to do is to attach the Wide World Importers file and then I’ll be able to browse. So I’m going to need to restore that from backup. And it’s very simple to do. If we just click on Databases and then right click you’ll see that one of the options that we have in the menu is Restore Database.

Now the first thing I need to do here is select my source. And the source for this is going to be Device. And what I now need to do is to browse and find the backup file. So I’m going to click Add. And what I need to do now is to find that backup file, which if you remember was in the course files folder. So I’m going to browse to that location. And of course if I browse to that folder I’ll find that file, Wide World Importers DW-Full Backup. So I’m going to select it and click on OK and that’s the only one I’m going to restore from so I’m going to click on OK and it loads it in. And you’ll notice that SSMS fills in the rest of this information for me. And I’m going to click on OK again. And after a while you notice that you’ll get this message that says Database Wide World Importers DW restored successfully. So I can click on OK and now you’ll see under Databases I have Wide World Importers DW. And if I click the plus I can now see the structure of that database.

Now it’s worth remembering that this is a SQL Server database and not an Access database, but there are some commonalities between the two. For example, SQL Server databases have tables. And if I wanted to look in a table I could browse to where it says Tables here and just expand that tree. And I can see all of the tables within that database. And if I wanted to look a bit further into a table I could browse that table. So for example, if we take this table just here, Dimension.Supplier I could right click and I could say select the top thousand rows if I wanted to. And this takes me and so I have my thousand rows in the bottom pane and then I have my code in the top. And if you are interested in SQL Server and SSMS then you can look a bit further into this. As I said, this isn’t a training session on SQL essentially but there’s a whole host of training courses and information out there if this does interest you.

And we’re going to use a little bit of this table in Access and it’s going to be used in our CRM table. And the table we’re going to use is this Dimension.Supplier table. So just keep that in your mind because we’re going to come back to that. And what I’m going to do is I’m going to close down this SSMS.

So I need to explain to you before we move on what a DSN or a Data Source Name is. So let's suppose you want Access to be able to use the World Wide Suppliers database and in some way you need to tell Access what it is and where it is. So we need to tell it it's an ODBC compliant database, what server it's on, we need to tell it what SQL Express instance it uses, and then what tables we want.

So first what we need to do is we need to tell it where to look. And the way we make other entities on our system able to see an ODBC compliant database is to create a data source name. And we do that from Control Panel. So bring up your Control Panel and we're going to jump into this first option here, Administrative Tools. And you'll see here you have two files, so ODBC Data Sources 32 bit and ODBC Data Sources 64 bit. So that's why it's important to remember what your system is actually running. And if you remember back to when we last came into Control Panel mine is 64 bit. So I'm going to need to run this as Administrator. So I'm going to right click and I'm going to select Run as Administrator.

Now there are three different types of DSN and in this case we're going to create one using the third type, a File DSN. So we're going to jump across to File DSN tab and you really don't need to worry about all of the other information in these other tabs.

Now what you see here might be different to what I see as it really completely depends on what's on your system already but we're going to be adding a brand new file DSN. So we're going to click on the Add button. And you have options for which driver you want to set up a data source for. So for us this is SQL Server. And you could at this stage then click on Next to move through that wizard, but if you know all of the information that you need to add, which we do because we made a note of it earlier, you can set it up through Advanced.

So if you look here it's telling us what we need to add into this little panel here. So we need to add the server and also the database. So I'm going to click underneath where it says Driver and we're going to type in Server, equals, and then we're going to grab our server name. And again yours is probably going to be different to mine because the server in my case is my laptop and this is my entire laptop name and then we have the instance which was SQL Express. And we also need to add the database. So Database, equals, Wide World Importers DW. Now if you are typing these in manually just make sure that you've got the spelling exactly correct before you move on. So I'm fairly happy with that. I'm going to say OK and I'm going to click on Next and we're going to give this data source a DSN name. And you can decide to save the DSN in any location you want to but there is a default location for DSNs and I'm going to save to that. Now remember if you are following along you won't be able to use my DSN because yours would be different. I'm going to click on Next, I'm going to do a final check of that information and if I'm happy with it I'm going to click on Finish and you can see there that my sample DSN has now been created. So I'm going to click on OK and we're now ready for the next step.

So I've opened up Access again and I have my CRM database open. And I want to access the data in that suppliers table from Access. So I'm going to jump up to my external data ribbon, I'm going to click on New Data Source, I'm going to say From Other Sources, and we're going to select ODBC Database. And up pops our little wizard and I'm going to say we're going to create a link, a linked table, and click on OK. And now I need to browse to my file data source DSNs. And if I scroll across I can see it's just there and click on OK. And now I get to choose my table. And if you remember we decided we were interested in this Dimension.Supplier table. So I'm going to select it from the list and click on OK. And you can see that Dimension_Supplier table has been added into my navigation pane. And once again note that it has that little blue arrow next to it which denotes that that is in fact a linked table. So I'm going to double click on Dimension_Supplier and there we go. I have access to the data in the SQL Server database.

Now one other thing to notice here in the navigation pane is that globe icon. And again that says that this table is linked by ODBC specifically.

Now of course I can right click and go into Design view if I want to but remember that I can't change anything related to that design. So I can click Yes to view the information but I can't make any changes to it.

Now if a linked table such as this doesn't have a primary key then Access will normally create one and you can see it's done that here. It's made the supplier key the primary key.

And when it comes to the data types also because the data types are slightly different between Access and SQL Server you may find that the data type as you see it in Access may not be exactly the same as the data type in the SQL Server Management Studio.

So that's it. We've successfully linked to an ODBC database. And as before if you want to remove the link you can right click and you have the option to delete.

That's the end of this module. I will see you in the next one.

Video: Exercise 02

Deb: Hello again and welcome back to our course on Access Advanced 2019. It's now time for Exercise 02 and I can't give you a sample answer for this because I would need to use a DSN from my system that wouldn't work on yours, but I'm going to show you my answer on the screen here anyway.

So I've made version 02 of my CRM database and at present it just has two linked tables. And the two linked tables are tables in the Wide World Importers database. We've got the Dimension Customer table and the Dimension Suppliers table.

So what I would like you to do is create a new database and just link to those two tables in your copy of the Wide World Importers database.

So to do this you're going to need to have created the World Wide Importers database in SQL Server from the backup file. And then you're going to need to create your own DSN and then use your own DSN to link to those two tables.

See how you get on with that and I will see you in the next section.

Section 6 – Splitting an Access Database

Video: Splitting an Access Database into Frontend and Backend

Deb: Hello again and welcome back to our course on Access Advanced 2019. Now I've already mentioned a few times that an Access database is a single file and due to this it creates a number of problems. Now one popular way of overcoming these problems is to split the Access database into two parts, a frontend and a backend.

So in this module we're going to talk about why you might want to split an Access database, we're going to go through the actual process of splitting up a database, we're going to talk through some of the issues around protecting the backend, and some of the issues around selecting a location for the backend.

So let's look at that first point, why split an Access database. And I'm looking at the Exercise 01 version of the Night Movies database. And you can see we have the navigation pane and we have our objects grouped under tables, queries, forms, etcetera. Now if several users are going to access this database and it's currently all in one file and they're going to access it from different locations, such as a network location, the more users that access it the more likely it is to get damaged perhaps by a network glitch or some kind of hardware error. And if that file is damaged then the database might be out of use. Now helpfully there's not really a reason why each user shouldn't have their own copy of most of this database. So each user could have their own copy of forms or queries in the database but one thing you can't have is your own copy of the data. The data must be shared. So what you could do is split this database into two and have the shared data in the backend and the nonshared data, so things like user interface components, forms, etcetera can be put into a frontend database.

And there are other reasons for splitting. So let's suppose this database is in a shared location and we have ten people using it from different locations. And every time one of these users wants to view or amend data they load a form and that form has to travel across the network. So maybe they use this form, maybe they change some data, maybe they run a report. All of that data has to travel, generating network traffic. So if you split the database so the backend is in the shared location and the frontend is on the user's PC only the data part generates network traffic. So as each user is just loading forms on their local device that reside on their local device it will reduce the amount of network traffic.

Another factor that can be important is generally speaking security is easier to organize if you have a database split. And we're going to talk a lot more about the security aspect a bit later on. But it is worth bearing in mind that when establishing security the way that you do it for the backend is a lot different from the frontend, which kind of leads me nicely into my next point.

If you have users who want to create their own reports or forms it’s generally easier to manage if each of those users has their own frontend. So rather than put them into a centralized location, cluttering the database up for other users, they can keep those customized elements to themselves in their own copy of the frontend database.

Now two other things to quickly mention and these aren’t really reasons for splitting a database but either or both of these can be consequences of splitting a database.

So one of them is that it sometimes makes it easier to do updates. So for example, if you fairly regularly bring in new forms or reports you can update the frontend database without necessarily changing the backend database. So users can essentially continue to use the database while others are being upgraded.

The other point is that if your ultimate aim is to move to something like SQL Server then starting out by splitting the database can be a useful step in that direction.

So let’s now get on with splitting our database and I’m not going to use an existing database. We’re going to use one of those in a little while in Exercise 03. But I am going to create a new small database which will make it a lot quicker and easier to demonstrate. And we’re going to create an asset tracking database. And for that I’m going to use one of these templates. And if I scroll down you can see it’s the first one here in the list. So I’m going to click on that template. So here is the information about that asset tracking template. You can see it’s provided by Microsoft. Now I’m going to go away and give this an appropriate name and save it into the course files folder. So join me again in a couple of seconds.

So there we go. I’ve given my file a new name so I can easily identify it and I’ve saved it into the course files folder. So I’m going to click on Create.

Now one thing to note here is that when this template has loaded you’re going to get this welcome screen. And I’m going to talk more about this a little bit later on in the course. But for the moment let’s just close down the welcome screen.

And another thing to point out is that this database has four tables and you can see them listed there. We have Assets, Contacts, Filters, and Settings.

And some things to bear in mind. If you’re going to split a database you should always take a backup of the database in case things go wrong so you can always restore back to an older copy.

The other thing to make sure to do is that everything is closed. So you want to close all of the objects in the database. So I’m going to go over to the little cross and just close down all of the objects in this database. And it’s always best to do that before you start the split.

So now we’re going to go up to the Database Tools ribbon. We’re going to go to this Move Data group and it’s not immediately obvious which option you should select here but if you hover your mouse over Access Database you'll see the little screen tip there says Split a database into two files, one containing the tables and one containing the queries and forms, which is exactly what we want to do. And this brings up the Database Splitter. Now before we perform the split let me draw your attention to some important paragraphs in this dialog box. So it says here This wizard moves tables from your current database to a new backend database. In multiuser environments this reduces network traffic and allows continuous frontend development without effecting data or interrupting users. If your database is protected with a password the new backend database will be created without a password and will be accessible to all users. You will need to add a password to the backend database after it's split. It then says that this could be a long process and advises you to make a backup copy of your database before splitting it. And I will say that having done this in the past even on a fairly small database with little or no data it can sometimes still take a little while for the split to work. So I just advise you to practice a little bit of patience. So we're going to go ahead and we're going to click Split Database.

Now an important point to note here is that the file name that is assigned to the backend database is the same as the file name that I gave the database myself. But you'll see there is a slight difference. If you look down in this file name you can see it's added a suffix after the name of underscore-BE, and the BE stands for Backend. So this is the default suffix that you'll get every time you're splitting a backend.

So I'm now going to browse to my course files folder which is where I'm going to store this. And once I've located my folder I am just going to click on Split. So my database is now split and I have this little message that says it's been successful so I'm going to click on OK. And now if you look in that navigation pane underneath that Tables group, and I'm essentially looking at my frontend database right now, each of those tables has a blue arrow that indicates a linked table.

Now a couple of other points to cover before I finish off this module. The first one is that the location of the backend database is very important. It must be a shared location, so possibly a network location. And then any user that needs to access the data can do so subject to any security arrangements that I've put in place. Now if I want to give all or any of the users their own copies of the frontend that's fine and in the next section we will review how to make sure the frontend is successfully looking at the backend.

So in the next module we're going to start using the split database and I will cover some of the important implementation points. So I'll see you over there.

Video: Converting Shared Table into Local Table; Moving Backend Database

Deb: Hello again and welcome back to our course on Access Advanced 2019. In the previous section we split an assets database into two parts, a frontend and a backend. And I am currently looking at the frontend database.

Now there are a number of consequences of this split that I'm going to about in this module and also a couple of other important points to be made aware of. So in this module we're going to look at the use of shared and local tables, we're going to look at the importance of backing up local data, then we'll look at saving an Access database in what's called Compiled Format, I will then show you how to move the backend database, and also the advantages of using Trusted Locations.

So let's look first at the four tables in this database and as I pointed out in the previous module all four of them are linked to tables. But if I open the Settings table the function of that table is very different to the function of other tables. So as you can see it has a single record and the record has two fields, and it's ID is one of them. Now the purpose of this record is to determine if that welcome screen is shown or not. And the fact that a single record has Show Welcome checked means that the welcome screen is shown.

Now there are two possibilities here, either that the design of the database is such that this determines whether the welcome screen is shown and the other is that each user of the database should have one of these records and each user can choose if they want to see the welcome screen.

Now the mechanism for changing this record is one of the things we're going to come back to later in the course, but for the moment the important thing is that rather than have this record in this table shared by all users of the database each should have their own version of this record and this table. And therefore this table should be a local table to them and not one of the shared tables. So what we essentially need to do here is to move this table back into the same database as the frontend, the queries and the forms, etcetera so that each user has their own copy of this table and this record within the table. So given that we have a shared table and that we want to convert it back to be being a local table that's a very easy thing to do in Access 2019.

So the first thing I'm going to do is I'm going to close this table and I'm going to go to Settings, I'm going to right click and you can see we have an option in our contextual menu that says Convert to local table. And you can see now in the Tables group that little linked table arrow has been removed. So that's converted that back to being a local table and it's now stored in the frontend database. And of course each user will have their own copy of that.

So clearly the use of a local table for this purpose is fairly straightforward to see. There may well be other settings. You may want to give local users local values for but you may also want to let users set up their own local data. So if for example, they are developing their own add-ins or forms and that requires them to have some local data then you can certainly let your users set up their own local tables as well. And it's quite normal for users to be accessing a combination of local and shared data.

And another important point is if users do have local data in their frontend databases you do need to consider the consequences from the point of view of backing up the data. If the data is if you like your responsibility you need to make sure your backup regime covers not only the backend database but also the frontend database. Whereas if you're giving user permission to create their own local data you may just need to make it clear to them that it's their responsibility to back up the frontend data.

Now the next feature of Access 2019 we're going to look at is the facility to save the database in compiled format. And I'll explain more about what that means in a moment. But if you have just split the database you can save the frontend in compiled form which will not only usually improve the performance of the database but it normally makes it a little quicker and smaller. But also you will find security advantages as well. For instance, users won't be able to modify the forms in the database.

Now in order to do this we save the database in a different format, a compiled format. And the standard extension for this is .accde. And in earlier versions of Access it was .mde. And as I said, in this compiled form all of the program code which is responsible for running forms, reports, etcetera, is in compiled form, so it can't be edited by users. Users can however create and modify tables and queries.

So let's save this current database in .accde format. So we're going to jump into File and we're going to go down to Save As and you can see here in Save As we have an option for Make .accde and it says File will be compiled into an executable only file. So we're going to select that option and click on Save As. And I'm going to navigate to the same folder. So we're going to save this into the course files folder. And you can see there the file extension .accde and I'm going to click on Save. Now you can see that we've got the welcome screen come up again so I'm just going to close out of that. Now let me also close this .accdb file and take a look at the .accde file in File Explorer.

Now what you can see here is that the .accde file is smaller than the .accdb file. And although functionally the database will be able to do the same things what users can do will be severely limited. So let's open the .accde. So you can see here that I have a warning message that's telling me it's not possible to determine that this content came from a trustworthy source. Now I know that this is a trustworthy source so I'm just going to click Open. And you also might see little popups like this saying Module not found. Now don't worry about that too much at the moment. We're going to look at that later on. So I'm just going to click on OK. And the asset list form opens. Now I'm going to close this form down. So let's now try and go into the design of the asset list form. So if I right click the answer is that we can't. You can see that Design view option is grayed out. And if I was to go to a report, so let's scroll down and choose this Asset Details report. If I right click again I can open it but I can't go into Design view. So in .accde format users are restricted in terms of their ability to change the design of the database, although generally speaking they can still run forms and reports. So essentially they can still use them, they just can't change the design. And we will see in this course that there are various ways that you can control what users can do and protect the database. Another consequence of these facilities is that you can provide many levels of security on the database as well.

So as you start to look at options for distributing Access databases using .accde format for the frontend it's an attractive possibility. You might want to say that you don't even want people to be able to see the navigation pane on the left hand side and that's something that we're going to look at as well.

Now something that may well happen is that you may need to move the backend database at some stage and this could potentially cause problems for your users. We've covered the tools for overcoming the problems this produces earlier in the course but I would like to give you a quick demo of what happens here. So I'm going to go to my course files folder.

So there is the backend database. I'm just going to move it to this little subfolder that I've called Covered for the moment. And then I'm going to open the .accdb. And of course it can't find it. So I'm going to click on OK and you can see that I've now got this Macro Single Step dialog box. Now again don't worry too much about this at the moment. I'm going to explain that a little bit later on. So I'm just going to click Stop all macros and I'm going to jump up to that External Data ribbon and I'm going to go to Linked Table Manager. So relinking these tables is going to be fairly straightforward. So I'm going to select all three tables by clicking Select All and I'm going to click on OK. And you can see it's prompting me for the new location. So all I need to do is to browse to that Covered subfolder. There's my backend database, click on Open and it says All selected linked tables were successfully refreshed. Click on OK and click on Close and now if we try to open the database again we can see that it opens fine. So again I can just close down that welcome screen.

Now the last thing I'd like to look at is setting up a trusted location. In fact you can set up more than one trusted location. Now your users may get a little frustrated by the fact that if you distribute .accde files like this as it opens you get a security notice. So if I open the database and then use it and close it again each time I open I keep getting the same message, which is potentially kind of annoying. So a way we can get around is to go to File, down to Options, and we're going to jump straight into the Trust Center page and we're going to click on Trust Center Settings. Then from here we're going to go to Trusted Locations and what I could do is click on Add a New Location and add my course files folder into my trusted locations list. So that means every time I open a file I'm not going to be prompted with that warning message. So I'm going to go ahead and I'm just going to add that new location in and I will join you over in the next section.

Video: Exercise 03

Deb: Hello again and welcome back to our course on Access Advanced 2019. It's now time for us to do Exercise 03. And what I'd like you to do is to take version A22 of the Night Movies database and turn it into your Exercise 03 database.

Now the database needs to be split into a frontend and a backend and the frontend needs to be saved in compiled form. You also need to make sure that when you open the frontend you don't get the security warning you saw in the preceding section. So here is my version. Let's now look at the File Manager Exercise files folder.

If we look at the folder it's very important to note that there two versions for Exercise 03 and each version has two files. So there is one called Exercise 03 X32.accde. That's an executable database file with a backend. And there is an Exercise 03 X64.accde. And that's a 64 bit executable file with its own backend. Now depending on if you're using a 32 or a 64 bit version of Access 2019 you'll be able to open one of those .accde files and not the other. And of course when you do open the relevant .accde file you will need to relink the tables in the relevant backend tables as well. So just be careful of that one as one of the two won't work for you.

That's it for Exercise 03. See how you get on with that and I will see you in the next section.

Video: Addressing Multiuser Issues when Sharing Data

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to be taking a look at how you can address multiuser issues when sharing data. And I've referred a number of times to the principle of data being shared between users.

Now whilst we have been looking at putting in place the mechanisms to do this there are a small set of issues that arise when users share data. To give you an example let's suppose you located a record in the database, perhaps looked at it on a form, and you're considering making a change to that record. And whilst you're doing that another user of the same database accesses the same record and also wants to make a change. Potentially either of you could make a change that will affect a change the other one has recently made or perhaps whilst one of you is making a change the other changes it. So in that case the first user would be looking at an outdated record. So these multiuser aspects are handled in Access using a number of different tools and techniques and we're going to look at many of these in this module.

So first of all I will look in more detail at the issues that arise when users share data, then I'll cover some of the advanced client settings in Access that can help you deal with these issues, and then we'll look at default record locking and record level locking.

Now when it comes to looking at issues when users share data one of the most important things to look at is the balance between the size of the database and the number of users and the level of churn in a database. And churn is the amount in which that data changes. So if you have a database, so maybe a large one that has a large number of users but where the level of change or churn is low then you can afford to approach the issue of change in a very different way to one where the level of churn is high. So for example, if you have a database with a very low level of churn you can probably take the approach that when a user is going to change a record the other users are locked out of that record. Whereas if you have a database with a high level of churn, so maybe you have a lot of users and at any time users might want to see records in that database, and if anyone wants to change a record it's probably going to be a little bit frustrating for other users not to be able to see records where one user is attempting to change one. So when you're looking at your approach to multiuser access it's important to take into account the level of churn in the data. So what we're going to do now is look at some of the advanced client settings in Access 2019.

So let's jump up to the File tab and we're going all the way down into Options. And we're going to go to the Client Settings page. And we're looking for the Advanced section. So let me scroll down and there we go. Now some of these settings are very important in a multiuser context and one of them is the default open mode. Now I must emphasize this is the default open mode so it's not a particular open mode for one database or one user. It is the default on this device. And the choices are Shared or Exclusive. And almost without exception this will be set to Shared because you would expect to open Access databases in shared mode. There are some times where you may need to open an exclusive use, usually for maintenance. So if you need to do something to the database where you need exclusive access, in which case you need to make sure that anyone else using the database is disconnected from it, then you would open the database in exclusive mode.

So let's suppose I wanted to open one of the exercises in the exercises folder in exclusive mode. I'm going to cancel out of here. So let's click on File and go to Open and I'm going to browse to my course files folder and let's say we want to open this one here, the A02.accdb. Now instead of just clicking Open if I wanted to open this in exclusive use I just click that dropdown on the Open button, you see I have an option there for Open Exclusive. And of course this relies on nobody else using this particular database at the time. And as soon as I open it in Exclusive mode it means that no one else is going to be able to go into it. Now I'm not going to do that for that for the time being. I just wanted to demonstrate it so I'm going to click on Cancel and jump back to my database.

Now I'm going to go back to our Options and down to Advanced again. Now with the next option we decide on the default for record locking. And I must emphasize again that this is only the default and you may vary from the default in a number of ways. But let me describe these three options for record locking. The first one is No Locks and we do not by default lock a record. If a user is looking at a record and beginning to make changes to that record it doesn't lock anyone else from being able to look at the record or potentially making changes to it. So this option is used when either there is little chance of two people trying to change the same record at once or where the implications of doing so are not that serious. And whichever record locking option you decide to choose you can in any case write VBA code to ensure there's no damage done. So that on top of whatever Access is doing you can still write code to ensure your database is safe.

Another situation where you might use the No Locks option is primarily when people are adding data to a database and there aren't really that many situations where the data gets regularly changed.

Now the second option for default record locking is All Records and this applies to a table. And when you apply an All Records locking option to a table it means that all of the records in that table are locked. So this is sometimes used as a halfway house for doing maintenance. If you need to do maintenance on a table but you don't want to lock people out of the database all together you can use the All Records record lock.

And finally the third option is Edited Record and this means that when a user selects a record or opens that record that record and that record alone is locked. So the assumption here is that someone else might look at that record while changes are in progress and we're going to stop them from making changes themselves. Now this is a safer way of doing updates but it can be frustrating if the person who gets the edited record lock then goes off to make a cup of coffee, leaves the record open on their screen and so that record is locked for a substantial period of time.

Now with this Edited Record option it may still be the case that you need to write VBA code to deal with potential conflicts. But it is a safer way of dealing with record locking where you have a database where there is a significant chance of potential conflicts when editing user records.

Now I have one or two more things to explain here and I want to particularly look at this checkbox underneath the default record locking option that says Open databases by using record level locking. I talked just now in terms of locking individual records but there is an alternative option to that and this is for us to use page level locking. The records in an Access database are actually accessed using pages of data and a page can typically have many records in it. So if you are doing edited record locking and you are asking Access to lock individual records if you check this box it will actually lock the page that the record is in, which means you may be locking other records around that record as well.

Now the reason you might do this is for performance reasons, because locking pages is a much quicker thing for Access to do. So if you have performance issues with an Access database sometimes switching from record level locking to page level locking can sometimes make things better.

And I also suggest that you take a quick look at the numbers underneath that checkbox. There is a set of numbers there and sometimes you might need to adjust those numbers to better suit the multiuser environment you are working in. Now let me explain one or two of them and you can look the others up in Microsoft Help.

So the second one down, Refresh interval in seconds. This is the interval in which data will be refreshed for you. So mine is set to 60 seconds or one minute. So in my case every one minute that data will be refreshed. So if someone has changed something that will be refreshed minute by minute. And of course you can change that refresh interval just by using those little scroll buttons. Notice that there is also an equivalent ODBC refresh interval. So I suggest you familiarize yourself with those and adjust those numbers as necessary.

In the next module we’re going to start to take a look at the user of macros. So I will see you over there.

Section 7 – Macros

Video: Basics of Macros

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to look at macros and you may well have used macros before. And we're going to go through some of the basics in a moment and then we're going to move on to the more advanced aspects of using macros in subsequent modules.

So macros in Access are pretty straightforward and the advantages you get from being able to use and understand macros far outweigh the amount of effort involved in getting to grips with them, at least the basics up to an intermediate level of their use.

So specifically in this module I'm going to give you a quick introduction to macros, we're going to do a quick comparison between macros and VBA, then we'll run through creating a very basic macro, and I'll show you the different ways you can run a macro. And we'll also look at how you can edit a macro as well. So let me start this brief introduction to macros by talking about what a macro is.

Now when you're using Access you normally perform sequences of steps. So these might be things like opening tables, locating records, making changes to data, and so on. And sometimes it can be helpful to record the sequence of steps and replay them again. And this is particularly useful if the particular task, the thing that you're trying to carry out is something you do on a regular basis. It can really make you a lot more efficient.

Now one option for recording a sequence of steps is to record them in a macro. So let's say you have four steps that you perform in order to carry out a specific task. You can record those steps in a macro. Now not only can you repeat that sequence of steps whenever you like by running the macro but if in fact what you need to do is perform that sequence of steps hundreds of times then you don't have to sit there and manually perform those things hundreds of times. What you can actually do is set Access up so that it runs that macro hundreds of times.

And macros can be useful not only for performing steps many times but for performing those steps in the same sequence many times. So in effect it removes the element of uncertainty or error or doubt.

One other very important factor in Access is that macros are not the only way of performing steps if you like automatically. There is another approach and it's a different programming language called VBA. And in Access you can use either macros or VBA.

Now if you've used other Microsoft applications most of them have macros and VBA as well. but one important thing to warn you about is that although this terminology is used through all of the Microsoft applications the exact nature of macros and VBA in each application does vary quite a bit. And in fact in Access macros and VBA are very different languages. And although you can do many things using either macros or VBA just about anything you can do with macros you can do with VBA. Whereas with VBA there are a vast number of things you can do with it that you can't do with macros. So to some extent macros are the less versatile tool but they are the easier tool to use. So we're going to be looking specifically at macros in this and the next few modules and then we'll look at VBA a bit later on in the course. So let's start out by creating a very basic macro.

I'm going to jump up to the Create ribbon and the last group on the end is Macros and Code. So let's click the Macro button and now we're in a space where we build up a macro as a sequence of actions. And if you look at the tab just here you can see that my macro that I'm about to create has been given a default name of Macro 01. And of course you can change this, which I'll show you how to do later on.

Now the actions that are included in the macro can be selected from the Actions catalog on the righthand side. So let me make that a bit wider. And we have some program flow actions. And you'll see one of them in there is a comment. So that means that we can put a comment in and I can also do things like include an If statement if I want to. So what an If statement would allow me to do is to carry out my macro with its series of actions on a conditional basis. So for example, you may need to check something at some stage and then the subsequent actions will depend on the result of that check. And you can also create submacros, and we'll talk about that a little bit later on.

Now the actions themselves are included in this catalog here. So for instance you have data entry operations. And if I click the little arrow to expand that I can see the options or the actions that I have. So from here I can do things like delete records, edit list items, save records. And you can see underneath I have quite a few other different catalogs there as well. So let's get started building out macro.

Now there is a tradition when it comes to programming that the first program that you ever create, in this case the first macro, is just to say Hello World. So we're going to stick with that convention as we build this macro. Now I know Hello World is not particularly useful but this will demonstrate many of the things you need to know to get you started.

Now notice that we have this little control here, this plus. And in the box it says Add New Action. So I can click the dropdown and I can choose the action I need here or I could choose it from my Actions catalog on the righthand side. Now on this occasion I'm going to use this dropdown list simply because it's right there in front of me. And the action that I'm looking for is Message Box. So I'm going to scroll down and again these are in alphabetical order and I can see there it is in the list. So let's select that one.

Now when I select this a couple of important things happen. One is that I'm presented with the arguments to go with this particular action. Now in the case of the message box there are four arguments, there’s four additional pieces of information that I need to provide. So the first thing is the message itself. And we’ve already established that our message is going to Hello World. I can select if I want a beep sound when this message comes in. So in this case I’m going to say No. I need to select the type of message. So again if we click the dropdown you can see I have Critical Warning, Warning, and Information. So I’m going to say that this is a warning with an exclamation point. And then we need to complete a title. And title you’ll see what this is in just a moment but I’m just going to put something in here for now.

Now the other thing that happened here is that Access presented me with my next action so I can go in and I can add another action if I want to. So I could do exactly the same thing. Select from this dropdown list or again I could choose from my Actions catalog. So let’s add another action in here. So I’m going to say Go to page, and the sequence in which Access adds these options is just going down the page from top to bottom. Now if I was to make one of my actions an If command or add another action that uses the If command that sequence might get more complicated but basically it’s always from top to bottom. Now I don’t actually want this Go to page action so I’m going to delete it and you’ll see just on the righthand side we have a Delete option. So very simple just click that and it’s gone.

Now in theory at this point I could run my very simple macro. I will however run into a very small issue. So I’m going to run this macro. I’m going to go up to my Design ribbon underneath my macro tools and the first button I have there is Run, so let’s click it. Now what Access does is it basically insists that I save the macro first. So I’m going to say Yes to save my macro. And I’m now required to give my macro a name. So much like everything else that we do in Access I like to have a naming convention. So I name all of my macros MCR and we’ll call this Hello World and click on OK. So now my macro exists and it’s run and you can see what I’m getting here is a message box, which is the action we selected, the message is “Hello World”, there was no beep when it came up, the type is a warning and you can see the yellow warning triangle, and the title, “This is important”. So it looks like my macro is working perfectly at this stage. So let’s click on OK and why don’t we run it again? And there we go. Works perfectly on the second time around as well.

Now one other thing to be aware of is if I go over to my navigation pane and just click the dropdown there we have an option for macros. So if I click Macros you can see it’s going to list all the macros that I have and there is my macro for Hello World. Now as you’ll see in the coming modules there are many ways of running macros. And in many situations macros can be invoked automatically when other things happen in your database. But the one other thing you need to know is how to change and edit an existing macro. And this is pretty straightforward stuff.

So I have the macro selected in the pane on the left and if I wanted to make changes to this message box command, if I click somewhere within you can see that I can now make changes. So I can change my message, I can change the beep, I can change the warning, I can also change the title. So let's make some small changes to our macro. I'm just going to change the title and I'm going to change the type from Warning to Information and let's add an action. I'm going to add a comment and I'm going to select it from the Actions catalog this time. And you'll see this comment appears within this sequence of action steps. So let's add a quick comment. Now once I've finished that I can perhaps have another action. So let's have another message box. I'm going to add a message. This time I'm going to keep beep as yes. The type is going to be Information and the title. So now what we essentially have here in our macro are two message boxes and the second one has a beep. So I think I'm just about ready to run my macro. And remember when you go to click Run it will ask you to save any changes that you've made. So let's click on Yes and see this macro run. So there's my first message box. I have the information icon. It says "Hello World" with a title of "This is a very important message". I'm going to click on OK and I get the second message box. "I hope you're well" with a title of "Have a nice day". And you probably didn't hear the beep because of the way that I am recording this video but let me assure you there was a beep when that second message came in.

So you've seen there running a macro and also editing a macro.

The final point I want to go through is just in the navigation pane if we click the dropdown and select All Access Objects you can now see your two linked tables and also the macros that you have. And as with most objects in Access once you've finished working on your macro you can right click and you can close it down. But remember it is always over here for you to run.

Now if at any point in time I wanted to go back in and edit this macro if I just select it in the navigation pane, right click and select Design view that's going to take me back in, then all I need to do is click within whichever part of the macro I want to edit and I have all my options available to me. So very, very easy to modify any existing macros.

In the next module we'll look at a special kind of macro, an AutoExec macro, so I will see you over there.

Video: Macros on Events; AutoExec

Deb: Hello again and welcome back to our course on Access Advanced 2019. In the previous module we created a very simple macro to display Hello World and we talked about some of the properties and uses of macros. In this module we're going to take a look at the ways you might use macros and also look at the very important AutoExec macro. So first we'll look at macros and events, then we will apply a macro to an on click event, we'll talk about the AutoExec macro, and we'll talk about the possibilities of using local settings in Access 2019.

Now when we ran Hello World in the previous module we clicked on the Run button in order to execute that macro, but there are various different ways to run a macro. And macros will run on the basis of various events occurring. So that might be clicking on the Run button but it also might be something like somebody clicking an OK button on a form, maybe opening a form, closing, deleting a record. Each one of those is an event and macros can react to respond to respond to events and in a simpler way you can write VBA code that responds to events as well.

So what I'm going to do is I'm going to create a form in the CRM database, I'm going to put a button on that form and I'm going to make a macro run when you click on the button. So I'm going to jump up to the Create ribbon and we're going to go to Form Design. And I'm going to put a very simple button on here using my control. So I'm going to grab this one here, the Button control. I'm just going to draw a reasonable size button. Now I don't want to utilize the wizard at this stage so I'm just going to click Cancel and I'm going to give this button a different caption. So we're going to say Click Here. I'm going to save the form and if we select All Access Objects in the navigation pane you can see the form just there. Now I'm going to create a macro. I'm going to save the macro. Now what I'm going to do is I'm going to make that macro run when I click on the button on the form. So to do that I need to look at the properties of the button. So I'm going to go into Design view for the form and I need to make sure that I have the button selected and I'm going to bring up the Property Sheet. And for this type of thing the tab that I want to go to is the Event tab. And with the Event tab I can specify what happens when certain events occur. So the most obvious event here is the on click event. So if I click the ellipsis it will take me into the Choose Builder so I can build a macro, an expression or possibly even some code. Now I've already prepared a macro, I've already recorded one so I don't need this builder. So I'm going to cancel out of there. So if instead I click on the dropdown in this On Click field you can see it lists out all of the macros that I've created. So I can just select my macro. So having assigned that I can now close the form and save and let me reopen the form. So there is the form. The only thing we have on this form at the moment is this button. So if I click on the button you can see my macro runs and I get my message box.

So not only can I do that as many times as I like but if I wanted that message to come up many different times on different forms I can assign that same macro to each of those events on many different forms. So essentially we would call this reusable content, essentially.

So in order to demonstrate this point I’m going to go into the Design view on this form again, I’m going to click on my button and make sure I’m in my Event tab in the property sheet. And what I’m going to do this time is I’m going to assign my macro to the On Mouse Move event. I’m going to close and save the changes and I’m going to double click to reopen up my form. Now watch what happens as I move the cursor. Currently nothing is happening. However watch what happens when I move my cursor over the button. You see my macro runs and I get that message box pop up. Now this might not be the behavior that you want but hopefully that demonstrates how easy it is to assign a macro to an event. And again, as we will see later on the Access Event module includes many events for various objects in Access.

So I’m going to click on OK. I’m going to go back in and I’m going to change this button so that it doesn’t fire when we move the mouse.

We’re going to turn our attention now to looking at a really important macro and that is the AutoExec macro. And many databases will have an AutoExec macro and you may get to the point where you want to write your own. So what I have open on the screen here is the Asset Tracking database and there is an AutoExec macro. And you’ll see in the navigation pane right at the bottom where we have macros you can see it sitting there, AutoExec. And it’s really important that this macro is spelt exactly right. And provided it’s spelt correctly that macro will be executed when you start up the database. So let’s look at the AutoExec macro for this database.

I’m going to right click and I’m going to go into Design view. Basically what we have here are two open form statements, although the second is within an If. So the first open form statement, and you can just click in here to see the different fields, it makes it clear what each parameter means. So you can see here it’s opening the form name asset list, it opens in Form view, and remember if you want to change anything you can click the dropdown and you have there Design, Print Preview, Datasheet, so on and so forth. If you want to apply a filter or a where condition you can apply that in there. And then we have Data Mode. And Data Mode you can add, edit or read only. Now I’m going to leave that blank for the time being. And then we have a Window mode. And if I hover over that it says Select the mode for the form window. So we currently have Normal and that means the form is in the mode setting for its form properties. We could also choose Hidden, we could have Icon, Dialog, so on and so forth, but I’m going to leave mine as Normal. So in terms of opening you can completely choose the parameters that apply.

Now you may have noticed each time you open this asset tracking database that the asset list is always open. And the reason is because this open form statement is in the AutoExec. So if you want to see a different form you know how to change that by changing the AutoExec macro.

Now the second option that we have on here is within an If statement. We are saying that if a condition is true then open the form. So let’s look at this If statement and when doing that we’re going to look at one option for dealing with local settings in an Access database.

Now earlier on in the course we changed the Settings table back from being a table in the backend database to being in the frontend database so that each user could have their own settings. So if we go up and look at the settings table in Design view you can see that we have two fields in each record. We have the ID and then we have the Show Welcome field. And that is set to yes or no. And having looked at this design in the settings table let's open it in Datasheet view. So you can see there we have the two fields in each record, ID and the Show Welcome field. So if I want to show that welcome screen when I first come in to the database I can make sure that I have that selected in there. So essentially we have one record and the value of Show Welcome is true. So let's go back to the macro and just look at this If statement.

So the If statement is saying get me the value of the Show Welcome from the settings table and if its true open the form Welcome. So that's basically how it works, what it's saying here. But you might be wondering what is this DFirst command that we have here. Well DFirst is a function used in Access to do what might seem like quite a strange thing. And the best way to get more information on this is to jump into Help.

So you can see here there are actually two functions with DFirst and DLast and it says you can use the DFirst and DLast functions to return a random record for a particular field in a table or query when you simply need any value from that field. And you can use the DFirst and DLast functions in a macro module, query expression or calculated control on a form or report.

So in this case we are only going to have one record in that settings table. And so the Show Welcome value in that field will be the only Show Welcome value in that table. That's why we can use DFirst. If we needed to find a specific value corresponding to some other specific filter or where condition we would need to do something a little bit more sophisticated than this. But the advantage here is that all we need to do is to find that Show Welcome value. So we can use the DFirst function. And in that function we first specify the expression that corresponds to the name of the field, so Show Welcome in this case. And secondly the domain where we're looking for that. So that would be the settings table. So what we're essentially saying is I don't care what the value and the reason is because there will only ever be one. So in this case there is a tick in that Show Welcome screen so the value will be set to True. So when I open the database I will see the welcome screen or the form. So this is an example of using local settings to allow users to determine certain aspects of how your database application runs on individual installations.

Note here how important it is that the settings table is in the frontend database and not the backend because if it was in the backend there's only one setting of Show Welcome for all users of the database.

So that's it for this module. In the next module we're going to look at more AutoExec macros, so I'll see you over there.

Video: More About AutoExec; Bypassing AutoExec

Deb: Hello again and welcome back to our course on Access Advanced 2019. In the previous module we've been looking at macros and we've also been looking at AutoExec. Now for much of this section we're going to be looking at some of the other things you can do with AutoExec macros and some key aspects of their use. So we're going to go through some more of the key points, I'm going to show you how to trust a project as part of running an AutoExec macro, we're going to look at the use of a login dialog, I'm going to show you how to hide the navigation pane, and finally how to bypass the AutoExec macro.

So first off we're going to focus on a specific example of an AutoExec macro and look at some of its features.

Now a couple of warnings before we proceed. As you start to learn more about macros and use of AutoExec you will discover it has a few shortcomings from the point of view of absolute security. Access is a little limited in terms of what you can achieve. Now for many people this isn't a problem, but if you do require a high level of security then it is sometimes necessary to move into using VBA rather than macros and in some situations it's not possible to achieve a high enough level of security in Access anyway.

Now one or two of these points I will come back to but ultimately if you can achieve the required level of functionality and security in Access it will very much depend on what your requirements are. So first off let's take a look at AutoExec. And the AutoExec we're going to look at is the Northwind Traders A2 database in the course files folder.

Now when you open this file there's a couple of things I want you to notice. I want you to notice the navigation pane on the left hand side and also the fact that we have this Login dialog popup box. So the idea here is that you come in and you select your name as an employee and then click the login to go into the database. Now we're going to come back to this a little bit later on so for the moment I'm just going to close out of this login dialog.

Now notice what you can see in the navigation pane. We can see lots of tables for customers, employees, so on and so forth. And if we scroll all the way down you can see right at the bottom we have our macros. And you can see one of them is AutoExec. So let's jump into Design view for the AutoExec macro.

Now in order to make this a little bit less chaotic to look at I'm just going to collapse some of these objects. So let's collapse up tables, queries, so we can just purely focus on this macros section. And the first thing I'm going to concentrate on here is this top section where it says Set displayed categories.

Now if I click this dropdown button in the navigation pane next to All Access Objects you see I have all of these categories. So I have things like creation date, modified date, filtered by group, so on and so forth. Now if I wanted to set up my database application so that some of these options are not visible, so maybe I didn’t want users to see created date, modified date, or filter by group I can do that. And I should emphasize we aren’t talking about what is selected when they open the database. We’re talking about what can be selected, what is available. So what I’m going to do here is I’m going to go over to the Actions catalog pane on the righthand side and I’m going to expand this user interface commands option. And you’ll see that one of them there is Set displayed categories. And what I’m going to do is I’m going to click on this, I’m just going to drag it and I’m going to drop it in between that If statement and the first set displayed categories. And you see I get a horizontal line there which is showing me where it’s going to be placed when I let go of my mouse.

So what I can now do from here is that I can click the Category dropdown and I’m going to say created date and I’m going to change the Show to No. I’m going to do this twice more. I’m going to grab, Set displayed categories, I’m going to drag it, I’m going to select from the category, modified date, and I’m going to say No, and do it one more time. I’m going to select the category and this time I’m going to say tables and related views and set that to No as well.

Now having done that, so I’m going to close down the AutoExec and I’m going to save the changes. I’m going to close down the database and I’m going to open it up again. Now again I’m just going to cancel this Login dialog. But if I now click on All Access Objects you can see that I no longer have in there those options that we set to no. So there’s no creation date, there’s no modify date and there’s no tables and queries. So that’s a demonstration of how you can control which options are available to users in that navigation pane.

Now I’ve just been in and changed the AutoExec back to only having the original set of displayed categories action in it and I want to now focus on the second action which is this one here, the If statement. And in this case we’re looking at a property of the current project. And just to make this a bit easier to see I’m going to collapse up some of these other categories. So you can see here it’s referring to the current project, then we have a dot and “is trusted”. So is this current project trusted? Now in this case it is trusted and the reason it is is because the database file is in my course files folder which we placed into a trusted location earlier on in the course. Now if it wasn’t trusted then what would happen is that this would say If not the current project is trusted then a form would be opened and the form is called the Startup Screen. So let’s have a look at that form Startup Screen.

If I just click to Open this is really just an informational form so it doesn’t really have any executable buttons in it. So what it says is Welcome to the Northwind Traders sample database. In order to use this sample click Options on the Message bar and select Enable this content. Alternatively open the database from a trusted location. Now in fact that would appear with the standard Access warning about trusting a project and really this screen is there for additional explanation. So it helps users understand what’s happening.

Now in order to demonstrate to what this would look like to a new user who'd not yet trusted this database or put it into a trusted location I'm going to make a copy of the database and put it into a location that's not trusted. And that's how it would look if it were opened from a nontrusted database. So that's a good example of a form that gives a user help but doesn't really provide any functionality.

So we're back now at the Northwind Traders AutoExec and we're moving on to the next action. So it says If the current project is trusted then open the form Login dialog. And it's important to point out that the login facility in Northwind Traders isn't really implemented in a way that would be viable as a workable login facility but it does have some useful features. So first of all let's look at some of the reasons why you wouldn't do logins using this approach.

So I'm going to close and reopen the database and the first thing that you're asked to do in this Login dialog is to select an employee. Now this would be quite unusual and not very clever to choose a username from the dropdown list because essentially anybody could jump on here and select anyone from this list. So I don't think you would arrange it in this way but the idea is that you select your username and you click on the Login button. So I'm just going to just select the default, Andrew Cencini, and I could then login.

The other thing that you wouldn't normally have on a Login dialog is a Close button, because again effectively what someone could do is just click on that Close button and carry on working in the database as normal. So it doesn't really suffice as a viable working option when it comes to logging in. But let's look at the Login dialog itself, so I'm going to close out of here.

So I've opened the Login dialog in Design view and you can see the two controls in here, the Select Employee and the Login. Now if I click on the larger one, so the Select Employee, and you look over in the property sheet on the right, so you can see here in the property sheet the selection type is combo box. So that's a combo box control. And if I click on the Event tab you can see that I have an embedded macro in here after Update. So basically when someone has made a selection on the combo box this macro is then fired. And an embedded macro is one that's actually been embedded into your database as opposed to creating an external macro and saving it as an MCR file.

Now if we click on the ellipses, so it has a SetTempVar action, it has an on error action, a re-query action, and a run menu command action. Now I'm not going to go through each one of these. I'm in fact just going to close this.

Now if I go to the Login button if you look at the property sheet you can see the selection type is a command button. And if we look down in the Event tab it says On Click and then we have another embedded macro. And if at any time if you wanted to specify an embedded macro rather than one as a separate macro file this is how you would access it. Now the important difference here between embedded macros and external ones that you create yourself is that embedded macros can't be reused. So by putting a macro into a separate file, like an MCR file, you can then call it from a number of different places. But if I only wanted it to be accessible in this location then this is a nice way of keeping the code with the control.

So what we've seen here are some of the elements of a login system. And what we really need to do is to take away that Close button in the corner of this Login dialog. And in reality, in a real life scenario, you'd probably want to add a lot more checking in to this Login dialog. So you'd want to check that the employee is valid or the user is valid and you might want to get them to do something like provide a password. And you definitely wouldn't want to allow them to just be able to click a Close button in the corner to bypass the entire login progress. Now we're not going to go through a lot of those things but it's definitely worth something thinking about.

Now as an Access user you will know how much you depend on the navigation pane when it comes to working within your database. But when it comes to users accessing a database, as we've already shown, you can control what's available. But one other option is to hide the navigation pane all together. So what I'm going to do now is show you how to remove that.

So I'm back in Design view for my AutoExec and the first thing I'm going to do is I'm going to remove this Set Displayed Categories action at the top here. I'm then going to jump across to my action catalog and I'm going to expand macro commands. And I'm going to select this one, Run menu commands. So I'm going to double click to add that in. And note it will add in that action wherever you are currently clicked. So you can move these around if you're not happy with the sequence. And I actually want to make this the first item so I'm just going to drag and drop it at the top.

So now with my new action I need to select my first command. And I'm going to select Window Height. I'm going to close and save, I'm going to close my database and I'm going to reopen it. Now you'll see when I reopen I still have that Login dialog there but there is no longer a navigation pane. So on the face of it I'm protecting my database from users fiddling around with those different elements.

Now of course users do still have the option of pressing the F11 key to bring that navigation pane back up but you can prevent that. So if you go into File and down to Access Options and go to the Current Database page one option here is User Access special keys. So if I uncheck this and click on OK it says to me You must close and reopen the current database so I'm going to click OK. I'm going to close down my database and reopen it and now if I try and use my F11 key to bring up that navigation pane nothing happens. Now of course there are ways of even getting around that but for most users this will probably suffice.

Now one last thing to show you and that is that you can suppress the running of the AutoExec macro when you open an Access database by holding down the Shift key. So again I'm going to close my database down and now when I go back to open my database, so you can see here it's the top one listed in my Recent. If I hold down my Shift key when I open it it now opens without the AutoExec being run at all. So I don't have that login screen popping up. Now not everyone is going to know about that option and it is possible to even suppress that as well but you would need to use VBA code.

So really what you need to do here is you need assess how secure you need to make your database. And usually a few simple steps like this, so hiding the navigation pane, is enough to achieve a reasonable level of security.

So quite a lot in that section. That’s it for now. We’re going to move on to the next section where we’re going to look at macro security, so I will see you over there.

Video: Macro Security Settings

Deb: Hello again and welcome back to our course on Access 2019. In this section we've been taking a look at macros. And in this module I want to look further into macro and code security. And quite a bit of this applies to VBA code as well as macros. So in this module we're going to take a look at an overview of code security, we're going to look specifically at macro settings, and then we'll look at trusted documents, trusted locations, and trusted publishers.

Now in principle in Access 2019 security is pretty straightforward provided you stick to one or two principles. So the first thing to bear in mind is that the main areas of risk to databases come from code being run. So one of the most important principles is that you should only have code running that you trust. So for example, if someone gives you a database from outside or you download a database from the web you do need to be careful of any associated code. And bear in mind some Access code can be very clever. And if that code include commands to do things like delete records in a table or corrupt data it could cause a lot of damage. So what we're really talking about here is restricting the ability of code to run. So let's jump into our macro settings.

I'm going to go up to File and down to our old favorite, Options, and we're going to click on the Trust Center page and we're going to jump straight into Trust Center Settings. And from here Macro Settings. And you'll see that we have four main options just there. And again just to reiterate, the macro settings apply to macros and VBA code.

So the first option here is Disable all macros without notification. So this means if you get a database from someone else the macros won't run and you also won't be told that the macros won't run. Now the second option, and this is normally the option that I have set as you can see here, is Disable all macros with notification. So this means if I open the database I've received from someone else the first thing I will be told if it has any code in it is that there is code. And it doesn't run I'll be invited to give permission in order for that code to run. So if I know it's from a trusted source I can just click Yes or Enable to run the code and off it will go.

Now the third option here, Disable all macros except digitally signed macros, this is normally more relevant to commercial situations, but you may as a private individual have a need for this option. So what happens here is that providers of macro code can digitally sign those macros to assure that they are safe to use and you would normally buy a digital certificate which would interact with the certificate supplied with the macro. Now as I said, for most people this won't be an option but just remember that that option is available.

Now the fourth option here is Enable all macros and I don't think I have ever had this set on any PC that I've used. So basically whatever macro it comes across it's just going to run that code. I would say this is probably the most risky out of all of them and I wouldn't recommend that you use this option.

So let's turn our attention now to Trusted Documents.

Now if you open a database in Access and trust it then the document is added to your trusted documents list. Now the odd thing about this is that you can't actually see the trusted documents list but basically as you go through trusting documents they are added to this kind of list that sits in the background. So once you've trusted it once the next time you open that database you're not going to be asked to trust it again.

Now there are some important controls here. You have a Clear button. So let's suppose you've been using Access for some time and you're a bit worried that the list contains maybe some old databases that you don't use anymore or maybe one proved not to be quite as trustworthy as you first thought. So what you can do here is to just completely clear that list out and essentially start again. So then each time you open a document you'll be asked to trust it to build up that list again.

Also note this checkbox here, Allow documents on a network to be trusted. So you may decide that that's an option that you want to do. So any documents that come from a network you want to trust them.

Now the next thing I want to look at is Trusted Locations. And we did speak a little bit about this previously because I added in my course files folder and my exercise folder into Trusted Locations earlier on in the course. So effectively anything I open that's come from these two folders is automatically trusted. And you have your options down here to add a new trusted location, you can remove, and also modify. And again I have a checkbox towards the bottom, Allow trusted locations on my network. And you can see that that isn't actually recommended, hence why I do not have it ticked.

And then the final one in this module is Trusted Publishers. Now I can see that I have something in there and really talking about these certificates that apply to these publishers is outside the scope of this course but if you do get a certificate from a publisher and you set it up in Windows you can see the details of the certificate by selecting the publisher and clicking View at the bottom.

So those are some things to bear in mind when it comes to your macro Trust Center settings.

That's it for this module. In the next module we're going to do an exercise so I'll see you over there.

Video: Exercise 04

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this section I'm going to set you Exercise 04 and in Exercise 04 we have quite a bit of do.

Now the starting point is version 23 of the Night Movies database in the course files folder. What I want you to do is to take a copy. Now it has a frontend and a separate backend and you need to create a new version. I've called mine Exercise 04 but you need to make sure the frontend connects to the correct backend. So if you just copy both of the version 23 frontend and backend files of course the copy of the frontend will link to the original backend. So you need to make sure you have a pair of separate frontend and backend files. So let me open up my new version, my Exercise 04 solution and show you the features I want you to include.

So I have a welcome screen and you can see there that I've included a clapperboard graphic. Now if you want to use one of your own graphics that's fine but I have also included the clapperboard as a JPEG file in the exercise files folder. So you might want to use that. I've done this as a modal dialog. You can see that I have a Close button but I don't have any navigation buttons associated with changing data. Then of course when the user closes the welcome screen they are back into the application.

Now that isn't all there is to do. I want you to incorporate a couple of other features.

Now let me bring up the navigation pane and there's no need to suppress the navigation pane in this exercise. I've included a settings table and the settings table is used so that users can enable or disable the welcome screen. There is also an AutoExec macro and of course you can use the AutoExec macro to invoke the welcome screen.

Now if I jump up into Options and go to Current Database there are a number of settings you can use in here to do some of the things we've looked at doing via AutoExec. For example, you can set up the first display form just here. So currently it says frmNavigationForm. Rather than specifically open a form in the AutoExec, and bear in mind on this occasion we're essentially opening two forms, that is the welcome screen and the AutoExec, and then from here we can specify which one we want to start working on.

Now if we move a little bit further down these options there is a facility where we can choose to display the navigation pane. And if you click on Navigation Options you can essentially specify what's available in that navigation pane. So this is a very flexible way of controlling that navigation pane. So there are a couple of facilities in Options that you may want to look at as good alternatives.

So that's about it on Exercise 04 except that when you have it working I'd like you to take some time just tidying it up a little bit.

So you may have looked at the version of the welcome screen that I showed you and you may have seen the name on the form as frmWelcome. Well notice it now says Welcome with an exclamation mark. The dialog is centered and I’ve sorted out inconsistencies in the color of the text. And the version you’re looking at now is version 04 in the exercises folder.

So see how you get along with that. We’re going to move on to customizing the ribbon and the Quick Access Toolbar in the next section so I will see you over there.

Section 8 – The Ribbon

Video: Customizing the Ribbon and Quick Access Toolbar

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to take a look at how you can customize the ribbon and the Quick Access Toolbar. And you may have seen how to customize these options before but apart from a quick review of the basics there are some other aspects to point out.

So we're going to start out with an introduction to customization and then I'm going to show you how to customize the Quick Access Toolbar or the QAT. We're going to talk about ribbon features in general and how to customize them. We're going to create tabs and groups and then I'm going to show you how you can import or export ribbons.

Now a little bit of background, ribbons were introduced in Access in 2007 and the ability to customize them was introduced in the 2010 version. So from 2013 onwards you can customize the ribbon extensively. And the ribbon you get in Access by default is very much a ribbon for designers and builders of databases. So if you're providing a database system to users it's likely many of them won't need to create or design forms. Now when it comes to a group like this, the Records group, you can probably envisage the majority of users wanting to be able to do things like delete records, save, create, but maybe not many users you would expect to go onto the External Data tab and use the options on here. For example, you wouldn't expect all users to go into the Linked Table Manager. So it's good to customize to the specific requirements of your users. So here I'm working on version 04 of the CRM system and we're going to first start out by customizing the Quick Access Toolbar.

And the Quick Access Toolbar is this little thing located in the top left hand corner. And if I click the dropdown on the end hopefully you're reasonably familiar with this. It does run throughout all of the Microsoft applications. And you'll see there frequently used commands or the ten or so most useful. You can just turn them off and on or disable or enable them by placing a tick next to them or not. So you can see I currently have three ticked, Save, Undo and Redo. And the ones that are ticked are currently sitting on my Quick Access Toolbar. So this is really just a quick way of being able to access commands that you use frequently without having to kind of dive around and jump into different ribbons.

Now currently I have my Quick Access Toolbar showing above my ribbons but you may prefer to show it below the ribbon. And if I click to change it you can see now it's under there. It makes it a little bit easier to access rather than having it tucked away in the corner.

Now if I wanted to customize that Quick Access Toolbar, for example if I wanted to add a command button which isn't one of these standard ones that are listed, all I need to do is click on More Commands. And that essentially jumps me into Access Options and straight to the Quick

Access Toolbar page. And what I have is here on the left I have all of the commands that are available. And if I click this dropdown at the top and select All Commands that shows me absolutely everything that's available in Access and these commands are listed out in alphabetical order, which does make it a little bit easier when you're searching for a specific command.

Then on the right I have essentially the commands that are currently on my Quick Access Toolbar. So you can see I have Save, Undo and Redo. So essentially all you need to do is scroll through this list and find the command that you want to add to the Quick Access Toolbar. So I'm going to look for Delete Record. And again let's scroll down to the D's and there we go, Delete Record. Select it, click the Add button to move it to the righthand side. Now if you're not particularly happy as to where this has inserted you can change the order of these commands just by using these up and down arrows. So in this case I'm going to move Delete Record all the way down to the bottom. So this is all very easy to do.

In the same way you can remove commands. So again if I decide I didn't want this Delete Record just make sure you have it selected in the Quick Access Toolbar section and click on the Remove button just to get rid of that.

So very, very simple just to customize that Quick Access Toolbar.

It's also worth noting that if you do want to add anything to this toolbar if you can see it on the ribbon, for example let's jump to the Home ribbon, if I wanted to add Cut I could right click and select Add to Quick Access Toolbar and that will be added onto there as well. So a few different ways to add commands.

Now when it comes to customizing the ribbon there are a number of things that you need to be aware of. So I'm going to start by going into the relevant page in Options and I'm going to select Customize Ribbon. Now you can see here under Main Tabs I have actually eight tabs which are currently checked. However if you look at my actual Access ribbons that are showing I only have one, two, three, four, five of them there. So there's a bit of a discrepancy and I'm going to come back to this in a moment.

So with these ribbons you can see I have the Home ribbon expanded and then underneath I have the different tabs. And you can see there it says Views, Clipboard, Sort and Filter. And again and we are currently clicked on the Home ribbon you can see that those correspond with the group name. So Views, Clipboard, Sort and Filter. If I expand Clipboard, for example, you can see I have the commands that are within that group. So Paste, Cut, Copy, Format Painter, and the Office clipboard. So this hierarchy the tab group command is absolute. So you can't put a command on a tab without there first being a group. So you have to have a group even if it only has one command in it. And you are very restricted in terms of being able to change these default groups and tabs.

Now if at any stage when you are making changes if you want to just reset back to default and disregard all of your changes there is a Reset button at the bottom which will do that for you.

Now one of the things that you can do is to create your own tab with your own groups and commands. So let's go back to looking at these eight main tabs or eight standard tabs.

You can see that one of the standard tabs is showing me Print Preview. But I don't have that one showing in my Access window. So let me just cancel out of this and I'm just going to go to a report in the navigation pane. So we have this one here report countries. I'm going to right click on it and I'm going to select Print Preview. So all I have now is that one Print Preview ribbon. So that is a standard tab in Print Preview mode. So as with any tab at any time which commands are available will depend on the status of things. So for example, I have an Excel button available here, I have a Text File button. So currently when I'm in Print Preview I am not able to show the margins but I am able to utilize all of these other commands. So let's close Print Preview and we go back to our standard five tabs.

Another important feature is if I were to deliver this Esprit de Tour database to a number of users and I had customized the ribbon that doesn't mean that the users will get the customized version of the ribbon when they open this database. And I will show you how you can send them a customized version of the ribbon a little bit later on.

So let's start by doing a couple of pieces of customization. Again I'm going to go up to file, down into Options. And we're on Customize Ribbon. Now if I wanted to just hide the External Data tab all I would need to do would be to come into here and uncheck that box. And if I click on OK you'll see that a lot of the time you will get a message that says that you must close and reopen the current database for the specified option to take effect. Now in my experience this isn't always true. So if we click on OK. There you go. You can see that that External Data tab is no longer visible. So if I want to deliver this as a customized ribbon users wouldn't be able to see the External Data tab.

Now in order to demonstrate and explain some of the other options I'm going to create a new custom tab. So back into Options and at the bottom here I'm going to click on New Tab. And the new tab is created and it has a default name of New Tab. And then underneath it's automatically given me a new group with the ever so insightful name of New Group. Now obviously these are customizable and we're going to change those.

So let's change the name of the tab first of all. I'm going to click on the Rename button. And this is just something that I like to do is that when I'm creating a custom tab so that I know that it's a custom tab and it's very obvious I like to have the display name in capitals. So this tab is going to contain commands that we would use for browsing. So I'm going to call this and click on OK. Now I'm going to go to this Home tab and this Sort and Filter group just here I'm going to move it to this new tab. So all I need to do is drag and drop. And if I now click on OK just to see what that looks like, and again we're just going to click just here, you can see I now have a new tab.

When I click on it I have that Sort and Filter group. And if we go back to the Home ribbon you can see that we no longer have Sort and Filter on there because we moved it to the Browsing tab.

Let's go back into Options, I'm going to expand my Browsing custom tab, and where I have this new group I'm going to rename this and this group is going to be my Find group. And this time I'm going to add a command to it from my All Commands list. So again if we scroll down to the F's and there we go, there is Find, and I'm going to click Add to add that to my Find tab, and click on OK and OK again. Let's go back to our browsing and you can see I have a group called Find. So we're starting to build up a custom ribbon.

Now when it comes to this customization there are rules about what you can and can't do. Now I moved Sort and Filter and you can move a whole group from a standard tab but you can't move individual commands. Now when you have your customized ribbons looking as you expect, when you are happy with them, you might want to distribute them to your users. And again if we jump back into Options and in that Customize Ribbon page at the bottom you have an Import and Export option. Now if someone had sent me or wanted me to use their customized ribbon I could choose Import to customization file but in this case I want to send my customized ribbon to someone else so I'm going to choose to export all customizations. I would then go through and choose a folder and you can see the file name here, Access Customizations, and the file extension is Exported UI. So you can use that on machines that are going to use this database. And if you want them to have the ribbon that goes along with it you can use this option so that they can do that as well.

So that's it on customizing the Quick Access Toolbar and also creating a new customized ribbon. I will see you in the next section.

Section 9 – Visual Basic for Applications

Video: Convert Macros to VBA; VBA Editor and Property Windows

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to talk about converting macros to VBA and utilizing the VBA Editor and the Property windows.

Now one thing you may have noticed is that even though I have closed down the database that we were using in the preceding section I still have my customized ribbon showing at the top there. So you can see that it's not actually part of the database or whichever database you have open. The customized ribbon will always be there unless you go in and remove it.

Now what I'm actually going to do here is I'm actually going to go back and just reset all of my ribbon settings just to get back to those standard tabs that we have that at the top. So join me in a few moments after I've done that. So back to the main content for this module.

Now this is the first of a few sections on VBA. And if you're not aware of what VBA is it means Visual Basic for Applications and it's the programming language for Access 2019. Now as I pointed out earlier if you don't see yourself as a programmer, either you don't have any interest in it or you don't feel you're suited to programming I would still say you should persevere a little bit with these modules because it's really useful to know. So we're going to take the next few modules fairly steadily so that if you are new you have the maximum chance of being able to follow what's going on.

So first of all I'm going to start out by introducing you to VBA, we're then going to talk about the differences between VBA and macros, we're going to look at another example of Hello World, and we're going to look in detail at the VBA Editor and the Properties window.

Now a word of warning. Maybe you've used VBA in Word or Excel and there are similarities but there are also significant differences. So even if you have used it previously in another Microsoft application it's probably still a very good idea to work through the first few modules on VBA. On the other hand if you've used VBA in Access before and in a fairly recent version you could probably afford to skip the first two or three modules.

So what is VBA? Well it's a version of the basic programming language, which is well established and easy to learn. And there are various reasons you might use it. So it can help you automate repetitive tasks and it can most certainly automate and perform tasks a lot more quickly than you can.

Now for a lot of the time when you're working in Access you don't really need to be able to write program code. Access does a lot of the hard work for you. And there are lots of features that don't require you to write your own code. But for almost all developers there comes a point in time where you might need to write your own code. So for example, Access might do something in your database and you would prefer it to do something slightly different and the only way you can get it to do that is to write code.

I’m going to turn now to the question of why we have both macros and VBA in Access.

Well you can do extra things with macros and you’ve seen how to display messages and we also spent some time looking at AutoExec. So you can do many, many things with macros. There are however some things that you just can’t do and with VBA you can do pretty much anything that’s possible. So what generally happens is that people learn the basics of macros, they start to use them to perform tasks but then they get to a point where there is something they need which macros can’t do and it’s usually at that point they revert to using VBA or learn about VBA. So in my opinion it’s definitely worth knowing macros and VBA but ultimately when it comes to more complex procedures the VBA language is more likely to provide you with a solution.

Now one other point to bear in mind, if you write your own code in VBA to some extent it’s more transportable. So if you develop a database and later it needs to be converted to something like a Microsoft SQL Server database the logic at least in your VBA code is more transportable to say an SQL Server database, whereas the macro language is a little bit more specific to Microsoft Access and therefore not very transportable. So let’s start out by making some VBA code and in fact we’re going to let Access make it for us.

So I’ve created a copy of the Hello World message box macro in this new database, which you’ll find in the course files folder. So let’s jump into Design view and remind ourselves what this looks like. So we can see the message displayed is Hello World. Again there is no beeps or no notification when that pops up. The type is Information and the title we’ve just put, This is generated by macro code. So let me close this again and I’m going to double click to run it. And there you go. Our message box has come up with all of those elements. Let me click OK and just jump back into Design view again. And if you now look at the Design tab on the ribbon one of the options is Convert Macros to Visual Basic. So what we’re going to do is we’re going to convert this macro.

Now I have a couple of options that I can choose from here. Now Add error handling to generated functions you will see what this is in just a moment but I’m going to leave that checked. And I’m also going to include any macro comments. So let’s try converting it. So notice we get a message letting us know that our conversion has finished. And if I click OK the VBA code has been created and we’re now looking at something called the VBA Editor.

Now the VBA Editor is going to take quite a bit of explaining but if you look at the tree structure you’ll see the name of the database and you’ll also see modules, converted macro. Now if I right click on my converted macro and select View Code you can actually see the VBA code that generates that message box. And there’s all sorts of things around it. So don’t worry about anything at the moment apart from the thing about halfway down the page where it says Message

Box. So you can see here it says Message Box, Hello World again enclosed in quote marks, comma, we then have VB information, so that's telling us the type, and we then have the title again enclosed in quote marks, This is generated macro code. So even if this is the first time that you're seeing VBA code that one line is reasonably simple to work out.

So just as our first foray into editing VBA code let's just make one simple change. So I'm going to make a change to the title here. So it says This is generated by macro code. I'm just going to say This is generated by VBA code. Now what I'm going to do is just test that that is working. So I'm going to run the macro essentially. Now to run it from here all you need to do is go up to your toolbar, and this is very much one of the old school style toolbars, and you'll see just under where it says Run you have a little Play button. And if you hover over it it says Run Sub/User Form. And there is a shortcut key for that of F5. So let's click Run and there we go. And you can see that that title has actually changed to This is generated by VBA code. So I can see that that is actually working correctly.

So what we essentially ran there was the VBA code instead of the macro code. If we jump back into the Editor, now the VBA Editor comes as a little bit of a surprise to some people as it doesn't use the standard ribbon interface that we're used to throughout all the Microsoft applications. It uses a more old school style conventional menu system. Now over the next few modules we will be looking at many aspects of the VBA Editor.

Now the first thing I'm going to do is just jump up to this View menu and you can see in here that there are various different windows that you can show or hide and also a selection of toolbars. So if I hover over Toolbars you can see that I currently have the standard toolbar showing. And what I'm going to do is I'm just going to switch that off to get rid of it. Okay? So if you find that you need a little bit more space, you want to turn those toolbars off, you can easily do that through View, Toolbars and turn it off and on as you like.

Now a window that can be useful is the Properties window. So if I jump back up to View and go down to Properties Window you'll see it appear in the bottom left hand corner. And this will basically show the properties of whatever I have selected in the project window at the top here. So this window at the top is referred to as the Project Explorer. And for whichever VBA project we're working on we'll see the components of that project displayed in here. So you can see at the top we have a component called ACWZ Tool and I really don't want you to worry about that at the moment. You can just ignore it. Then we can see the name of our database, so that's the database I currently have open. And within that database the component that the VBA Editor is interested in. And these are the code components. And they're essentially arranged into modules. So VBA is arranged into VBA modules and typically in a project you may have one module with loads and loads of code or many different modules each fulfilling different purposes. So let's start with this first module here.

Now I don’t particularly like the name of that because I use a standard naming convention. So I’m going to rename this module. So I’m going to go down to the properties window just here and you can see I can click in here and make changes to the name. So I’m going to change the name of this module. And my VBA modules always begin with the prefix BAS. So I’m going to change this to basHelloWorld. So there we go. I’ve created my first VBA module and renamed it.

Now when you’ve finished working in the VBA Editor you can simply close it using the cross in the top righthand corner and it will jump you back to your database. And if I close the macro version of Hello World and if you look in the navigation pane you can see that I now have a section in here which is showing my VBA code or my new module, basHelloWorld. So if I want to edit my macro I can click on the macro, right click and jump into Design view. And if I want to edit my VBA code if I click here, right click and open in Design view it’s going to reopen that VBA Editor.

So that’s got us started on VBA. As I said, I’m taking it really slowly. In the next module we’re going to learn a lot more about VBA and using the VBA Editor so I will see you over there.

Video: VBA Modules; Running Code and Error Handling

Deb: Hello again and welcome back to our course on Access Advanced 2019. So so far we've been looking at VBA and we started taking a look at a few more details about the VBA Editor. So we're going to start out this module by taking a look at a few more details about the VBA Editor, then we're going to look at some VBA modules and I'll explain aspects of running code and also compiling code, and then finally we will start looking at branching and error handling.

Now in the preceding section I showed you how to open the VBA Editor by going into Design view for a module, but you can actually open it by jumping up to the Create tab and in the Macros and Code group on the end here clicking on Visual Basic. And I pointed out to you that it comprises a number of windows of but has many other features. And you can see here we have at the bottom something called the Immediate Window. And of course you can close any of these windows just by clicking on the cross in the corner. So let's close down that Immediate Window for now. And if I wanted to show that immediate window again I can go up to my View menu and I have Immediate Window and that will bring it back again.

Now all windows can be resized as well. so if you just hover over the edge until you get that double headed arrow you can drag in and you can drag out again. And you can also do the same thing vertically as well. So I can drag up and down. So you can really resize these to whatever suits you and what you're working on at that time.

You can also move these windows around. So if you're not particularly happy with the way that they're laid out, so for example if I wanted to move this Properties window which is currently in the bottom left hand corner if I just hover my mouse over the heading, click, I can then drag it around. And I can drag it all the way over here and I could drop it if I wanted to. And then I could resize it, so on and so forth.

Now another thing to point out when you're moving these windows around is that some of them are what we call dockable. So again if I pick up this Properties window and I was to move it over here it's just going to be a floating panel. If I wanted to display it so nice and neat like these other windows you'll see if I drag this and kind of hover over the dividing line I get this kind of ghosted outline which means that when I let go of my mouse it's going to dock it in position. So it's not going to be a floating window on the page. It's going to dock it to this particular layout. So you'll see this quite often when you drag it around to different places. So if I was to drag it up to the top of the code window you can see that if I was to let go it's going to dock it above my code window. I could also drag it down. And again I could put it above the immediate window. And if I drag it back to its original position, if I drag down the screen until I get those lines again you can just see it right at the bottom, that ghosted line, and let go it's going to dock it back in its original position.

Now we've looked at a few different windows here but there is one that we haven't looked at. I'm going to jump up to View and I'm just going to show you the Object Browser. And again you can see this is a little floating window. And you'll find that you can use the Object Browser for lots of different things. And it is a really good way of finding out what the structure of various different VBA statements is. Now I'm not going to go into detail with this now but one thing I do want to point out is as with most other windows you can see a Close button in the top righthand corner. So if I wanted to close the Object Browser I can just click Close.

Now if you go to the Tools menu and go down to Options you'll see that one of the tabs that you have there is Docking. And this tells you which windows are dockable. And you can see that the Object Browser is one of them. Now I'm going to untick that so you can see the difference. And now if I go back to View and select Object Browser you can see it doesn't appear as a floating window with a Close button in the corner. It actually kind of replaces my VBA code window. So just be aware of that. You can choose which windows are dockable. So by now you should be kind of getting a reasonable idea of the layout of the VBA Editor.

So now I want to talk about VBA modules. In Access, generally speaking, the modules associated with a database have a hierarchical structure. So at the top level they're generally divided into two groups, Class Object and Standard Modules. Now class objects contain the code associated with a particular object in a database. So let's suppose I created a form, maybe a form for maintaining information about movies, and I wrote some VBA code that was attached to some of the buttons on that form. Now that code would be associated with that form and therefore would be considered class object code. However, the second type of code is the sort of code we've got here and it's code that's not really associated with a particular form or report. It's what's called Standard Module Code. And at the moment within this database we have one standard module, basHelloWorld.

Now within that module we can have a number of pieces of code and in this module we have one function and the function is called mcrHelloWorld. It begins with a function statement and ends with an end function statement. So everything from Function to End Function is one VBA function within vasHelloWorld. And in fact this module could have many different functions in it.

Now we're going to take a look at that function in a moment but before I do you'll see there are other things within this module. Starting at the top we have a couple of what we call declarations. One of them says Options compared database and one says Option explicit. So for the moment all you need to do is make sure you have those in your modules. So let me briefly explain both of these to you.

Option compared database says that within this code whenever it comes to comparing strings to characters, so supposing I wanted to sort data into alphabetical order it will use the same comparison method that's used in the database itself for the particular thing I'm trying to sort.

Now you don't need to worry about this at the moment. And as your knowledge of VBA increases there'll probably be times when you'll need to look at the consequences of that, but all I need you to do right now is make sure you have that statement at the beginning of the module.

The second statement, Option explicit, says that if you're going to declare variables within your program code you must declare those first. You must tell Access VBA what those things are. And again I will explain that a bit more later on.

Now below those two declarations we have comments. So where it says mcrHelloWorld in green, anything you see in green that has a single quote at the beginning is a comment. And by putting that single quote first that will make whatever you type after it essentially a comment. And comments are kind of dormant in modules. They don't run anything. They are purely there for informational purposes. And I would highly recommend that if you are writing your own VBA code that you do throw in quite a few comments here and there so that people can understand what your code is doing.

So basically within this module we have some declarations at the top and then we have a comment that indicates the name of this module. Now bear in mind that this isn't the actual name of the module because we changed that earlier. So I'm going to jump in here and I'm just going to change that from mcrHelloWorld to basHelloWorld. And what I'm going to do is I'm just going to put some more useful information in there. So another comment.

Now another thing you can do is have empty lines. So Access won't do anything with those. So don't worry if you've got empty lines in your code as well.

Now the rest of this module is essentially a function. And there are other things that can appear in modules. So notably subroutines, and again we're going to talk about those a bit later on. But let's now look at running code.

Generally speaking your code will be run in response to events such as people clicking on buttons or opening forms, etcetera. But if you just want to run a particular function there are a number of ways to do that. So you can place your cursor inside the function and use a button, the Play button. So again if we hover up, and you can see as we hover over it says Run sub/user form. So let's run that function. And there we go.

Now let's suppose when I was writing this function as I had made a mistake and I had forgotten the correct command for message box. So I'm just going to come in here and change this and misspell it essentially. Let's see what happens now when I click on Play to run that function. I get a compile error. And what Access is saying is that the message box function is not correct. It doesn't recognize it so it fails to compile. So just bear in mind that when you click Run the first thing Access does is try to compile the function. And compiling means that it takes the word you have there and it turns it into its own machine code. When it does that if it sees something it doesn't recognize it will give you a compile error. So let me just jump back in and correct that and let's run it again. And there we go.

Now one way of avoiding that kind of nasty surprise is to compile your code before you run it. So let me show you a different sort of error. So I'm going to jump into my code and I'm going to type in Today = Wednesday. So what I'm trying to achieve here is that it should display Wednesday. And what I'm going to put in here is instead of Hello World again, I'm going to delete that out, and I'm going to put Today. So that should display Wednesday, and this is of course generated by VBA code. Now before I run it I'm going to compile. So I'm going to jump up to the Debug menu and I'm going to say Compile. And what VBA complains about is that the variable is not defined. It doesn't understand what Today is. Well that should make you remember that option explicit that we discussed earlier. Option explicit means that I will declare any variables before I use them. So what I should really do here is say Dim Today as String. So what this is actually saying is that I'm introducing a variable called Today and it's a string variable. And remember String is a character string. So when I run this what it should say is that Today is Wednesday and I'm going to include it in my message box statement. So let's compiling again and this time no error message. So I'm going to run it and there we go. Wednesday in my message box.

So that shows you the advantage of compiling. You'll pick up errors before you run the code.

Now the last thing I'd like to look at is branching and error handling. Now with this function the first thing in it is a statement. So it says On error go to mcrHelloWorld_Err. And mcrHelloWorld_Err is a label within the function. And you can tell it's a label because it has a colon after it. So what this says is that if at any point an error occurs when the function is running go down there. And what happens there is that we display a message box Error$. And Error$ will be what the error that has occurred is. So having displayed the message box we then resume at mcrHelloWorld_Exit and that's another label. So that's the last thing it does, it exits the function. So if no errors occur when running this function the condition is not triggered. So what essentially happens is this. Dim Today as String, today is Wednesday, display the message box, the user clicks OK, and then we go into mcrHelloWorld_Exit. So what happens in this function is that if an error occurs we branch off into part of the function that handles errors. But whether it occurs or not ultimately we end up in the same place. We end up exiting the function.

Now I'm going to show you more about branching and error control over the coming modules. That's it for this module. I will see you in the next one.

Video: Creating Modules; Subs, Functions, Intellisense and Variables

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to continue looking at VBA and rather than use code that's being generated or converted for us we're going to start to write our own code, which is very exciting.

So I'm going to start by creating our own VBA module, I'll then explain the difference between subs and functions, I'll introduce you to Intellisense, and then we'll talk some more about variables. So let's start out by creating a VBA module.

Now as I explained before there are two types of modules, class modules and standard modules. Now class modules generally relate to forms or reports and Access generally creates these for you and within these modules you keep the code and that mostly corresponds to events related to those forms and reports. For example, there may be code in the class module for a form that shows what happens when someone opens a form or clicks a button. Now we're going to concentrate on standard modules and in general terms the code in standard modules doesn't necessarily relate to a particular form or report or to a particular event. And again in general terms the code in standard modules is accessible from anywhere within Access.

So let's jump up to the Create tab and then if we look all the way across in the Macros and Code group we have an option here for Module. Now as soon as you do that the VBA Editor opens and your module is created. And note that your module has a default name of Module 01. Now that's not particularly meaningful so the first thing I'm going to do is go in here and change the name of this module. And you should remember how to do this from the preceding section. So if we jump down into the properties area I'm going to rename this to basTest01. And you'll see when I click anywhere else it renames that in the project explorer.

Now as I pointed out earlier on if we jump into Tools and Options again we have some editor options. And one option that we have in there is Require variable declaration. And that's currently checked. Now if I uncheck that and click on OK what I'm going to do is I'm going to go in and I'm going to create another module. Now if you're in the VBA Editor window, so you don't have to close this down and go back to Access to create the module, you can just right click on one of the modules, go down to Insert and select Module. And you can see now I have Module 01 again. But note that this is slightly different. There is no longer an option explicit at the top. If we just jump back to basText01 for comparison you can see that has option explicit. So that is what that little checkbox controls.

Now it's worth pointing out that whether you've selected that checkbox option or not you could just jump into here and type Option Explicit manually if you wanted to. Similarly if you had Option Explicit in here and you wanted to remove it you just could come into the code and delete it out again.

Now what if I want to delete a module. Well if I select Module 01 and right click I see my contextual menu. And you see I have an option here that says Export File. Now if you've written a load of code and you think that you don't want it anymore you can export it instead of deleting. So if you think to yourself, "That's a lot of code. I don't feel comfortable deleting it. I might possibly need it at a later date." You can choose to export it out to a file. However if you want to remove it you would go to the Remove Module 01 option. And what it will say is it will then pop up a warning and ask you again if you want to export this module before removing it. Now in this case we have nothing in here so I'm going to say no I don't want to export and that module has now gone.

Now something else I mentioned earlier on, code can be arranged in modules in units, which are either subs, which is an abbreviation for Subroutine, or in functions. And there is an important difference between the two.

When you put code into a sub and execute it what you're saying to Access is do that for me. But when you put code into a function what you're saying is tell me this. Now the tell me this might involve the code doing something but ultimately the code will be coming back to you and telling you something. So basically with a function you get a return value, you get something back with it. Whereas with a sub you don't.

Now in this first demonstration we're going to create a sub. Now in order to insert a sub into this module I only really need two statements, a sub statement and an end sub statement. And you need to give the sub a name. And there are some fairly restrictive rules when it comes to naming subs. So for example, the name must begin with an alphabetic character. Generally you can't use punctuation characters, although you can use some, such as underscores and I'm going to use one of those in a moment. And the name can't be more than 255 characters long. And what I would suggest is that you try to make all of your subs have names that reflect what they actually do.

So what I'm going to type into this module is subTest_01. Now in a moment note the black type. When I hit Enter what Visual Basic will do is it will look at what I've typed and if it's happy with it it will confirm it. So let me just hit Enter. And you can see what it's done there. So basically it's happy. It's changed the type from black to blue and it did two other important things. First of all it put a pair of round brackets after the name and those must always be there. So it might be that later today you put some parameters in there, but even if you have no parameters you still have to have those open and close round brackets. The second thing it did was to put an end sub statement in because it knows that one is going to be needed. So at the moment that is a perfectly legitimate sub. However it doesn't do anything. So if I wanted to run it I would run it in the same way as I ran the Hello World code but it's just not going to do anything because we don't have anything between our two subs. So let's now make it do something.

Now before we start to add some things in an important thing to note here. Let's hit Enter a couple of times. It's a really good idea to space out your code because it makes it a lot more readable for people.

So now it's time to start writing some VBA code.

Now generally speaking I indent the code to reflect the structure of the code. So I'm going to press my Tab key which is going to give me a first line indent so that the subtext and the end sub stand out from the rest of the code. And I'm going to type in the message box command. And I'm going to hit Space. Now when I press Space Access applies the Intellisense command. So it's recognized the command and it prompts me with a list of parameters. So I need to open a round bracket and my prompt is Hello World. And then after that there is a list of other arguments for the message box. And you'll see that the ones listed there in square brackets mean that those are optional. They're optional parameters. Now all I'm going to do here is I'm going to keep it simple. I'm going to close my round bracket and I'm going to press Enter. And I want to make sure that Access is happy with this so I'm going to compile it. And I don't get a compile error so I feel that I can safely run this. And there we go, Hello World. So that's a very simple piece of VBA code written from scratch.

Now if I wanted to make a change to the module I can just click in it to change it and I'm going to change this to do a little bit of arithmetic. So what I want the message box to display here is the result of a calculation. So I'm going to remove Hello World and I'm going to say 2 multiplied by 3. I'm going to hit Enter and you can see that Access has added in some additional space for me. I'm going to compile, looks okay to me, and now if I click on Run I get the result of six. So these numbers here, two and three, are referred to as constants. They are numbers that don't change. Now generally speaking in VBA we work with the opposite of that essentially which is variables, things that do change. So what I want to do now is to reintroduce the concept of variables.

So I'm going to introduce three variables, A, B, and C. So I'm going to go to the line above message box, I'm going to press the Tab key. So the first variable I'm going to declare is A, then I'm going to press the Spacebar, As, I'm going to press Space, and now Intellisense pops up a list of the possible object that A might be. So I want it to be an integer. So if I just hit the I key it takes me down to I. If I do N, T, eventually I get to Integer. So I can then select it from the list. I'm going to hit the Enter key and say A=2. So I've now declared the variable as an integer variable and on the next line I've given it the value of 2.

Now let me introduce a different kind of variable. So we're going to go to the next line and say Dim B as Integer. And this time B=3. Now C is also an integer but the value of C is the result of A multiplied by B. Now looking at my code I can see that I have my message box in the middle there so I'm just going to take it, I'm going to cut it, and I'm just going to paste it again underneath.

Now instead of message box showing me the result of 2 multiplied by 3 I want it to show me the result of C. Now as usual I could compile this if I wanted to but if I'm reasonably confident that my codes going to work I can just click on Run which I'm going to do in this case. So let's click Run and there we go. It works.

Now throughout the balance of this course when we use VBA we will be using variables a lot and also many types of statements. And two of the most common you have seen here. There are Dim statements where we declare variables and also what we call Assignment statements.

That's enough for this section. I will see you in the next one.

Video: Scope, Variable Naming and Strings

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to look at Scope, we're going to look at variable naming, we're going to look at the use of strings, we'll take a look at some more features of Intellisense, and finally we're going to look at continuation characters.

So what I've done here is I've created a second module and I've included two subs, sub A and sub B. In sub A I've declared an integer variable A with a Dim statement and given A the value of two. And in sub B I've given A the value of three. So let's try to compile this and see what we get. I get an error and it tells me that variable A in sub B is not defined.

Now the declaration of A in sub A only has a scope of sub A. So it only applies to sub A. It doesn't apply anywhere else. Now if I took this Dim statement out of sub A and placed it immediately after option explicit A is now declared for the whole of this module. So its scope is the whole of Test 02. So let's compile again and there we go, no error. So not only does it compile okay but the A that I've declared is the same A in sub A that is in sub B. So they're both referring to the same variable.

Now when I declare a variable at the module level like this instead of using Dim I can use what's probably a slightly more informative title in this situation and that is to call it a private A. So let's change that. I'm going to put a bit of space in here and I'm going to change Dim to Private. So what we're saying there is that although it's known throughout this module it isn't known to other modules. So for instance, if I went back into Test 01 and what I've done here is change the sub there so that it's got it's own A with a value of one this A will be different to the one in Text 02. So let's just make sure that this will compile okay, which it does.

Now let's suppose that what I really wanted to do was use the same A for both of these modules. So let's suppose I want an A to apply throughout my project. What I can do is this. I can declare A as a public variable. So I'm going to remove this, I'm going to go back to Test 02 and I'm going to change Private to Public. Now Public A as an integer means that I have declared an integer variable called A and it's public, therefore it can be accessed by any module in my project. So let's compile. And again it compiles correctly and each of the subs that is accessing this this A is accessing the same A. and the scope in this case is the project. So you can have a scope of project, the module or the sub or function.

Now another important factor is that not only variables have scope. Subs can have scopes as well. So let's jump back into that first module and instead of only applying one to the variable A here what I'm going to do is display a message box. So let's run it and there we go. So that's the sub Test 01. Let's go back to the second module and what I'm going to do is I'm going to try and call that sub that's in the first module from the second module. And all I need to do to do that, and let's just remove some of this spacing in here, make it a little bit neater, is I type Call Test_01. So let's try running that. And there we go.

Now you'll notice that I left some brackets out just here. So I would only really apply brackets if I had some parameters that I wanted to call, but in this case I didn't so I'm perfectly fine just to leave those brackets out.

Now let me jump back into the first module again. Now if I change this to say Private Subtest 01 that means that the scope of this sub is just the first module, which means that only code in the first module can access that sub. So now let's jump back into Test 02 and try and run it. And now I get an error because it can't find Sub 01 because although it's still there it's now private to that first module. So of course what I can do is I could jump back into Test 01 and I could either remove the word Private or I could change it to Public. And if we now go back into Test 02 and click Run it runs perfectly fine. So as you can see you can have scope for subs as well.

Now the next thing I'd like to look at is the naming of variables in Access VBA. Now you've already seen I tend to prefix with short sequences of characters that describe what the objects are. And commonly used naming convention used throughout IT is called Hungarian Notation, which was created by this guy called Leszynski. And if you look at the Wikipedia page and scroll down you can see we have all of those standard prefixes. So some of these we've used throughout this course. So we have FRM for Form, MCR for Macro, BAS for Module, so on and so forth. So a lot of people who use Access use Hungarian Notation.

In terms of the various restrictions on names, such as the length, punctuation, etcetera there are quite a few links that you can get to on the Microsoft support site that gives you guidelines for naming. But it is very important to make your names informative.

Now what I'm going to do is make a quick change to those modules that I've just been demonstrating. So in reality I wouldn't really call A just A. I would give it the name intA. So let me just change that wherever it's used. And although I tend to use BAS for the prefix on my modules a lot of people do use MOD, M-O-D. So it's entirely up to you which you use. And if you have any local standards then I would advise you to follow those.

Now as you'll see as we work through there are many different data types in VBA. So what I'm going to do here is just demonstrate another one. So I'm going to say Dim strprompt as, and again we get our list come up. Now this is a list of data types. It's not only data types in here but you will find all of the data types as well. And the data type that I want to use is String. Now string data type is the one we use to hold character strings. Now in the case of a string variable we assign a character string to it and in this case what I'm going to assign to strprompt is this. Now notice that the string constant is enclosed in double quote marks. And I'm going to introduce another variable and this is going to be an integer variable. And I'm going to give this a value of 12. So now what I'm going to do is in my message box is I'm going to put these two strings together using the concatenation character. Now strictly speaking what I'm going to do here is a little bit cheeky. It's not quite right but let's do it anyway.

Now concatenation if you've never used it before is essentially the ampersand symbol and it's what you can use to connect two strings together. So let's modify our message box.

Now what I've got there in the argument of the message box, one thing, but it's one thing that's formed by concatenating two other things, strprompt and intMonthsInYear. So let's compile it. And that's a good sign. I get no compile error so let's run. And it runs. So even though I've cheated a bit there it does still work. So you've seen there the use of strings and concatenation characters. And there are many, many useful tips and tricks surrounding this but I really don't want to bombard you at this stage.

One of the others I want to look at relates to Intellisense.

Now remember earlier when we typed in Message Box and we were given a list of arguments so let's suppose now that you couldn't remember the command for Message Box. So if I delete that out and you just type in M nothing happens. But if you hit Control-Space you get a list of the available commands starting with M. And you can then go through and scroll through this list. And once you've found it double click to select it. So that is there to help you, Control-Space.

Now one other thing you're sure to need is the ability for commands to occupy more than one line. So depending on how you have the editor arranged you may not have enough width to show all of your commands. For example, let's just move this window in, make it quite a bit smaller. And let's pretend that our strprompt is quite a lot longer than this. So as you can see I've now got a long prompt which doesn't quite fit. Now something you can do here is you can split it onto a second line.

So first of all you need to choose ap lace in your string where you want to essentially break it onto another line. And I'm going to do that just after the word Year. So the first thing I need to do is just enclose that in quotes and then I put in a space, underscore. And what underscore is is it's a continuation character. So it means just continue this on the next line. And then what you might want to do is indent it so that it looks a bit neater, it's a bit more lined up, you can do whatever you like with that.

And in order to make it pick up where it left off I need to put in the ampersand or the concatenation character. So essentially what I'm doing here is I'm basically concatenating two separate strings together. And remember your second string needs to be enclosed in double quotes like so. So now by using that method it just makes it a lot easier to see if your window is either quite small or if you have very long strings. So it's worth having a little play around with that as well.

It’s worth noting that you’re not restricted to using continuation characters like this. You can split up any lines of code in this way.

So that is the end of this module. I will see you in the next one.

Video: Standard Functions; Logical, Case and Loop Statements

Deb: Hello again and welcome back to our course on Access Advanced 2019. This is the final introductory module on VBA and after that we're going to be using VBA in some Access applications. So in this module we're going to take a brief look at VBA standard functions, we're going to look at conditions and branching, we're going to start to delve in to case statements, we're going to apply looping with For Next, and finally we're going to explore the options for Do Loops.

Now you've already seen an example of a couple of functions in the course and you can write your own functions to do all sorts of things, but Access VBA comes with a whole raft of standard functions that can do lots of different things for you.

Now generally speaking with these functions to find the function you want you can simply look it up using Help. And in this section I'm going to look at some examples of Date and Time functions.

Now the very simplest Date and Time function is a function called Date. And if in this message box statement I just type Message Box and Date, Date is not a variable. It's actually a call to a date function. And because there are no arguments on the date function I don't need to put in any brackets. So let's try and run that. And as you can see the date is November the 28th 2019.

Now let's look at a different date function. There is a function called Weekday and this function tells us which day of the week a specific date is. So I'm going to delete out Date and I'm going to replace it with Weekday. Now remember that shortcut key as well. If I can't quite remember what this function is called I can do Control-Space and it will bring up that Intellisense list. And I can see just there the one I want, Weekday.

Now you can see Weekday has two arguments. The first argument is the date I want to test. So for example I could put in today's date to get that. And then I need to specify the first day of the week, whether that's Sunday, Monday, Tuesday, etcetera. So let's deal with that first argument first. So the first argument is the date that I want to test. So I could use today's date and to get that I just need to type in Date. So let's now type in a comma. Now note, once I've selected my date which is the function Date what's highlighted now in the list of arguments is the second argument. But I'm given a list of different options for that second argument. So from here what I would do would be to select the first day of the week. So I'm going to select Monday. And I'm going to close the bracket. So what I'm essentially saying here is message box, display the day of the week for today, counting Monday as the first day of the week. So shall we try and run this and see what happens? And there we go. It gives me a result of four because today is Thursday. So it's counting Monday as one all the way through to Thursday. So that looks like it's working pretty well.

So what I'm going to do now is to introduce some conditional logic into our sub. And this logic will check what day of the week it is and then display a message if it's a working day. So what I'm going to do here is I'm going to put in a new line before our message box and I'm going to type in If. And then what I'm going to do is check the value of the weekday function and if the value of the weekday function is one, two, three, four, or five then it's going to display a message to say that this is a working day.

Now at this point you might be tempted to think, "Well we could just take Weekday and cut it but you get a small problem here and that is as soon as I click away from that If statement Access is not happy because it's not a valid statement. It actually checks my code as I go. So it recognizes this isn't a valid statement because currently it just says If and I've gone and clicked somewhere else.

Now you do have a couple of options here as to how to get around that. One is just to put up with it and cut it if that's what you really wanted to do. And a second option is where you stop Access from checking what you are typing as you go. Now in this case we're going to stick with that first option. So I'm just going to click on OK just here, I'm going to Cut using Control-X and I'm going to go up here and paste that into my If. Now in order for this to work out if this is a working day or not I just need to check is the weekday is less than or equal to five. So I'm just going to put in here Is less than or equal to 5. Then if it is display a message box that says it's a working day today. And then finally I'm going to put an End If statement in order to terminate this particular construct. So let's try and run our piece of code. And there we go. It's telling me it's a working day today and as today is Thursday that appears to be completely correct.

Now we're going to develop this logical construct further but before we do let's look at that option B I was speaking about earlier. If you go up to Tools and then down to Options the first one in the Code Settings group says Auto Syntax Check. And I normally have this turned on and that's the thing that causes those messages to pop up related to syntax errors. So you can choose to leave that on if you want or if you find them a little bit annoying you can untick that box and it won't check your code as you write it. So that's completely down to personal choice.

Now I'm going to make a change to this piece of code. I'm going to add an Else condition. So let's jump into our code again and we'll make some room for it in here. And we're going to say Else. I'm going to copy this message box line of code and I'm going to paste it and let's just tidy that up, making the indents the same. And I'm going to change the text to say, It's the weekend. So basically what we have here is if the weekday is less than or equal to five then display a message box that says it's a working day today. If it's not display a message box that says it's the weekened.

So what we've done here is to use this conditional statement to make the logic of this sub follow one of two alternative routes. So we've essentially achieved a branch in the logic of our code.

But another point is that you are not restricted to only having two options. So for instance, I could change this code to do this sort of thing.

So what I have now is a significantly more complex logical structure where I'm checking the weekday for values one to five and displaying the appropriate day of the week. And if it's not in that range I just say it's the weekend.

Now there is an alternative way of doing this using case statements. Now really a case statement is a generic term for this type of statement. In VBA we actually call it a select case statement. So what I'm going to do is I'm going to go up to the top of this sub, I'm going to Tab across and I'm going to put in a select case statement at the beginning. So we're going to say Select Case and then we specify the expression we're going to check. And in our case that is, and the end of the select case statement is indicated by an End Select statement. And in between we can have any number of case statements. So let's put one in.

So what we're saying here is that we're testing the value of that weekday function. And if it's one we display the message box It's Monday. So let me put in another couple of options.

Now notice how I can list the options such as Case 6, comma, 7 there, or I can specify a range such as Case 2-4. And I can sort of have a catch all as well, like this. Now of course we would expect the weekday function to return a value in the range of one to seven but I just put a message in there to say something horrible has happened, which is always a good idea when you're writing code.

Now one other really important thing that you should know how to do in VBA is loop, to repeat sequences of steps over and over again. And there are two main ways of doing this in Access VBA, although in both cases there are some variants on those two main approaches. The first is what's called a For Next Loop. And for this approach we will use a counter variable to count how many times we are doing something. So I've set up a variable, intcount, as an integer variable. So it doesn't currently have a value. And I've also inserted in this new sub a command message box which will display in a message box the value of this count. So what I'm now going to do is not only do that a number of times but I'm going to change the number of intcount as I do it. So intcount is indeed going to count.

So first of all I'm going to go into my code and let's just make a little bit of room, and we're going to start with For, then I specify what counter I'm going to use, and then I say Equals 1-5. Now having typed that if I hit the Enter key I'm within the For Loop. And there is my message box command. Now what I'm going to do is I'm going to indicate the end of the loop. And the end of the loop will be immediately after the message box command. So let's organize this a bit better. Let's indent message box. And all we need to say here is Next. So what a For Next Loop does is it just repeats whatever is in between For and Next according to the count mechanism you've set up. So this isn't restricted to counting in ones. So I could say something like 1-10, step 2. And that will go from one to ten in steps of two. And as soon as it gets beyond ten it's going to stop. So let’s run it. So we have one, three, five, see it’s going up in twos, seven, and this will probably be the last one. I would expect when we go on to eleven because it’s after ten we’re going to exit, which we do.

Now the other type of loop is a Do Loop and there are really two versions of this, each with a few variants. There’s a Do Until and a Do While. And I’m going to demonstrate a Do While and I’ll leave Do Until for you to look up in your own time.

So the first thing I’m going to do here is I’m going to give intcount a value before we start the loop. And instead of this For statement I’m going to put in a Do statement. So we’re going to say Do While intcount less than or equal to 10. And for Do While the code that terminates this isn’t Next. It is Loop. Now an important difference here is that we have to manually increment the value of intcount. So having displayed intcount the first time with the message box command what I would need to do next is intcount equals intcount plus two. And this will have the same effect as the For Next does. So let’s give it a whirl. There we go, number one, number three. So again it’s still going up in twos and there we go, we exit. So see how you go with that and maybe have a go at Do Until. I will see you in the next module.

Section 10 – Access Events Model

Video: Adding Events to Assets Database

Deb: Hello again and welcome back to our course on Access Advanced 2019. Now that you have a basic knowledge of VBA it's time to start to look at Access Events and put that VBA knowledge to good use. So the first thing we're going to take a look at is just an introduction to Access Events, and as an example of handling events we're going to look at a splash screen with a timer, we'll look at class modules, and finally we'll look at the various Access builders such as the macro builder and the VBA builder.

Now when you're using an Access database there are probably hundreds of events occurring behind the scenes whenever you are doing a variety of different things, be it moving around, clicking buttons, adding records and so on. Now when you build an Access database for most of these events there are default actions which are added automatically by Access. And when a button is clicked an action happens. However, there will be occasions when you'll want it to do something other than that default action. So now that you have the ability to write code you can affect what happens when those events occur. So to get started we're going to start with one or two unusual types of events and then later on we will look at handling some of the more normal things that happen in Access.

So for this we're going to be using version A01 of the asset tracking database. And if you remember that has a backend and a frontend in one single file. Now I've already created the database and I've made a change to it so let's open the database and watch what happens.

Now notice how that welcome screen disappeared as if by magic. So I didn't close it. Now the reason is because I've added a timer to the screen. And when the timer expires it triggers an event and the event closes the form. So to see this in action let's go into the Design view of the welcome screen. And just note in the property sheet that we have Form selected and then about halfway down we have this on timer and timer interval. So currently my timer interval is set to 5,000 milliseconds, which is essentially five seconds. By default this is zero but if it has a nonzero value then a timer will fire after the specified number of milliseconds. So as I said, that form stayed open for five seconds and then it closed itself down.

Now on timer is the event that occurs and there is an event procedure. And just note the even procedure is in square brackets, which means that it's not a separate piece of code. And I need to click on the ellipses to open up the relevant code. So let's just do that.

Now the code is a single piece of VBA code and you can see the sub on the right there. And it has a default name of Form_Timer. So this is the code that reacts to the timer on this form only. And it only has one line of code but it is a very important line. If you're running code to do specific Access things like close forms, open forms, etcetera and this largely corresponds to the commands available when you write macro code then the way you process those Access commands is by proceeding them with Do CMD, Do Command, dot. So when you see Do CMD-dot it means do this thing and the thing in this case is close. So when the timer fires that subform timer runs and the Access command close occurs. Now you might be thinking to yourself, “Close what? It just says close. It doesn’t say exactly what it’s closing.” The important thing here is that the code is attached to a form and by default what will be closed by the close command is whatever this piece of code is attached to. And this piece of code is attached to the welcome form. So I opened it, five seconds later this piece of code closed it.

Now what I’ve just done there is a little bit silly as this welcome screen contains lots of useful information and some helpful instructional videos. So I wouldn’t normally just want to close that. But if you have something that is there to be shown whilst the database is getting ready or maybe just to display a version number or copyright information, so maybe not something that users need to do, just some information that they need to see, usually a timer is quite a good idea. And this is usually referred to as a Splash screen.

Now if we go back into the VBA Editor and just take a look at Project Explorer you can see that we now have a couple of class objects or class modules. So for the welcome form it has its own class module which contains the code for that form. And what you can see on the right is a private sub. So it’s only a timer for this form. And you can see that sub, but in fact I could have many pieces of code associated with this form.

Now if you look above the option compare database you have a couple of dropdowns. So let’s click on the one on the right. So if I want to react to all sorts of possible events for this form then I can choose which event handler to create over here. So let’s suppose I want something to run when I open this form. And what Access does is it creates a form open event procedure and that code. So whatever I decide to put in there, I can put any reasonable to execute in there.

Now if I insert a procedure like that but think, “Hmm, I don’t know what I want to do when the form opens” you can just leave it there because it doesn’t do anything so it’s not going to do any harm. But I tend to clear them up. I like to declutter. So in general I will usually delete any empty subs that I might have.

Now if on the other hand you want to attach some code to something else on the form, so not the form itself but something else, if you use this dropdown on the left you get a list of the things that are available for you to add event procedures for. And I’m going to add an event procedure for one of the other objects on the form. So I’m going to leave all of this as it is. I’m going to close down the VBA Editor and just jump back to the form. And what I’m going to do is I’m going to go over to this property sheet and I’m going to change the timer interval from 5,000 milliseconds to 20,000. So we’ll give ourselves some time. And I’m going to select a piece of text on the form. So this one just here, Feel free to explore. And I’m going to add an event procedure.

Now when you're talking about this text what you're talking about is actually a label. And you'll see in the property sheet underneath it says Selection type Label. And there are very few event procedures that can actually be assigned to labels. So you can't put a timer on a label but you can have On Click, On Double Click, On Mouse Down, On Mouse Up, and On Mouse Move. And I'm going to put an On Double Click event procedure and I'm going to use the code builder to do it.

So let's click on the ellipses and we're going to choose our builder and we're going to choose Code Builder. And it takes me back into the VBA Editor. And it's already created the sub for me with the name according to a standard. So it's used the name of the label and then an abbreviation for the double click event. And now I'm going to put my code in there. So I'm just going to close after here, so Do CMD-dot, and notice Intellisense helping me out here. And I can double click for Close. And I'm just going to do a little bit of tidying up just here just to make it all nice and neat. So what I'm basically saying here is that if I double click on that label that will close the form as well.

So let's test it. Let's close the VBA Editor, let's close the form and save our changes, and we're also going to close Asset List. So I'm going to reopen that form just by double clicking it in the navigation pane. So now what I would expect is that if I double click on this label here, Feel free to explore, it's going to close, which it does.

Now the final point I want to mention here is related to the builder again. So let's say that I'm going to attach some on click code, and if we click the ellipses again we have a choice of macro builder, expression builder or code builder. So let's go to macro builder this time and let's do a close form here, close window. And our object is going to be form. We'll just say Welcome is the object name and no. So I'm going to close my macro, I'm going to save the changes, close, save the changes and let's just open up that welcome form again and this time instead of double clicking on that field again I'm going to single click and it closes it down. So that all again appears to be working fine.

That's it for this module. I will see you in the next one.

Video: Exercise 05

Deb: Hello again and welcome back to our course on Access Advanced 2019. It’s now time for Exercise 05. And in order to describe what you need to do I’m going to start with my solution.

Now with Exercise 05 we have a splash screen which will display for ten seconds and then it’s going to disappear, as you saw it just do there. Now there are several differences between this splash screen and the welcome screen we had before. One of them is that it has a timer, so it only displays for ten seconds, and another difference is that if you double click on that splash screen it will close. And when we created the welcome screen before we inserted a picture onto the form. And with this splash screen I would like you to make the picture the same clapperboard picture or your own picture, but I want you to make it the background image for the form.

Now another change I want you to make is that I want you to change the settings table. Firstly, I want you to make it once again part of the backend database. So you’ll need a different backend database for Exercise 05. And the only setting that you need in there is a setting that shows the current version number of the database.

Now let me just open that form again. I want you to display the version number. So clearly as you continue to work on the database you should update that version number each time. And that version number will display on that splash screen. So essentially the user will no longer have the option of displaying that form or not. It will now be a splash screen that always appears.

That’s Exercise 05. I will see you in the next module.

Video: Common Events; Form and Report Events

Deb: Hello again and welcome back to our course on Access Advanced 2019. So far I've introduced you to general concepts related to events and then I set you Exercise 05. In this module I'd like to look in more detail at the events that occur in Access and in some cases where we have quite a complex sequence of events the reference material you need to be aware of so you can check the sequence of events.

So first of all we're going to look at Help on events, we will then look at common events in Access and then specifically at form events, we'll then take a little explore around report events, and finally we look at an example of a form event. However, from now on most of the things we're going to do will be covered as part of looking in more detail at topics such as forms.

So let's start by looking at the most common events that occur when you're using an Access database. And there is a page within Help which sets out the order of events for database objects, and this applies to all recent versions of Access, including 2019. So I want to start by looking at the first paragraph on this page. So it says Order of events for database objects. A single action such as moving from one control on an object to another control can trigger several different events which occur in a particular sequence. Knowing when events occur and in what order they occur is important because it can affect how and when your macros or event procedures run. For example, if there are two event procedures that are to be run in a certain order you want to make sure that these events that they are associated with occur in that same order.

Now as I pointed out most of the events occur and what needs to be done in response to those events will be done. If you need something different or additional to happen it's important to understand what the sequence of events is. Now ultimately it's quite likely you will need to know what these sequences are and a lot of the sequences can be quite complicated. So the most important thing really to know here is where to find this information. And that's what we're looking at here.

So what I want to look at first are some of the common events that occur when using Access. And the first section I want to jump to is this one here, Order of events for keystrokes and mouse clicks. Now I won't go through all of these sections but I will choose a couple of examples and we're going to start there with the first keyboard event.

So when you press and release a key while a control on a form has the focus or use the Send keys action or statement to send a keystroke the following sequence of events occurs. So imagine you are in an Access form, the cursor is sitting in one of the controls in the form and what you're going to do is just start typing. The sequence of events that occurs is this. Key down, key press, key up. So key down corresponds to the point where the user presses the key on the keyboard, key press corresponds to the pressing and releasing of a key, and key up is the event that happens when the user releases the key. So essentially you have a sequence of three events. Now depending on what's going on you may need to identify what happens on the key down event, the key press event, and the key up event, or you may not. You may only need to identify what should happen on the key up event. The important thing is you understand the sequence and you know you may need to differentiate between those three separate events. If you need something different to the default to happen you need to understand this.

Now something else that's mentioned in here is that send keys action. And if you've never come across send keys before you can use the send keys command from your code to send keystrokes to your Access application as though they were types on the keyboard. So that is what send keys will allow you to do.

If we move further down the page and just go to mouse events, mouse events will occur very commonly as well. If you start at the top of that section you can see we have a mouse down, mouse up, click event. And following on from that if you look at the next section if a control has the focus and you click another control to move the focus to this second control the following sequences of events occur. So then we have first control, exit, lost focus, second control, enter, got focus, mouse down, mouse, up, click. So again the chances are that you would rarely need to break up these sequences and react to the individual events but if you do it's very important to be able to look at that sequence and see exactly what is happening.

I'd like to look next at the order of events for forms and subforms and I'd like to draw your attention to two or three important aspects. Now the first part of this section talks about the events when you open or close a form, but bear in mind if you have more than one form open and you're switching between two open forms the sequence of events you need to be aware of is a different sequence. So you aren't loading and unloading the forms. The main events you will deal with is activate and deactivate event. So be cautious of that.

Another important aspect when dealing with forms is what happens in relation to data on the forms. And of course the data might be being updated from individual controls. And there's also a section here on working with subforms and the sequence in which forms and subforms are loaded and unloaded, which it's very important again to understand.

And just look at this little note towards the end, because the events for a subform occur after the main form is closed certain events, such as canceling the closing of the main form from an event in the subform will not occur. You may need to move these types of validation tests to an event on the main form. So if something happens in a subform that says, Hang on. Don't close the main form maybe there is a validation problem. You might already be too late. The main form might have already been closed so as it says there you may need to move these types of validation tests to an event on the main form.

Closely related to the form events we have events for controls on forms. And typically in an Access database application you will have forms that contain many controls and we're going to look at examples of this, but it is worth you having a read through this Help page we're reviewing. So where it talks about moving the focus to a control, changing data in a control and the order of events, etcetera the sorts of things you might do within a control is you might change the color of the control depending on what a user has entered. And there are many, many events that can occur with individual controls on a form. And then the form containing those controls may also have its own level of events and corresponding event procedures.

The other thing to mention here is that there is an event model related to reports. Now of course users don't generally interact with reports the way they interact with forms but when reports are created various things can cause problems. For example, if someone created a report and there are no records for the report rather than receive a messy error you can control using event procedures how that lack of any data is presented to users. And earlier on we looked at a couple of examples of event procedures. So let's look at one or two more of them now.

Now what you're looking at here is the copy of the answer to Exercise 05. And I've actually saved a copy of this into the course files folder so you'll find it under version A24 of the Night Movies database. Now what I'm going to do here is I'm just going to open the splash screen. And you may have wondered about that welcome in the top corner. It's set up as the caption of the splash screen. And we also have one or two event procedures assigned to this form. So what I'm going to do is I'm going to right click and we're going to go into the Design of this form. I'm going to bring up the property sheet and we're going to jump to the Format tab. And you can see the first field there, Caption Welcome. So say that I didn't want that caption I could just simply come in here and delete that out. I'm going to close and save my form. I'm going to open it again. What you discover is that we get a default caption, which is essentially the name of the form. So if I want to make it blank what I can do is write an event procedure that fires when the form opens. Now the next question is which event procedure will I need attached to that procedure? So let's go back to our Help document.

So now we're back in the Help document we can see the sequence of events. So it says Open, Load, Resize, Activate, Current. So in theory I could attach this procedure to any of those events. So what will determine the best choice will be what else happens as part of each of those events. But often a good option would be to put it on the last of those events, so the current event. So what I'm going to do is write a current event procedure.

So with the form in Design view if you look at the property sheet with the event tab selected each of those five events is available and I could write a procedure for any of them. So you can see at the top we have On Current but if you scroll down we also have On Load, On Click, On Activate, and On Resize as well. So let's go into On Current by clicking on the ellipses. We're going to select Code Builder. Note here the name is very important because it's the name of the sub, which is Form, and secondly it's the current event. So what I want to do when this event fires is I want to make the caption of the form empty. And that will be a space in this case. So to refer to the form in VBA I call it Me, then I enter the property of me that I want to change. And in this case that is, and if I just start typing I want to change the caption. And I want it equal to, double quote, space, double quote. So that will essentially make the caption empty. So let me click on that, we're going to close the splash screen, save the changes, and double click to reopen. And there we go. We have no caption at the top. So I can see here that my event procedure has worked really well.

That's it for this module. Please join me in the next one.

Video: Switchboard and Navigation Form; Error Handling

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to talk a little bit more about the switchboard and navigation forms. And for this example we're going to use a different sample database and this one will have more advanced features, including event related features and features that depend on code. So in general this database that we're going to be using is a lot more advanced than the ones we have been using in the past.

Now to develop an advanced database takes quite a bit of time so I'm not going to show you everything that you need but I'm certainly going to show you how to get started and then I'm going to move forward in time and show you a later version when many features have already been added. Hopefully that should give you enough to see the advantages that these features offer. So in this module we're going to look at a student database and then we're going to look at switchboards and navigation forms, we're going to look at coding using interaction and we're going to look at how you can handle errors in code.

Now one of the standard databases supplied with Access 2019 is the students database. Now you could create one of these yourself but all I've done is taken the standard one and adapted it by adding more data to it. And you'll find the version with the data in it in the course files folder. And this is what I'll be using for many of the examples throughout this section. So let's first open the database.

Now notice there is an About form and you'll see in a moment that disappears. And what we have underneath is a form called Main Switchboard. So we're going to start out by talking about switchboards and navigation forms.

Now I've explained to you previously that you might want to hide this navigation pane from users. And remember you can collapse it by pressing the F11 key. And you might also decide that rather than let users use the ribbon you might want to hide that as well. Now I'm not going to go in and permanently hide it but I am going to collapse it up. Now imagine if I had permanently removed the ribbon and also permanently deactivated the navigation pane users are now very restricted in what they can do with this form. And notice that there isn't even a Close button on this form. So what you might choose to do is to restrict what users can do by having a form that gives them access to the things they can do but restricts them from accessing things you don't want them to do.

So in this particular switchboard all that the students can really do is click on Students and that opens the student list form. And if we go back to the main switchboard they can also click on Guardians and that opens the guardian list form. Now I'm going to close these down by just clicking on the cross in the corner. You can see they can click on a Close form button to close this entire college student database and they can also click on About which will just pop up that About this application splash screen that you saw at the beginning there. And then finally at the bottom we have an Exit button which will essentially exit you from Access all together.

So that's a switchboard. It's a highly controlled form where you're only allowing users to do certain things that you specify and it's quite a graphical way of doing it. It's nice and easy for people to use and understand. Bearing in mind that in reality we would hide the navigation pane, so users couldn't access that, and if we did provide users with a ribbon it might be a highly customized ribbon. Again, just directing them to certain tools and utilities that we want them to use. So what you can see here is just one way of limiting what users can do. And sometimes, and I think you'll agree, it's a little bit less confusing to have this kind of arrangement than to have the whole navigation pane available with dozens of forms, reports, and macros. That can get a little bit confusing for some people. This is a much simpler way of displaying that information.

Now over many versions of Access there have been mechanisms put in place to produce this kind of control center for an Access application. And one of the earliest attempts was to create a switchboard and this is just an example of a very straightforward type of switchboard.

Now the mechanism to manage switchboards is normally hidden these days. You used to be able to find it on the Database Tools ribbon, but now it's no longer there. So if you do want to find that Switchboard manager the best thing I can suggest is to add it to your Quick Access Toolbar. So let's quickly jump into Options and do that. So I'm going to search for Commands not in ribbon, and again these are in alphabetical order. So if I'm looking for a Switchboard manager there it is. I can click Add and OK. It says I must close and reopen the database. But again I don't actually have to do that because I can see that it's already been applied to my Quick Access Toolbar.

So let's click on this Switchboard manager. Now when I click on Switchboard manager you can see here it's got Main Switchboard Default and that's because I've been in and I've created a main switchboard using this New button. Now if you don't have any switchboards set up in your database when you click on Switchboard manager you will get an error message, basically saying it can't find a switchboard and then it will ask you would you like to create one. If you click Yes it will then essentially create what you can see here. It will say Main Switchboard Default. So it creates a dummy switchboard and it will take you through the sequence of setting up a switchboard to access whichever part of the Access database you'd like to access. So just be aware of that.

Now another option for managing navigation is the use of the navigation form. And we've spoken about the navigation form previously.

Now we're looking at a recent version of the Night Movies database and here you have a choice of two forms that you can use, Movies and Genre. And each can be used to maintain information. So if you want to create your own navigation form, if you go up to the Create ribbon and in the form section you have a dropdown next to Navigation and there's a number of different options that you have in there. So at the moment I essentially have vertical tabs on the left. So you can see where I have Movies and Genre. They're running vertically down the page. But I could have horizontal tabs, vertical tabs right, horizontal tabs two levels, all different ways that I could lay out my tabs. So I'm going to select this option, horizontal tabs and vertical tabs left and it creates a new form. And I can put tabs running across the top or running down the side. And I can specify what each tab does. For example, which form will be opened when I click on the tab. So I could essentially use this to navigate around the database.

Now as with switchboards I'm not going to work through an example of this. There's plenty of information in Help in order for you to practice this in your own time. And in general you'll find that some people prefer switchboards and other people prefer navigation forms. And it normally comes down to personal preference, but these are definitely two options to look further into.

So having taken a look at those options let's go back to the subject of events and let's get back to this main switchboard in the college students database.

Now at the very bottom we have the Exit button and that has some very interesting behavior. So I'm going to go into the design of this form and I'm going to click on this Exit button and we're going to look over in the properties sheet. And you can see here in the Event tab the on click option has an event procedure. So let's look at the code behind this event procedure. So we'll click on the ellipses. So first it uses the message box function not the message box subroutine. The significance of that, as I mentioned before, is that if you use a function you get a return value. And what is happening in this sub is that Access is saying to the user, "Are you sure you want to exit?" That's the message in the message box. And the way we set up the options for the user is using the second argument in the message box function. So we have VB Yes/No and that means show the user a Yes button and show the user a No button. And they will either click Yes or No. Now if they click Yes the return value of the message box function will be VB Yes. So what we're saying here is that if the message box returns a value of VB Yes then do command Quit, which will essentially close Access. Now note the third argument is going to be the title in the message box. And if the user clicks No instead of Yes the answer to this If condition is false, then the sub does nothing at all. So essentially what will happen is nothing. So if the user clicks Yes they quit, if they click No nothing happens.

So let's try this. I'm going to go back to Form view and let's click Exit. And there you can see it says Exit database, so that's the title of our message box, it says, "Are you sure you want to exit?" and the user could either click Yes or No. Now if I click No I would expect nothing to happen, as per that sub in the macro code. And there we go, nothing happens. Let's do that again but this time we're going to click Yes and we exit Access.

Finally in this section I'd like to look at a straightforward example of error handling. So from this switchboard it's possible to click either Guardians or Students to see the relevant form. And if I go into the on click event for guardians you'll see that it's a lot longer than the one for students. And note in here that we're using a string variable called strformname, which is declared as a public string variable at the top of this module. But if I scroll down and look at the command students click procedure I've given that variable strformname the value from students list. And basically the on click involves doing an open of the student list form. However, the guardians on click procedure although it's doing the same thing has error handling. So at the beginning it says on error go to and then I have a label, command guardians, underscore, click, underscore Err. And you can see that label a little bit further down in the sub. So following that label there is a command to put out a message box with the value Error$, which is basically the text of the error. Once the message box has been displayed we fall out to command guardians, underscore, click, underscore, exit, colon. And then out of the sub all together. If an error doesn't occur then given that the form name has been assigned to the value guardian list and we've done an open form successfully the form is open. And what happens within that sub is that control is passed to the exit label. So basically what happens here is that we take control of the situation if an error occurs and we display an error message.

Now I'm going to show you the difference between these two now by changing the name in these form names so that both of them fail but let's look at the difference.

So what I'm going to do here is I'm going to change the names of the guardian list and the student list. So I'm just going to remove a letter from each of these. So technically neither of them should run correctly but remember the guardians has error handling built into it, whereas students doesn't. so this is really to demonstrate the difference between these two. So let me close down this window and we will go to Form view. Now let's click on Students. Now we get as expected a Visual Basic error, a runtime error 2102. It says the form name frmStudentLst is misspelled or refers to a form that doesn't exist. So it doesn't recognize the name of that form which would be correct because I removed a letter from it. Now you could maybe work out what this means and maybe jump into the code and start debugging but it's not particularly informative and you may not want students to start going in and trying to resolve this.

So let's see the difference when we click on Guardians which has that built in error handling. So I'm just going to click End. Let's click on Guardians. And we get a much neater message box that says the form name Guardian List is misspelled or refers to a form that doesn't exist. So it's a much more controlled, gentle error message. The user realizes something has gone wrong but they don't have an option to go into the code. So we've handled the error more elegantly in this scenario.

So that's a very simple example of error handling to get you started.

That's the end of this module. I will see you in the next one.

Video: Exercise 06

Deb: Hello again and welcome back to our course on Access Advanced 2019. It's now time for Exercise 06 which really is a significant step up in terms of your capabilities within Access. And it's quite a long exercise and there are quite a few different things for you to do. So we're going to introduce some new things and one or two elements will involve you doing some research. So this is a bit of a more challenging exercise.

Now whilst I'm working through this I'm going to keep the ribbon collapsed and I'm going to keep the navigation pane hidden. So taking the Night Movies database one step further let me show you my answer.

Now don't worry too much about the alignment and the placement of the controls at this point in time. What I have is just a home page. And it's a single form and it has five buttons on it, Movie Maintenance, Actor Maintenance, Genres, About, and Exit. And it also has a clapperboard graphic which is not completely opaque and you can find this graphic in the exercises folder or as always you could use your own if you would prefer.

Now one or two things you need to make sure of. You need to keep the frontend and the backend separate but make sure you have linked the tables to and from the correct backend database. So you should have an Exercise 06 frontend database and an Exercise 06 backend database in the folder. And the frontend should be using the tables in the same backend database. So what I'm trying to say is go in, go in to your linked table manager and make sure those tables are pointing to the correct backend database.

Now we don't have a splash screen here. We have an About screen instead. So I'm going to click on the About button and notice that it has a Close button. And I've removed the facility whereby double clicking will close the splash screen. You can also see that the version is shown in the bottom righthand corner, and you may want to go an extra step further and put something like a copyright notice on there. So to close this splash screen I just need to click the Close button.

Now a couple of other things, one of them is that this is nonmodal. So if I want to look at something else as a genre I can do it without closing the About form. So let me bring up About again. If I want to look at Genres I can still do that. And I've also suppressed the caption on the form but I've done it using a macro rather than VBA.

So in this particular case I've created the switchboard as a standard form. Notice that I have suppressed its Close button in the top righthand corner. And instead I've added an Exit button. I've also added a button for actor maintenance but we don't yet have an actor maintenance function. So when I click on Actor Maintenance what it says is this facility is not yet implemented. And I only have an OK button just here. Now in the case of Exit I have a Yes and a No button.

Let's now take a look at some of the internals of what I've done here. And if I take one of the buttons, so let's say the Movie Maintenance button and look at the code notice that I've commented it. So make sure you comment your code. Code comments are used to identify who's done something and also keep track of changes. And it's useful also to date any changes that you make. So get familiar with how to comment VBA and macro code.

Something else I suggest you're careful about is giving controls meaningful names. So if we jump up to this property sheet here and click the dropdown you can see some of the names that I've assigned. For example, cmdActorMaintenance. That's pretty obvious that that command, it's pretty obvious that that's the command button that accesses Actor Maintenance. Then I have a command button to access Genre Maintenance, one to access Movie Maintenance, and so on. And you can see as we move further down I have IMG and then my clapperboard image. So that's clearly stated that that is an image. And also any labels I have the prefix of LBL. So I would recommend that you do make sure that these controls are named in a meaningful way otherwise you'll just be working with the default names like Label 01, Label 02, Command 01 and that makes it really difficult to identify what that particular control does.

Now something else here, at the moment I have buttons to fire up two functions, the Movie Maintenance function and the Genre Maintenance function. And in fact I've written the Genre Maintenance button action not as VBA code but as an embedded macro.

Now you don't really know how to do this both ways but at this stage I suggest that you try out both. So there is my macro for opening the Genre Maintenance form, including error handling. And again I would suggest that you jump onto the Microsoft support site because there's plenty of documentation on there about the On Error macro action. So it explains the structure of a macro with error handling. And there's some useful examples in there that are very similar to the one I've included. So again I suggest you refer to Help to learn more about that.

Previously the Night Movies database was using a navigation form and I'm switching to the use of a switchboard from this point on. Now you may decide to delete the navigation form or you may decide to leave it there for future use.

And one final point, if you've never done any programming and you've managed to have a go at Exercise 06 which involves some quite sophisticated features you really should give yourself a pat on the back because what you've achieved by getting an example like this working is that you've moved significantly forward in your use of Access. So from this point on your Access databases can become more sophisticated, more flexible and much more powerful.

That's it for now. I will see you in the next section.

Section 11 – Forms

Video: Formatting Bound and Control Form

Deb: Hello again and welcome back to our course on Access Advanced 2019. In the next few modules we're going to be taking a look at forms in greater detail. And in this first module I'm going to go over one or two of the basics of forms and introduce you to a few features and techniques that you may not be familiar with.

Now we're going to use the student database for this and for most of the next few modules in fact. And this is a relatively advanced database and quite a bit of work on it has already been done.

So first I'm going to look at bound forms and controls, we're going to review record selectors and navigation, we're going to take a look at morphing controls, and we're going to look at some shortcuts for formatting controls on forms.

So let's start by looking at one of the forms and we're going to take a look first at the student list form.

Now the first feature of the student list form is the fact that it is bound. So if I go to the Data tab in the property sheet you can see here it says Record Source it's bound to query students extended. So this is a query within this database which will return a number of fields. And I can select which fields I want to appear on the form. And if you look at the student list form now you can see that it has many fields already. And if you jump up to the Design tab in the Form Design Tools you have an option in the Tools group for Add existing fields and that gives me a list of the fields that are returned to the query. Now this form has been designed with a selection of those available fields. Conversely many of the fields on the form correspond to fields in the field list of this query. But they don't all have to do that and in many cases you may have other fields. Or in this case if you look at Open its value is an If expression. And basically what it says is if the ID of the current record is null, i.e., we don't have a student selected, we display New in that field. Whereas if we're looking at a student we display Open in that field. So many of the fields on the form are also bound.

Now let's select the Room field and let's just bring up the properties for that particular field. And you'll see the control source is Room. So that field is bound to the Room field which comes from the query which is the bound record source for this form. So even if a form is going to be used to maintain data it doesn't have to be bound to a record source and the individual controls don't have to be bound to an individual control source either. The whole thing can be done manually. You can use VBA or macro code. And as you come to a form you can load data into the form, allow users to change data, and when the user is finished you can perform the relevant database operations.

So whereas the binding gives you the highest level of automation you can do everything manually if you want to.

So having looked at a form that's bound to a query let's create a new form but this time I'm going to select an existing query first. And the query I'm going to select is this one here, query guardians extended. I'm going to go to the Create tab and I'm going to say Create Form. Now this form is created based on the selected query. And you can see that I have a number of fields. So Access has essentially taken its best guess at what sort of fields I need to create and each field is bound to the relevant control. And the whole form is bound to that query.

So this is that form in action. And if I right click on the tab at the top you can see I have Form view which I'm currently in, Layout view and Design view. Notice that I haven't allowed Datasheet view. So let's jump into Design view and I'm going to make sure that I have the form selected. I'm going to jump across to the Format tab and here where it says Allows Datasheet view I'm going to change that to Yes, I'm going to close and save my changes, and let's open it up. And by default it's in Form view. Now on the left you have a selector and what you would typically use this selector for is if you wanted to delete this entire form you could make sure you have it selected and press your Delete key. And also at the bottom you have a record selector. So navigation controls at the bottom. If a filter is applied that will be indicated here and you also have a search facility. So let's jump into Datasheet view which we now have because we enabled it for this form. And you can see that the selectors in this case are these little gray boxes on the left hand side. And as I click in them it will select a record. And we still have that navigation running across the bottom to navigate between our records, our filter and our search.

Now note that when I'm in Datasheet view if I scroll down to the last record anything beyond that is marked as New. So if I wanted to create a new guardian I can click in here in the normal way and add that guardian in.

Now one useful thing, instead of adding a new guardian using Datasheet view and let's say that instead of adding one I was looking at this record here, James Jones, if I switch back into Form view it's the James Jones record that I will see. Now if I wanted to add a new record whilst in this view I could click on this button just here, New Blank Record and that will give me a new blank record and I can go in and add in the details of the new guardian.

Now when you create a new form such as the one you can see in front of you it's often the case that you need to change some aspect of it. And one aspect you might need to change is the type of control. Now this form was created automatically based on a query. And one of the controls on it for country region is a straightforward textbox control. So it would be much better if that allowed the user to select from a list of available countries. So I'm going to change that control and we often refer to this as morphing a control. So I'm going to jump back into Design view and I'm going to go to the Data tab. Now notice that I'm not changing the control source for it but I want to change it to a different type of control. So I'm going to right click on the field and you can see that in this contextual menu I have a Change To option. And I'm going to say Change to combo box. And now I can adjust the properties for this particular combo box. So I'm not going to allow people to add other countries in so I'm going to say limit to list, yes, and I'm also going to say allow value list edits, we're going to set that to no. I basically don't want users modifying or changing the list in the combo box.

Now there’s a couple of other things I might need to change in here. So for example the row source. What I would effectively normally do now is create a new table with a list of the countries and perhaps call it something list tblCountry. But I’m sure you’re reasonably familiar with setting up things like combo boxes. So if you’re looking for something additional to do this might be something that you want to have a little practice with.

One thing that tends to happen in an Access database is that you can end up with a number of forms with many, many fields to format. And it can become a very long and laborious job. And quite often what happens is you need to slightly change the formatting of something and you have dozens of fields that will be affected by that change. So one very useful shortcut to be aware of is the use of the Format Painter.

So let’s suppose we take the guardians form and I decide I need to indicate the mandatory fields. So let’s make one of those the last name field. And I’m going to change the formatting. So I’m going to go to the Format tab in the properties sheet and I’m going to make some changes. So I’m going to change the font size just here to 14, I’m going to make it bold, and I’m going to apply a background color. And I’m going to use one of the theme colors in here. So let’s just say, let’s just do like a medium red color. So this essentially will tell the user that any fields which are showing in red are mandatory fields.

Now it might be that I have quite a few mandatory fields on this form so I want to apply that same formatting to those fields. And as I said, it can be quite laborious if you have to go through and do those steps that we just did on each field that you want to make mandatory. So in this case I can use the Format Painter. So I’m going to go to the Home tab and in the Clipboard group we have the Format Painter. Now if you just want to copy this formatting once then you only need to click the Format Painter once. And as soon as you’ve copied that formatting once the Format Painter will deactivate. However if you have a few fields to format and you want to carry on using the Format Painter double click your mouse and then the Format Painter won’t deactivate until you either click on Format Painter again or press the Escape key on your keyboard. So I’m going to double click on Format Painter and I’m going to make business phone, zip and email address mandatory fields. So I’ve been able to copy that formatting really, really quickly. So the Format Painter is a really useful little trick to know. And you may be familiar with it from other Microsoft applications. It is throughout all of them.

Now as I said, you can see my cursor still has that little paintbrush icon which means I’m still in Format Painter mode. I’ve now finished painting my formatting so I’m going to press the Escape key on my keyboard to come out of that mode.

One other little formatting shortcut that can save you a lot of time, let’s suppose you’re going to make a form and you have a number of controls to put on that form. And with the default formatting you have to change it every single time. So let’s take a textbox as an example. So let’s jump up to the Design tab and in the controls I’m going to select the Textbox control. Now if you look at the property sheet what you can see here, it says Selection type default textbox. So what you can do is with this default textbox selected in the property sheet you can go in and you can make various stylistic or formatting changes to the default textbox. So again you could go in and you could change the color, you could change that font size again, you could change the font weight from normal to bold but these changes are being made in the default. So what that essentially means is that every time you select a textbox and draw it it’s going to apply those default settings. So it means that you don’t have to apply that formatting every time you add a textbox because you’ve changed the way the default works. So that is something that can also save you a lot of time.

So that is it for this module. I will see you in the next one.

Video: Split Forms; Macro Error Handling

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to look at a special type of form called a Split form. And in particular we will see how a split form is used in the student database.

Now one of the reasons for doing this is that we're going to use it as an opportunity to look at some more coding related to forms. And we're going to concentrate primarily on macro code. So first up a quick review of split forms, then we're going to look at macro error handling again and then we're going to look at what we mean by a dirty form. Then we're going to look at a new function, the NZ function, and finally we're going to go through an example of re-querying a data source.

Now when you look at data in the student list it actually looks like you're looking at data in Datasheet view. Now you can't see all of the fields for the students. This is just a selection of information. But this is actually split form view not Datasheet view. The visible window is essentially split into two parts. One part is Datasheet view and the other part gives you a form where you can access the data and make any changes to it. So you can add new students and so on. In this student list the reason it is a little confusing is because you can't see the other form.

So let's just jump into Design view and what you'll notice here is if you look over in the property sheet the default view is actually Split Form. And you can change the properties of the split form towards the bottom of the Format tab. So for example, split form splitter bar is currently set to No and I'm going to go in here and I'm going to change that to Yes. And also note that in split form Datasheet view edits are allowed. So I'm going to save this change that I've made by closing down this form and selecting Yes to save the changes and let's reopen it. So now what has happened is that I have a splitter bar. And you can see I can now drag it down and that will reveal the form underneath. So I can make any changes to the student details from here. So for example, if I was to select this student in the bottom half of the pane, so LeClerc, I can go in and I can make changes to his details. So his first name is listed as Carl. Well let's pretend that we've made a mistake here and this is actually Carla. I can then go to another record, I can make changes there if I wanted to, and if we go back to LeClerc you can see my change is still there. So everything is fine. So that's the basis of how split forms work.

Now let's say that the designer and builder of this student list has decided against doing things this way and what I'm going to do now is change the student list back so that the splitter bar isn't there and we're only looking at the Datasheet view part.

Now it's worth noting that I can still edit in Datasheet view. So if I go back to the record that I changed, so this one here LeClerc, so if I wanted to change the first name back from Carla to Carl and note that I don't actually have the first name field showing, what I can do is jump up to Show/Hide Fields and I can check First Name and I now have that first name column. And it's simply a case of coming back into here and deleting out that A.

So why has the designer of this form decided against letting you get to the other half of the split form? Well the main reason is because the person wants to use instead a dialog for making changes to student details. And that's what this Open option is there for. So again if we use our example of Carl LeClerc, let's click on the Open link and it brings up a dialog. So now not only can I change the first name and other information I can also look at things like guardian information. I can see medical information, attendance information, the courses they're attending, so on and so forth. So I have access to a lot of information about each student. And all of these are arranged in tabs in this dialog box. So what I want to look at now is how the mechanism for this works.

As you should recall the student list form is bound to a query. And the field we're using on the form here to access the dialog is not actually bound to one of the fields returned by that query. It has this interesting control source that I mentioned earlier where it says If the ID is null, so in other words if we are in a row that doesn't correspond to an existing student, display the word New in round brackets. And if it isn't, i.e., we're in a row that does correspond to an existing student, display the word Open. So that explains what you see in that field but what actually happens? We need to look at the Event tab field for the textbox. And what actually happens is determined there by an embedded macro. Now as is the case with many macros in Access when you first open them up they might have different sections that are collapsed up. So a quick way of just expanding all of these so that we can actually see the code is to use our ribbon and you can see we have an Expand All. And there we go. Now this can get quite complex and I'm going to concentrate on explaining this macro for most of the rest of the rest of this module.

The first thing to explain is the two On Error actions here. And the macro starts with an On Error action and it's an on error action that takes people by surprise sometimes. So it says On error go to Next. And what happens is that Access records the details of any error that occurs and it stores them in something called the Macro Error Object. And I will come back to that in a moment. It doesn't take any other action itself. It just lets things continue. The important thing to remember is that On Error action persists until a different On Error action occurs. And further down in the macro we have a different On Error condition. And this effectively replaces the first one. So it says On Error go to Fail. And this error condition means that Access stops the current macro and displays its own error messages. And really you'd only normally use this if you were pretty confident that something isn't going to go wrong. And of course you always have a third option in here which is Macro Name and we looked at that earlier on in the course so we have covered all three possibilities for macro error handling.

The next that happens in the macro after the On Error condition is specified is that Access checks if the form.dirty is true. And the word Dirty is the property of a form and it actually determines if a record has been changed since it was last saved. Now bear in mind when this macro runs we will usually have a student selected and we may well have changed some of the information related to the student. And we can check whether the current record has been modified by testing the value of form.dirty. And if form.dirty is true then we run the menu command Save Record. And that will save the current record. So any changes we've made will be saved to the database.

Now the next part is particularly important because we're going to open the form in a moment but before we do that we need to check to see if any errors have occurred. And bear in mind the error handling is in the hands of the macro because we've said to Access on error go to Next and the way that we check if any errors have occurred is to check that macro error object. And we normally check it by saying if the macro error object, macroerror.number.

Now the next part is important because we're going to open the form in a moment. But before we do that we need to check to see if any errors have occurred. And bear in mind the error handling is in the hands of the macro because we've said to Access on error go to Next. And the way we check if any errors have occurred is to check that macro error object and we normally check it by saying if the macro error object, so macroerreor.description, so it's property number is nonzero and bear in mind that will only happen if an error has occurred, we're going to display a message box with a beep and the message will be the macro error property description. So what we would normally be talking about here is a situation where that save record action has failed and the description will describe the error that has occurred. If there is an error, so this is still inside the same If condition, we display the message box and we stop the macro. And at that point the user would see the message but the dialog would not open.

In the next part of the macro we get to the point where we're ready to open the student details form. And we change the on error condition to On Error go to Fail because we really don't expect anything to go wrong. And if it does we're going to leave that in the hands of Access. What we need to do is open the form frmStudentDetails. That's the dialog you saw earlier on. And we're going to look at this form in more detail in the next module.

But one important thing we need to do when we open the form is to tell that form the ID of the currently selected student. Now that is pretty straightforward because it's held in a field called ID but there is a potential problem. And that is when we don't have a student selected the ID value will be null. And the way the student details form works and the way we open it is by using a Where condition. But the Where condition needs to be quite clear in terms of handling situations where the ID is null. In order to do that we use a special function called the NZ function.

So if we refer to Help it says you can use the NZ function to return zero, a zero length string, or another specified value when a variant is null. So if we jump back to our macro code what that means in this case, and if you look at that Where condition the very last part says ampersand, NZ, ID zero. And what that does is to say if the ID is null don't make it null, just make it zero. And the Where condition reads equals, and don't get too confused by the two equals, there just ignore the one on the left hand side. The rest of it yields a value based on equals ID, equals If ID. The rest of it yields a value based on equals ID equals. If ID is not null it will just be the ID of the selected student. If the ID is null it will just be zero. So in the case of an existing student, say with an ID of ten that will say equals ID equals 10. In the case where we don't have a selected student it will say equals ID equals zero. The form student details will need to understand what that means and we'll be looking at that in the next module, but just bear in mind that when we open student details form we will be passing the ID of the current selected student. And if not we'll be passing an ID of zero. So that explanation seems a lot more complicated than it needs to be but that is pretty much how it works.

Now in anticipation of the fact that the user is going to change some data about the selected student the final action in the macro is the re-query action. And what the re-query macro action does is update the data in a specified control on the active object by re-querying the source of the control. If no control is specified the action requires the source of the object itself. So what is going to happen is that the query that gives us content of the student list form will be rerun after the user has made their changes and we'll have an updated set of data in the student list.

So that explains what is happening in the datasheet part of this split form.

That's it for this module. Please join me in the next one.

Video: Creating Form with Tab

Deb: Hello again and welcome back to our course on Access Advanced 2019. In the previous section we looked at the student list in the student database and a macro was invoked from the student list to invoke the opening of the student details form either to modify the details of an existing student or to add a new student. And many of those features are really important and you'll find yourself using them over and over again.

In this module we're going to turn our attention to the student details form and specifically the dialog that's invoked. And we will look at the Tab control and then we'll look at several of the individual controls.

Now in the next module we're going to look in detail at one of the most complex controls because this will also demonstrate many important features of Access and programing in Access.

Now just in case you aren't familiar or haven't come across the Tab control before and specifically if you're not aware of the structure within those controls let me show you a quick example.

So I'm going to jump up to the Create ribbon and in fact I am now going to pin my ribbon so it's always there and I'm going to create a new form. So now you can see I have my Form Layout Tools ribbon and on the Design tab in the Controls group I have all of my different controls that I can select. And this one just here that looks kind of like a folder, if you hover it says Tab Control. So I'm going to click on Tab Control and then I'm just going to drag a reasonably size rectangle over my page. And there is my Tab Control and you can see that it's made up of two different pages, page three and page four.

Now one other thing to know is that with the selection I currently have if you look in the property sheet you can see selection type is Tab Control. If I click on one of the pages, so let's say page three, my selection type is now Page. And I have different things I can change within that Page control. I could also if I wanted to put other controls within each page. So if I'm dealing with, for example, information about students I could split the student information over a couple of pages. But I could also on any page get information from other tables. And that's what happens in the case of the student details form. So let me close this temporary form and let's take al look at the student details form in Design view.

So here you can see that I have five pages, so General, Guardian Information, Medical Information, Attendance, and Courses. And if you take the courses page the page has some properties, but within that I have a student courses subform. So let me just select the subform and within the subform I have of course all of the normal structure. So I have a form header, I have details, if I scroll down I have a form footer as well. And then of course within here I have those individual fields. So let's click in the Details section and then what I'm going to do is select the form, so this is the form within the page, and what you'll see is that the record source here is a query. And if I open that you will see it's getting information about student courses but the important factor is that it is student courses for this student. So notice the join here from the students table and of course in the detail subform overall we will have a specific student selected.

So we'll be looking at the courses that that student is currently taking. So that's how we access the data for this specific subform. If I go to individual fields I can find out what data is held in each field within that subform within that page within that tab control. And that is pretty much the structure of how the student details form works. That's the structure of how you would build such a form.

Of course I may just remind you that in a form-subform situation the relationship between the form and the subform is very important. So if I select the student courses subform and note that in the Data tab there you have the linkage between master fields and child fields, which is the ID in this case. That is of course the part of the data structure that connects the main form to this subform. Let's now look at the student details form and look at some of the features and controls in more detail.

As you know by now the form has its record source as a query and the query is called query students extended. This query is very important. It creates the field student name and that field is used in the form header here. So this is the part of the query that creates the student name and you can see it creates the name using nested If statements. And you need to look at this expression carefully and make sure you understand how it all works because it's very typical of the type of expression you might need to use with nested functions. In this case the first question, the first If condition is, is the last name null? If the last name is null we then ask a second question, is the first name null? If the answer to that is Yes as well then the student name will be just a space. If the answer to that second If condition is no, the first name isn't null, then the student's name would just be the first name. On the other hand if the answer to the first If condition is null, so there is a last name, then we will ask the question is the first name null? If the answer to that is Yes then the student name will just be the last name. If the answer is no then the student name will be the first name concatenated with space and then the last name. So that all sounds quite complicated and I would urge you just to take the time, maybe copy that out to another file and just really look at it and make sure you understand what I'm saying there so you can understand exactly what is happening.

Now one of the controls on this form that currently says Unbound is one we're going to look at in the next module, but we are going to look at some of these others and we're going to start with Save and New. And if we go to the embedded macro that's a good example of a standard type of control you might use often. So we have on error go to Next. We do a check to see if the form is dirty and if it is we save the record. We check error condition. So did an error condition? If an error did occur we're going to flag it with a message box and stop the macro. If not we set an on error go to Fail and then we do a go to record and the go to record is new. And what this means is that we will go to the last record slot in the record set which is the new record at the end. And then at the bottom we have a go to control first name. That means not only do we get ready to add a new record but we position the cursor in the first name field already for the user to start entering data. So essentially we're getting things ready for the user by placing the cursor into the correct control.

Let's look now at the email command. Again, it's an embedded macro and on this occasion it uses the action email database object which has many arguments. And you can specify subject, message text, CC fields, all of those good things that you would expect in an email. And in this kind of student details form you would probably expect an email to be about any number of subjects. So it might be about attendance, administration, fees, etcetera. So there is very little that prefilled into the arguments. Note however that we do have an expression in the To argument. So I suggest that you carefully check that expression.

Basically it’s going to include the student name and be followed by an email address if there is one. In fact if I were writing this particular macro myself I might check that there is an email address and maybe flag an error to the user if we don’t have an email address for the student. But in this particular case it doesn’t do that. It just tries to send an email anyway.

One other argument that does have a value is Edit Message. And what this means is that the user will have the option of editing the message and in fact just about all the other arguments are entered or selected by the user when they click on that button.

So let’s jump back into Form view and you can see that I have the record Sven Anderson selected. If I click on Email and you can see that I have Sven’s name there with his email address is square brackets, but everything else I need to specify myself. So that’s the use of an email action.

So we’ve been through a few more of the controls and I would like to go through the rest, in particular save as Outlook contact. And as you can probably guess this involves another embedded macro. So in this case the on error condition that’s at the start says go to macro name. And in that submacro all it does is beep. It doesn’t give any indication of what’s gone wrong and then it runs the menu command Save As Outlook Contact. So quite a simple one there.

So I would suggest that you go through and look carefully at all of these buttons. So for example, Close, we’re going to look at that embedded macro, and a lot of these are things we’ve talked about in this module so you should be able to work out exactly what most of these are doing. So go through all of those forms and make sure you are clear.

Now there is one other interesting control and that is the one up here. And if we go to the Data and take a look at that control the Data is essentially an expression but quite a tricky one. What it’s trying to do here is indicate a potential duplicate. So if you entered a first name or a last name for a student here or indeed changed the name so that it appeared that you had a duplicate student then it would indicate here that you had a duplicate. So again I’d like you to take some time, just work through that expression and make sure you understand how that duplicate student is identified.

So, that’s it for most of the fields on the student details form. Next we’re going to look at one particular one that’s quite complex but it involves some very useful features, tools and techniques. So I will see you over there.

Video: Go To Combo Box and Its Properties

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to do a complete review of the student details form and we're going to concentrate on one of its controls, its properties and the code associated with it. In many ways how this control works is a very useful model for many other things you may want to do using code in forms in Access 2019. And we're going to be looking at that control, the Go To Contact combo box.

So let's look at it in action. Now again I'm going to select my student Carl LeClerc and I'm going to click Open. And this gives me the student details form for Carl LeClerc. Now if I want to go to another student I can click on this dropdown and I can see a big, long list. And I can see the full names, the email address and the home phone number. And there are a couple of things to notice. So in the email address if there is no email address it's not blank. It just says Unknown. And it's also worth noting that Carl isn't actually in this list. And the last person in the list is Tom Perkins. And you've probably noticed already that these are not in alphabetical order. So if I select a student, for example, Robert Simon, once I select him I can see his details loaded into that student details form. So let's look at some of the key aspects of that control.

The first important point to note about that control is that it's an unbound control.so it doesn't correspond to a field in a student record. And the control source is of course empty. However as a combo box it needs a row source. And where do we get the rows of that combo box from? We get those rows from this select query. And in this select query you can see that four fields are returned. We have ID, student name, email address, and home phone. And the ID field is not shown in the row source but the other three are and I'm going to come back to that point in a moment. However, on the ID field there is a criterion specified and the criterion is that ID is not equal to NZ form, exclamation point, ID, comma, zero. Now when you see that punctuation character, the exclamation point I normally say Bang. And I know a lot of people say Bang instead of exclamation point. So I'm going to say that as we work through this module.

So what form bang ID means is that the value in the ID field in the current form was the ID for Carl. So when we run this query we're running the query using query students extended but we're saying where the ID is not equal to Carl's ID and whichever student is currently displayed will be excluded from the row source list. So we don't try to jump to the same student we're currently one. We always exclude the student we are on unless we haven't got a student selected.

Now the next point I want to make is probably something you already know because I'm sure you've probably dealt with combo boxes in the past. And we're still looking at the Data tab for that control. The bound column is column 1. So when we make a selection from the list it's the first column that yields the value that we want. Now let's go to the Format tab and you can see here we have Column Widths. And we have four columns represented here, zero inches, two, two and two. And because the width of that first column is zero inches that's why you don't see the column. So it's still there and we can still use the value of the student ID as the ID value to jump to in the list of students but we only see the other three columns because their widths are set to two inches each. So that's essentially how you can jump to a value you can't actually see.

So let me just go back to the working student details form. So if I were working in the form for Carl and I wanted to go to a different student I can either go up here and click in the Combo Box or I can click the dropdown and select my student from there. Either way it’s this control that has the focus. And when I click on this control and select a student, so let’s say Jane Smith, up comes her details. So there are actually two event procedures involved in that so let’s look at those next.

So I’ve selected Go To Contact Combo Box and there are two events procedures. We have on got focus and an after update procedure. The on got focus is an embedded macro and it’s pretty straightforward so let’s take a look at that one. So I set up an error condition that says Go to Next and what it says is if screen.activeform.name equals form.name then do a re-query. So essentially what that is saying is take a look at what’s happening on the screen, whatever form is active on the screen if the name of the field on the form is equal to the name of this field on this form I’m going to do a re-query. And what that does is to re-query the data that is going to be used to populate this field. So I essentially update my data on that basis.

If I go to the after update procedure, which is also an embedded macro as far as Access is concerned I’m making a selection from a list. So after I’ve made that selection I execute an after update event procedure. And this is a considerably more complex macro. And within this macro the first thing is to check to see if the contents of the active control is null. And that shouldn’t happen but if it did then the rest of the macro wouldn’t work. So what we do if the contents of the active control is null is stop the macro so that we don’t run into any problems. The next few actions after that are pretty routine and you should recognize them down to this one where it says Set temp var. And in this case what we’re doing is setting the value of a temporary variable and we’re going to call it Active control value. And what we’re going to store in that value is the value in the active control and of course the value is the ID of the student that has been selected. The reason we’re saving that value as a variable is because we’re going to need it in a couple of actions from now. And in the mean time we’re going to remove it from the screen.

I want to look at one or two of the actions in the Action catalog now. So let’s take one of the categories, so database objects here and we’re now looking at all of the available actions. If you look up at the Macro Tools ribbon, the Design tab, in that Show/Hide group you’ll see you have a button there that says Show All Actions. And you’ll see a number of additional actions here. So we’re now looking at all of the available actions and the additional ones have a warning triangle. So essentially they carry an element of danger. In order to execute those actions you would normally need an Access database to be a trusted database. So when we use these actions we always check for trust first of all. And in the next action in this macro what it does is to check the is trusted property of the current project. So if current project is trusted then we set the value of the active control to null. So we’re actually clearing the ID out of the combo box control and returning it back to null again.

And the next point you may have noticed a filter was applied that restricted the form to the selected student. In order to find another student we need to remove that filter from the form. So the next command is if the filter is on run the menu command remove filter sort. So the student details form will now show all students. Then the next action after that is to search for the record.

And for the first one that we find we have a Where condition. And the Where condition says that the ID of the student is stored in the temp variable active control value. So that takes us to the student we've selected in the combo box. And then finally having done that we do a bit of a tidy up, we remove the temporary variable active control value.

So that's it for event procedures and that is the end of this section. I will see you in the next module for an exercise.

Video: Exercise 07

Deb: Hello again and welcome back to our course on Access Advanced 2019. It's now time for Exercise 07 which is a considerably more substantial exercise than the ones you've seen so far.

So based on what we have covered in the last few modules at the end of it you should have a Night Movies database that is headed in the direction of being a more usable movies database. Now improvements will still be needed but one of the aspects of learning Access is to realize that the more you find out about it the more you realize that the way you need to do them will depend on your own preferences and circumstances in which you expect your users to be working. So basically you need to build up your knowledge and then look at the options for improving. And working through these exercises will in fact put you in a much better position.

So with Exercise 07 I've done it using macro code. And you may prefer to use VBA code and in fact most experienced Access developers would use VBA code. But at the moment I would like you to persevere with the macro code at the moment.

You could even once you've got your basic database working convert the code to VBA code. So take a look at the VBA code and maybe from that point onwards you could work with the VBA code instead.

Now what you're going to be doing is hooking up actor maintenance. Now I've done that here in my answer. So if I click on Actor Maintenance up comes my actor list. Notice the open a new feature, which I set up to work the same way as the student list, and I've implemented a couple of additional buttons in the form header. There is a new actor button and a Show/Hide columns button. Now Show/Hide columns we looked at briefly previously. So take a look at that, look at the macro command that's been used, find out how that works, and implement that here. So for example, if I didn't want to show the died date all I would do would be to click on Show/Hide columns and uncheck Died Date and that column has disappeared.

Another feature I would like you to check is that the Search at the bottom of the screen there works. And then of course the most fundamental aspect, so let's choose an actor. So I'm going to choose Lee J. Cobb and if we open his record we now have the actor details form. So you can see there we have screen name, given names, gender, date of birth, ID, family name, country of birth and date of death. And note the screen name there. That's independent of the given names and the family name.

And one of the things I've implemented in the actor details form is a Cancel button. When you click on Cancel it will ask me if I'm sure I want to cancel. And if I say Yes it will undo any changes that have been made. And then if I close the dialog no changes are made to that record. Similarly if I go for the new actor button and set up a new actor if I try to save this I'm going to get an error message and that's because I haven't stored a value in the gender field. Now I don't particularly like this error so I might want to change this error and really make it my own. So think about that as well. I'm now going to complete a gender. I'm going to say Male. I'm going to complete the country of birth and I'm going to cancel out of this dialog box and close.

Now let me open another actor. I’m going to open Viggo Mortenson and we also have another page in here. So we’re currently on the General page but we also have a Movies page and this lists all of the movies that have featured Viggo. And you can see here he was in Lord of the Rings Return of the King. So this particular Movies tab is a subform that doesn’t have the facility to add roles for this actor but of course it could. So again if you wanted an additional exercise you could experiment by enabling an Add button there.

And of course the Go To facility works with the list of all the available actors in the database.

So now I think you’ll agree you’re starting to get a reasonably usable database but one of the consequences of that is that the more you do the more you realize how much you could do. But for now that’s it for Exercise 07 and I will see you in the next section.

Section 12 – Advanced Use of Queries

Video: Basics of Queries

Deb: Hello again and welcome back to our course on Access Advanced 2019. Over the next few modules we're going to look at queries. And in this first module we're going to do a quick review of the basics of queries, we're going to take a look at Structured Query Language or SQL as you might know it, we're going to talk about aliases, we're going to explore the use of expressions, and we're going to look at the Order By clause.

So first let's start out with the basics of queries. I'm going to jump up to the Create tab and in the Queries group you'll see that I have two buttons, a Query Wizard and Query Design. Now I already have a table selected. So you can see in the navigation pane it says tblStudents, but that doesn't really matter at this stage. If I click on the Query Wizard I get a new query form and I have four different options that I can choose from.

So the first one is a simple query and that is exactly what we're going to do in a moment so we'll come back to that. The cross tab query we're going to talk about a little bit later on. And the other two I'm going to assume that you're familiar with, so find duplicates query and find unmatched query. If you're not I would suggest that you use the Help feature in order to find out what those two are. So those are the options that you have within the Query Wizard. Now I'm just going to cancel out of there.

The other option we have is Query Design and that opens up the Query Grid at the bottom here. So I'm going to create a very simple query. So from my Show Table list I'm going to choose the Students table and I'm going to add it to the grid and I'm going to close this Show Table dialog. And all I'm going to do is I'm going to take all of the fields and drop them down to the bottom. So I'm going to use this asterisks which means all fields and I'm going to drag it down to here. And really what that says is get me all the fields from the students table. And if I wanted to run this I can click on the Run button on the Design ribbon and there are the results. All of the students and all of the fields. So that is a very, very simple query.

So we've seen two views of a query but there is a third view. If you go up to the View button on the Home ribbon you can see Datasheet view which is the one that we're looking at now, Design view which we are familiar with, and we also have an SQL view which actually expresses the query using Structured Query Language. Now as you can see this is very, very tiny so we will need to make that a little bit bigger.

So in order to do that we jump up to File and we go into our Options. And I'm going to choose the Object Designers page. And you can see here in Query Design I can choose the size of my font. So I'm going to change that from 8 to 14 and I'm going to click on OK. Now it's telling me I must close and reopen the database so I'm going to do that and there we go. That is a lot easier to see.

Now given you can achieve good results using the Query Grid why would you choose to work in SQL? Well learning SQL means you are learning the most popular language for working with databases. So overall if you learn SQL much of what you've learnt could be used with, for example, Microsoft SQL Server database. And there are a couple of other reasons. One of them is that there are a few things that can be done in SQL that you can't do in the grid. And also if you use the grid some things are quite difficult to achieve. So in general sometimes it's quicker and simpler to use SQL. So let's look at one or two features of this select statement.

Now there is a tendency to write the statements with the clauses beginning on new lines. So for instance select tblStudents.* and then the from tblStudents begins on the next line. Now you can do it like this but this is also unnecessary. I could quite as easily do it like that.

Another important point here is that since all the data we want is in the student table we don't need the prefix on the asterisks. So if I wanted to get all of the fields I can just use asterisks. So now we're saying select all from tblStudents. And it's also worth noting that SQL statements generally end with a semicolon. So let's run this and see what happens. And as I would expect the results are the same. I get a list of all of the students.

Now notice a couple of the fields, the last name and the first name. If we go back into SQL view, if I wanted to get specific fields here and the fields have spaces in the names what I can do is put the field names in square brackets. That, for example, would give me the values in those two fields. So first name in brackets, last name in brackets. Let's run the query and there we go. First name and last name.

Next I want to introduce the concept of an alias. So let's jump back into our SQL view and I could introduce some aliases into this string. So I could change this to say select first name as first and select last name as last. And first and last in that case are aliases for the real names. Let's run the query. So of course I get the same data but notice the column names are now just First and Last.

Now I'm not restricted to renaming the existing field. I can create new fields in this way. So let me make a full name field out of first name and last name. And to do that I'm going to use an expression. So we're going to say first name, ampersand, quote, space, quote, last name as and then we're going to call full name. And because that has a space in it we need to put it in square brackets. So select first name, ampersand, space, ampersand, last name as full name from tblStudents. Let's run that. And there we go. So you can see what's happened here. It's concatenated the first name and the last name, put them into one column and that column has the name Full Name. And again if jump back to the SQL view just to review you can work out exactly how it's done that.

So now I have working full names. If you think back to the students extended query that was a bit cleverer than this as it coped with a situation where either the first name or the last name was blank. So let's start by covering a situation where the first name is blank.

So what we're going to say is if, and for that we use the I, If, open round brackets, is null. Now that's another function so we need round brackets for that. And the thing we're checking is first name and that needs to go in square brackets because it does have a space in there. So if the first name is null what will the full name be? Well, it will just be last name. Whereas if it isn't null then the full name will just be first name, ampersand, space, ampersand, last name. So I need to add into here the last round bracket for the If function. So I now have an expression which deals with the full name whether the first name is null or not. So let's just try running that and that looks fine.

Now whilst that's all well and good that's not quite enough on its own because of course we're also going to need to cover the situation where the last name is blank. So what we're going to do is embed all of this inside another If statement and I'm going to check if the last name is blank.

So there we go. So it's quite a long piece of code but let's sort of walk through it. So if the last name is blank in theory the full name will just be the first name. So if the last name is not blank the expression I want is the whole of the expression I had before. So let me run this one and see what I get. And there we go. I get those full names.

Finally let's look at one of the features you will have used in the grid. And the first thing I'm going to do is I'm going to put that carriage return back to put from tblStudents on a new line. And in fact to make this a bit neater I'm going to drag that property sheet in very slightly.

And what I'm going to do is I'm going to add an additional clause into the select statement. And I'm going to use the Order By clause which basically sorts the specified records by a specified set of criteria. And in this case I'm just going to sort by the last name of the students. So on a new line I'm going to say Order by and then we say what we want to order by. So in this case it is last name. So I have my select statement there covering almost all of the situations where the first name and the last name are blank from the table students and I'm going to order it by last name. Let's click Run and there we go. So that's the use of the Order By clause.

That's the end of this module. I will see you in the next one.

Video: Syntax of SQL Statements

Deb: Hello again and welcome back to our course on Access Advanced 2019. In the preceding section we started to look at SQL statements in more detail. And in this module we're going to go to town with many aspects of SQL statements.

Now the SQL statements we're going to look at are fairly universal when it comes to features and when you learn how these work you'll find they work in other database systems and not just in Access. So we're going to start out taking a look at the use of Wildcards, then we're going to look at dates in queries, we're going to look at aggregation in queries and Group By, and finally we will look at Having in SQL statements.

So in my first query I'm going to look at the full name of any students with the last name Peterson. So let's run that. And there we go, I get two students. Let's go back into the query itself.

Now one other possibility here is instead of putting an equal sign is to use the term Like. So let's try that instead. So let's try running that and I get exactly the same result, which is good. Let's go back into the SQL code again. So now what I'm going to do is I'm going to say I want the name Like P. So let's run that and see what happens. I get no results. So let's jump back into the SQL. But what if I put P*? So using a wildcard. Let's run and now I get all of the students whose name starts with P. So that P* basically means show me all of the names where the last name begins with a P. Let me change this to J and Run and I get two results.

So now I'm going to do the same thing but for the case of first name. And run again and now I get all of the students where their first name begins with J.

Now I'm going to do something slightly different and I'm going to introduce some other wildcards. So instead of having J* I'm going to replace that asterisks with three question marks. And the difference here is that a question mark represents a single character. So let's run this and see what we get. So there you can see I now have a list of students whose first names are four characters long and begin with the letter J.

Now apart from using asterisks for any number of characters and a question mark for a single character you can also specify a choice of characters. So let's look at a different example.

Now this time I'm going to base this query on email address. So what I'm saying here is give me the full name and email where the email has any number of characters, a semicolon and any number of characters. So let's run this and we only get one student, Julie Spencer. So I found the student with a semicolon in her address but how about looking for a semicolon or an underscore. So let's go back into SQL. Now if you want to offer a choice what we need to put in here is asterisks, square bracket, semicolon, underscore, close the square bracket, asterisks. So what we're essentially looking for here are students that have either a semicolon or an underscore in their email address. So let's run that and see what we get. And we get two students. So Maria has an underscore in her email address and Julie has a semicolon.

If you'd like some more information about wildcard characters then it is definitely worth checking out the Help within Access. And you can see all I've searched for here are examples of wildcard characters. And you can see those in the table below. You'll also see there is a link through to applying criteria to a query. And it's also worth noting that you can apply wildcards in expressions as well. So there is more information about that in there. So definitely something you should have a little read through.

Now something else you'll need to do in an SQL statement is refer to a date. And if you want to specify a date constant in an SQL statement you need to embed it between hash symbols. So suppose I want to find all of the students born before June the 30th 1993. So I'm using the date of birth field which is a field of type data and to specify a constant instead of using quotes it has hash symbols. And all I'm going to do is add my date in between those hash symbols. Now the format in which you enter this date will very much depend on your locale. So I'm using U.S. locale so my date is going to be June the 30th 1993. And let's run that and see what we get. And there we go. I can see I have 23 students who were born on or before June the 30th 1993.

What I want to look at now is one of the predicates you can use in SQL statements. And predicates are sort of additional pieces of information. And they are more specific or standard case of a special statement. So we're back in our query and what I'm going to do is I'm going to change it completely.

So what we have here is select state/province from tblStudents. So fairly straightforward enough. Let's run it. And there we go. We have all of the states and provinces that are referred to in the student table.

Now let's jump back into SQL view and instead of saying select I add the predicate distinct. So it will select the individual states and it will only name each state or province once. So let's run that and watch what happens. There we go. We have a much shorter list.

And don't forget that we can always add an order by. So if I wanted them in descending order I could do something like that. And then if we run I now get those states listed in descending order.

So that's the use of the distinct predicate. And remember you can always dive into Help where you have all of the information there related to predicates. So again definitely worth checking this out, having a read through, making sure you understand all of those different options.

So now I want to look at aggregation in SQL and in particular the group by clause.

So let me make a small change to this query. So you can see there I've added the last name field into my select distinct statement and I've also changed the order by from descending to ascending. And it's worth noting that ascending is the default so I didn't necessarily have to specify this but just so you can see what that looks like there it is. So let's run this. And no real surprises there, I hope.

If I go back into SQL view something else I can do in here is to group the results. And I can use a group by clause. So let's jump in here and add that in. So we're going to say Group By state/province. And instead of just showing the list of last names what I can do is to count how many last names there are for each state. So we can say Count and let’s run this. And there you go. Now you can see a count of the number of states. And it’s worth noting that there is one student without a state specified.

Now aggregation of information and grouping can be done at multiple levels. So don’t forget with the student table apart from having state/province we also have level. So let’s go in and make further changes to our query.

So what I’m doing here is introducing the level of the student into the selection. And I’m going to make level part of the group. So I’m going to group by state/province and then level. Let’s take a look at that and there we go. So for example you can see that Connecticut has two juniors. And in New York we have one junior and five seniors, etcetera.

Now one important thing to be able to do is to start combining all of the elements of SQL statements that you’ve seen so far. So for instance, instead of getting a strange field name coming out of count let’s rename that count. Now I’m also going to add a Where clause. So I’m going to say I want to get these counts but I only want them for students born before December the 31st 1993. So I’m going to drop a Where clause in after the From. So there’s my Where clause. Let’s run and see what we get. And there we go. We now have the count of only students born prior to December the 31st 1993.

Now I’m going to make one more change here. I’m going to change Order By. So instead of ordering the output of this query by state/province I’m going to order by count and in descending. And I’m going to change the predicate from distinct to top. And let’s go for a little bit of a strange number. I’m going to say I want the top seven. So I’m asking for here the top seven where they’re ordered by count ascending. So let’s try that. And there are the top seven counts.

Now the very last clause I would like to cover is the Having clause. So let me remove the top seven predicate and I’m going to leave that Where in place but in Group By I’m going to add a Having clause. And the Having clause specifies which group records are displayed. So what I’m going to say in this case is select the students born prior to December the 31st 1993, group them by state/province and level and I want the student count but I only want the records included where the count is greater than or equal to two. And I’m still going to order them by count descending. So let’s take a look at that. And you can see I have seven records. So for New York state I have five students at senior level, New Hampshire two students at senior level, so on and so forth.

Now the very last thing to look at is just to go into the Design view of the query because it’s worth spending some time looking at this because you can create a query like that using Design view. Now I find it easier to understand the SQL but not everyone does. Some people like Design view. And if you take a look at this in the grid at the bottom you can probably work out exactly what this query is doing in this grid view. So if that is a way that you find it easier than you could definitely use that way. As I said, personal preference for me. I find it just a little bit easier to come into SQL view and do something like that. But again, entirely up to you.

That’s it for this module. I will see you in the next one.

Video: Use of Multiple Where Criteria & Select Statements Function

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to continue looking at SQL statements, specifically Select statements. And we're going to look at multiple Where criteria and then some special functions, the In function and the Between function and also the Not operator, and finally we're going to look at null conditions. So let's start by looking at multiple Where conditions.

So let's suppose I want to find all of the students in the student table that have a Gmail or a Yahoo account. So if we look at the students table and look at the email addresses I can see some people in here have Gmail, some people have Yahoo accounts. So I essentially want to cherry pick these students out of this list. And it's also worth noting here that they're not all Yahoo.com addresses. We have one just here that is a Yahoo.co.es. So we want to make sure we pick up those as well.

So what we might do here is say look in the email address field for either Gmail or Yahoo but not the .com part so that we pick up those other ones which have a slightly different extension. So we can do that with Like and wildcards but we have two conditions to check and we're essentially looking for either of them. So let's jump back to our query.

So the way that we would formulate the Where clause is like this. So what we're saying here is where email address is like asterisk, Yahoo, asterisk or like asterisk, Gmail, asterisk. So let's give that a spin and see what we get. And there we go. We get ten records. And you can see that because we've used those wildcards it's actually picked up those Yahoo.co.es addresses as well.

Now I'm going to introduce a third Where condition after the second one and note here I'm selecting the first name. And the third condition is that the first name is like C*. So it begins with a C. So let's run that.

Now you might be a little bit surprised by this outcome. So it's found six students but only two of them begin with the letter C. Now the reason for that is that these operators have an order of preference. And for the main operators that sequence is Not and then it's And and then it's Or. So what happens with this Where statement is it looks for students first of all where the email address is a Gmail one and the first name begins with a C and then it treats the Or as a separate lower ranked operator. So let me put in brackets to show you what it's doing. So that's what it's doing. It's doing the And first and then it's done the Or. So somebody who wants to be included on the grounds of having a Yahoo account doesn't need their name to begin with a C. If on the other hand what we really wanted was a student with a Yahoo or Gmail account but they must have a first name beginning with C as well then what we really need to do is this.

So the way I've now set this up it means that the brackets now get priority. So you can see here if I highlight it to make it easier to see this is what is going to be done first. So essentially the brackets take precedent over the And or the Or. So you can see this pair of round brackets. Those are executed first and we're saying that the email address is Yahoo or Gmail. Once we've established that we then have an And requirement that the first name begins with a C. so let's give that a whirl and see what we get. And there we go. So we now get just two email addresses, one Yahoo, one Gmail and both of those names begin with a C.

So what you've really seen there is an example of the order of precedents. And many people writing SQL statements know and rely on the order of precedents and many people always put in those brackets. So the way I like to think about it when I'm working in SQL is I just really try and think about what do I want it to execute first. And whatever that might be that goes within the round brackets. And again if you've used an application like Excel when you're compiling formulas using functions you may be familiar with that structure of putting brackets round what you want it to calculate first of all. So it works in a similar kind of way.

Now bearing in mind what I've just said about the order of precents it's interesting to look at the query in the grid. So in this case it's saying that these two criteria Like C and Like Yahoo or Like Gmail, so in this case it's saying that these two criteria Like C* and Like *Yahoo* or Like *Gmail* these two are standard. And even in the grid you can see what the relationship is between the criteria or groups of criteria. And sometimes looking at these conditions in the grid can help you understand how exactly complex conditions will work.

Now let's look at another example. Let's suppose I want to find all students who are from California, Kansas, New York state, or Connecticut. So I could have a Where condition where state/province equals California or Kansas, etcetera but that might start to get very, very long. Now there is an alternative and it's called the In function.

Now in this case I've included the city and the state/province in my select. And what I do with the In function is to say that where the state/province is in and then I just need to list all of the options within my round brackets, making sure that each of those are in single quotes. So let's try running that. And there we go. I have a list of 13 students who are located in those states. So I think you can agree that using that In function is a much shorter way than using multiple Where conditions.

Now another useful function you have is the Between function. And I'm going to demonstrate this by looking for students whose date of birth is between two given dates. So what I've got here is our Where condition. So where date of birth is between and then I just need to put in my two dates. And remember that date constants we need to put in between hashes. So I'm going to say the first of the 1st 1994 and 12/31/1994. So let's give that a little test drive and there we go. I have three students that were born in 1994.

Now another useful feature of Access SQL is the Not function, which essentially negates whatever Where conditions you want it to. So if I wanted to find the students not born in 1994 I could put Not and then everything I want it to be not in brackets. So let's take a look at what that would look like.

So basically what I'm saying here is where not and then I put everything that I want it to be not within round brackets. So let's give that a test drive and there we go. It's telling me I have 32 students who were not born in 1994.

Now instead of putting all of that in brackets with a Not before I can also do it like this. So instead of saying Where Not I've removed my round brackets from the outside and I've moved that Not to here so now it says Where date of birth is not between these two dates. And effectively that would do exactly the same thing.

Finally let's look at one other useful thing to be able to do in a select statement and that is to look for data that is null or not null. Now null in general is different from space. So sometimes a text field will have a space in it. That doesn't necessarily mean it's null. And some fields can have null values and some can't. And sometimes fields that have a null value can be quite deceptive. For instance, I'm looking at the students list here. And if you look at the two students where the email address is unknown, so that's Jane Smith and also Robert Simon. Now the word Unknown is not what's recorded as their email address. Their email addresses are essentially null. The reason it says Unknown in there is due to the formatting of that field. So if I go into the design of the student table you should be familiar with this kind of format statement. And this says if the field is null put Unknown there. Now I can look for instances of null in a field by using the term Null. So for instance, I could say select first name, last name from the students table where email address is null. Let's give that a whirl and there we go. I get those two student records.

So we've looked at quite a lot in this module. We've now come to the end. I will see you in the next module.

Video: Exercise 08

Deb: Hello again and welcome back to our course on Access Advanced 2019. It’s time now for Exercise 08 and this one is pretty straightforward.

So all I want you to do is create a query and for the movies in the Night Movies database it works out which is the most popular genre for all of those movies and puts them in ascending order. So you can see we have the genre Action at the top and the number of movies 22. We then have Drama, number of movies 18, so on and so forth.

That’s it for Exercise 08. As I said, very straightforward. I will see you in the next section.

Video: Action Queries – Part 1

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this section we're going to move on to talking about action queries. And action queries involve doing something to change data. So typically this will be maybe something like an update or a delete or even making a new table. And as part of action queries we will look at parameter queries. So we're going to start out with update queries, then we'll look at parameter queries, and then we'll look at a fairly complex string of manipulation examples.

So in an update query what we do is update data in the database. Now the update might be to a single record or maybe all records in a table, or maybe two records in different tables. And I'm going to demonstrate this with a very simple example. Of if you cast your eyes over to the state/province column in the students table, and it's all the way over on the righthand side, let's suppose what I want to do is replace the two character code in the state/province field with the full state name. So let's suppose I want to replace AZ with Arizona wherever it occurs in the table. In order to do that let's create a new query.

Now I'm going to start out with a select query and we're going to select the student's last name and I'm going to select students where the state code is equal to AZ. Now one very useful aspect of running a select query first is that I can make sure that I'm getting the correct students. So let's run that and see what I get. And I get two of them, Burgstrom and Pink. And I've already had a quick check in the students table and I can confirm that those are correct.

Now if I go back into Query Designer what I'm going to do is to change the type of this query from select to update. So if you glance up onto your Query Tools Design ribbon you can see in the query type we currently have Select selected, which is the default. But what I'm going to do is I'm going to change this to an Update query. Now potentially either of the fields that I've got indicated there can be updated. And the one I want to update is the state/province field. And I want to update it from AZ to Arizona. So let's run this query again. It says you're about to update two rows. So I'm going to click Yes and now if I jump back to the students table and you can see here where we have Arizona for Burgstrom it's been update from AZ to Arizona. And if we go all the way down we have Hermione Pink and she has also been updated from AZ to Arizona. So I'm fairly happy with how that's working at this stage.

So I'm now just going to look at this query in SQL view and you can see what it's doing here. So it's saying update tblStudents set tblStudents.state/province equals Arizona where tblStudents.state/province equals AZ. So it's basically saying wherever you find AZ update the table with Arizona.

So that is a really straightforward kind of update and there are a lot more complex ones but that is the essence of an update query.

Now having developed this update query we could change what we're updating. So let's say we're looking for the CT state code, which translates to Connecticut. So let's make a change to our code. I'm going to replace AZ with CT, and the full name Connecticut. So let's just try running that. It says you're about to update four rows so I'm going to say Yes. Back to our students table and you can now see that all of the CT's have been updated to Connecticut.

So that's all fine but once you've developed a query like this you can add parameters. So you can create a query where one or more of the values in that query are parameters and the user is required to enter those parameters when they run the query.

So in this particular query we're looking for students where state equals CT. If instead of that we want the user to be able to specify both the code and the name each time the query runs we use parameters. And we identify them by putting them in square brackets. So when it comes to update to instead of literally putting Connecticut we would put in square brackets Enter state name. And let me just widen that column so you can see it. And in the criteria, again in square brackets Enter state code. So let's now run this and see what we get. There you go. I'm being asked to enter the state name. So let's say California, click OK, state code, CA, click OK. It says you're about to update two rows. I'm going to say Yes. And now if we jump back to the students table and look in our state/province column I can see there those are my two California records updated.

Now it's worth noting there that the wording in these messages corresponds to the words that I've put in those square brackets. So just be aware of that.

Now let's look at this in SQL view. So here it says Update tblStudents. Set tblStudents.state/province equals and then in square brackets Enter state name, where tblStudents.state/province equals Enter state code.

So let's now look at a more complex update example. I'm going to update some area codes in the phone numbers. So that home phone column. And I would like to point out that these are not real phone numbers but I have tried to make them have the correct structure. And you'll see if you look towards the bottom a couple of them have an area code of 212. And what we want to do is update those to 211. And I'm going to use an update query but it's more complex than the one we've done before because we're only looking for part of the phone number. So the first thing I'm going to do here is I'm going to find the students that have a 212 area code. And this is a good way to start a query by doing a select first to make sure you're identifying the correct records to update. So let's start with that select statement.

Now in order to find the students with a 212 area code you might argue that we could put in Like and then something like star-212-star. However note that the area codes are generally in round bracket. And if we put star-212-star we may pick up phone numbers where there is a 212 elsewhere in the phone number. So it's really important that we put these in round brackets. What I'm going to do here is just make a small change. We're going to delete that out. We're going to put in quote marks and in round brackets we're going to put in 212.

Now we haven't quite finished here because currently this is just going to pick up 212 but I want it to pick up the full phone number. So I want everything that comes after it as well. So let's now try running that query. And there we go. There are four students where the home phone number is a 212 area code.

Now look at those numbers. What we need to do is replace that 212 with 211 but keep the rest of the number. So we have an element of string manipulation that we need to do here. And there are many string manipulation functions in Access so let's first of all look at some of the functions that are available.

So I'm going to jump back into Design view and I'm going to change the query type from Select to Update. And the first thing I'm going to do is I'm going to save this query. And I'm going to save it with the name query Area Code Change. So now I'm going to put my cursor in the Update To field, I'm going to right click and I'm going to select Build. And that's going to jump me into my Expression Builder. And what I'm going to do in here is build an expression which will change that area code. And I'm going to use one of Access's own string manipulation functions. So in the expression elements tree there I'm going to just click the plus next to Functions and I'm going to look at the built in functions. And the expression category that I'm interested in is text because essentially what I'm saying is that within a string of text replace it with a different piece of text. And one of the text functions if we scroll down is Replace. So essentially I can use Replace to replace a piece of text within a string with a different piece of text. So when I double click on Replace you can see that I now have six arguments. And it's worth noting here that if you're not really sure what these arguments mean, if you look down to the bottom of this dialog box in the Expression Builder you do have a link there which will take you through to Help on the Replace function.

So the first argument here is the string in which we're going to make a replacement. So in this case that would be the student's home phone number. So, let's replace this with square bracket, home phone, close square bracket. So that is our string. What do I need to find? Well that is 212 in round brackets. What do I need to replace it with? Well I need to replace it with 211 in round brackets. And next we have start. So it's basically asking me where do I start looking. Well I want to start looking at the beginning of the list. And that is actually an optional function so I can leave that out. We then have count which basically says which of these 212 occurrences will it be? So if you have multiple occurrences of the find string in the string you were searching you can say find the third occurrence, for example. In our case there will only be one occurrence, and again that is an optional argument so I can leave that out. And finally we have a compare argument. And you only need to go for the default value in terms of compare. So essentially these three arguments can be deleted.

So that's it. That's my expression. I'm going to click on OK. Let's take a quick look at this in the grid. So I'm going to widen that column out. So we have Replace home phone number 212 with 211. So let's try running that. It's telling me I'm about to update four rows. I'm going to click Yes. Let's jump back to our students table. And if you look down here these two were 212 fields and you can now see we have 211 in there instead. So it looks like it's worked as expected.

That's the end of this module. I will see you in the next one.

Video: Action Queries – Part 2

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to continue looking at action queries and there are three more types of action queries that we haven't yet looked at. We have Delete queries, Make Table queries, and Append queries. So let's start with a Delete query.

Now I'm in the courses table just here and we've been told that we're no longer going to run courses that are less than 30 weeks in duration so I need to delete the courses in this table that are less than 30 weeks long. And when I'm doing a deletion by query I will normally take precautions to make sure that if something goes horribly wrong I haven't lost a lot of valuable data. And there are a number of ways of doing this. One way is to take a copy of the table and another way is before I delete the records make a table with just the records I'm going to delete in them to keep them safe. Now I am actually going to start by writing a query that writes away the records I'm about to delete to a separate table and then later on I'm going to show you how you can put those back. But for now let's just assume that I've done that and right now we're just going to delete the records that are less than 30 weeks long. And in the same way that we did for update queries it's a good idea to start with a select statement to make sure that you're getting the right courses. So let's start with that select statement.

So that's my query in Design view. Let's now look at that in SQL view. So we can see we have our select statement, so Select tblCourses.weeks and select all of the courses from tblCourses where tblCourses is less than 30 weeks. So let's run it and see what we get. So there we go. I get three courses where they are less than 30 weeks long.

Let's jump back into Design view. Now this time I'm going to change the query type. So again currently we have Select. I'm going to change this to a Delete query. And let's look at the SQL for the delete query. So now you can see it's changed here from Select to Delete and everything else is pretty much the same. So it's going to delete from the table courses all courses where the number of weeks is less than 30 weeks long. So let's run this and it says you're about to delete three rows from the specified table. Click on Yes. And if we jump back to our courses and have a quick look down we can see those deleted records. They're showing there with #deleted in them. And if I jump up to my Home ribbon and just do a quick Refresh All you can see that gets rid of those records. So they really are deleted.

Now I'm going to save this query and there we are. No more short courses.

Next we're going to look at Make Table queries. And I've been asked by our admin department to make a mailing list for students. So sometimes when you need a selection of information about a particular entity, in this case about students, for confidentiality reasons it may be better to make a table with just the information the user needs rather than give them access to a table that contains lots of other information about the student.

So in this case if all they need is a list of names and addresses I can make a table for them with just that information. And that is what I'm going to do in the next query.

Now there are a couple of things to warn you about first here. First of all, when you run a Make Table query it makes a table. If you already have a table with that name the table with that name that you already have will be deleted. And secondly is that once you've run a Make Table query it doesn't leave any connection going forward. So for example, if I run a Make Table query now to make a name and address list if I add some students to my table later on they won't be automatically added to the list that I've made for the admin department.

Now for the purpose of this exercise I'm working in version 05 of the student database and I've put those state/province names back to the two character code. And what I'm going to do is I'm going to construct an address for each student comprising their first name and their last name, then their address line, then city plus state code on the third line. So what I'm going to do is I'm going to go up to Create and we're going to go to Query Design. And this is going to be based on the students table and the first thing I'm going to do here is change the query type from Select to a Make Table query. And I get this little Make Table dialog box come up and the first thing I need to do is give my table a name. So I'm going to call this tblStudentAddresses. Now note that I already have a table with that name. So this is going to override it. And now what I start to do is to build those fields for my new table.

Now the very first field will be the first line which will be first name, space, last name. So what I really want here on my first line is going to be first name and space and last name. So essentially that concatenated string will form the first field in a record in my new table. And when I click away Access gives that field a default name of Expression 01. Now that names not particularly meaningful so I'm going to rename that and I'm just going to call it Full Name. Now for the next field I'm going to use the student's address and let's style it the same way. Let me drag that out very slightly and I'm going to give it a name of Address Line 01. And then finally, and again let's widen this column, going to have Address Line 02. So I think that looks okay. Let me try running this query. It says the existing table tblStudentAddresses will be deleted before you run this query. I'm going to say Yes to continue. So there we go. Let's look at our table and I'm going to widen out these columns and there we go. That looks pretty good to me.

Now let's go back into the query one more time and look at this in SQL view. So what we have here is select first name, space, last name as Address Line 01. City, space, state/province as Address Line 02. And insert them into table Student Addresses from the table Students. So reasonably simple to work out what it's doing there.

And if we jump back to the student addresses I just want you to note the field names in this table. So Address Line 01, Address Line 02, and Full Name.

Now the final thing I want to look at here is the Append query. Now I mentioned that normally before I delete records I save them somewhere and in fact the three short course records I deleted from the courses table I put into a separate table for safe keeping and I'm now going to add those back into the courses table. So these three records have been saved into the short courses table. And what I want to do is to append them to the course table.

So I'm going to jump up to Create and we're going to go into Query Design. I'm going to select the table Short Courses. I'm going to say I want all of the fields. I'm going to make this an

Append query. And I'm going to select to append them to a table in the current database, which is the courses table. And click Run. It says you're about to append three rows. And now if I go to the table Courses I can see that those courses that were under 30 weeks have now been added back into this table. And note that they've been added back in to the position that they came from.

So that is examples of all of those different query types. So over the last couple of sections you've had a read good look at all of the different types of action queries. It's time now to move on to the next module so please join me for that.

Section 13 – Packaging Your Application

Video: Steps in Packaging Access Application

Deb: Hello again and welcome back to our course on Access Advanced 2019. In this module we're going to talk about the steps you need to take in order to package an Access application. And these are the steps that you might consider when packaging your application for your users. So some steps we've covered already, such as providing a switchboard or navigation form but there are some others that are equally important and you really need to kind of take a look at these and select which ones are appropriate for you. So I'm going to divide these steps up into three groups. The first group are steps you might take in order to protect the data in your database, the second group are the ones that are taken to stop a user from breaking the database, and the third step includes steps you can take to make the database more user friendly.

Now during this course you've seen features for logging into a database, you've seen how to control access to certain functions but the most important protection you can put on a database is to encrypt it. Now if you don't encrypt your database anyone who knows just a little bit about Microsoft Windows can normally get in and see the data in your database. So encrypting it makes it a lot more difficult for them. Not impossible but a lot more difficult.

Now one of the most important things about encrypting a database is that you need to have it open in exclusive use. You then apply a password and once it's encrypted and protected with a password someone's going to need to know that password in order to see the data. Now I'm going to use a fairly simple password and if you try and open a database from the course files or exercise files folder and it asks you for a password then it's the same password and that is pa55wd.

Now what I'm going to do here is I'm going to close down my database and instead I'm going to first open it for exclusive use. So we're going to go to File, Open, I'm going to go to Browse, and I'm going to navigate to my database. And once I've found it and you can select any of these student databases in here, I'm going to click the dropdown next to Open and I'm going to say Open Exclusive. So what that means is that this database is now open exclusively to me so I know that nobody else is in it or concurrently get into it. Once I've opened it in exclusive use I can then encrypt with a password and that's a fairly straightforward thing to do. we go to File and on this Info page you can see the second option we have there is Encrypt with Password. And this is where we enter our password. So I'm going to use that one that I mentioned earlier, so pa55wd. We need to verify our password and click on OK.

Now you will get this little message popup which kind of sounds really terrible, Encrypting with a block cipher is incompatible, but you really don't need to worry about this. Just click on OK.

Now let me close this database down and I'm going to try and reopen it again. And you can see its now asking me for a password. So that's all working okay.

Now what if I wanted to decrypt this database or essentially do the reverse? Well, again very straightforward. We go to File, we go down to Info and this time we have Decrypt Database.

And again it's telling me I must have the database open for exclusive use. So let's close down again, I'm going to go to File, Open, I'm going to Browse, I'm going to select my database, Open Exclusive, it's asking me for the password, and there we go. My database is open. So again now it's open in exclusive use. I can decrypt it if I want to. Enter the password in again and there we go. That encryption has now been removed. So a very straightforward process.

Now as I mentioned in the case of a split database we will need to encrypt the backend as well. And in the case of the student database most of the data is in the backend so it's really important to encrypt that.

Now I've encrypted the backend database already and I did it in exactly the same way as I just showed you. So let me now try and open the frontend database.

Now when I open the frontend database I haven't had to enter a password because if you remember I removed it and although it all opens normally it I try and do something, so if I try and click on say this students icon I get an error. And that's because the backend database is encrypted with a password. So essentially I can't access any of that information that's sitting in the backend. So the way that I would resolve this is that I would need to delete the links to the tables in the backend database and recreate them.

So you can see here all of the linked tables. They have those little blue arrows next to them. So what I'm going to do is I'm going to delete this linked table, so tblStudents. I'm going to say delete. And it says Do you want to remove the link to the table tblStudents. I'm going to say Yes. And now what I need to do is relink the student table from the newly encrypted backend database. And now it's asking me for the password. So pa55wd. It's asking me what table I want to link. So tblStudents. And of course now having relinked that if I click on if I click on Students I can now see the student list.

Now one other thing you may do is that you might want to protect your VBA code from prying eyes. And the way you can see code if you want to is to go to the Create tab and anyone can simply click on Visual Basic and you will be presented with the VBA Editor where you can go in here, you can look at the code, you can make changes. But you can protect your code. So what you can do is go up to the Tools ribbon and you'll see one of the options in there is the name of your database. And you can see that we have a tab there called Protection. And what you can do is that you can lock the project for viewing and also apply a password. So I'm going to say Lock Project and I'm going to use the same password and click on OK.

Now I'm going to close the VBA Editor and I'm going to reopen the database. So now if a user is using this database if they click on Visual Basic they can essentially access the editor but they can't see any of the code. And if they do try to view any of the code they're going to be asked to enter a password. So if I type in that password, there we go. I now have my class modules back and I can see my code again. So a really important little option to be aware of.

Now if I want to remove that password all I need to do is go back into Tools, go back down to my database, go to that Project tab, and literally just uncheck the options in here and delete out the password. Okay? So really, really simple.

Now many of the features of protecting a database do kind of overlap but when users do have access you may want to do things like suppress access to the navigation pane. But there is also something else I've mentioned a few times and that is how to stop people seeing the ribbon.

So what we're going to do here is we're going to jump into Visual Basic and if you look on the left hand side in our modules one of them is Mod Hide Ribbon. And this contains a single function, Hide Ribbon. So it basically says don't show the ribbon. And what you can do is you can run this function during the AutoExec. So what I'm going to do here is I'm going to add this little command into the AutoExec. So I'm going to add a new action of Run Code. I'm going to add the function name and I want this to run first so I'm going to move this part of the code up to the beginning of the AutoExec. I'm then going to close the AutoExec and I'm going to save the changes. Now what I'm also going to do is I'm going to go into the Options for this database and I'm going to suppress the navigation pane. So I'll do that, I'll close down the database and then I'll reopen it. Now what you can see is no ribbon at the top and also no navigation pane. So we've really been able to lockdown what users can do in this database. They're pretty much limited to just the buttons that they have on this splash screen.

Now the final thing I'd like to talk about here is user friendliness. Now there are always helpful things that you can do in a database to really help out your users and one of those things might be something like a message box. One is also making sure that users have a version number to refer to. And as I said before about message boxes it's always good to make sure that there is really meaningful information in those message boxes so that users aren't confused when they come across an error or a prompt. They can really understand exactly what's going on. So I would suggest that when you are designing your databases think about how it's going to look to the user at every single stage and just make things as simple and as easy as possible.

I would also say it's really important to be consistent with regards to the colors that you use in your database, also the style and the arrangement of buttons. If a database looks as though its been carefully thought out and put together you'll find that your users will often have more respect for it. So really do try to be consistent.

Those are all the points I wanted to mention and this is now the end of this section so I will see you in the next one.

Video: Exercise 09

Deb: Hello again and welcome back to our course on Access Advanced 2019.

We’re down to Exercise 09 which is our last exercise. We’ve made it all the way through. And what I want you to do in this exercise is I want you to package up the Night Movies database. I want you to protect the front and backend with a password. And I want you to use your own password. And when I open up my answer I see no ribbon or navigation pane. I can just see my switchboard. I have an About button that gives me access to the version number. And of course I can exit using the Exit button.

So that is it. That is all that is required for this final exercise. I will see you in the next module.

Section 14 – Conclusion

Video: Course Close

Deb: Hello again and welcome back to this final module in our Access Advanced 2019 course.

So, we've come all the way to the end of our course and if you've made it through to the end I would hope that your skill level in Access is at a pretty high level at the moment and you can really start creating some well thought out databases utilizing the abundance of skills that you've learnt in this course.

I very much hope you have enjoyed working through this course as much as I have enjoyed delivering it and presenting it to you. And I very much hope to see you online again.

My name is Deb. Bye for now.