Video 5.3

The template below will help you structure your video using the PowerPoint slides and content flow given. You need to use this as a reference to create your video. Please **DO NOT FILL** the section below, it is for your reference only.

|  |
| --- |
| Plan your narration using the template below |

|  |  |
| --- | --- |
| **Introduction** | |
| **Action on Screen (Code /imagery to support the narration)** | **Narration (The corresponding explanation to the Action on Screen)** |
| [Mandatory Slide] This is the introduction slide of the section where the name of the section would be displayed. | *Hi and welcome to Microsoft Access – Building Meaningful Relationships (RDBMS)*  *section 5.3*  *Different Types of Joins* |
|  | *In this video, I’ll be covering the following:*   1. *Getting ready and prepare for a Joining table* 2. *Know the rules about the Data Types (& their Sizes where applicable)* 3. *Learn about the 3 types of Joins there are in Access* 4. *And Understand the impact over 3 different variations of Joins*     *Let’s get started…* |
|  | *You are now aware of having multiple tables in an Access database and why they are in small logical units.*  *To help connect and test joining tables together it would be useful to have the following ready and in place:*   * *Back up your database as a precautionary step – just in case you need to re-instate a version.   This can simply be done by closing the file and use the normal copy and paste file action in Windows.*   *This may seem a little excessive but…if your database is empty (with no records), you will ideally need to have some test data in tables in order to test the joins between tables so that you can see the potential impact between joins later on.*   * *Ensure you have set all primary and secondary keys for all the main fields that will be joined together across tables.  Remember, at least the primary key should be set for a unique identifier field for the ‘one-side’ of a relationship.  The secondary key is optional but recommended to be also set.* * *You backed up database may be empty and that’s fine,… but you really need to see the impact between tables with matching ID’s between tables.  Therefore, using the same ID Field’s data value across two tables with some general data for a record is recommended.* * *Quickly run the Compact & Repair utility provided in Microsoft Access.  This will reset and clear memory and any indexing, in particular where the ‘AutoNumber’ data type is used in your database giving a clean database state.  Nothing will actually change other than it will quickly close and re-open the same database file ready for you to continue.* * *You can call this utility by going to the ‘Database Tools’ tab on the Ribbon bar and click the ‘Compact & Repair Database’ icon.* |
|  | *In order to join two tables together, there are certain rules that should be observed, and the following will act as a guideline when getting your fields ready:*   * *Data Types:* * *The two fields being joined should be of the same data type. After all, it’s the same bit of data between the two … so wouldn’t it be?* * *Therefore, A Text data type should join with its corresponding Text data. You cannot join a Text to a Number data type or a Number to a Date/Time data type.* * *Field Sizes:* * *Should be really of the same Size where applicable.* * *Remember, the only two data types where the fields size applies is for a Text or a Number. All other data types do not have this option.*   *A Text and Number are the typical data types used which means their size will matter. Again, because it’s the same bit of information being used across two tables, it should be the same.*   * *However, there are some exceptions: A unique identifier field which is set as an ‘AutoNumber’ needs to join to a non-unique identifier which cannot be an ‘AutoNumber’ so it will allow you to join to a ‘Integer’ or ‘Long Integer’.*   *Also, when joining the same data type with different sizes, may cause what is known as casting and conversion challenges and may prevent you from correctly completing this task too.*   * *By changing a Number who size is Integer to another field whose size is a Byte (a smaller range) may not fit the larger range value to it.* * *Likewise, a Text size whose size may be 10 characters long will to fit to a field which is set as just 5 characters.*   *It therefore, makes sense to keep it the same.*   * *Fields Names?* * *Ideally, keep the field the names the same so it can be easily matched and viewed in other parts of your Access database. Names are not sensitive when joining fields together.* |
|  | *Access will work out the type of relationship for you based on where the primary key is set for a field and make it the ‘one-side’ state of a join.*  *There are three types of relationships that Access will interpret for you…they are:*   * *One-to-One – which is where both fields are set as the primary key and therefore, they are unique across both tables.   This may sometimes be used when wishing to extend a table based on the same unique ID record to help keep the number of fields to a manageable level, and only be called when it really is required as well as help to increase the performance for larger volume databases.* * *One-to-many (or many-to-one) – is the most common type of join in Access… and here in this example, it uses a unique Customer ID being called many times for each order they place (in the Orders table) by their corresponding Customer ID (which is a secondary key).  You will typically use this type of join.* * *The third type is called a many-to-many relationship and is in fact based on three tables.  You cannot have a many-to-many over two tables and is in effect a back-to-back of two one-to-many tables in a relationship.  In this example, a unique order (the order ID) joins to the ‘Order Details’ table for its many items which in turn is using the many product ID’s as uniquely stored on the Products table.* * *So, in essence there is a default many-to-many relationship between the Orders table and the Products table using the order Details table as the pivoting table in between.  Again, the primary key will drive this type of relationship.* |
|  | *Access can also be flexible and provide more powerful reporting results by manipulating the relationships between two tables.*  *Sometimes, there may not always be a matching value found across two tables and the impact of this can be either misleading or confusing when running reports.*  *So, there are three variations of join types to help overcome either missing or unknown values.*  *The first type of join is known as the ‘Inner-Join’ or ‘Equi-Join’*  *For example,*   * *Here’s our Customers table - the one-side* * *And the Orders table – the many-side* * *Where it overlaps,* * *we can associate this as the inner-join* * *where records are matched between the two tables* * *Any records in the customers table may not have any orders (yet) and will be excluded from this type of join in a query as well as…* * *Any records in the orders table that cannot be matched to an existing customer would also be excluded in a query too.*   *This is the default type of join when two tables are joined* |
|  | *The second type of join is one of the two Outer-Joins, known as a Left-Join (which is just a direction of a join).*  *Using the same two tables as an example,*   * *We have the Customers table being the one-side* * *The Orders tables being the many-side of the join* * *Where it overlaps this time, you will see that it includes not just the matching of records between the two but…* * *For the Left-Join…* * *It also includes all the many side’s order records which means…* * *It will show only matching customers and ALL orders whether there’s a matching Customer or not.* * *The Customers table still excludes any non-issued orders*   *a Left-join pushes the emphasis on the ‘many-side’ of a join showing all its records.* |
|  | *The reverse of this completes the final type of join which is still an outer-join and known as a Right-Join.*  *Using the same two tables as an example,*   * *We have the Customers table being the one-side* * *The Orders tables being the many-side of the join* * *Where it overlaps this time, you will see that it includes not just the matching of records between the two but…* * *For the Right-Join…* * *It also includes all the one side’s customer records which means…* * *It will show only matching orders and ALL customers whether there’s a matching Order or not.* * *The Orders table excludes any non-associated orders or unknown customers*   *a Right-join pushes the emphasis on the ‘one-side’ of a join showing all its records.* |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Steps or Tasks** | | |
| **Action on Screen (Code /imagery to support the narration)** | | **Narration (The corresponding explanation to the Action on Screen)** |
| **Summary** | | |
|  | *To summarise:*   1. *How to get ready and prepare for a Join* 2. *Know the rules about Data Types (& their Sizes)* 3. *Learned about the 3 types of Joins* 4. *Understand the impact between Inner and Outer Joins* | |
| [Mandatory slide] – Next Video | *In the next video, we can now implement Access joins for your table and queries.* | |