Forms, quires, and reports in access

Objectives : At the end of this lesson you shall be able to
• **explain database relationship and forms**
• **explain quires and reports in as access**

In a relational database (Access), the data in one table is related to the data in other tables. In general, tables can be related in one of three different ways: one-to-one, one-to-many or many-to-many. The relationship is used to cross reference information between tables.

One to One

In a one-to-one relationship each record in one table has at most one related record in another table.

In a one-to-one relationship, each record in Table A can have only one matching record in Table B, and each record in Table B can have only one matching record in Table A. This type of relationship is not common, because most information related in this way would be in one table. You might use a one-to-one relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table.

One to Many

A one-to-many relationship, often referred to as a "master-detail" or "parent-child" relationship.

A one-to-many relationship is the most common type of relationship. In a one-to-many relationship, a record in Table A can have many matching records in Table B, but a record in Table B has only one matching record in Table A.

Many to Many

In a many-to-many relationship, a record in Table A can have many matching records in Table B, and a record in Table B can have many matching records in Table A. This type of relationship is only possible by defining a third table (called a junction table) whose primary key consists of two fields the foreign keys from both Tables A and B. A many-to-many relationship is really two one-to-many relationships with a third table.

A many-to-many relationship means that for each record in one table there can be many records in another table and for each record in the second table there can be many in the first.

Many-to-many relationships can not be directly represented in relational database programs and have to be built by using two or more one-to-many relationships.

Defining relationships

You define a relationship by adding the tables that you want to relate to the Relationships window, and then dragging the key field from one table and dropping it on the key field in the other table.

•   The kind of relationship that Microsoft Access creates depends on how the related fields are defined:

•   A one-to-many relationship is created if only one of the related fields is a primary key or has a

•   A one-to-one relationship is created if both of the related fields are primary keys or have unique indexes.

•   A many-to-many relationship is really two one-to-many relationships with a third table whose primary key consists of two fields the foreign keys from the two other tables.

move a table that appears in the relationship map:

•   Place your mouse over the table you want to move.

•   Hold down the left mouse button, then drag the table to a new location.

•   Release the mouse button to drop the table in its new place.

Understanding the relationship map

The relationship map lists all of the tables that were selected to relate, as well as all of the fields that were previously set up for that table. Notice that the first field has a key icon next to it. This is the primary key for the table.

Fig 32

Primary and foreign keys

A primary key is the first field in each table of the database. You may recall that this field auto-numbers by default, so every record in the table has its own unique number to identify it. Access uses this number to quickly pull information together when you run queries or reports, which are covered later.

In the example above, the primary key for the Customers table is Customer ID, the primary key for the Orders table is Order ID, and the primary key for the Books table is Book ID.

A foreign key is a field that is the primary field in its own table but that shows up in another table. If you look closely at the Orders table, the fields Customer ID and Book ID appear there, as well as in their own respective tables. These fields are the primary key in their own tables, but in the Orders table, they are considered foreign keys.

Fig 33

Forms

A form is a database object that you can use to enter, edit, or display data from a table or a query. You can use forms to control access to data, such as which fields of data are displayed. For example, certain users may not need to see all of the fields in a table. Providing those users with a form that contains just the necessary fields makes it easier for them to use the database.

create a form with a single click,

1. Open the table or query upon which you want to base the form.

2. To create a form on which all fields from the underlying table or query are placed, displaying one record at a time, on the Create tab, click Form.

Fig 34

Forms in Access are like display cases in stores that make it easier to view or get the items that you want. Since forms are objects through which you or other users can add, edit, or display the data stored in your Access desktop database, the design of your form is an important aspect. There's a lot you can do design-wise with forms in Microsoft Access. You can create two basic types of forms -

- Bound forms

- Unbound forms

Bound Forms

• Let us now understand what Bound Forms are ?

• Bound forms are connected to some underlying data source such as a table, query, or SQL statement.

• Bound forms are what people typically think of when they think of the purpose of a form.

• Forms are to be filled out or used to enter or edit data in a database.

• Examples of bound forms will typically be what users use to enter, view or edit data in a database.

Unbound Forms

Let us look into Unbound Forms:

• These forms are not connected to an underlying record or data source.

• Unbound forms could be dialog boxes, switch boards, or navigation forms.

• In other words, unbound forms are typically used to navigate or interact with the database at large, as opposed to the data itself.

Types of Bound Forms

There are many types of bound forms you can create in Access. Let us understand the types ?

Single Item Form

This is the most popular one and this is where the records are displayed - one record at a time.

Multiple Item Form

This displays multiple records at a time from that bound data source.

Split Form

The form is divided into halves, either vertically or horizontally. One half displays a single item or record, and the other half displays a list or provides a datasheet view of multiple records from the underlying data source.

Form Controls

Every object on a form or report is a control and has its own set of properties (displayed in the Property list) that determine the characteristics of the data it contains. Some controls are linked to fields in the underlying table or query and some are merely text or graphical objects not related to any data source. Two of the most commonly used controls in a form or report are text boxes and labels.

1) Label: displays unchanging or informational text often times as headings or titles. Labels do not come from a source in the database such as a field or expression. This is called an unbound control because it is not bound to any data source.

2) Text Box: displays data as text and is abound control. A bound control displays information that is stored in a field from the data Text boxes are used to display, enter, and update values from fields in your database.

IT & ITES : COPA - (NSQF Level - 4): Related Theory for Exercise 1.9.62 - 1.9.66

Form Sections

Forms consist of header, footer, and detail sections. In forms, there are two types of headers/footers. Form headers and footers are visible on every page of the form. Page headers/footers only appear on a printed copy of a form. There are never visible in Form View. Information about individual records appears in the detail sections.

Sizing Sections

Adjust the size of the area for any of the sections (headers, footers, detail), by dragging the horizontal border for that section with your mouse. To adjust line spacing in the detail section, drag the Detail border. There is no other line spacing control.

Working with Queries

Queries are questions you ask of your database. They allow you to select certain fields out of a table, or pull together data from various related tables and display it together. Queries can be used to perform calculations, and to edit the data in your tables. Queries can also form the basis of forms and reports.

Query Results

*   If a query contains fields from two or more related tables, the results will only include records that have related records in all the tables. (In other words, in the query shown above, customers without records in the Orders table will not appear in the query results.) To display such records, use an outer join (see page 6) in your query.

*   WARNING! Any change you make in a query (adding, deleting or editing data) WILL BE CHANGED IN THE TABLES TOO!!!

Sort query results

In Design View, click the Sort row in the column you wish to sort by. A dropdown arrow appears; choose Ascending or Descending. If you sort by more than one column, the sorts will be done left-to-right.

Create a concatenated field in a query

Concatenated fields combine information from two or more fields into one. Ex: Combine first and last names into one field, called "FullName". In Design View, click in a blank column and type the following:

FullName:[FirstName]&" "&[LastName]

(Type a space between the two sets of quotation marks, and use square brackets around the field names.)

Create a calculated field in a query

In Design View, click in a blank column and type a mathematical expression, typing field names in square brackets. Any of the normal mathematical operators (+ - * / ^) can be used. If desired, give the expression a name as in the sample below. The following expression calculates a total cost by multiplying quantity times price:

Item Total:[Quantity]*[Price]

Use the Expression Builder to create calculated or concatenated fields

Fig 35
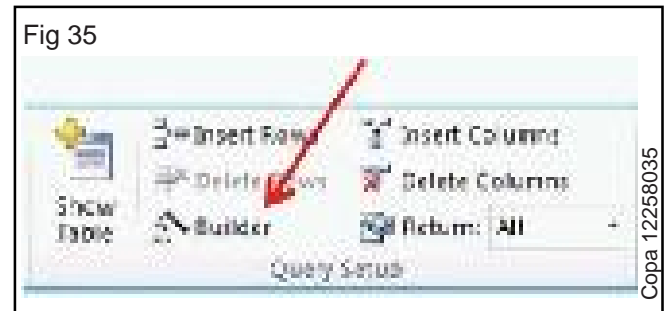
You can simplify the process of creating concatenated fields and other expressions with the Expression Builder. Create a blank column in your query grid and then click the Builder button in the Query Setup group on the Query Design Ribbon. The Expression Builder opens.

Type the expression in the large box at the top. One advantage is that you have way more room to type and see your expression than you have at the top of a query grid column!

A second advantage is that you can add fields to the expression by double clicking them in the Expression Categories column rather than by typing them. This removes the possibility of typos.

Fig 36
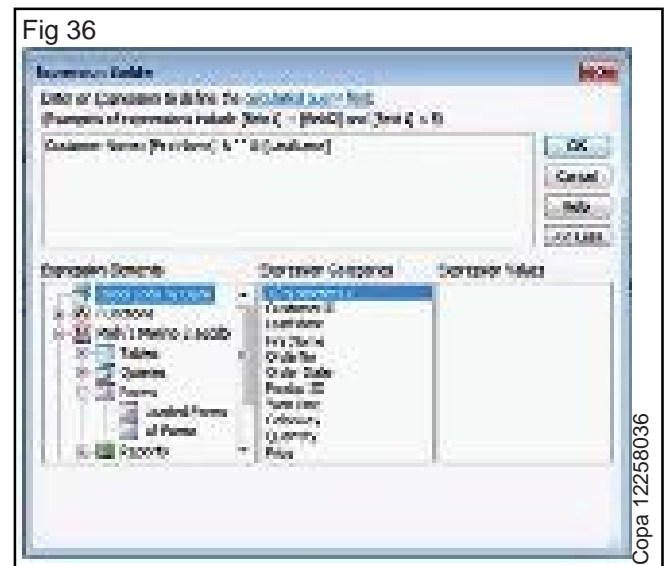
Specify criteria in a query (Filtering)

In Query Design View, click in the criteria row for the field you want to filter, and type the criteria (ex: type "OH" in the criteria row of the State field). When you run the query, only records matching the criteria.

Fig 37
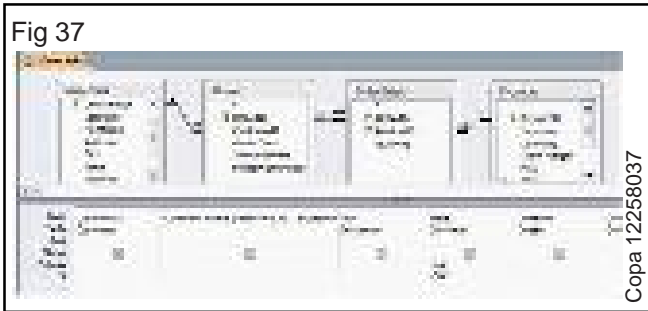
Copa 12258037

Using Multiple Criteria

You can type criteria in more than one field.

- Criteria typed on the same line must all be met for a record to be displayed ("AND" logic). Ex: "OH" in the state field and "Smith" in the LastName field pulls up only Ohio-dwelling Smiths.

- If criteria are typed on different lines, records matching any of them will be retrieved ("OR" logic). Ex: "WA" and "OH" on different lines in the State field retrieves records for both states.

Criteria Ranges

Criteria ranges broaden and customize a search. Type a criteria range instead of a single value in the Criteria row of the query grid.

Criteria	Example	Description and Result
<	<14	Finds records where the field is less than a certain value. Records will have values under 14.
<=	<= #2/2/2006#	Finds records where the field is less than or equal to a certain value. Records will have dates on or before February 2, 2006 Note: dates are enclosed between # signs.
> >=	>100 >=100	Similar to above: Finds records where the value is greater than 100. The second expression displays records where the value is greater than or equal to 100.
=	="Diana"	Finds records where the value in the field is Diana. Note: text is enclosed in double quotes.

IT & ITES : COPA - (NSQF Level - 4): Related Theory for Exercise 1.9.62 - 1.9.66

Expressions like these may be combined - for example, a criteria of <49 Or>=100 would return records with values below 49, or 100 or above, but not ones with values between 49 and 99.99.

Between __ And __	Between 1 and 4	**Finds records where the field is between the endpoints.**
Is Null	**Is Null**	Finds records where the field is empty.
Is Not Null	**Is Not Null**	Finds records where the field is not empty.
Like	**Like U***	Finds records where the criteria match parts of the field.
	Like "[U-X]*"	The * wildcard stands for any number of characters, and
	Like "*Korea*"	the ?wildcard stands for a single character.
	Like "*ina"	The sample criteria will return records where the field
	Like "Chi??"	1) starts with U; 2) starts with U, V, W or X, 3) contains
	Like "*4.99"	the letters 'Korea'; 4) ends in the letters 'ina' as in ballerina; 5) has five letters and begins with the letters 'Chi' as in Chile or China; 6) is a number and ends in the digits '4.99' as in 24.99.

Create a parameter query

Parameters make queries flexible by letting users choose their own criteria each time they run the query. In the criteria box, type a phrase in square brackets; Access will use this phrase to prompt the user for a value. Ex: In a State field, type [Enter a state]. If the user types CA, only California records will be retrieved.

Specify a range in a parameter query

In the criteria box, type the following:

Between [____] And [____]

Inside the brackets, type text telling the user what kind of information to enter.

Inner Joins vs. Outer Joins

Excel normally uses inner joins in its queries. These queries return only those rows from both tables in the join that match on the joining field (e.g., customers in the Customers table with corresponding orders in the Orders table.)

If you need to display all customers, with or without orders, as well as any available order information, you need to use an outer join.

In the query tray, double click the join line between two related tables. The Join Properties dialog box will appear:

Fig 38
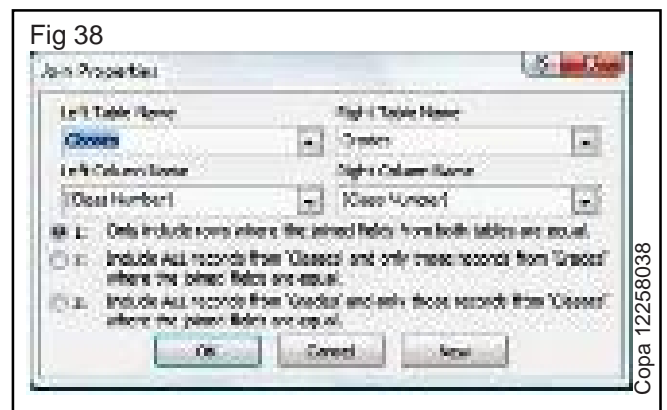
Select button 2 or 3 depending on which field's records you want to display completely.

Action Queries

Action queries modify groups of records in one fell swoop. Types of action queries include:

- Update Queries (modify all records in a table that meet specific criteria - ex: change the tax rate for all Ohio customers from 5.75% to 6%)

- Delete Queries (remove records from a table - ex: delete all customers whose last order date is more than 2 years ago)

- Append Queries (add records to an existing table - ex: at the end of the year, add the year's orders to the end of an archive table)

- Make-table Queries (create a new table - ex: extract a subset of records into a new table, for a subordinate who needs Ohio customers only)

Other Specialized Queries

Instructions for creating and running these special query types can be found in the Access help manual by clicking the Help button in the upper right corner of the screen (Internet connection required).

Find Duplicates query: Lets you locate duplicate records in a table by displaying a field containing duplicate values alongside other fields for comparison, to help you tell if the record is really a duplicate.

Find Unmatched query: Lets you compare two tables and identify records in one of the tables that have no corresponding records in the other table.

Crosstab query: This is a special type of Totals query that presents summary information in a compact format that is similar to a spreadsheet.

Union query: Lets you review all of the data that is returned by several similar select queries together, as a combined set.