

Access 2007

Intermediate



- Courses never cancelled
- UK's most regular instructor led courses
- Onsite/bespoke training UK wide
- Free after training Access support
- No fuss online booking
- Complete downloadable reference material



E&OE

Best Training reserves the right to revise this publication and make changes from time to time in its content without notice.

Quick reference: Access shortcut keys

Command	Keystroke
Add new record	Ctrl +
Builder	Ctrl-F2
Check/uncheck box or option button	spacebar
Close	Ctrl-W
Copy	Ctrl-C
Cut	Ctrl-X
Cut current line and copy to Clipboard	Ctrl-Y
Cycle through sections	F6/Shift-F6
Cycle through tab of each object's type (toggle)	Ctrl-Tab/Shift-Ctrl-Tab
Database window	F11
Delete current record	Ctrl -
Edit/Navigation mode (toggle)	F2
Exit subform and move to next/previous field in next record	Ctrl-Tab/Shift-Tab
Extend selection to next/previous record	Shift-Dn/Up
File/Save As	F12
Find	Ctrl-F
Find Next	Shift-F4
Find Previous	Shift-F3
GoTo	Ctrl-G
Insert current date	Ctrl ;
Insert current time	Ctrl :
Insert default value	Ctrl-Alt-spacebar
Insert new line	Ctrl-Enter
Insert value from same field in previous record	Ctrl '
Menu bar	F10
Move to beginning/end of multiple-line field	Ctrl-Home/End
Move to current field in first/last record (Navigation mode)	Ctrl-Up/Dn
Move to first field in first record (Navigation mode)	Ctrl-Home
Move to first/last field in current record (Navigation mode)	Home/End
Move to last field in last record (Navigation mode)	Ctrl-End
Move to left edge of page	Home or Ctrl-Left
Move to page number/record number box	F5
Move to right edge of page	End or Ctrl-Right
Next window	Ctrl-F6
Open combo box	F4
Open in Design view	Ctrl-Enter

Quick reference: Access shortcut keys

Command	Keystroke
Paste	Ctrl-V
Print	Ctrl-P
Property sheet	Alt-Enter
Refresh combo box	F9
Replace	Ctrl-H
Requery underlying tables in subform	Shift-F9
Save current record	Shift-Enter
Screen left/right	Ctrl-PgUp/PgDn
Select/unselect column (Navigation mode)	Ctrl-spacebar
Switch to Form view	F5
Turn on Move mode	Ctrl-F8
Undo	Ctrl-Z
Undo previous extension	Shift-F8
Zoom box	Shift-F2

Course Objectives

1. Create relational databases
2. Work with related tables
3. Define data entry rules
4. Use advanced query features
5. Create advanced queries
6. Create advanced form design
7. Use advanced report features

Creating relational databases

Unit 1 objectives

- Normalize tables to reduce data redundancy
- Set relationships between tables
- Implement referential integrity between related tables

Creating relational databases

Unit 1 objectives

- Normalize tables to reduce data redundancy
- Set relationships between tables
- Implement referential integrity between related tables

- # Creating relational databases
- ## Unit 1 objectives
- Normalize tables to reduce data redundancy
 - Set relationships between tables
 - Implement referential integrity between related tables

Your notes: Unit 1

[illegible]

[illegible]

Visual summary: Unit 1 objectives

Normalization

First normal form:

- Break fields down into smallest parts.

Employee_code	Name
E-01	Malcolm Pingault
E-02	Shannon Lee
E-03	Melinda McGregor
E-04	James Overmire

Employee_code	First_name	Last_name
E-01	Malcolm	Pingault
E-01	Malcolm	Pingault
E-01	Malcolm	Pingault
E-01	Malcolm	Pingault
E-02	Shannon	Lee
E-02	Shannon	Lee
E-02	Shannon	Lee
E-02	Shannon	Lee

- Combine fields containing redundancy (Build tall not wide.)

Employee_code	Name	Region	Project_number_1	Earnings_1	Project_number_2	Earnings_2
E-01	Malcolm Pingault	East	SL-99-01-01	\$20,000	SL-99-02-02	\$18,000
E-02	Shannon Lee	South	SL-99-01-01	\$21,000	SL-99-02-02	\$20,000
E-03	Melinda McGregor	West	SL-99-01-03	\$25,000	SL-99-02-01	\$23,750
E-04	James Overmire	North	SL-99-01-03	\$32,960	SL-99-02-01	\$26,250

Employee_code	First_name	Last_name	Region	Project_number	Earning
E-01	Malcolm	Pingault	East	SL-99-01-01	\$20,000
E-01	Malcolm	Pingault	East	SL-99-02-02	\$18,000
E-01	Malcolm	Pingault	East	SL-99-03-01	\$19,000
E-01	Malcolm	Pingault	East	SL-99-04-03	\$28,500
E-02	Shannon	Lee	South	SL-99-01-01	\$21,000
E-02	Shannon	Lee	South	SL-99-02-02	\$20,000
E-02	Shannon	Lee	South	SL-99-03-03	\$20,500
E-02	Shannon	Lee	South	SL-99-04-02	\$30,750

Visual summary: Unit 1 objectives

Second normal form:

- Find all fields related to only part of composite primary key

Sales_projects_done		
	Field Name	Data Type
?	Employee_code	Text
	First_name	Text
	Last_name	Text
	Region	Text
?	Project_number	Text
	Earnings	Number

- Group these fields into another table

Sales_projects_done						
	Employee_code	First_name	Last_name	Region	Project_number	Earnings
	E-01	Malcolm	Pingault	East	SL-99-01-01	\$20,000
	E-01	Malcolm	Pingault	East	SL-99-02-02	\$18,000
	E-01	Malcolm	Pingault	East	SL-99-03-01	\$19,000
	E-01	Malcolm	Pingault	East	SL-99-04-03	\$28,500
	E-02	Shannon	Lee	South	SL-99-01-01	\$21,000
	E-02	Shannon	Lee	South	SL-99-02-02	\$20,000
	E-02	Shannon	Lee	South	SL-99-03-03	\$20,500
	E-02	Shannon	Lee	South	SL-99-04-02	\$30,750

Sales_employees				
	Employee_code	First_name	Last_name	Region
	E-01	Malcolm	Pingault	East
	E-02	Shannon	Lee	South
	E-03	Melinda	McGregor	West
	E-04	James	Overmire	North

Sales_employees		
	Field Name	Data Type
?	Employee_code	Text
	First_name	Text
	Last_name	Text
	Region	Text

- All remaining fields should directly relate to both of the fields of the composite key and so forming a table which is in 2nd Normal Form.

Visual summary: Unit 1 objectives

Third normal form:

- Table is in the first normal form
- All fields relate to the primary key

Office_data						
Employee_code	First_name	Last_name	Region	Dept_name	Dept_code	
E-01	Malcolm	Pingault	East	National sales	NSL	
E-02	Shannon	Lee	South	National sales	NSL	
E-03	Melinda	McGregor	West	National sales	NSL	
E-04	James	Overmire	North	National sales	NSL	
E-05	Roger	Williams	West	Human resources	HR	
E-06	Annie	Philips	West	Human resources	HR	

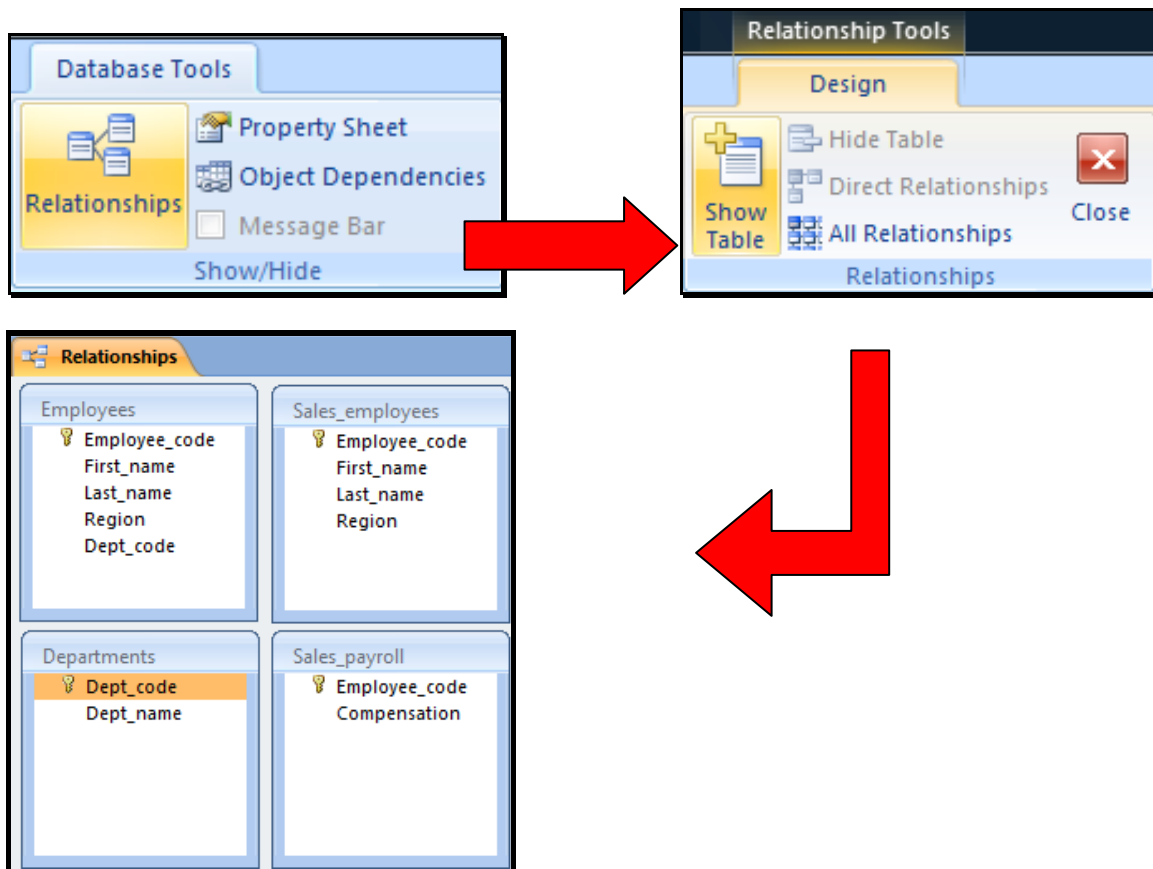
Sales_projects_earnings		
Employee_code	Project_number	Earnings
E-01	SL-99-01-01	\$20,000
E-01	SL-99-02-02	\$18,000
E-01	SL-99-03-01	\$19,000
E-01	SL-99-04-03	
E-02	SL-99-01-01	
E-02	SL-99-02-02	
E-02	SL-99-03-03	
E-02	SL-99-04-02	

Sales_projects_earnings	
Field Name	Data Type
Employee_code	Text
Project_number	Text
Earnings	Number

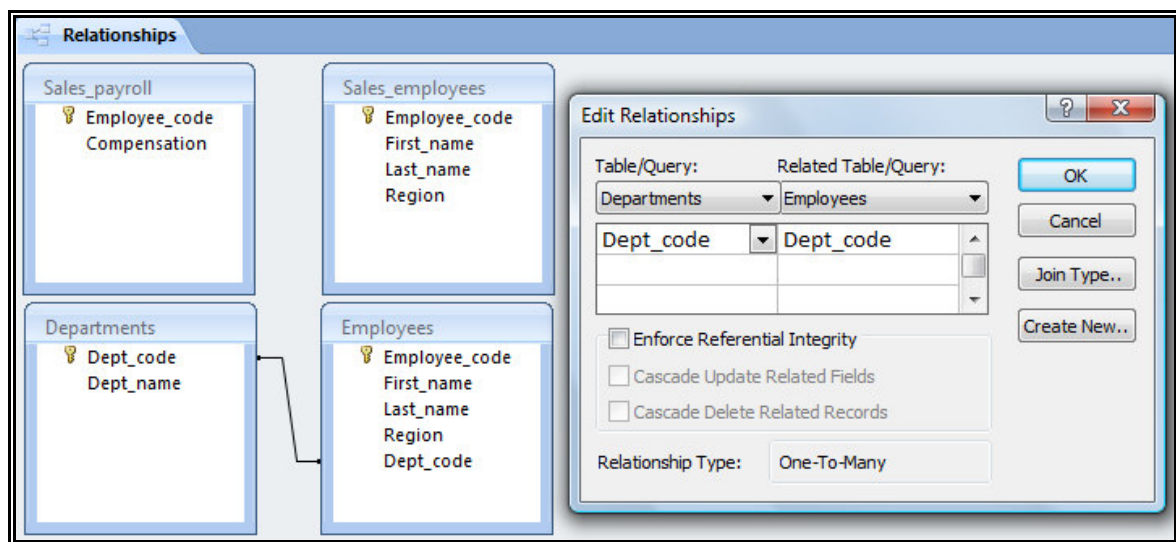
Employees				
Employee_code	First_name	Last_name	Region	Dept_code
E-01	Malcolm	Pingault	East	NSL
E-02	Shannon	Lee	South	NSL
E-03	Melinda	McGregor	West	NSL
E-04	James	Overmire	North	NSL
E-05	Roger	Williams	West	HR
E-06	Annie	Philips	West	HR

Departments	
Dept_code	Dept_name
ACCTS	Accounting
CUST_SUPP	Customer support
HR	Human resources
MKTG	Marketing
NSL	National sales

Relating Tables

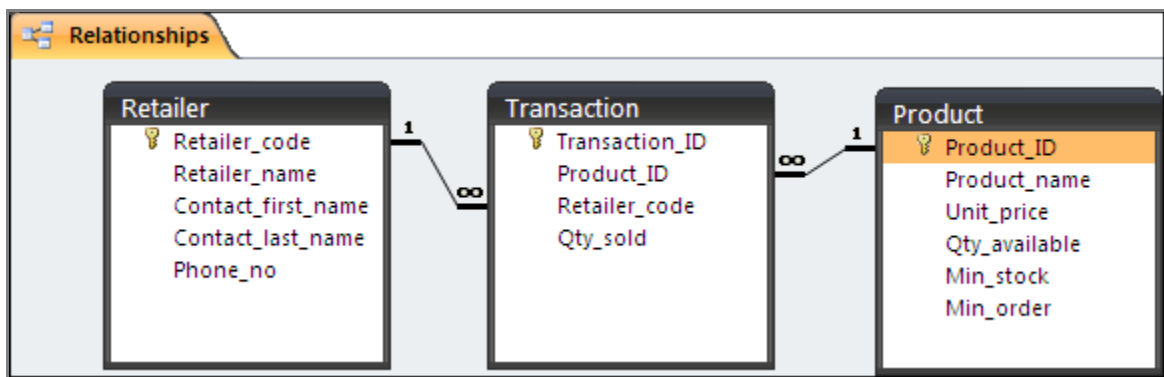


By dragging similar fields from different tables over each other a relationship can be built.



Unit 1 Practice Activity

1. Open **Relational Databases.accdb**.
2. Open the Relationships window.
3. Create a one-to-many relationship between the Retailer and Transaction tables. The relationship should include all of the records from Retailer and only those records from Transaction where the joined fields are equal.
4. Create a many-to-many relationship between the Retailer and Product tables by creating a one-to-many relationship between the Product and Transaction tables.
5. Save the relationships.
6. Enforce cascading deletes between the Retailer and Transaction tables.
7. Update the Relationships.
8. Test cascading deletes between the Retailer and Transaction tables.
9. Enforce Cascading updates between the Retailer and Transaction tables.
10. Update the relationship.
11. Test cascading updates between the Retailer and Transaction tables.
12. Close the database.



Online support forum and knowledge base

www.microsofttraining.net/forum

Visit our forum to have your questions answered by our Microsoft certified trainers.

Working with related tables

Unit 2 objectives

- Use the Lookup Wizard to create a Lookup list field
- Use Design view to modify Lookup field properties
- Use a subdatasheet to add data to related tables

Working with related tables

Unit 2 objectives

- Use the Lookup Wizard to create a Lookup list field
- Use Design view to modify Lookup field properties
- Use a subdatasheet to add data to related tables

- # Working with related tables
- ## Unit 2 objectives
- Use the Lookup Wizard to create a Lookup list field
 - Use Design view to modify Lookup field properties
 - Use a subdatasheet to add data to related tables

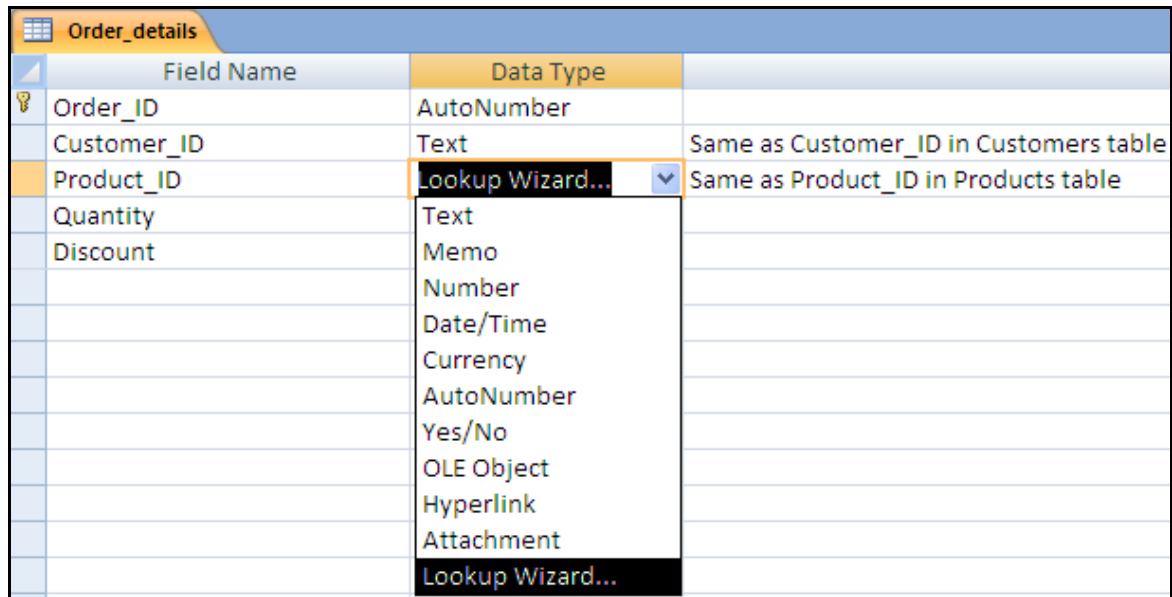
Your notes: Unit 2

[illegible]

[illegible]

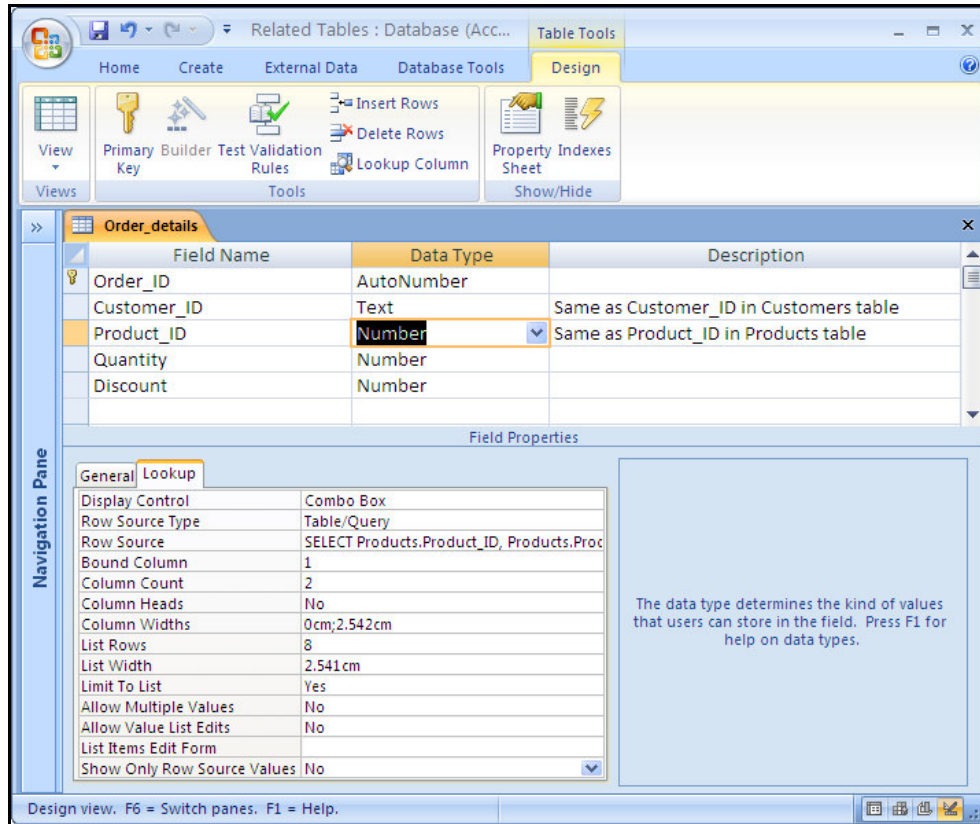
Visual summary: Unit 2 objectives

Using the Lookup Wizard



Field Name	Data Type	
Order_ID	AutoNumber	
Customer_ID	Text	Same as Customer_ID in Customers table
Product_ID	Lookup Wizard...	Same as Product_ID in Products table
Quantity	Text	
Discount	Memo	
	Number	
	Date/Time	
	Currency	
	AutoNumber	
	Yes/No	
	OLE Object	
	Hyperlink	
	Attachment	
	Lookup Wizard...	

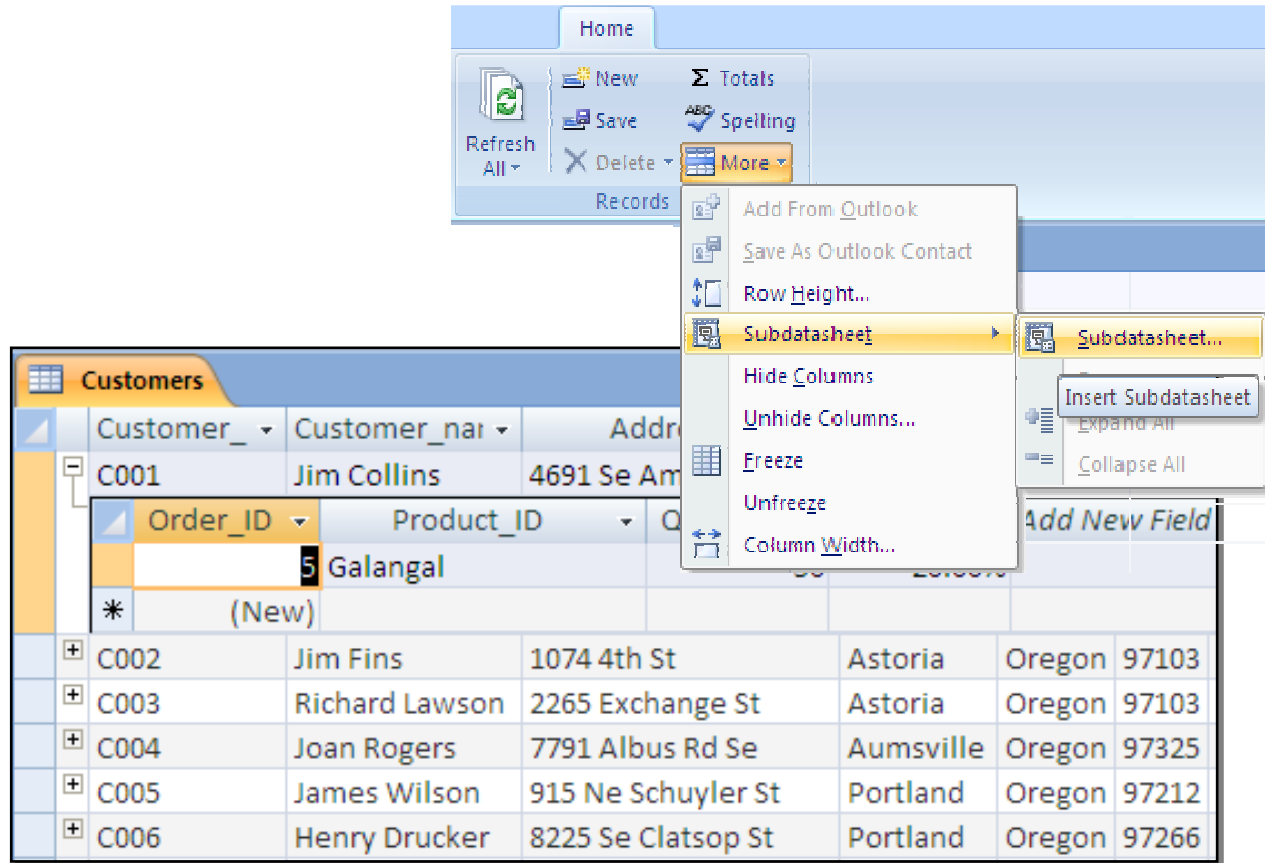
Modifying Lookup Fields



Visual summary: Unit 2 objectives

Adding Data to Related Tables

Using a subdatasheet.



Unit 2 Practice Activity

1. Open **Working_with_related_tables.accdb**.
2. In the Transaction table, create a Lookup list for Retailer_code by using the Retailer_code field from the Retailer table.
3. In the Transaction table, create a Lookup list for Product_ID to display product names from the Product table.
4. Update the table.
5. Add a record to the Transaction table by using the Lookup lists as shown below:

Transaction_ID: T011

Product_name: Cinnamon Ground

Retailer_code: R005

Qty_sold: 350

6. Update the table.
7. Change the caption of the Product_ID column in the Transaction table to Product.
8. Update the table.
9. View the contents of the Transaction table.
10. Compare the table with the example shown below.

Transaction			
Transaction_ID	Product_ID	Retailer_code	Qty_sold
T001	Annatto Seed	R001	150
T002	Cloves	R002	400
T003	Asafoetida Powder	R002	110
T004	Basil Leaf	R003	140
T005	Anise Seeds	R004	120
T006	Cinnamon Ground	R005	100
T007	Lemon Grass	R006	100
T008	Italian Parsley	R007	321
T009	Anise Seeds	R008	200
T010	Basil Leaf	R004	250
T011	Cinnamon Ground	R005	350

Practice activity continues on the following page

Unit 2 Practice Activity continued

11. Close the table.
12. Insert the Transaction table as a subdatasheet in the Product table.
13. Add a record in the Transaction table for the Product_ID P005 in the product table as shown below:
Transaction_ID: T012
Retailer_code: R007
Qty_sold: 400
14. Update and close the table.
15. View the record in the Transaction table that you entered in Step 13.
16. Close the table.
17. Close the database.

Product				
	Product_ID	Product_name	Unit_price	Qty_available
+	P004	Asafoetida Powder	\$1.49	700
-	P005	Anise Seeds	\$1.49	900
		Transaction_ID	Retailer_code	Qty_sold
		T005	R004	120
		T009	R008	200
		T012	R007	400
		*		
+	P006	Basil Leaf	\$1.89	1500
+	P007	Carob Pods	\$2.49	800



Online support forum and knowledge base

www.microsofttraining.net/forum

Visit our forum to have your questions answered by our Microsoft certified trainers.

Defining data entry rules

Unit 3 objectives

- Use the Input Mask Wizard to create input masks for fields
- Use Design view to set properties for a field
- Use Design view to set validation rules for entering data in a field

- Use the Input Mask Wizard to create input masks for fields
- Use Design view to set properties for a field
- Use Design view to set validation rules for entering data in a field

- [illegible]

[illegible]

Visual summary: Unit 3 objectives

Working with Input Masks

The screenshot displays the Microsoft Access interface. At the top, a tab labeled 'Customers' is active. Below it is a table structure showing the following fields and data types:

Field Name	Data Type
Customer_ID	Text
Customer_name	Text
Address	Memo
City	Text
State	Text
Zip	Text
Phone	Text
Fax	Text

Below the table structure is the 'Field Properties' window. The 'General' tab is selected. The 'Input Mask' property is highlighted with a red circle and is set to 'L000;0;#'. Other properties visible include:

- Field Size: 50
- Format: (empty)
- Caption: (empty)
- Default Value: (empty)
- Validation Rule: (empty)
- Validation Text: (empty)
- Required: Yes
- Allow Zero Length: No
- Indexed: Yes (No Duplicates)
- Unicode Compression: Yes
- IME Mode: No Control
- IME Sentence Mode: None
- Smart Tags: (empty)

Setting Properties

- Required
- AllowZeroLength
- Field Size

The screenshot displays the Microsoft Access interface. At the top, a tab labeled 'Customers' is active. Below it is a table structure view with two columns: 'Field Name' and 'Data Type'. The table contains the following fields:

Field Name	Data Type
Customer_ID	Text
Customer_name	Text
Address	Memo
City	Text
State	Text
Zip	Text
Phone	Text
Fax	Text

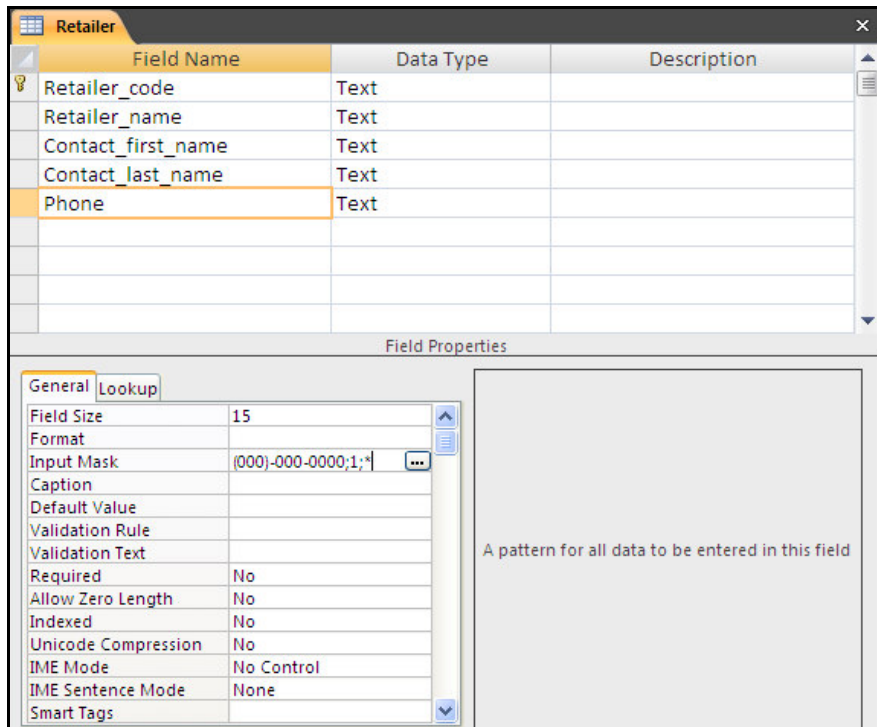
Below the table structure is the 'Field Properties' task pane. It has two tabs: 'General' and 'Lookup'. The 'General' tab is selected. The properties listed are:

Property	Value
Field Size	10
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

Red circles highlight the 'Field Size' property (set to 10) and the 'Required' and 'Allow Zero Length' properties (both set to 'Yes').

Unit 3 Practice Activity

1. Open **Defining_data_entry_rules.accdb**.
2. Create an input mask for the Phone field in the Retailer table. The input mask should ensure that only 10 numbers are entered in the field and should resemble the input mask shown in the example below. Ensure that although the input mask will display numbers in a specific format in the table, only the numbers (not the literal characters) are stored in the table. Use an asterisk (*) as a placeholder.



3. Update the table.
4. Add a new record in the Retailer table with the following details:
Retailer_code: R013
Retailer_name: Spice Life
Contact_first_name: Greg
Contact_last_Name: Jefferson
Phone: 4156489301
5. Update and close the table.
6. Set the Required property for the Product_name field so the Product table to Yes.

Practice activity continues on the following page

Unit 3 Practice Activity continued

7. Set the Default Value property for the Min_order field of the Product table to 50.
8. Create an input mask for the Product_ID field of the Product table to ensure that the data in the field always starts with a letter followed by three digits. Use a placeholder of your choice.
9. Create a validation rule and the appropriate validation text in the Product table to ensure that the Product_ID field contains only four characters with 'P' as the first character.
10. Update the table.
11. Test the properties and validation rule by entering sample data.
12. Update and close the table.
13. Close the database.



Online support forum and knowledge base

www.microsofttraining.net/forum

Visit our forum to have your questions answered by our Microsoft certified trainers.

Using advanced query features

Unit 4 objectives

- Use Design view to create join queries
- Create queries to add, delete, and modify data in tables and to create new tables

Using advanced query features

Unit 4 objectives

- Use Design view to create join queries
- Create queries to add, delete, and modify data in tables and to create new tables

- # Using advanced query features
- ## Unit 4 objectives
- Use Design view to create join queries
 - Create queries to add, delete, and modify data in tables and to create new tables

Your notes: Unit 4

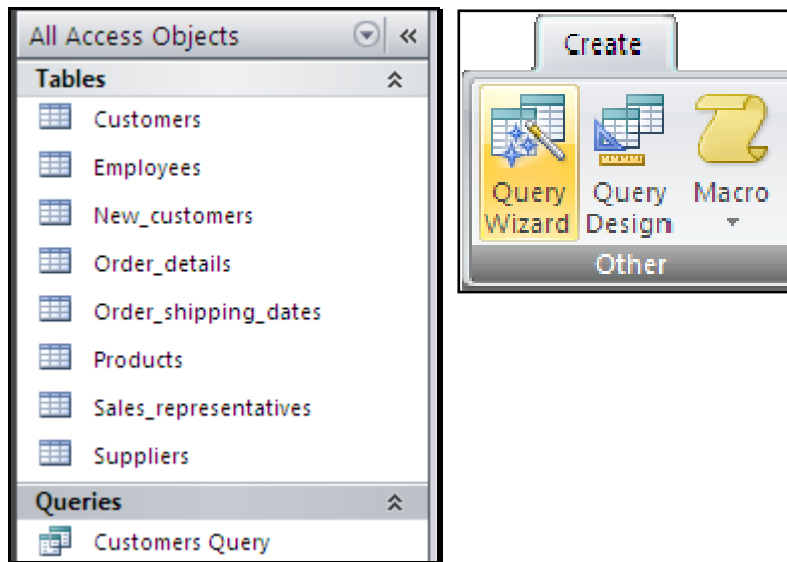
Your notes: Unit 4

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

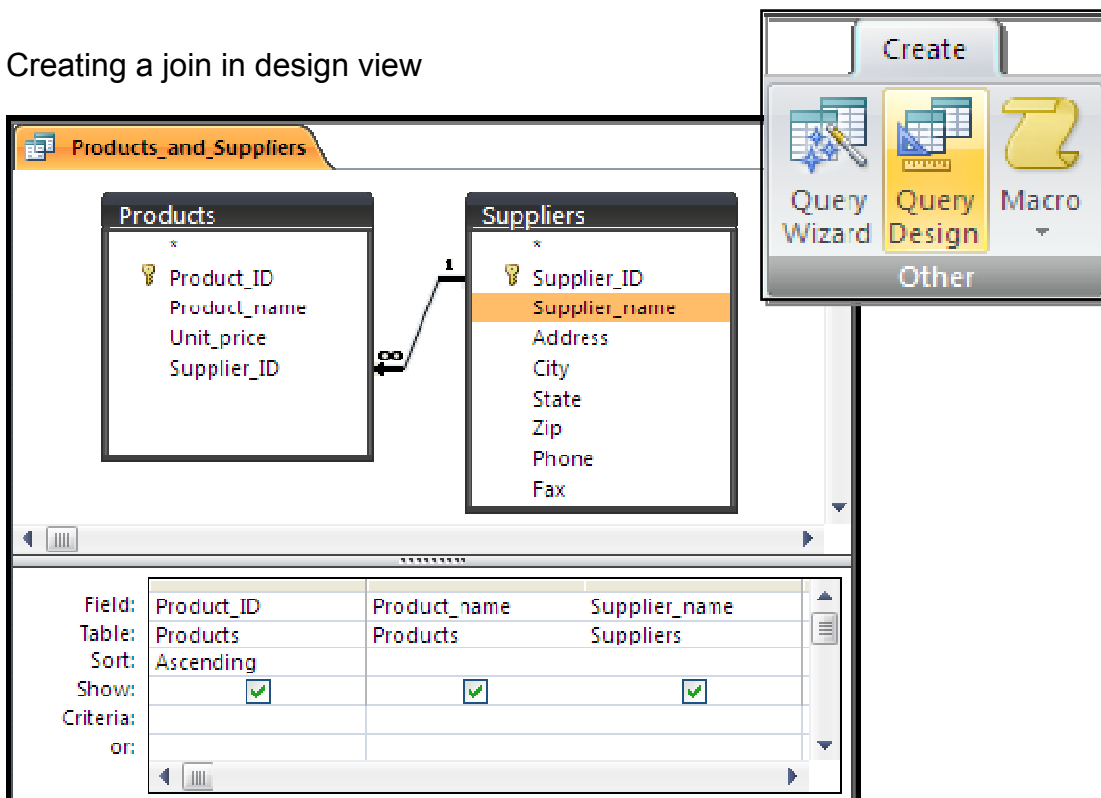
Visual summary: Unit 4 objectives

Joining Tables in Queries

Simple Query Wizard

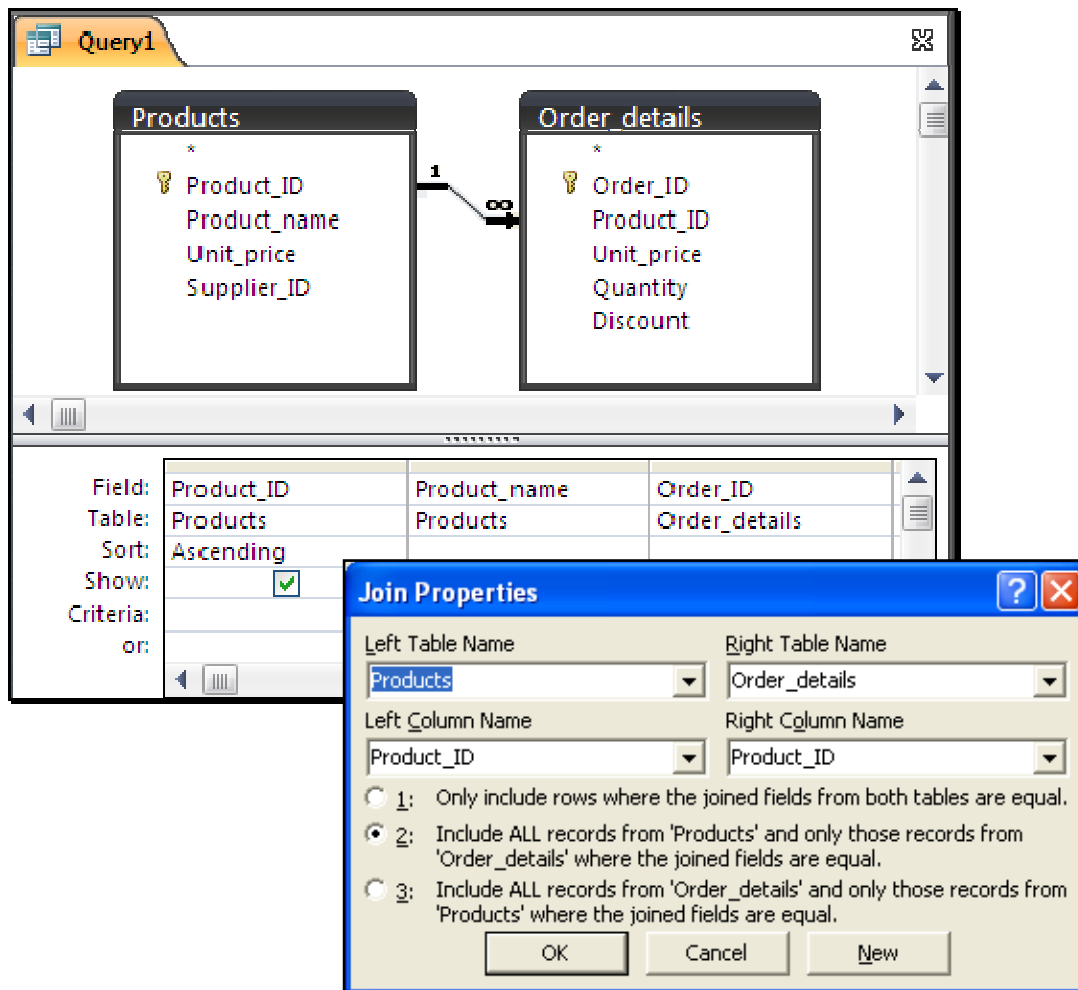


Creating a join in design view



Visual summary: Unit 4 objectives

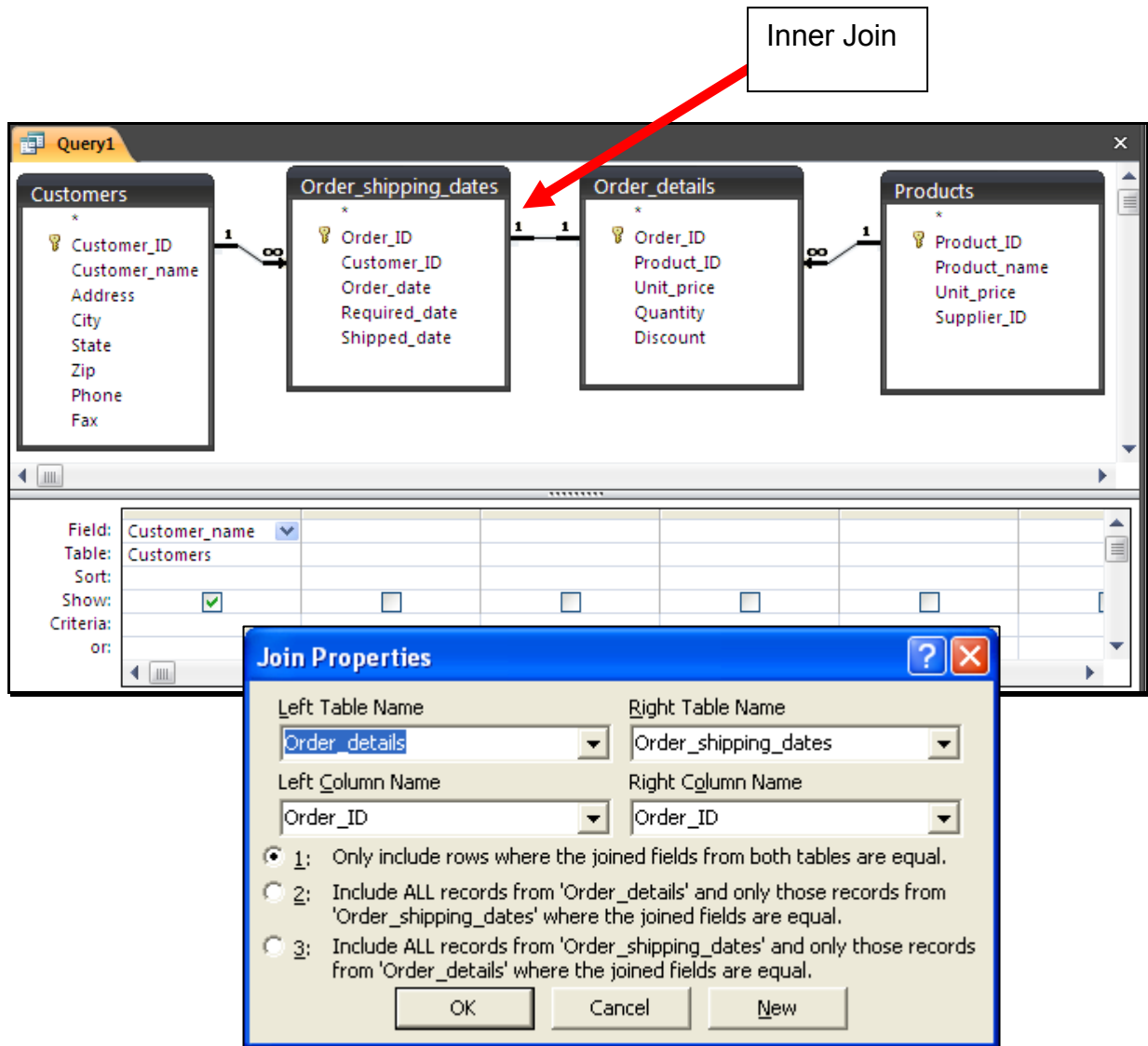
Outer Joins



Visual summary: Unit 4 objectives

Inner Joins

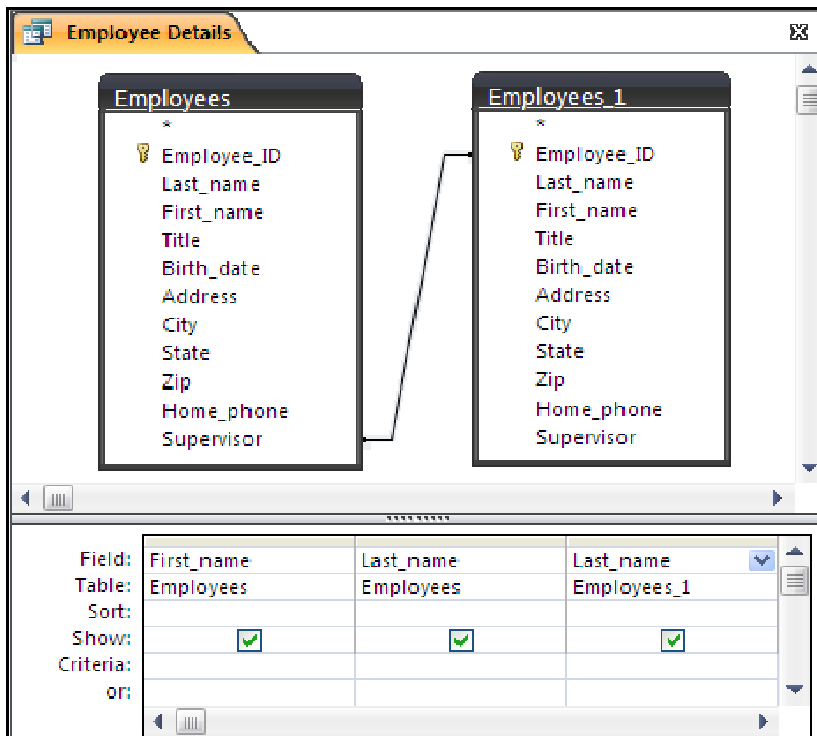
Create an Inner Join by using an intermediate table/s. The results of the matching records will be obtained first for this inner join and the results of the outer joins will be calculated thereafter.



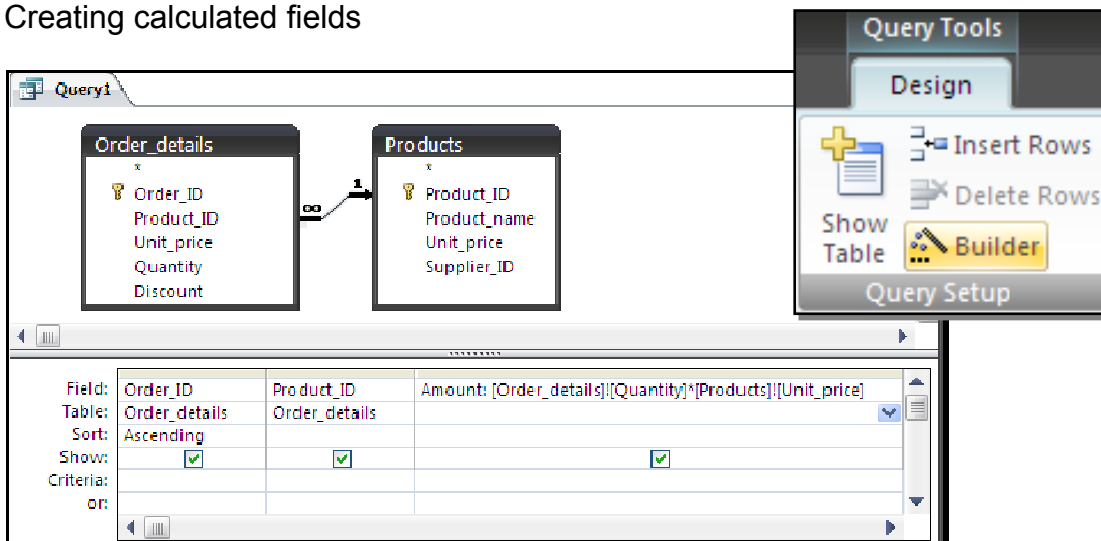
Visual summary: Unit 4 objectives

Self-join queries

You can create a self join by using two copies of the same table. A self-join is a query that displays matching records from the same table when there are matching values in two fields.



Creating calculated fields

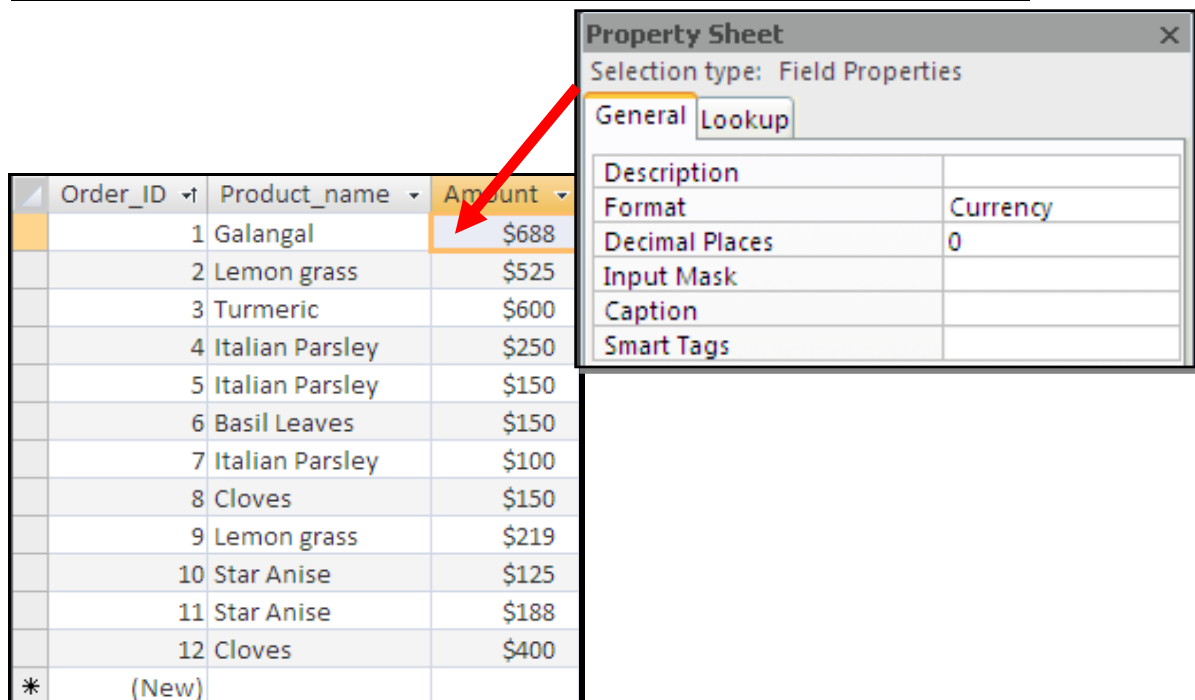
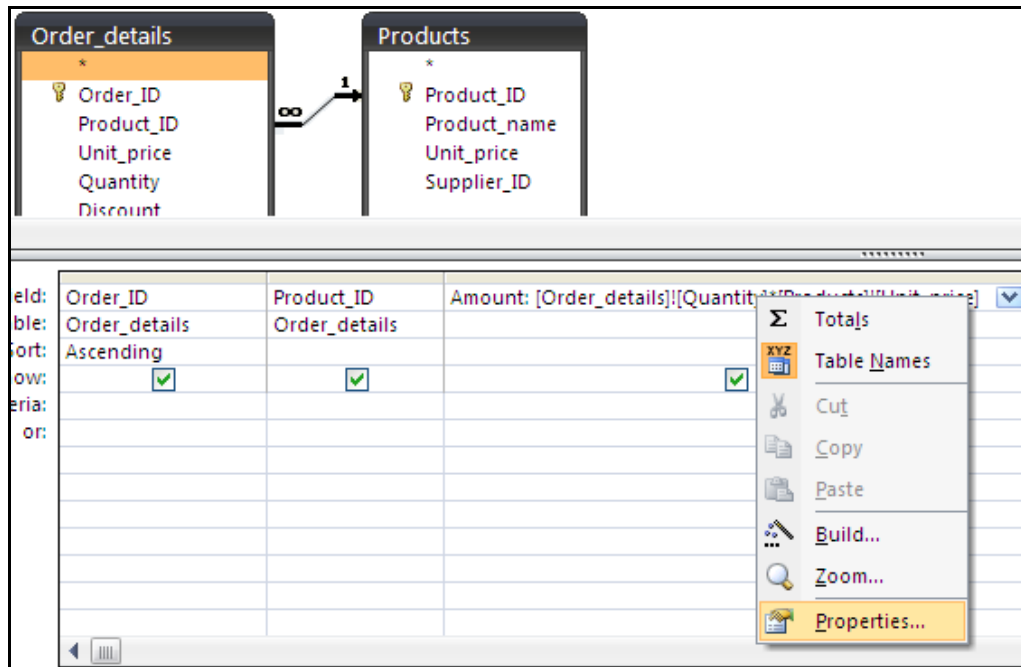


Amount: [Order_details]![Quantity]*[Products]![Unit_price]

Visual summary: Unit 4 objectives

Modifying Formats

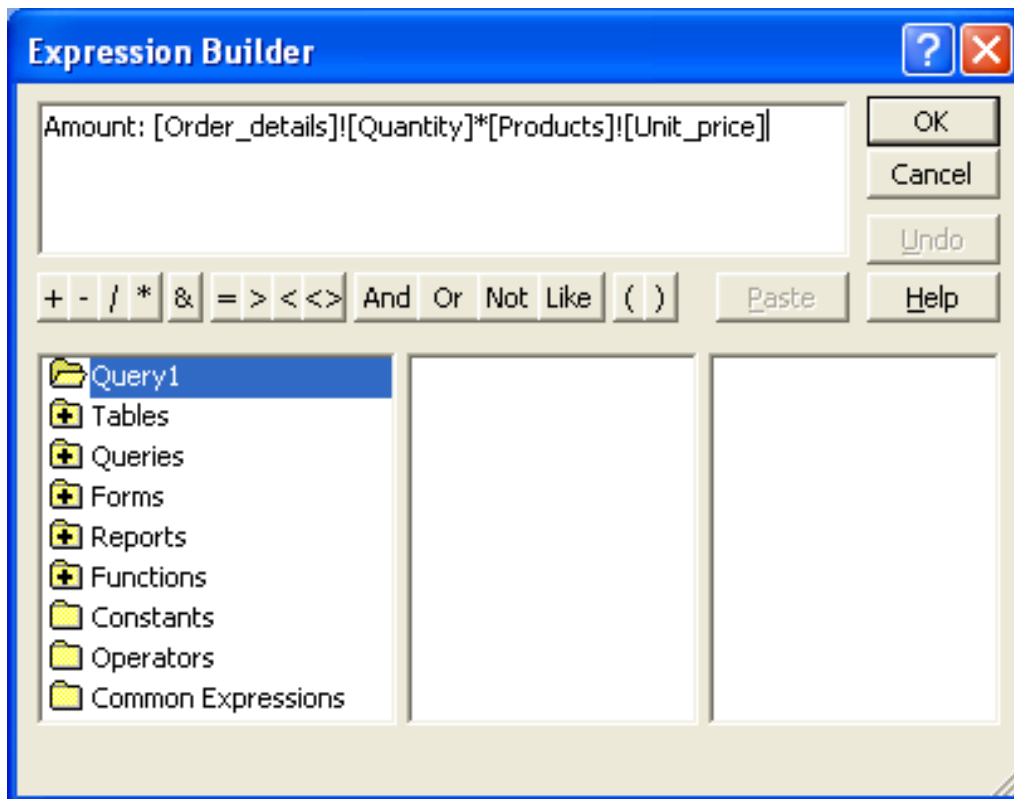
In query design view, right-click on the field that you want to format and choose properties.



Visual summary: Unit 4 objectives

The Expression Builder

Calculated fields can be built in the expression builder.

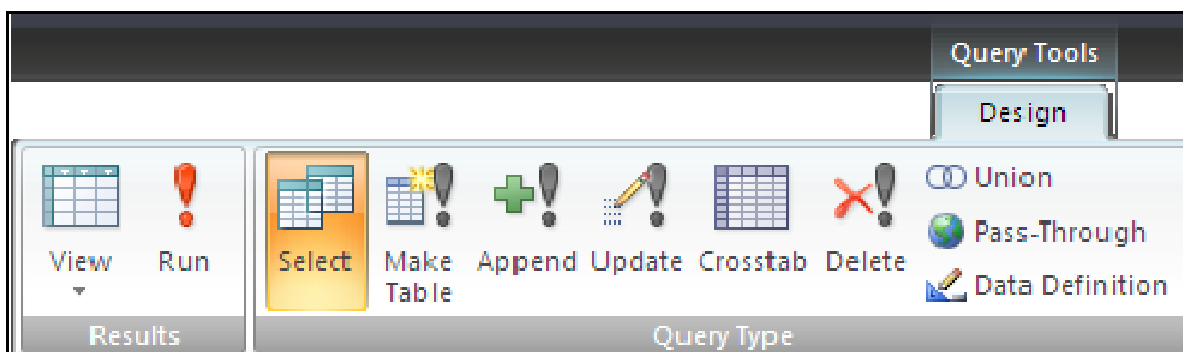


Visual summary: Unit 4 objectives

Using action queries

Action queries are queries that perform certain actions in tables. There are four types:

Append query	This type of query is used to append data from one table to another.
Delete query	This type of query is used to delete records from tables based on specified criterion.
Update query	This type of query is used to update data in different tables at the same time.
Make-table query	This type of query is used to create a table from the result of a query.



Unit 4 Practice Activity

1. Open **Advanced_query_features.accdb**.
2. Using the Product and Transaction tables, create an outer join where all records from the Product table and the Transaction_ID appear in the query results.
3. Create a query that displays the Product_name and the difference between Qty_available and Min_stock for each product from the Product table.
4. Create a query to append all data from the New_products table to the Product table.
5. Use an update query to increase the discount percent of all transactions in the Transactions table that have Qty_sold greater than 250 by 5%.
6. Create a new table by using a make-table query to include the Transaction_ID from the Transaction Table, Product_name from the Product table, and the value of each transaction. Be sure all of the records from the Transaction table and only the matching records from the Product table appear in the new table.
7. Close the database.



Online support forum and knowledge base

www.microsofttraining.net/forum

Visit our forum to have your questions answered by our Microsoft certified trainers.

Creating advanced queries

Unit 5 objectives

- Use queries to join fields, and find unmatched records different tables
- Use parameter queries to view results based on criteria entered while running the query
- Create single and multiple-field indexes to quickly sort and filter data in a table

- Use queries to join fields, and find unmatched records different tables
- Use parameter queries to view results based on criteria entered while running the query
- Create single and multiple-field indexes to quickly sort and filter data in a table

- Use queries to join fields, and find unmatched records different tables
- Use parameter queries to view results based on criteria entered while running the query
- Create single and multiple-field indexes to quickly sort and filter data in a table

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

Visual summary: Unit 5 objectives

Summarising and grouping values

Summary functions

A summary function is used to calculate the aggregate value (e.g. sum, average) of data which has been grouped, e.g. the total sales of a particular product.

Product_name	Total Discount
Galangal	£618.75
Lemon grass	£525.00
Turmeric	£570.00
Italian Parsley	£225.00
Italian Parsley	£150.00
Basil Leaves	£127.50
Italian	
Cloves	

Order_details

- * Order_ID
- Product_ID
- Unit_price
- Quantity
- Discount

Field:	Product_ID	Total Discount: [unit_price]*[quantity]*(1-[discount])
Table:	Order_details	
Total:	Group By	sum
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		

Visual summary: Unit 5 objectives

Concatenation

You can combine values from different fields into one field.

Employee_ID	Title	Name
E001	Customer Service Representative	Kim Leong
E002	Market Analyst	Solena Hernanadez
E003	IT Consultant	Thomas Boorman
E004	Senior Buyer	Ron Timmons
E005	Project Management Consultant	Kathy Sinclair
E006	VP Financial Services	Ann Salinski
E007	VP Sales	Jack Thomas
E008	Manager, IT	Elise Sethan
E009	Business Consulatnt (External)	Susan Gianni
*		

Query1

Employees

- *
 - Employee_ID
 - Last_name
 - First_name
 - Title
 - Birth_date
 - Address
 - City
 - State
 - Zip
 - Home_phone
 - Supervisor

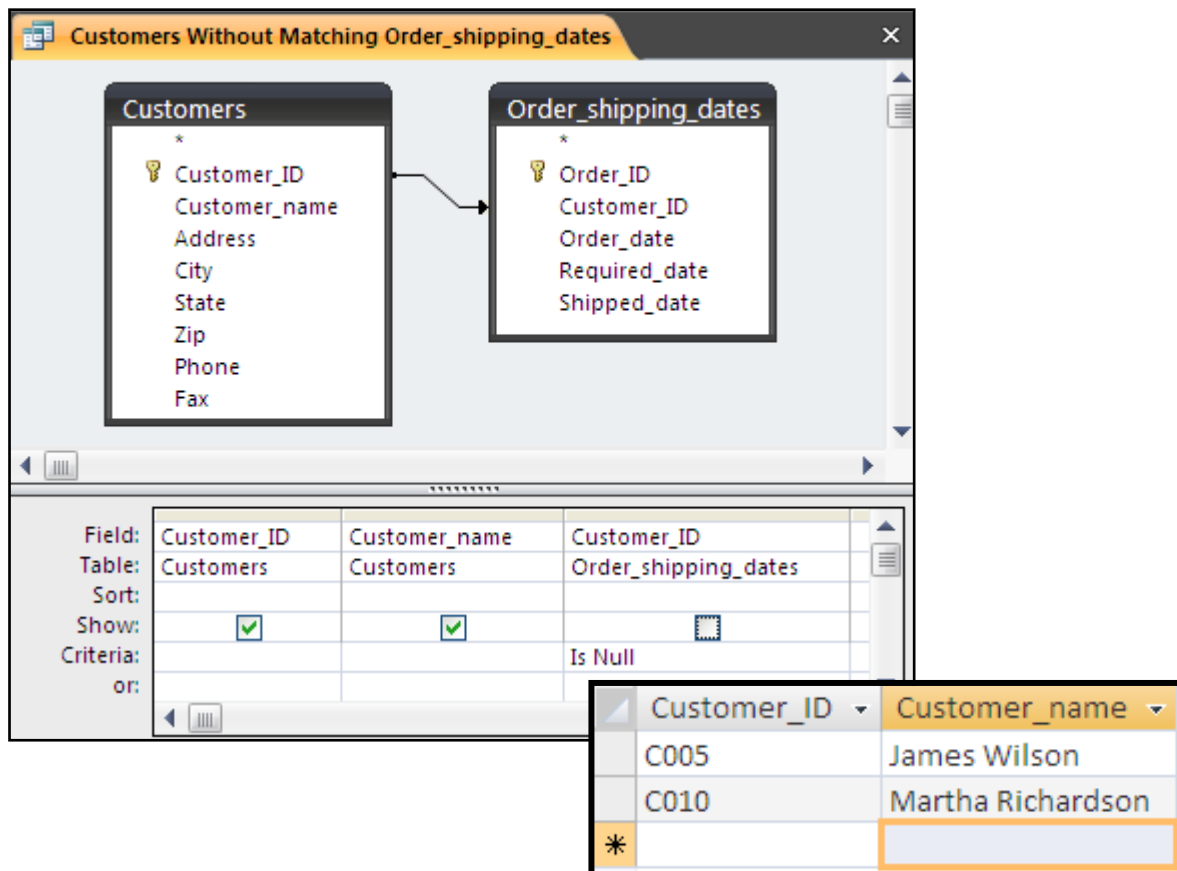
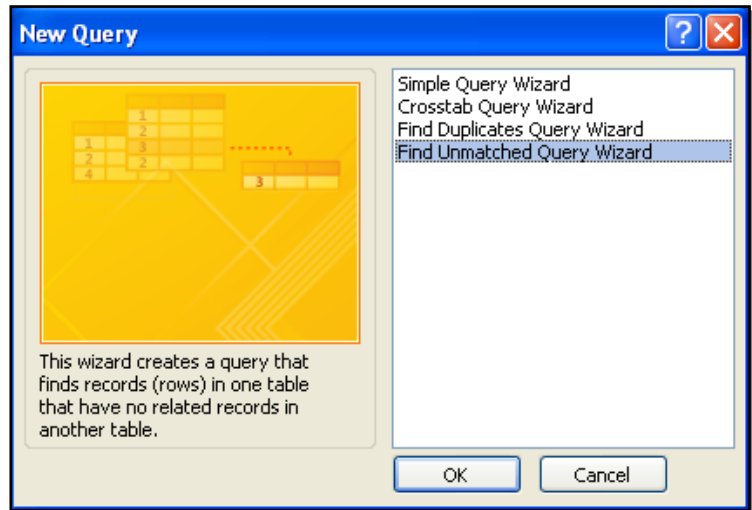
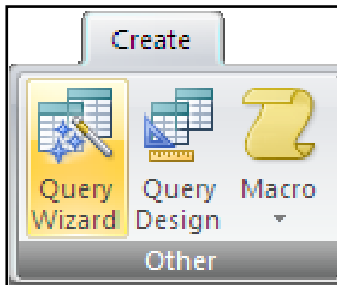
Field:	Employee_ID	Title	Name: [First_name] + " " + [last_name]
Table:	Employees	Employees	
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Name: [First_name] + " " + [Last_name]

Visual summary: Unit 5 objectives

Find unmatched queries

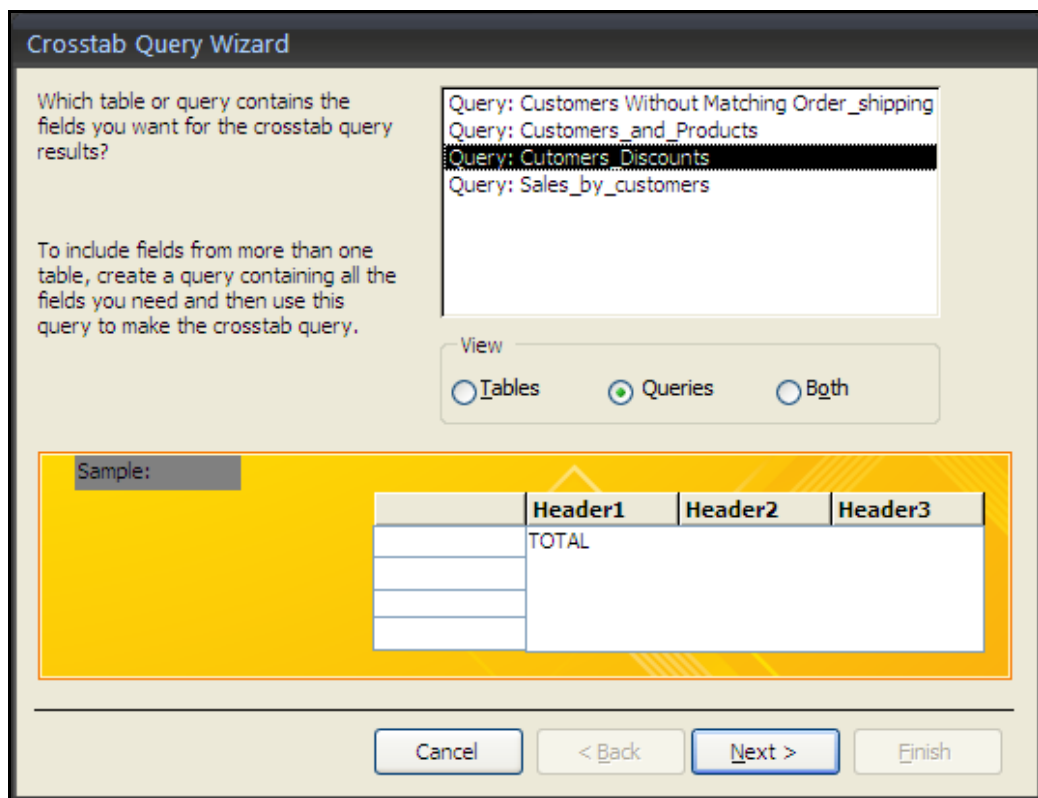
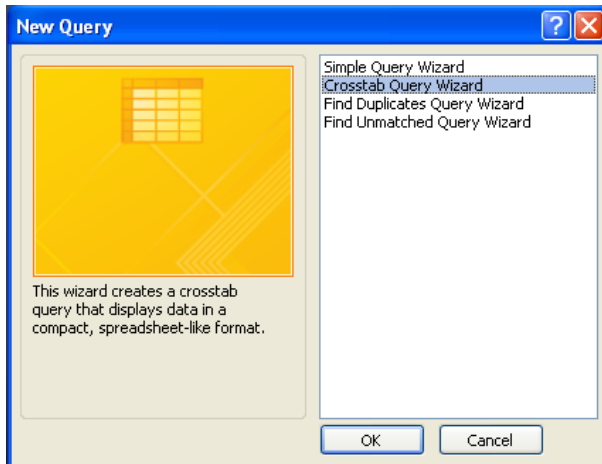
You can use *find unmatched query*, to view records that don't have a matching record in another table. For example which customers from the customers table haven't placed any orders in the orders table.



Visual summary: Unit 5 objectives

Creating cross-tab queries

You can display a summary of values based on two different types of information by creating a cross-tab query.



Visual summary: Unit 5 objectives

Crosstab Query Wizard

Which fields' values do you want as row headings?

You can select up to three fields.

Select fields in the order you want information sorted. For example, you could sort and group values by Country and then Region.

Available Fields:

- Order_ID
- Customer_name
- Amount
- Total Discount

Selected Fields:

- Product_Name

Sample:

Product_Name	Customer_name	Customer_name	Customer_name
Product_Name1	Avg(Order_ID)		
Product_Name2			
Product_Name3			
Product_Name4			

Cancel < Back Next > Finish

Crosstab Query Wizard

Which field's values do you want as column headings?

For example, you would select Employee Name to see each employee's name as a column heading.

Available Fields:

- Order_ID
- Customer_name
- Amount
- Total Discount

Sample:

Product_Name	Customer_name	Customer_name	Customer_name
Product_Name1	Avg(Order_ID)		
Product_Name2			
Product_Name3			
Product_Name4			

Cancel < Back Next > Finish

Visual summary: Unit 5 objectives

Crosstab Query Wizard

What number do you want calculated for each column and row intersection?

For example, you could calculate the sum of the field Order Amount for each employee (column) by country and region (row).

Do you want to summarize each row?

☒ Yes, include row sums.

Fields:

Order_ID
Amount
Total Discount

Functions:

Avg
Count
First
Last
Max
Min
StDev
Sum
Var

Sample:

Product_Name	Customer_nz	Customer_nz	Customer_nz
Product_Name1	Avg(Order_ID)		
Product_Name2			
Product_Name3			
Product_Name4			

Cancel < Back Next > Finish

Crosstab Query Wizard

What do you want to name your query?

Customers_Discounts_Crosstab

That's all the information the wizard needs to create the query.

Do you want to view the query, or modify the query design?

☒ View the query.

☐ Modify the design.

Customers_discounts_Crosstab : Crosstab Query

	Product_name	Total Of Discou	Greg Huns	Harry Wilkins	Henry Drucker
►	Basil Leaves	22.500000894			22.500000894
	Cloves	28.750000149		20.000000298	
	Galangal	68.750001024	68.750001024		
	Italian Parsley	15.000000224			
	Lemon grass	0			
	Star Anise	20.312500652		12.500000186	
	Turmeric	30.000000447			

Record: 1 of 7

Visual summary: Unit 5 objectives

Using parameter queries

A parameter query displays results based on criteria specified when you run the query.

Single-criterion parameter queries

The screenshot illustrates the process of running a single-criterion parameter query in Microsoft Access. The top window, titled 'Query1', shows the 'Products' table with fields 'Product_name', 'Unit_price', and 'Supplier_ID'. The 'Criteria' row for 'Unit_price' is set to '> [Price is greater than]'. Below this, an 'Enter Parameter Value' dialog box is open, displaying the prompt 'Price is greater than' and the input value '1.5'. The bottom window shows the results of the query, listing products with unit prices greater than \$1.50.

Product_name	Unit_price	Supplier_ID
Oregano	\$3.00	S001
Galangal	\$2.75	S004
Lemon grass	\$1.75	S002
Turmeric	\$4.00	S003
Italian Parsley	\$2.00	S001
Cardamom Powder	\$3.50	S005
Cloves	\$2.00	S003

Visual summary: Unit 5 objectives

Multiple criteria parameter queries

Using “Between” and “And”.

Between [Lower Value] And [Higher Value] (eg Between 21/10/2006 And 31/10/2006)

Wildcards in parameter queries

Like [First letter of name]+”*” (eg Like “S*”)

Using indexes

Employees

Field Name	Data Type	Description
Employee_ID	Text	
Last_name	Text	
First_name	Text	
Title	Text	Employee's title
Birth_date	Date/Time	

Field Properties

General | Lookup

Field Size	50
Format	
Input Mask	
Caption	Last_name
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	No
Indexed	No
Unicode Compression	No
IME Mode	Yes (Duplicates OK)
IME Sentence Mode	Yes (No Duplicates)
Smart Tags	

An index speeds up searches and sorting on the field, but may slow updates. Selecting "Yes - No Duplicates" prohibits duplicate values in the field. Press F1 for help on indexed fields.

Unit 5 Practice Activity

1. Open **Advanced_queries.accdb**.
2. Create a query that displays the **Retailer_code** and the **Total Sales_value** for each Retailer from the **Transaction** and the **Product** tables.
3. Create a sum of the **Sale_values** and group the query based on the **Retailer_codes**.
4. Compare the result of the query that you created in Step 2, with the following.

	Retailer_code	Total_sales
►	R001	£187.50
	R002	£965.00
	R003	£280.00
	R004	£985.00
	R005	£250.00
	R006	£200.00
	R007	£481.50
	R008	£300.00

5. Create a query called **Retailer_details** that displays the **Retailer_code** and both **Retailer_name** columns. The **Contact_name** column should contain the values from the **Contact_first_name** and **Contact_last_name** fields of the **retailer** table (*hint*: concatenate the two fields).
6. Create a cross-tab query from the **Transaction** table that displays the total quantity of each product sold to each Retailer.
7. Create a parameter query from the **Transaction** table that displays the **Product_ID** and **Qty_sold** fields for all products that have a **Qty_sold** value that's less than a specified value.
8. Create a parameter query from the **Product** table that displays the **Product_name**, **Qty_available**, and **Min_stock** fields for all products that have the **Min_stock** value between two specified values.
9. Close the database.

Creating advanced form design

Unit 6 objectives

- Use controls to add graphics to a form
- Use controls to add calculated fields to a form
- Add a combo box to a form
- Add unbound controls to a form

- Use controls to add graphics to a form
- Use controls to add calculated fields to a form
- Add a combo box to a form
- Add unbound controls to a form

-
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

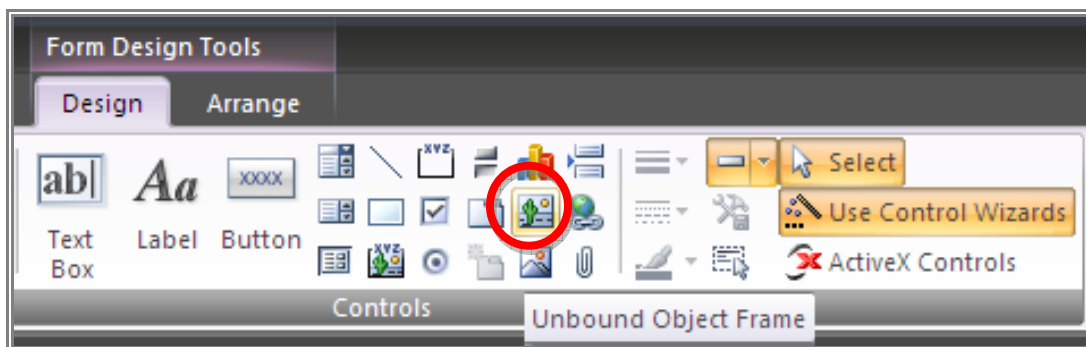
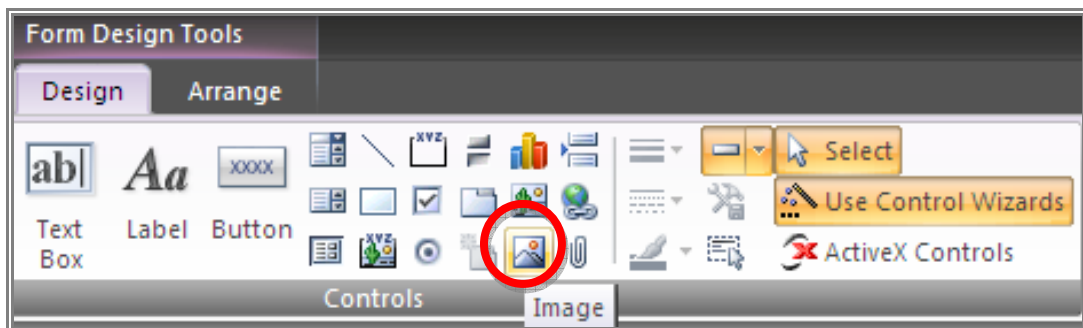
Visual summary: Unit 6 objectives

Adding graphics

Control Types

- Unbound Controls** These controls are not linked to any field in a table or a query. They are used to enhance the appearance of a form or to display information that isn't linked to any field in a table or a query. Graphics are unbound controls because they aren't linked to any table or query.
- Bound Controls** These controls are linked to a field in a table or a query. They are used to display a field value, to accept a value in a field in a table or a query, or to modify the value of a field in a table or a query.
- Calculated Controls** These are used to display a calculated value based on one or more fields in a table or a query.

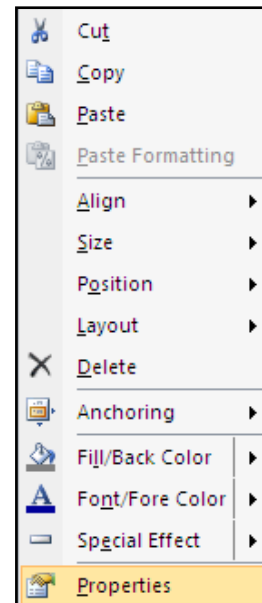
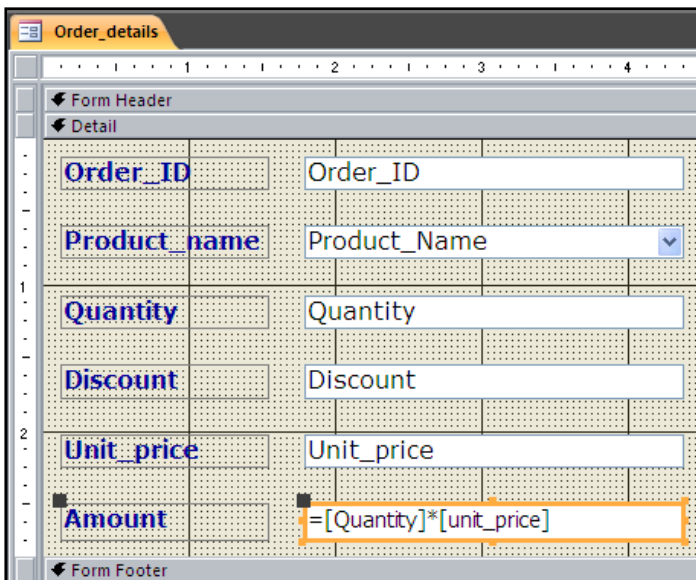
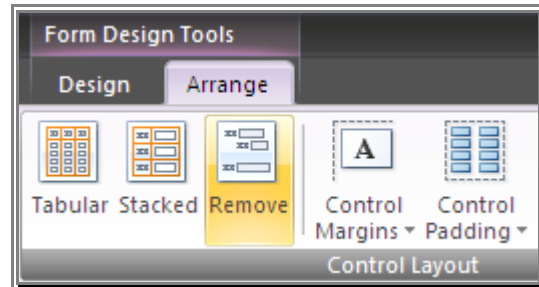
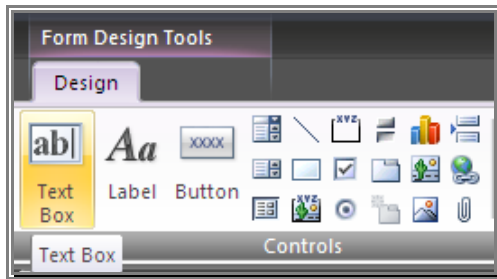
Image controls & Unbound object frame controls



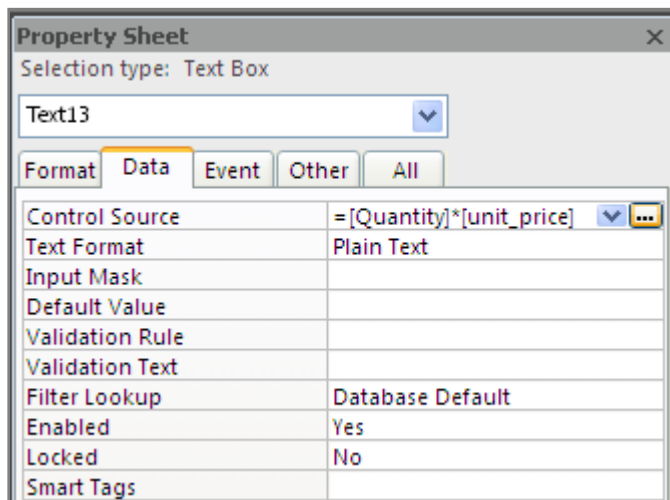
Visual summary: Unit 6 objectives

Adding calculations

Creating a calculated control in a form, use the Arrange tab to align boxes.



Right-click on field box and choose **Properties**, type in the formula under **Data** tab → **Control Source**



Visual summary: Unit 6 objectives

Aligning controls in a form

Select all the controls (CTRL + A), choose **Stacked**.

Order_details

Form Header

Detail

Order_ID

Product_name

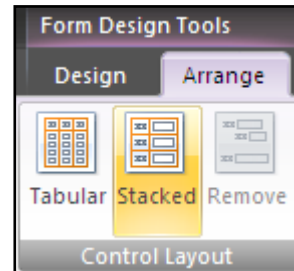
Quantity

Discount

Unit_price

Amount

Form Footer



OR

Select controls individually (SHIFT + Click), choose appropriate alignment.

Order_details

Form Header

Detail

Order_ID

Product_name

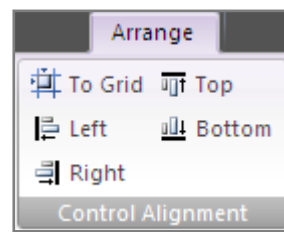
Quantity

Discount

Unit_price

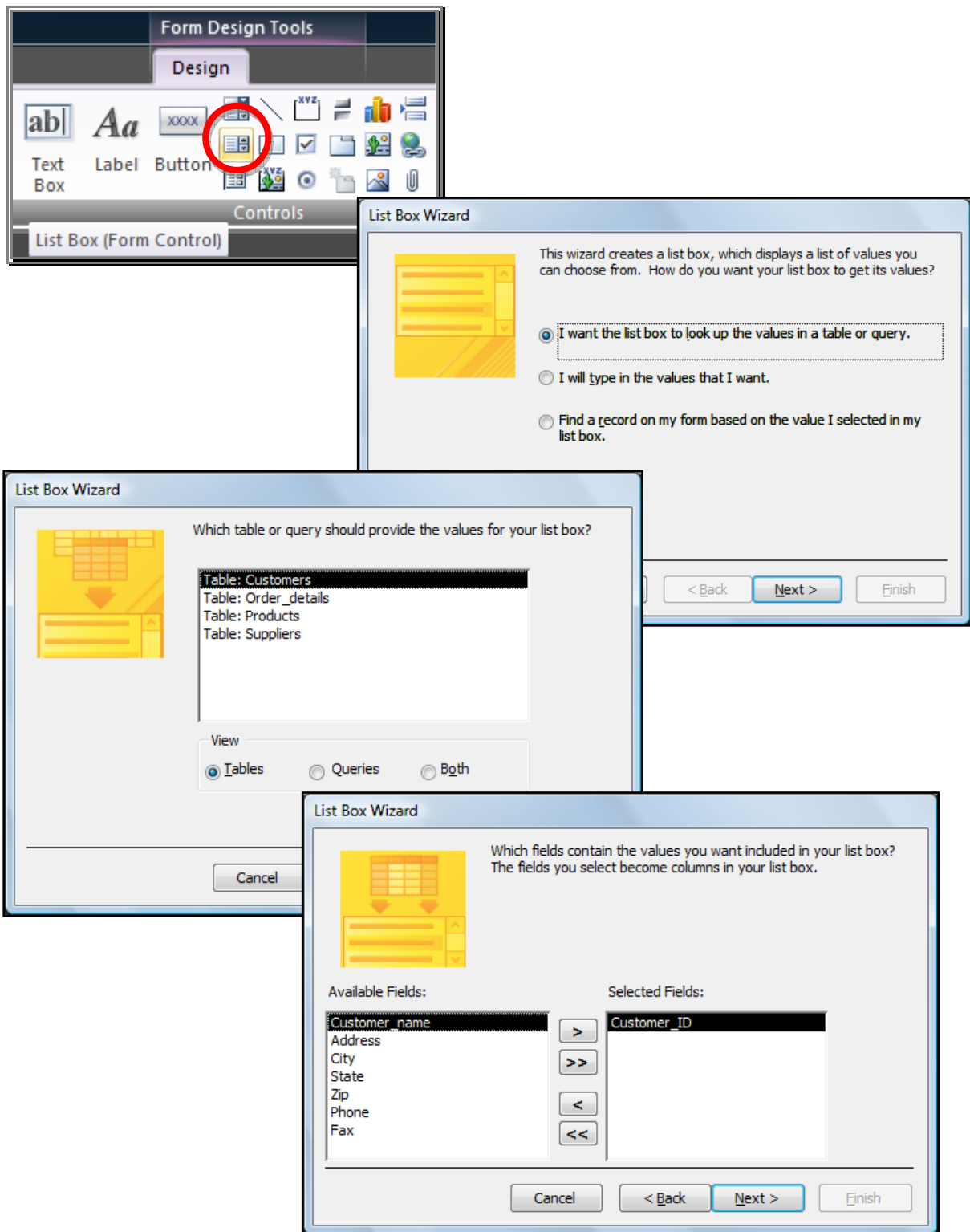
Amount

Form Footer



Visual summary: Unit 6 objectives

Adding combo boxes



Visual summary: Unit 6 objectives

List Box Wizard

How wide would you like the columns in your list box?

To adjust the width of a column, drag its right edge to the width you want, or double-click the right edge of the column heading to get the best fit.

Customer_ID				
C001				
C002				
C003				
C004				
C005				
C006				
C007				

Cancel < Back Next > Finish

List Box Wizard

Microsoft Office Access can store the selected value from your list box in your database, or remember the value so you can use it later to perform a task. When you select a value in your list box, what do you want Microsoft Office Access to do?

☐ Remember the value for later use.

☒ Store that value in this field: Customer_ID

Cancel < Back Next > Finish

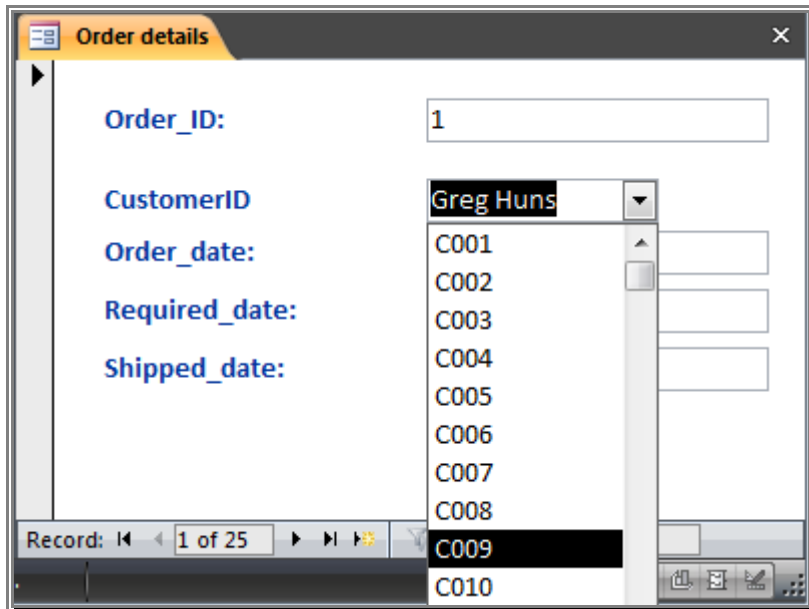
List Box Wizard

What label would you like for your list box?

Customer_ID

Those are all the answers the wizard needs to create your list box.

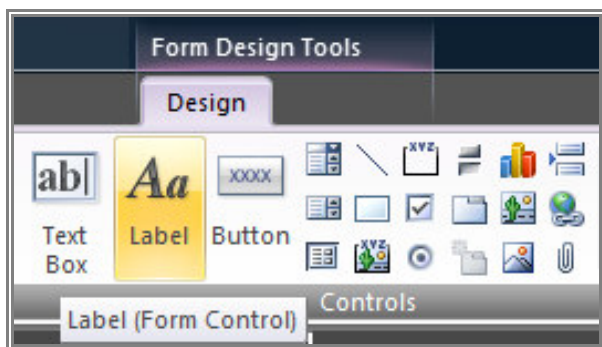
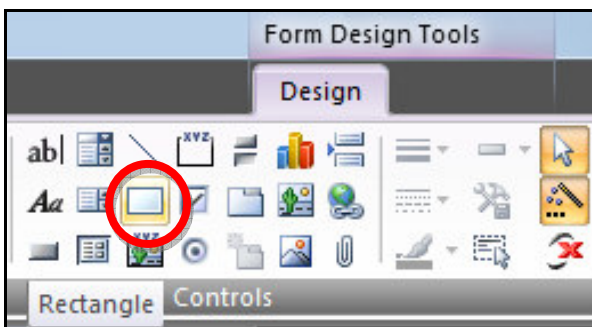
Visual summary: Unit 6 objectives



Modifying the properties of a combo box

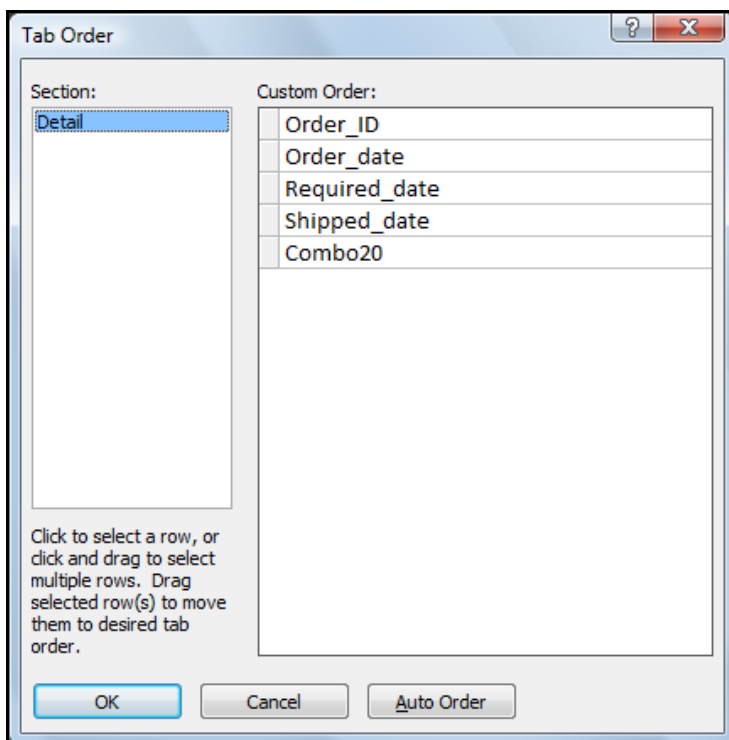
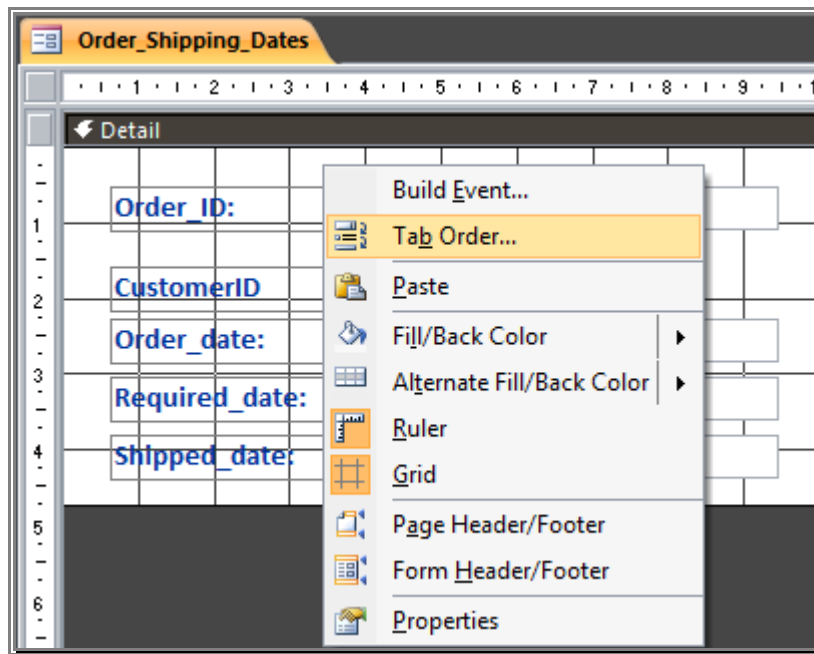
In design view right-click on control and select properties.

Adding unbound controls



Visual summary: Unit 6 objectives

Tab Order



Unit 6 Practice Activity

1. Open **Forms.accdb**.
2. Modify the **Sales_by_retailer** form by adding the following to the Header section, as shown in the example below:
 - A label control with 'Outlander Spices' as the caption.
 - An image control using the **Spice_picture** graphic.
 - A rectangle around the label and image controls.
3. Change the tab order of the controls in the Detail section to Sales, Retailer_code and Retailer_name.
4. Update the form and switch to the form view
5. Close the form.
6. Close the database.



Online support forum and knowledge base

www.microsofttraining.net/forum

Visit our forum to have your questions answered by our Microsoft certified trainers.

Using advanced report features

Unit 7 objectives

- Customise headers and footers, and set properties to group data and modify a report's appearance
- Use functions to add calculated values in a report
- Embed a subreport in a main report

- Customise headers and footers, and set properties to group data and modify a report's appearance
- Use functions to add calculated values in a report
- Embed a subreport in a main report

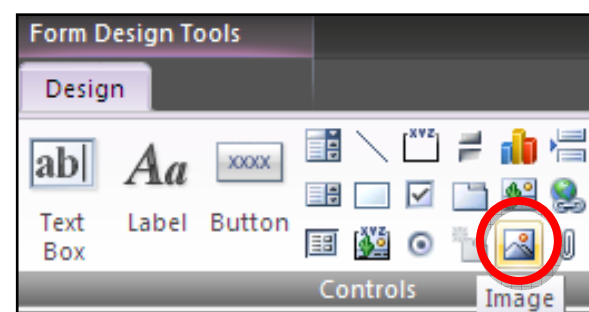
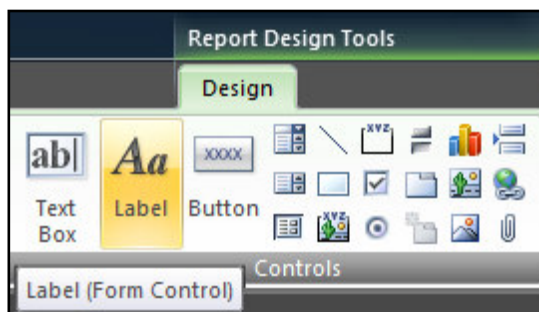
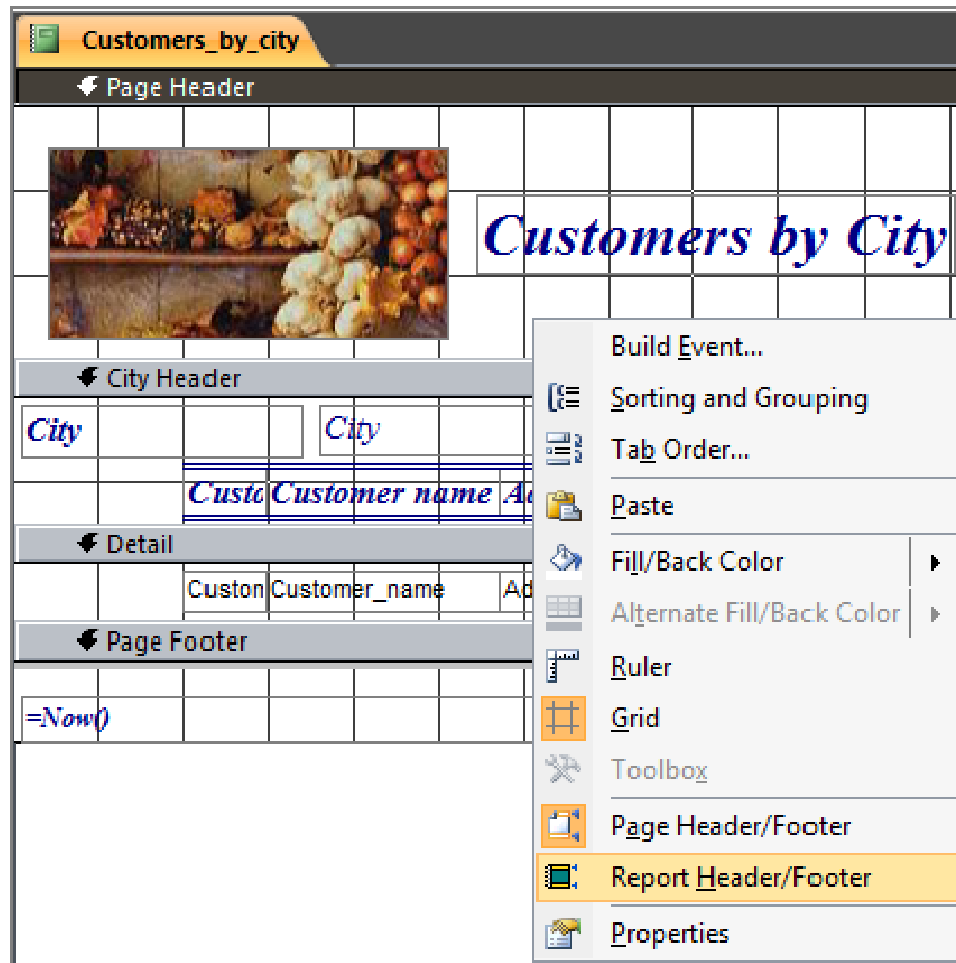
-
- This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

Visual summary: Unit 7 objectives

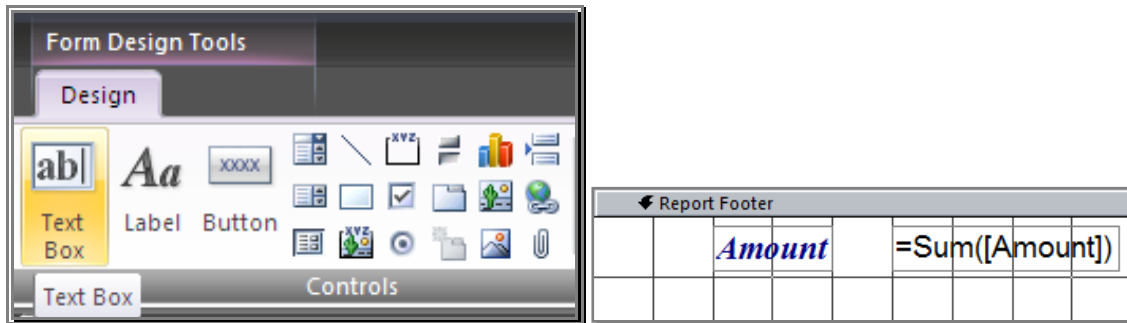
Creating Customised Headers and Footers

Report Header



Visual summary: Unit 7 objectives

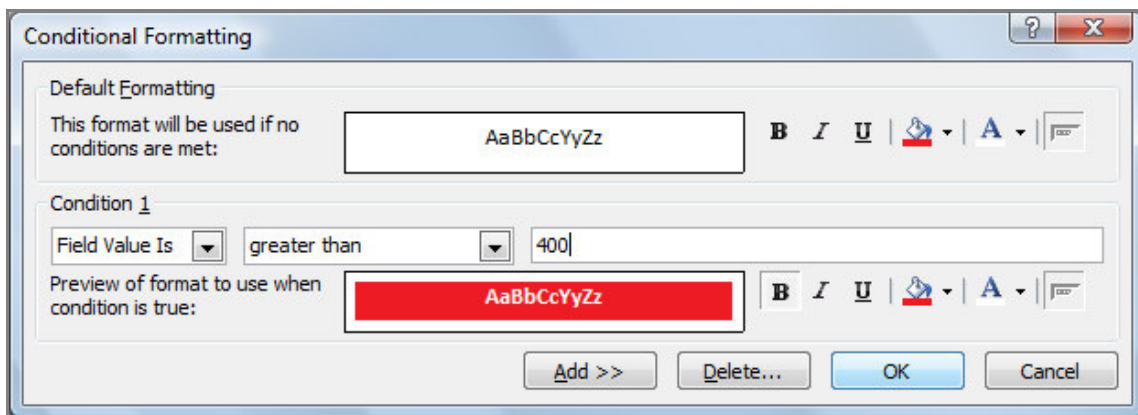
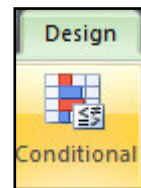
Report Footer



<i>Chinese Star Anise (Whole)</i>	19	\$125.00
<i>Chives</i>	20	\$300.00
<i>Cilantro Flakes</i>	12	\$400.00
<i>Cilantro Flakes</i>	16	\$800.00
<i>Cilantro Flakes</i>	22	\$400.00
<i>Amount</i>		\$7,115.00

Conditional Formatting

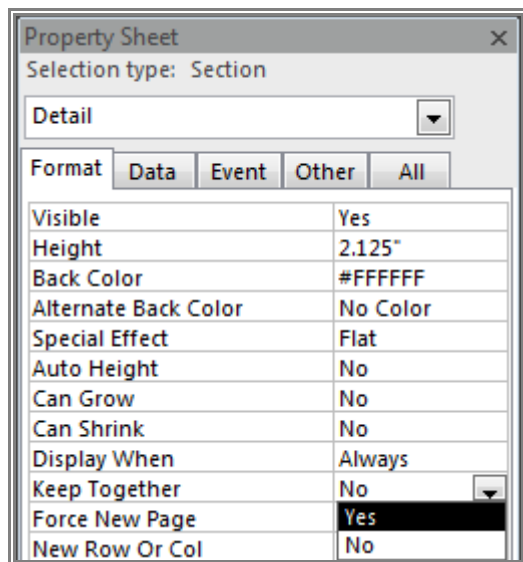
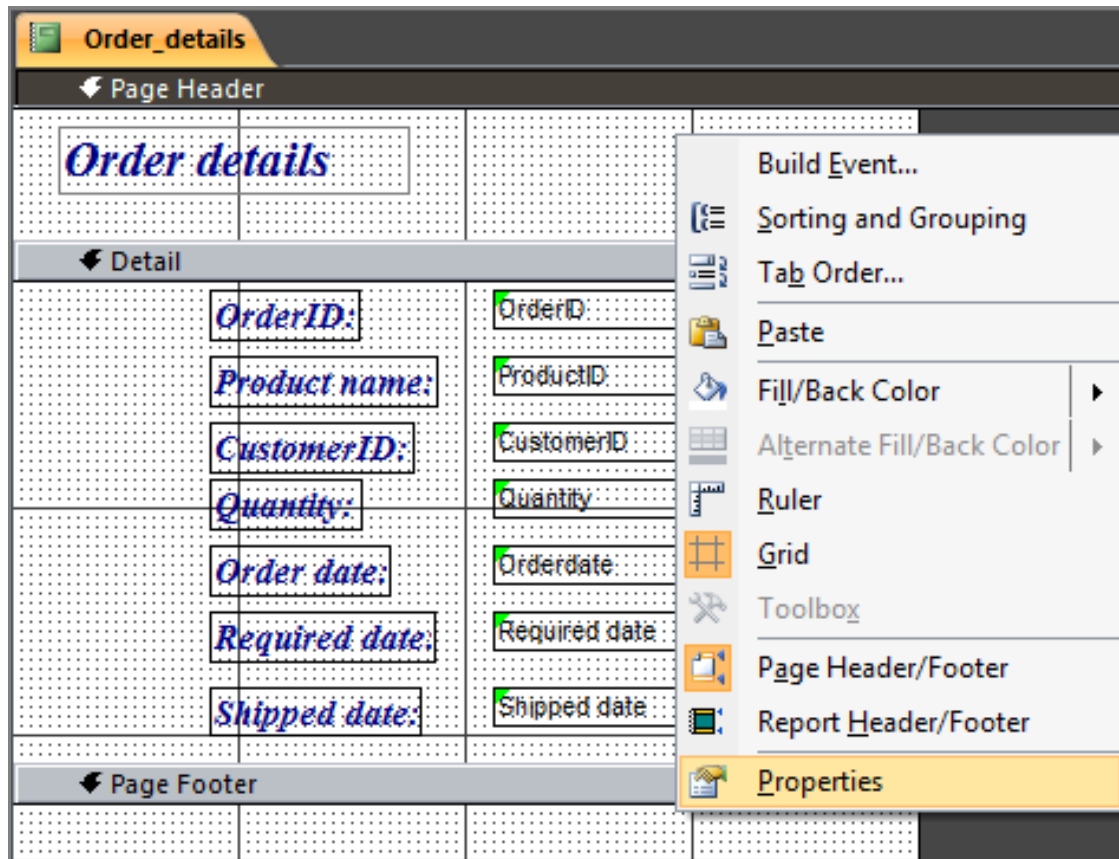
Attention can be drawn to specific data in a report by using conditional formatting. This feature only applies formatting to the value of a field if a specified criterion is met.



Visual summary: Unit 7 objectives

The keep together property

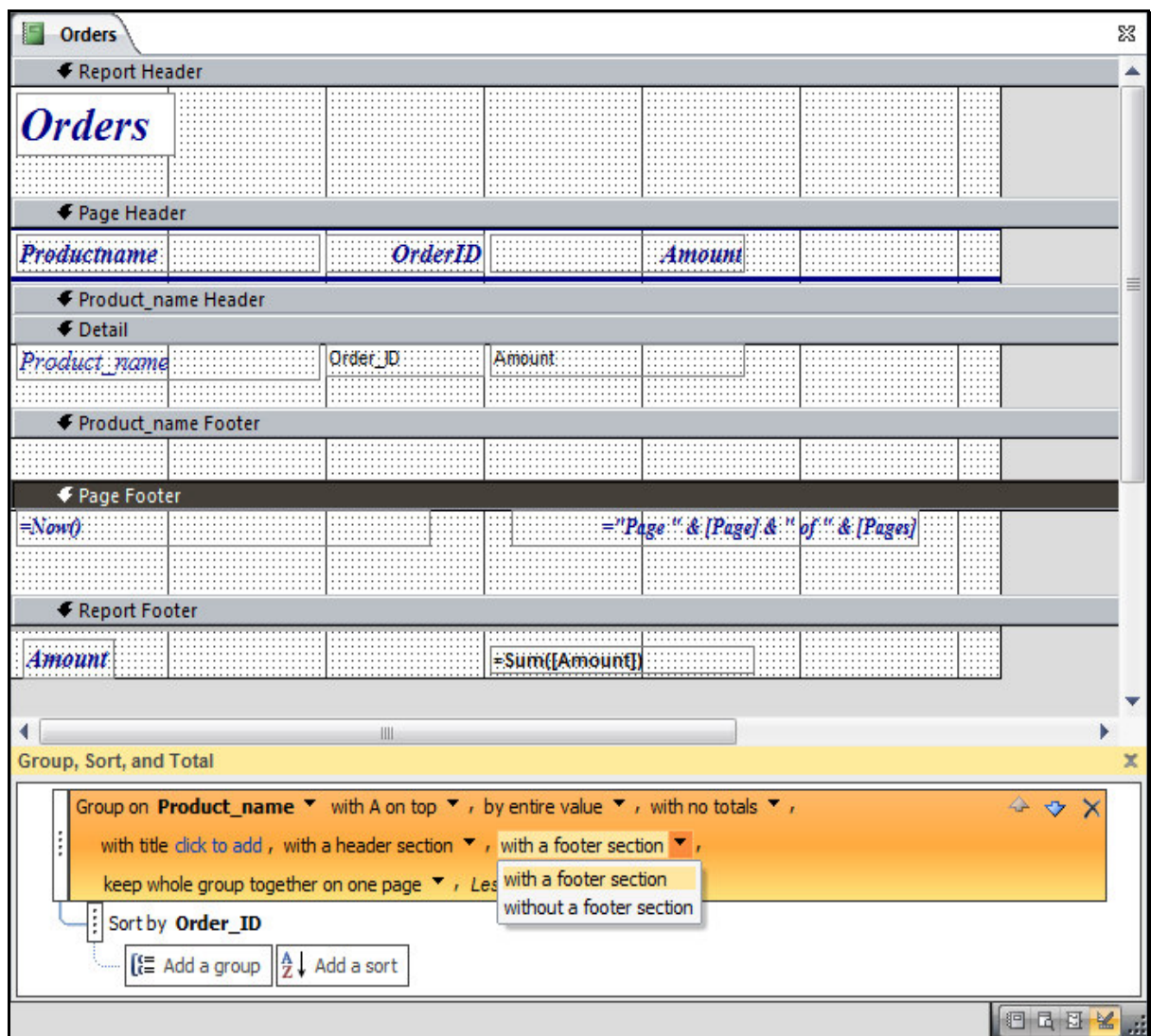
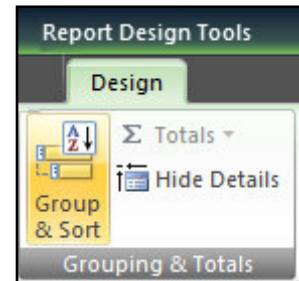
This can be used to ensure that a complete section of a report is always printed on the same page.



Visual summary: Unit 7 objectives

Group Footers

When reports are grouped based on a given field you can use the group footer section to add information particular to that group, such as the group total for example.



Visual summary: Unit 7 objectives

Forcing a new page

The screenshot displays the Microsoft Access design view for the 'Order_details' table. The table is structured with a 'Page Header' section containing the text 'Order details' and a 'Detail' section containing the following fields: OrderID, Product name, CustomerID, Quantity, Order date, Required date, and Shipped date. A context menu is open over the 'Detail' section, and the 'Properties' window is open at the bottom, showing the 'Force New Page' property set to 'None'.

Table Structure:

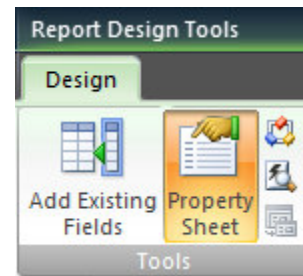
Section	Field
Page Header	Order details
Detail	OrderID
	Product name
	CustomerID
	Quantity
	Order date
	Required date
	Shipped date

Properties Window:

Property	Value
Visible	Yes
Height	2.125"
Back Color	#FFFFFF
Alternate Back Color	No Color
Special Effect	Flat
Auto Height	No
Can Grow	No
Can Shrink	No
Display When	Always
Keep Together	No
Force New Page	None
New Row Or Col	None

Visual summary: Unit 7 objectives

The hide duplicate property

The screenshot shows an Access report titled 'Orders'. The report has a 'Report Header' section with the title 'Orders'. Below this is a 'Page Header' section with three columns: 'Productname', 'OrderID', and 'Amount'. The main body of the report is in 'Detail' view, showing a table with columns 'Productname', 'OrderID', and 'Amount'. The 'Productname' field in the first row is selected. A 'Property Sheet' window is open on the right, showing the properties for the selected 'Text Box'. The 'Hide Duplicates' property is set to 'Yes', and the 'Can Grow' property is set to 'Yes'. The 'Format' tab is selected in the Property Sheet, showing various formatting options like Gridline Width, Margins, and Padding.

Report Header		
Orders		
Page Header		
Productname	OrderID	Amount
Product_name Header		
Detail		
Productname	OrderID	Amount
Product_name Footer		
Page Footer		
Report Footer		

Property Sheet
Selection type: Text Box
Productname

Format | Data | Event | Other | All

Gridline Width Bottom	1 pt
Gridline Width Left	1 pt
Gridline Width Right	1 pt
Top Margin	0"
Bottom Margin	0"
Left Margin	0"
Right Margin	0"
Top Padding	0.0208"
Bottom Padding	0.0208"
Left Padding	0.0208"
Right Padding	0.0208"
Hide Duplicates	Yes
Can Grow	Yes
Can Shrink	No
Display When	Always
Reading Order	Context
Scroll Bar Align	System
Numeral Shapes	System

Visual summary: Unit 7 objectives

Adding calculated fields

DateDiff function

DateDiff("interval", [date1],[date2])

For interval – **y** for difference between years.

m for difference between months.

d for difference between days.

IIF function

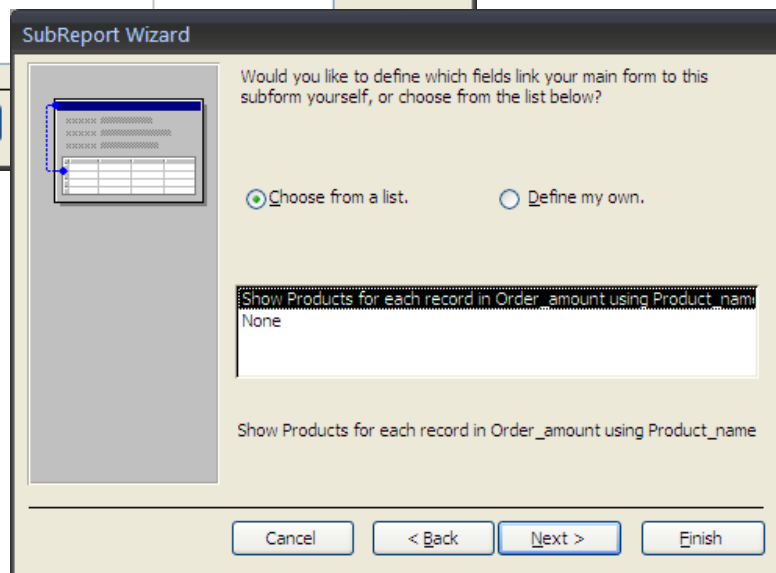
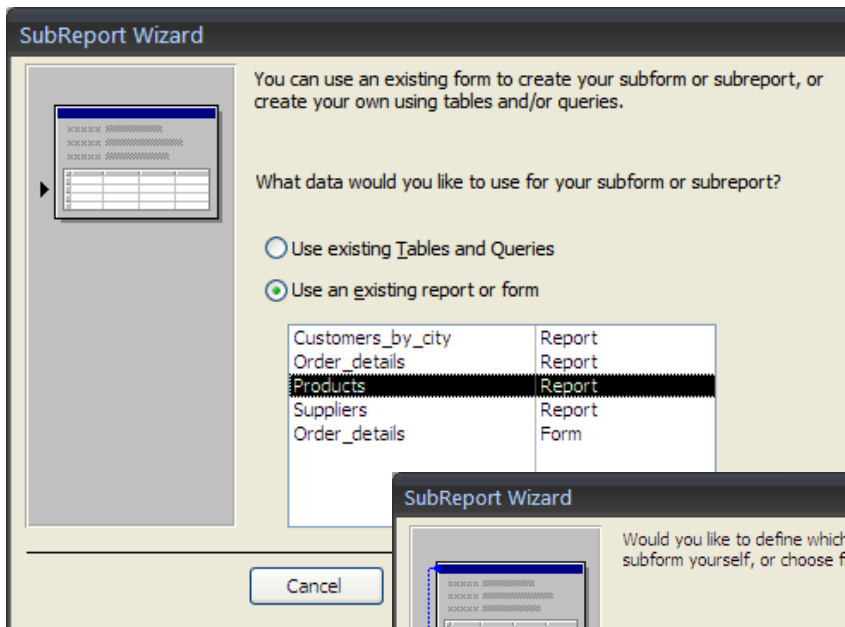
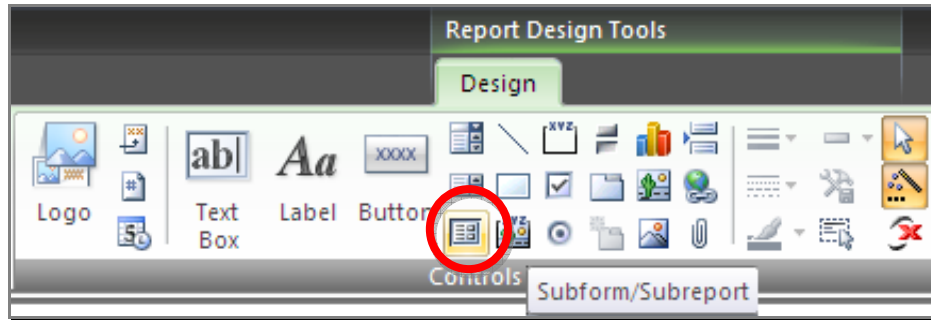
IIf(condition, value if true, value if false)

Visual summary: Unit 7 objectives

Working with SubReports

You can use a SubReport to display data from two reports.

Embedding a SubReport



Unit 7 Practice Activity

1. Open **Advanced_reports.accdb**.
2. Display the total quantity sold for each product in the Transaction_ details report by adding a group footer based on Product_ID.
3. Display the data in a different format whenever the Qty_sold is greater than 200 by adding conditional formatting to the Qty_sold field.
4. Update the report.
5. Switch to Print Preview.
6. Close the Report.

ProductID	TransactionID	Kiosk code	Qty sold	Requireddate	Orderdate	Discount
P001	T001	R001	150	1/20/2001	12/15/2000	0
		Total Sold	150			
P002	T006	R005	100	1/3/2001	12/3/2000	0.05
		Total Sold	100			
P003	T002	R002	400	12/27/2000	11/24/2000	0.15
		Total Sold	400			

7. In the Transaction_dates report, add a text box control to display the difference between Order_date and Required_date.
8. Update the report.
9. Switch to the Print Preview to view the report.
10. Close the report.
11. Close the database.

Transaction_ID	Product_name	Order_date	Required_date	No. of Days
Transaction_ID	Product_ID	Order_date	Required_date	= [Required_date] - [Order_date]



Online support forum and knowledge base

www.microsofttraining.net/forum

Visit our forum to have your questions answered by our Microsoft certified trainers.