

# Simple Invoicing Desktop Database with MS Access 2013/2016

David W. Gerbing  
School of Business Administration  
Portland State University

July 7, 2018

# Contents

<b>1</b>	<b>Create a New Database</b>	<b>1</b>
<b>2</b>	<b>Customer and Product Tables</b>	<b>2</b>
2.1	Customer Table . . . . .	2
2.1.1	Name the Customer Table . . . . .	2
2.1.2	Save the Changes . . . . .	2
2.1.3	Primary Key . . . . .	2
2.1.4	Data Type . . . . .	2
2.1.5	Index Fields . . . . .	3
2.1.6	Add Remaining Customer Fields . . . . .	3
2.1.7	Enter Customer Data . . . . .	4
2.1.8	Print Customer Data . . . . .	5
2.2	Product Table . . . . .	5
2.2.1	Create Table . . . . .	5
2.2.2	Define Fields and Primary Key . . . . .	5
2.2.3	Add Description and Price Fields . . . . .	5
2.2.4	Add Products . . . . .	5
2.2.5	Print Product Data . . . . .	6
<b>3</b>	<b>Database Structure</b>	<b>6</b>
3.1	Order and OrderLine Tables . . . . .	6
3.1.1	Create the Order table . . . . .	6
3.1.2	Add the Order Table Foreign key . . . . .	6
3.1.3	Save and Close the Order Table . . . . .	7
3.1.4	Create the OrderLine Table and Fields . . . . .	7
3.2	Table Relations . . . . .	7
3.2.1	Print the Database Structure . . . . .	9
<b>4</b>	<b>The Invoice</b>	<b>9</b>
4.1	Create the Invoice Form . . . . .	9
4.1.1	Invoke the Form Wizard . . . . .	9
4.1.2	Add the Fields . . . . .	9
4.1.3	Add the Fields from the Order Table . . . . .	10
4.1.4	Add the Fields from the Customer Table . . . . .	10
4.1.5	Add the Fields from OrderLine and Product Tables . . . . .	10
4.2	Edit OrderLines Subform . . . . .	11
4.2.1	Shorten Width of OrderLine Fields . . . . .	12
4.2.2	LineTotal Calculation . . . . .	13
4.2.3	Order Subtotal Calculation . . . . .	13
4.2.4	Display Subtotal Calculation with the Currency Format . . . . .	13
<b>5</b>	<b>Enhance the Invoice</b>	<b>14</b>
5.1	Shorten Blank Space on the Subform . . . . .	14
5.2	Change Customer ID Label . . . . .	14
5.3	Add Company Name to Top . . . . .	14

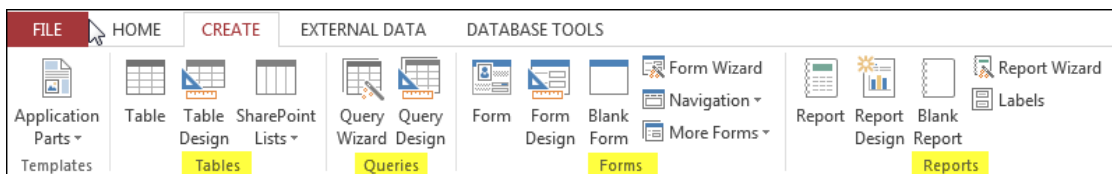
<b>6</b>	<b>Process an Invoice</b>	<b>15</b>
6.1	Display the Invoice for Data Entry . . . . .	15
6.2	Print the Invoice . . . . .	15
<b>7</b>	<b>Query the Database</b>	<b>16</b>
<b>8</b>	<b>Database Report</b>	<b>17</b>
8.1	Grouping Variables . . . . .	18
8.2	Calculate Totals . . . . .	20

# Simple Invoicing Desktop Database with MS Access 2013

A Database Management System (DBMS) is a set of procedures and tools to store and retrieve information. The database itself is the stored information. The types of information stored in the database are defined by the corresponding data structures. The **database structure** overall consists of the **tables**, their constituent fields, and the relations between them. *All* information in the database is stored in these tables.

A DBMS consists of more than just the data. The DBMS also includes forms, queries and reports. The **forms** are the displays for screen and print that allow entering new information into the database tables and displaying the existing information. The **queries** are searches of the database that extract specified information. The **reports** are formatted displays of the extracted information, for screen and for print.

These four database constructs are available from the **Create** tab in Access.

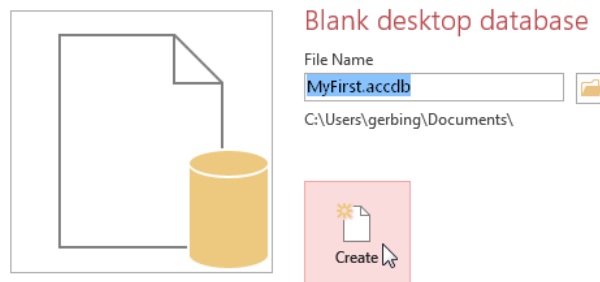
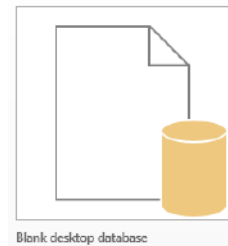


Following are the directions for creating a simple database to generate an invoice for selling products to customers.

## 1 Create a New Database

There are two ways to interact with a database: design and use. The database user can be, and most often is, oblivious to the underlying design principles. However, someone has to design the database, to create the tables and their relations, to build the forms, and implement queries and reports. The designer interacts with the database in various design modes that the most users never experience. These interactions are typically iterative, as initial designs are continually modified as a reflection of user experience.

After starting the Access application, to begin a new data base click the **Blank desktop database** icon. At the bottom-right of the opening window, specify the name of this new, blank database, here named *MyFirst*. Note where the database is stored, by default, inside the **Documents** folder. If, for example, you are working on a public computer, you will likely wish to store the file on a file system that you can access from anywhere, such as on your H: drive. Click the browse icon to change the location. Click the **Create** button to proceed.

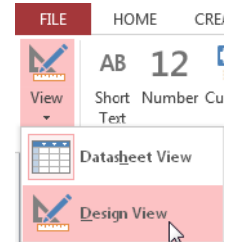


## 2 Customer and Product Tables

To construct an invoicing database, there needs to be customers who wish to purchase the products to sell. The first steps for the design of this database are the creation of the respective tables to hold the customer and product information.

Nothing can happen in a database without at least one table to store information. A new database opens with a table called Table 1, created and ready for additional modification.

A DBMS requires different views of the database from which to work. Each **View** accomplishes a different task, with its own functions, rules and menu system. The most fully featured editing of the database structure takes place in the **Design View**. To move to this View, click the **View** drop down menu at the top-left corner, select **Design View**.



### 2.1 Customer Table

#### 2.1.1 Name the Customer Table

The initial database structure consists of only a blank table named Table 1, with no fields. When moving to the **Design View** a prompt appears to save the table, with an option to provide a new name. Call the first table in the Invoicing Database the Customer table.



#### 2.1.2 Save the Changes

At any point, the changes made to a table or other database entities can be accomplished by right-clicking the corresponding tab. For the Customer table, right-click the tab named Customer, and select **Save** to save changes up to that point.



#### 2.1.3 Primary Key

After naming the table, a window appears with the fields for the table listed by rows. The first field is the table's primary key field, which can be reset by clicking the **Primary Key** icon at the top left in the toolbar. The default **Field Name** of the primary key field is **ID**.



#### 2.1.4 Data Type

Each field in a database stores information in a specific way, defined by its **Data Type**. Most data fields tend to be **Short Text** fields for generic text, but there are several other possibilities shown in the accompanying table. Change the data type by clicking the **Data Type** entry for a row and then select an option from the resulting drop down menu.

Data Type
AutoNumber

Data Type	Description
Text	Alphabetical/numerical data, up to a maximum of 255 characters, such as names, addresses, room numbers, zip codes, etc.
Memo	Up to 32,000 characters of text.
Number	Numbers for arithmetic calculations.
Date/Time	Dates and Times.
Currency	Dollars (\$).
AutoNumber	An automatic counter that assigns a number each time data is entered into a new field.
Yes/No	Binary data, such as True/False or Yes/No.

The primary key field has a default **Data Type** of **AutoNumber**. For the Customer table, the primary key field serves as the Customer ID. For a database to function successfully, each primary key field must possess specific properties, such as uniqueness. In terms of the Customer table, no two customers should have the same ID. This uniqueness property is assigned by default to primary key fields, as indicated by the **No Duplicates** term in the **Indexed** row of **Field Properties**.

General	Lookup
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates) ▼
Text Align	General

### 2.1.5 Index Fields

A database **index** functions similarly to the index in the back of a textbook. The database index allows the DBMS to more quickly locate data in an indexed field and to sort and otherwise process that data. However, the index comes at a cost. The index must be constructed and continually maintained as data is entered and modified. So only fields such as primary and foreign key fields, and other fields that are queried relatively often, such as LastName fields, are indexed.

### 2.1.6 Add Remaining Customer Fields

To add new fields to the table, just place the cursor in the first blank row in the first column, the **Field Name** column, and enter the name of the field. The Customer table should contain the following fields in addition to the Customer ID: **FirstName**, **LastName**, **Addr**, **City**, **State**, and **Zip**. To add the **FirstName** field, just enter in the name of the field and accept the default **Data Type** of **Short Text**.

Customer	
Field Name	Data Type
ID	AutoNumber
FirstName	Short Text ▼

The default **Field Size** of a **Short Text** field is 255 characters, a bit large for a first name. The standard length of a first name field is around 15 characters, so change accordingly. Now the database designer must make some decisions. Should a **FirstName** be required? If not, leave the **Required** attribute set at the default **No**. Also, first name fields are generally not indexed because their values are generally not searched for and/or sorted.

General	Lookup
Field Size	15

General	Lookup
Field Size	30
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	No
Indexed	Yes (Duplicates OK)

Next enter the **LastName** field. Set the length of the **Short Text** field to approximately 30 characters. Because the operation of customer based DBMS performs queries and sorting the last names, index this field, but do allow duplicates. Also, require this field to be present on any **Customer** record, with at least one character of content.

The remaining **Customer** table fields can now be entered, all as **Short Text** fields. No field need be longer than 30 characters, and the **State** field need only be 2 characters wide.

### 2.1.7 Enter Customer Data

The customer table has been created, so now customer data can be entered. In a production environment, data entry and modification are usually accomplished with forms, each form constructed to fulfill a specific purpose, such as entering **Customer** data. Access also provides a kind of “no frills”, more direct means for entering data without constructing a form. To do this, move from the **Design View** to the **Datasheet View**, such as by clicking the **View** drop-down menu at the top-left under **File**, or by right-clicking the **Customer** tab and selecting the **Data Sheet** option.

	Field Name	Data Type
🔑	ID	AutoNumber
	FirstName	Short Text
	LastName	Short Text
	Addr	Short Text
	City	Short Text
	State	Short Text
	Zip	Short Text

The **Datasheet View** lists the fields horizontally.

ID	FirstName	LastName	Addr	City	State	Zip
* (New)						

Each row represents the data for a specific record. In this example, the **ID** field is set to the data type of **AutoNumber**, so no value is entered into that field. Begin by entering a customer’s first name, and then continue for the rest of the fields for that customer. Data can be entered at any one time for as many customers as desired. Here data for two customers are entered.

ID	FirstName	LastName	Addr	City	State	Zip
1	Sam	Smith	44 NE 44th St	Portland	OR	97201
2	Sally	Surich	55 SW 55th St	Portland	OR	97231
* (New)						

### 2.1.8 Print Customer Data

Printing from Access is the same as from any application. Make sure the window is open and selected, with the data in view, here from the **Datasheet** view. Then, from the **Office Button** choose the **Print** option.

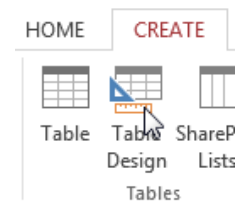
When finished entering customers, right-click the yellow Customer tab above the ID field and **Save** and then **Close** the window.

## 2.2 Product Table

Before creating an invoice, there has to be customers who wish to purchase the products the company has to sell. Next create the Product table and enter some products. Follow the same procedures outlined for the Customer table. First create the Product table, then enter the data in **Datasheet View**.

### 2.2.1 Create Table

To create a new table go to the **Create** tab at the top-left of the Access window and click the **Table Design** button.



### 2.2.2 Define Fields and Primary Key

A Product table includes a Product ID that uniquely identifies each product. Within the database, the Product ID serves as the Product table's primary key. In the first row of the table as viewed in **Design View**, enter ProductID as the field name and select **AutoNumber** as the **Data Type**. To set as the primary key, right-click in the cell for the **Field Name** and select the **Primary Key** option.

Field Name	Data Type
ProductID	AutoNumber

Primary Key

### 2.2.3 Add Description and Price Fields

To complete this minimal product table, also include a Description field, which contains a word or brief phrase that describes the product, as well as the selling Price to appear on the invoice. Keep the default **Data Type** of **Short Text** for the Description field, but limit to about 30 characters. Set the **Data Type** for the Price field as **Currency**.

Product	
Field Name	Data Type
ProductID	AutoNumber
Description	Short Text
Price	Currency

### 2.2.4 Add Products

Again, switch to **Datasheet View** as was done with the Customer table on page 4. Save the table with the name of Product. Then add some products to the company store.



ProductID	Description	Price	Click to Add
1	Jacket Liner	\$199.00	
2	Pant Liner	\$199.00	
3	Gloves	\$139.00	

### 2.2.5 Print Product Data

Printing from Access is the same as from any application. Make sure the window is open and selected, with the data in view, here from the **Datasheet** view. Then, from the **Office Button** choose the **Print** option.

## 3 Database Structure

### 3.1 Order and OrderLine Tables

The minimal structure of an invoicing database contains four tables, two more in addition to the already created Customer and Product tables. One of these tables is the Order table, which stores the individuals orders. Another table is the OrderLine table, which contains individual products ordered as well as the corresponding quantity for each order. However, unlike the previous Customer and Product tables, create the tables but do *not* enter data directly into the tables. Instead, an Invoice form will be created, and only through this form will data be entered into the Order and OrderLine tables.

#### 3.1.1 Create the Order table

Create the Order table and add the respective fields to each table. The Order table contains the primary key OrderID as well as the OrderDate. Set the primary key for the table. The data type for OrderDate is Date. Right-click the table tab above the **Field Name** field and **Save** the table. When prompted, name the table Order.

#### 3.1.2 Add the Order Table Foreign key

The Order table also contains a foreign key that links a selected order with the customer that placed that order. Name the foreign key fkCustID and assign the same attributes as the corresponding primary key CustID in the Customer table except that a foreign key is not unique. In this case, one customer will hopefully have many invoices. A useful convention places the primary key in the first field and any foreign keys in the immediately adjoining fields.

**Caution** → The attributes of a foreign key match the attributes of the corresponding primary key except that the content of a foreign key is *not* Unique.

Since the corresponding primary key has a **Data Type** of **AutoNumber**, set the **Data Type** of the foreign key to **Number**.

Order	
Field Name	Data Type
OrderID	AutoNumber
fkCustID	Number
OrderDate	Date/Time

Also, make it required and indexed, though with duplicates allowed.

General	Lookup
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (Duplicates OK)
Text Align	General

### 3.1.3 Save and Close the Order Table

Right-click the tab now named Order and once again **Save** the data. Only this save is necessary, however, as with any computer application, it is advisable to save often, not just when the entered information is complete. After saving, **Close** the window. Generally, when the immediate task for a window is completed, the window should be closed to avoid cluttering the interface and to make sure everything is saved and out of the way before continuing with the database operation as a user.

### 3.1.4 Create the OrderLine Table and Fields

Follow the above procedure for the OrderLine table. The OrderLine table is a many-table to both the Order table and the Product table, which requires two foreign key fields. These foreign key fields respectively link each OrderLine record to a specific Order as well as to a specific Product. Again, the foreign key fields are required and indexed, but with duplicates. The **Data Type** of the LineTotal field is **Currency**.

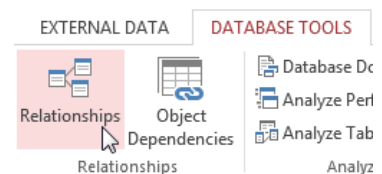
There should also be a primary key for this table, even though it is not used directly in the database, and is not shown in the following figure. Name the primary key field something such as LineID, and assign a **DateType** of **AutoNumber**.

OrderLine	Field Name	Data Type
	LineID	AutoNumber
	fkOrderID	Number
	fkProductID	Number
	Qty	Number
	LineTotal	Currency

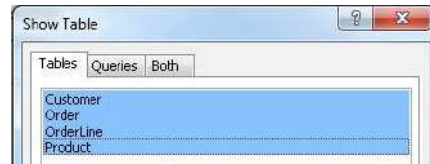
When finished, **Close** the window that contains the OrderLine table. Again, do not enter any data, a task accomplished with the Invoice form to be constructed later.

## 3.2 Table Relations

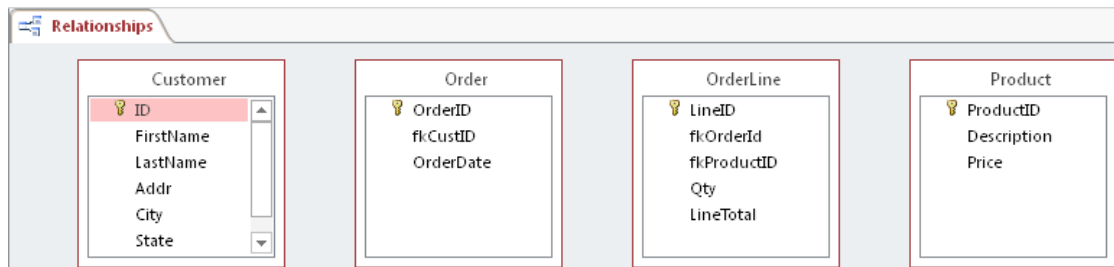
A crucial element of database structure is to relate the tables with the appropriate one-to-many relationships. To accomplish this, click the **Database Tools** tab at the top of the screen and then click the **Relationships** button.



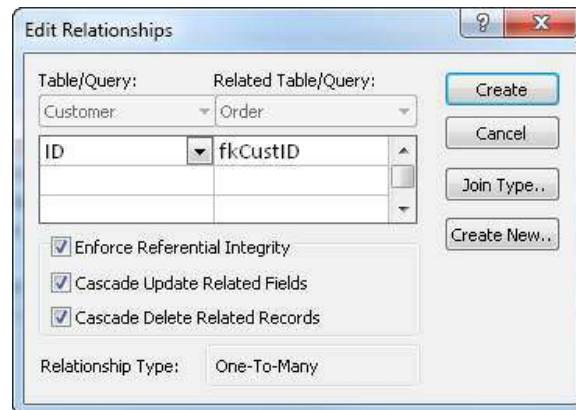
Then select all four tables in the resulting Show Table window. To do this, click on the first table, Customer, then hold the **Shift** key down and click on the last table in the list, Product. Then click the Add button.



The result is all four tables and their fields displayed in the Relationships window.



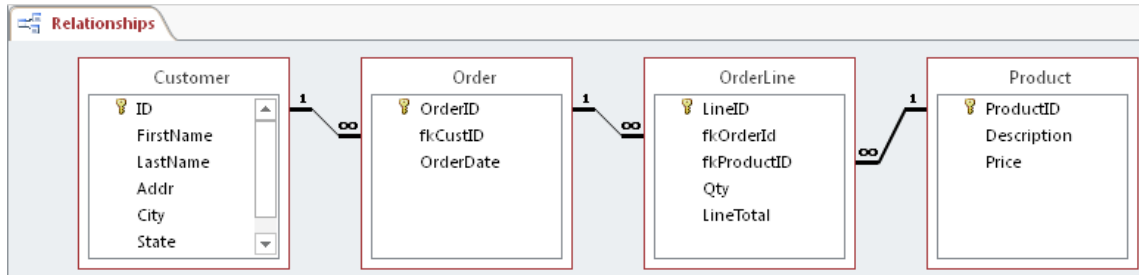
Now relate the tables with their corresponding one-to-many relationships. Click the Customer ID field, hold the mouse down and drag to the foreign key for Customer ID in the Order Table. Release the mouse and the Edit Relationships table appears. Check the Enforce Referential Integrity as well as the Cascade Update and Cascade Delete checkboxes. Then click the Create button.



Similarly, define the one-to-many relationships that related the Order table to the OrderLine table and the Product table to the OrderLine table. Relating all four tables results in the following database structure, as shown in the Relationships window. This figure illustrates the Access representation of the general invoicing database structure.

**Caution** → Check that your database structure *exactly* matches this structure in the accompanying figure before continuing.

These four tables and the corresponding three one-to-many relationships define the minimal structure required to produce a working invoicing database. Each of the four tables



should contain the fields specified in the preceding figure. When complete, Close the Relationships window.

### 3.2.1 Print the Database Structure

To print the database structure diagram, under the **Database Tools** ribbon, make sure the **Relationships** button is pushed so as to display the data base structure with the relationships and tables displayed. Then go to **Design** ribbon which appears when the relationships are displayed and select the **Relationships Report** button. Click the button, which causes the database structure diagram to appear in a window ready to print. Then print, such as to a pdf, as usual. To print a pdf, choose the **PDF** or **XPS** button toward the top-right of the **File** ribbon.

## 4 The Invoice

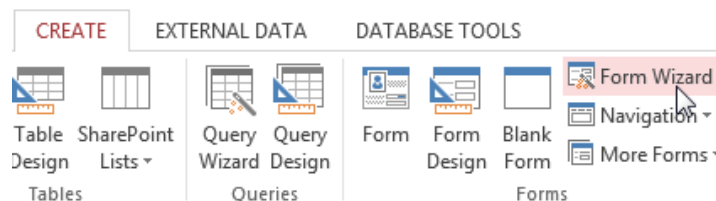
There is no Invoice table. Instead the Invoice is a form organized as a main form and subform. Each form is primarily identified with a single table. The main form is tied to the Order table, and the subform is tied to the OrderLine table. Use the **Form Wizard** to construct the Invoice form, including the main form and the corresponding subform.

### 4.1 Create the Invoice Form

Forms can be constructed manually, from “scratch”, but the **Form Wizard** simplifies and speeds up the task.

#### 4.1.1 Invoke the Form Wizard

To create a form, click the **Create** tab at the top of the screen and select the **Form Wizard**.



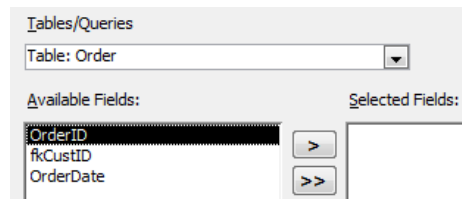
#### 4.1.2 Add the Fields

Add the fields to display on the Invoice form table by table. Include all fields in the database on the form except (a) the primary keys of the Customers and Parts tables, which have the

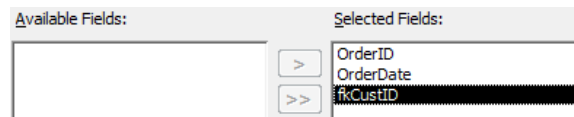
same values as the corresponding foreign key fields, and (b) the OrderLine primary key for which there is no interest to display the corresponding value. The **Form Wizard** displays the fields on the form in exactly the same order as the fields are selected. Accordingly, select the fields in the same order as they appear on the main form and subform, respectively, as shown in the accompanying figure. The fields on the form can be rearranged later at any time with the **Form Editor**, but it is easier to specify their order before constructing the form.

### 4.1.3 Add the Fields from the Order Table

Begin by adding fields from the Order table because those fields are the first fields listed on the Invoice.

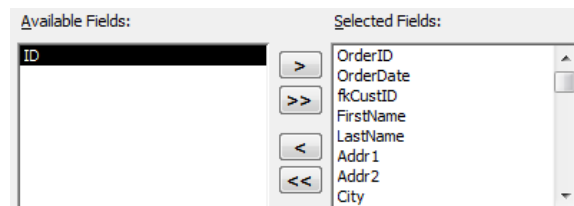


Then select the fields from the Order table in the order of appearance on the Invoice. The Order ID and OrderDate precede the customer information.



### 4.1.4 Add the Fields from the Customer Table

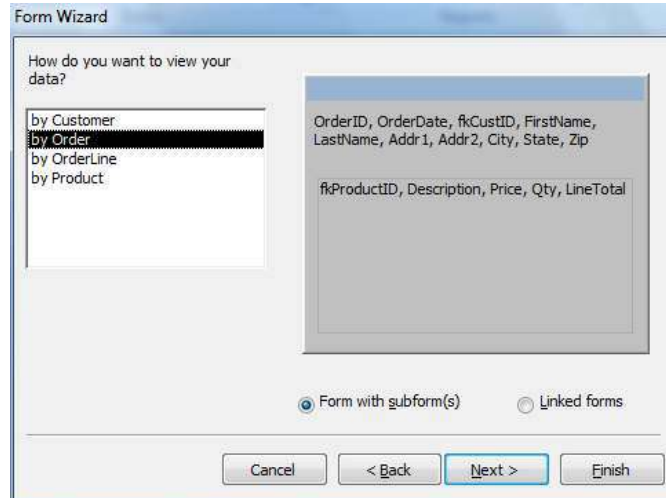
Now switch to the Customer table and add the relevant fields, except for the primary key or ID field. For automatic lookup to work on the Invoice form, the Customer ID appears on the form as the Order table foreign key. As a database user, as opposed to a designer, enter a value and the relevant customer information will automatically appear given the pre-defined relationship that links the Customer and Order table information.



### 4.1.5 Add the Fields from OrderLine and Product Tables

Similarly, from the OrderLine table add the fields Qty, LineTotal, and the foreign key for Product ID, all of which will appear on the Invoice, here on the OrderLine subform. From the Product table, add Description and Price. Because Access knows the one-to-many relationship between the Order and OrderLines tables, Access automatically generates the

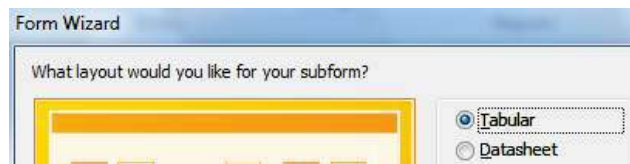
OrderLines subform when fields from the OrderLine table are included in the list of all fields on the Order form. When finished adding the fields, click the **Next** button. The following **Form Wizard** window appears. The form should be viewed by **Order**, and the **Form with subform(s)** radio button should be selected.



**Caution** → Check that the fields entered *exactly* match the order and field names in the accompanying figure before continuing.

If there is not an exact match, click the **Back** button and redefine the ordering of the fields by moving fields out of the form and then back again at the proper location.

Click the **Next** button. Select the **Tabular** radio button in the following window.



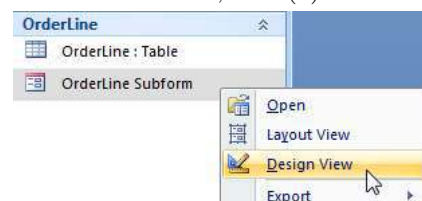
When finished with the resulting dialogue boxes, Access automatically creates two forms, the main form and the corresponding subform.

Choose to either begin entering data or further edit the forms. For our purposes, our order form is not ready to receive data without further editing.

## 4.2 Edit OrderLines Subform

The OrderLines subform needs editing in Design mode before the subform can function properly as part of an order form or an invoice. Separately edit the subform to (a) adjust the sizes of the fields (Text Boxes), (b) calculate the LineTotal for each line, and (c) calculate the subtotal across all the order lines.

Close down the resulting window and call up the newly created OrderLine form for editing. Right-click the form name and select **Design View**.



Form Wizard

What titles do you want for your forms?

Form:

Subform:

That's all the information the wizard needs to create your form.

Do you want to open the form or modify the form's design?

Open the form to view or enter information.

Modify the form's design.

Cancel < Back Next > Finish

### 4.2.1 Shorten Width of OrderLine Fields

The first task is to shorten the width of the fields. The default values are much too large. To do this, click a field display, between the **Form Header** and **Form Footer**, then, to simultaneously select the corresponding header field, hold the **Cntrl** key down and click the corresponding text box that contains the header. For example, as illustrated, consider `fkProductID`. The corresponding header field, which contains the name of the corresponding column, also shortens. For the `fkProductID` field, change the text of the header. This technical name does not need to be displayed for the database user. Choose something more natural such as `PrdID`.

OrderLine Subform

Form Header

fkProductID	Description			Price	Qty	
fkProductID	Description			Price	Qty	Line

Form Footer

After shortening the fields, the OrderLine subform appears approximately as follows.

OrderLine Subform

Form Header

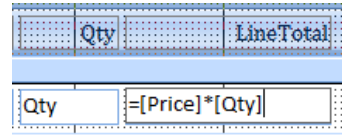
fkProductID	Description		Price	Qty	LineTotal
fkProductID	Description		Price	Qty	LineTotal

Form Footer



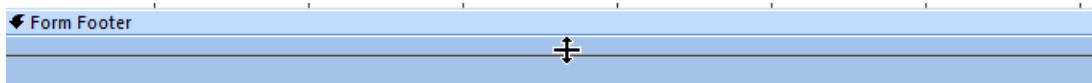
### 4.2.2 LineTotal Calculation

The LineTotal field is a calculated field. That is, its value is not entered or read from existing data stored in a table, but is instead calculated from data in other fields. To instruct Access to calculate LineTotal as the product of Price and Qty, specify the expression `= [Price] * [Qty]` for the LineTotal field.



### 4.2.3 Order Subtotal Calculation

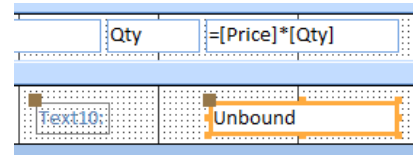
Next calculate the Order Subtotal as the sum of all the individual LineTotals. To do this, add a Text Box to the Form Footer of the subform. To add the Text Box to the Form Footer, first open some space in the footer area. Move the cursor to the footer area and then drag down.



Now add the Text Box. Go to the Ribbon within the Design tab and locate the icon for the Text Box, near the top near the Design tab.

**Caution** → Add the subtotal expression to the footer of the subform, *not* the detail area.

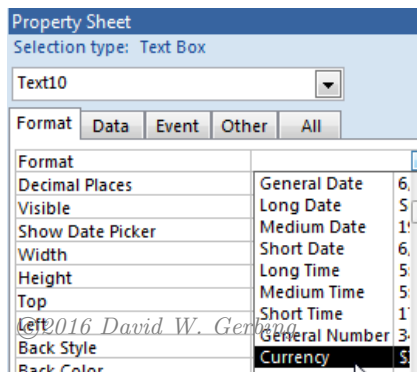
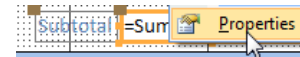
Drag the Text Box icon down to the footer area. Access also adds a corresponding Label. The Text Box itself is labeled as Unbound because the value it displays is not linked to any table in the database. The attached Label is a static label that does not display data, but just provides information to the user regarding the contents of the Text Box.



Change the text of this Label to Subtotal. Specify the following expression in the Text Box: `= Sum([Price] * [Qty])`. The Sum function sums all the individual LineTotals. To drag the label separately from the Text Box to a new position closer to the Text Box, select the Label and then drag by the little gray box at the top left of the Label.

### 4.2.4 Display Subtotal Calculation with the Currency Format

The subtotal is in \$'s, so change the default display format accordingly. Right-click the Text Box and select Properties from the bottom of the resulting menu.



The resulting Property Sheet lists the many properties that describe the object on the form, here the unbounded Text Box. At the top of the list, under the Format tab is the property of the same name, Format. Under the associated drop-down menu, choose the Currency option.

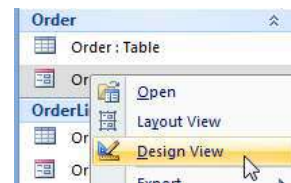
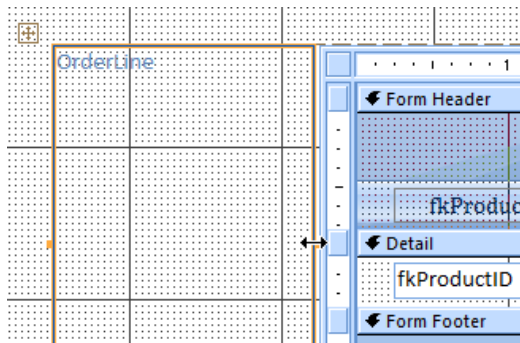
Now save the OrderLine form and close its window.



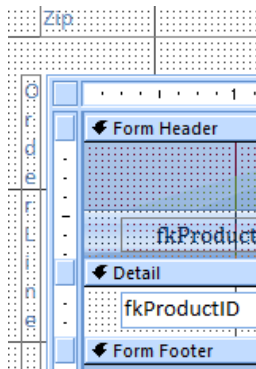
## 5 Enhance the Invoice

### 5.1 Shorten Blank Space on the Subform

Go to **Design View** for the Invoice and move the subform to the left. First reduce the space to the left of the subform. Select the label called **OrderLine**.



Then narrow or eliminate the label, which initially takes up way more room than necessary.



### 5.2 Change Customer ID Label

By default, Access labels each **Text Box** with the name of the corresponding field from the corresponding database table. The Customer ID field is the foreign key field from the Order table, called **fkCustomerID**. The user does not need or wish to see the technical jargon, so change the label name to something like **Customer ID**.



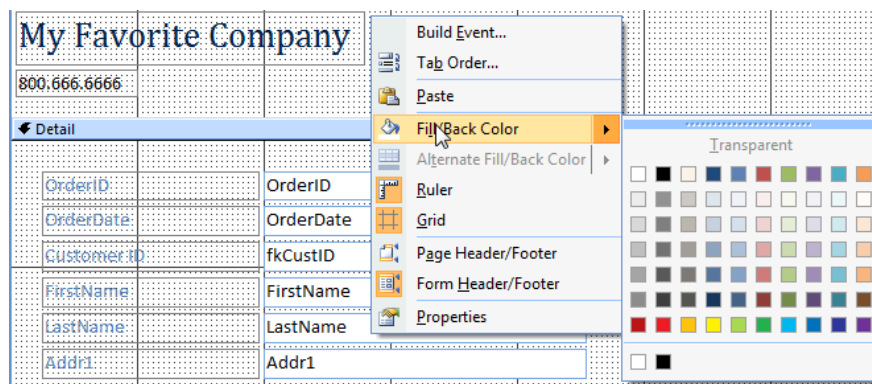
### 5.3 Add Company Name to Top



The default label at the top of the Invoice is the name of the table for which the Invoice form is associated, the **Order** table. To provide a more appropriate label for the customers who receive these invoices, change the name to **Invoice** and add company contact information. To add more labels, choose the **Label** icon from the **Ribbon** in **Design View**.

Drag to the top of the Invoice form and add the Company Name. Add another label for the phone number, web site, etc.

As an option, change the background color and/or foreground color of the label by right-clicking the label and then choose **Fill/Back Color** from the resulting menu.



Now save the Invoice.

## 6 Process an Invoice

### 6.1 Display the Invoice for Data Entry

Now the system is ready to generate an actual Invoice. Open the Invoice form. Add the date. Then add a valid Customer ID. Automatic lookup of the customer's information from the Customer table leads to the display of that information on the Invoice form. Then, move to the subform and add a Product ID. Again, automatic lookup ensures that the relevant Product information, Description and Price, are displayed. Then add a Quantity. Do this as for many products as desired.

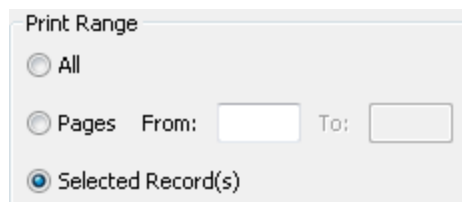
### 6.2 Print the Invoice

When directed to print, by default, Access will print all the records for the specified table. Moreover, the resulting information runs across pages. For example, if there is more than one invoice stored within the system, then the second invoice will automatically print after the first, and if the first invoice did not take up a full page, the second invoice will begin on the same sheet of paper below the first invoice. Of course when printing invoices, the goal is to have one invoice printed per page.

In actual practice, the database designer would usually put a Print button on the page somewhere, such that pressing the button would cause just that record (invoice) to print. Implementing this button requires code, which is not difficult, but not something to pursue here.

When the Invoice displays the information for a customer's sale, just that Invoice can be printed with the usual File → Print sequence. From the print dialogue box, select **Selected**

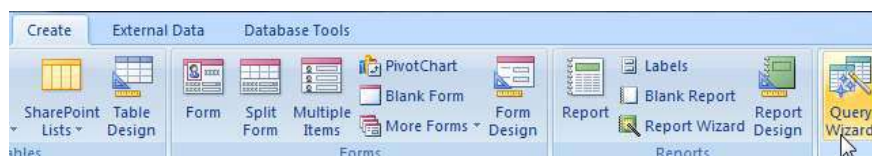
Record(s) in the lower left-hand portion of the dialogue box. Otherwise, all Invoices in the database are printed.



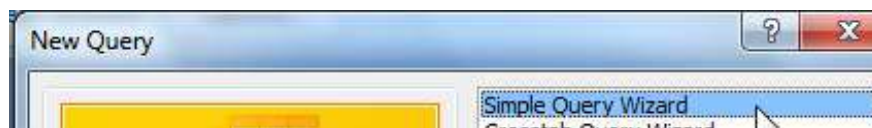
## 7 Query the Database

A primary purpose of a database is to locate specific information about some of the data in the table(s). A query is a search for general or specific data in a field or fields in your database. Examples include the first and last names and birth dates of all employees, just employees with a last name of Smith, just the people from Oregon, salaries > \$100,000, etc.

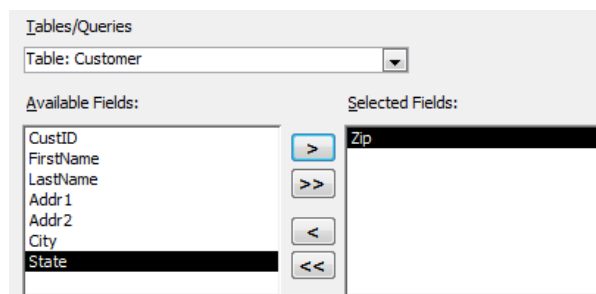
Consider the query to locate all customers in a specific zip code, 97201, a zip code with at least two customers. Begin with the **Query Wizard**, found under the **Create** Tab.



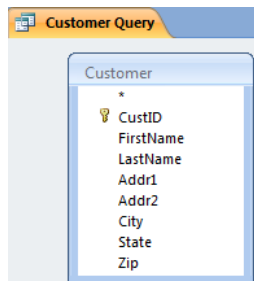
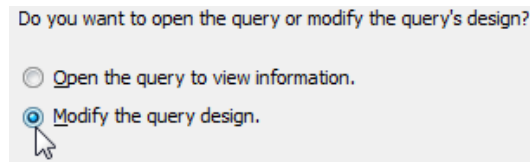
Select the **Simple Query Wizard** option.



In the **New Query** window, select the table or query about which to generate the report, and then select from the available fields. Select all fields that you wish to appear on the query as only these fields will appear.



After defining the information to appear on the query, the next step is to specify the criterion for each field, specifically here to specify the desired zip code. Otherwise, the query will simply return all the records of the table, albeit with just the specified fields listed. So, return to the **Design** environment.



To select additional fields, double-click the relevant fields from the resulting field list. Now add the relevant criteria, here for the zip code field.

Field:	[Zip]	FirstName	LastName
Table:	Customer	Customer	Customer
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	= "97201"		
or:			

This simple query is now designed, so now **Run** the query. Do so by saving the query and closing the query window. Then in the Access List of Objects on the left-hand side of the screen, identify the new Query object. Then right-click on the object and select **Open** to run the query.

The results are a list of all customers in the database with the specified zip code of 97201. The information displayed for each customer has been defined in the design of the query.

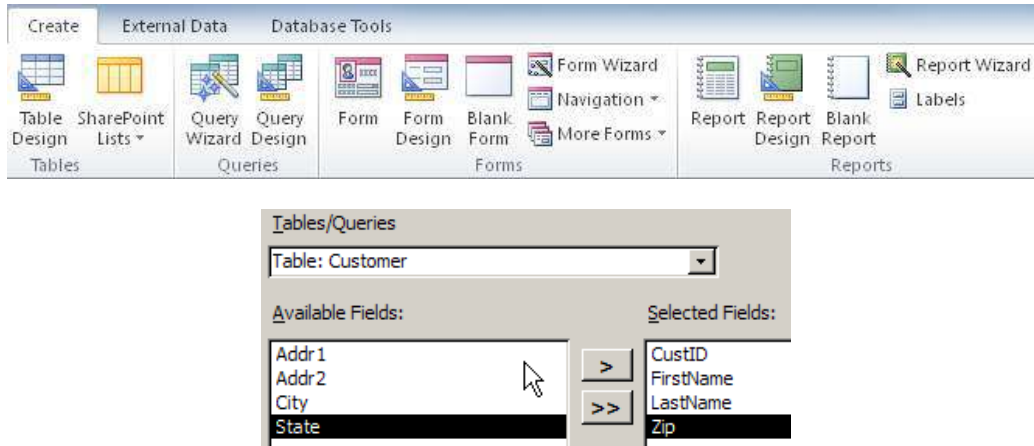
Zip	FirstName	LastName
97201	Sam	Smith
97201	Ben	Bat

Of course, this query would work just as well in a database with tens of thousands of Customers, and, about as fast.

## 8 Database Report

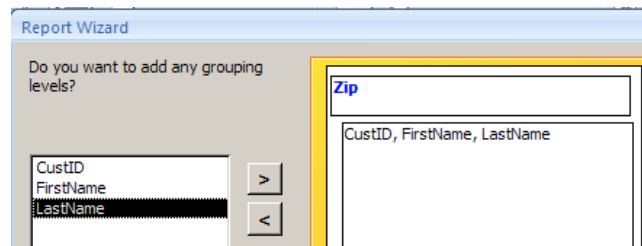
To create a template for an Access report, go to the **Create** tab and select the **Report Wizard**. The first choice is to select a table upon which the report is primarily based. Here select the Customer table. Then select the fields that appear on the report. Fields from additional related tables can be also selected. The report is formatted similar to the format produced by the **Forms Wizard**, and can be similarly edited.

After selecting the Customer table, select the fields to appear on the report.

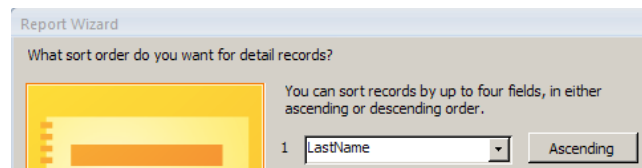


## 8.1 Grouping Variables

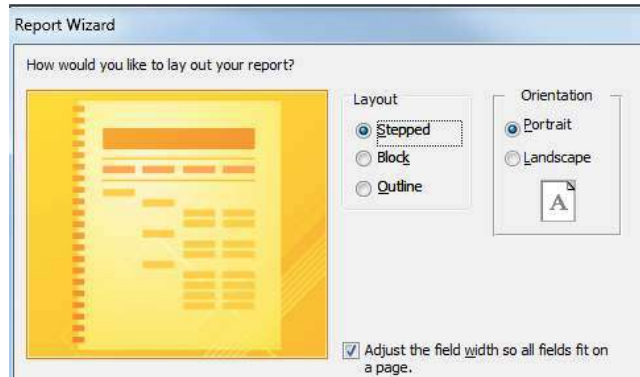
The grouping levels define the break points where a new grouping is started. Set Zip as a grouping variable and the report will generate a break point for each different zip code.



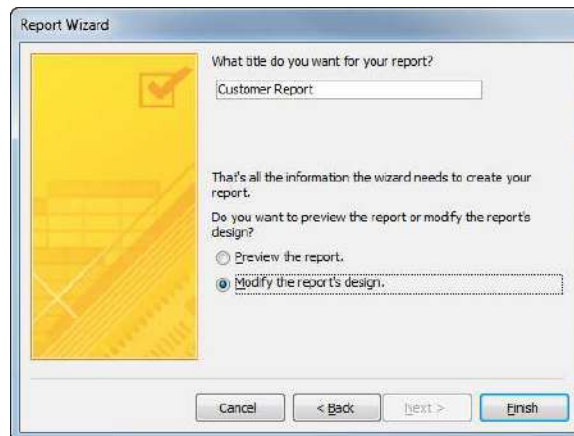
Within each grouping, there is usually a specified sort order, such as alphabetical. Here sort the customers by LastName within each zip code.



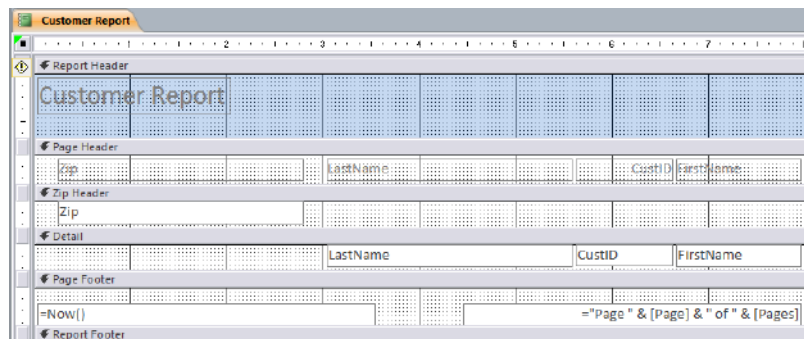
Choose the layout as desired. The default **Stepped** is fine.



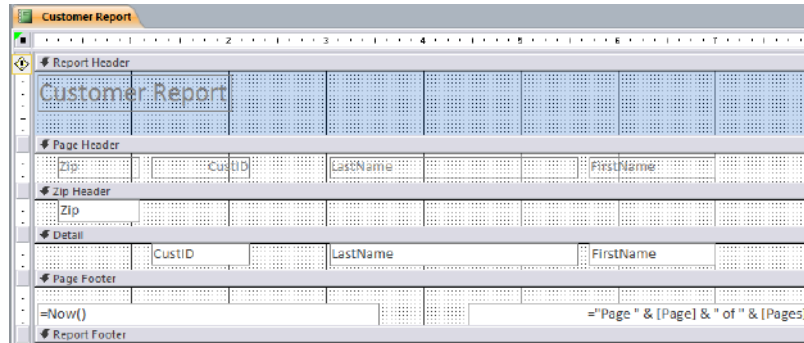
For the title, enter a meaningful title such as **Customer Report**.



The **Form Wizard** has completed its work, but now the form should be manually adjusted in **Design View**. The **Zip** fields are too wide, and the **CustID** field is in the wrong place, or at least not the most useful place, in between the **LastName** and the **FirstName**.

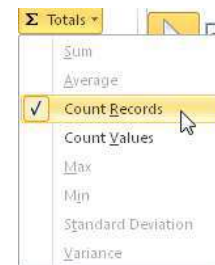


Edit the report with the same tools and process as editing the **OrderLine** subform. The following is one reasonable result.

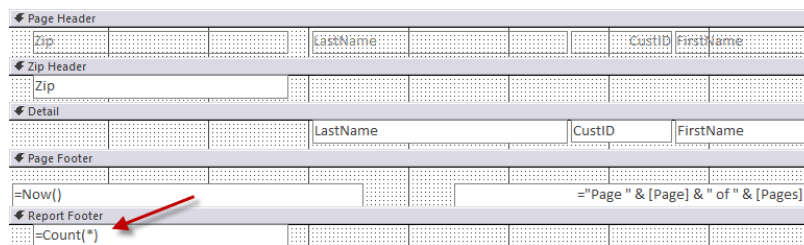


## 8.2 Calculate Totals

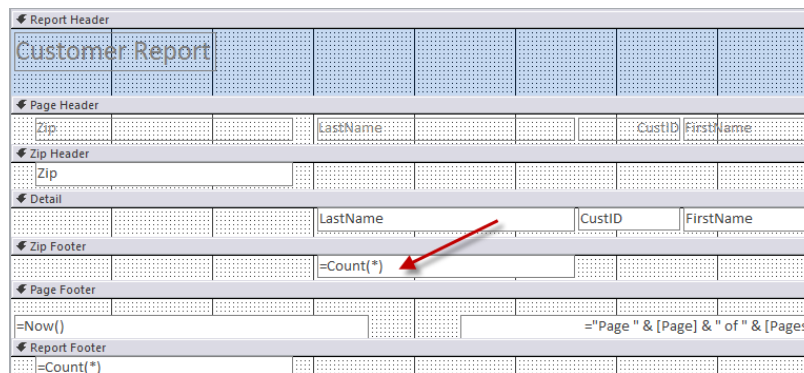
We want our report to do some basic arithmetic, providing simple summaries of interesting features of our data. With your report open in **Design View**, click the field/text box that you want to summarize, which here is the Zip text box right under Zip Header. In **Design View**, go the Design tab, where the Totals button is located in the Grouping and Totals section. Click the Totals button to show a drop down menu to select the desired function, here **Count Records**.



Access adds a text box to the report's group and report footer section that contains the expression that performs the calculation.



To get the count of records, that is, customers, *within* each zip code, go to the LastName text box in the Detail section. Click to select, and then again go to the Totals button and select **Count Records**.



Now polish the report. Align the Count fields and place a label at the top of the fields, as shown below. To add a label, such as the word Count, do what you did for adding a title to your invoice: Choose the **Label** icon from the Ribbon in **Design View**.

Report Header						
Customer Report						
Page Header						
Zip	Count	LastName	CustID	FirstName		
Zip Header						
Zip						
Detail						
		LastName	CustID	FirstName		
Zip Footer						
	=Count(*)					
Page Footer						
=Now()			="Page " & [Page] & " of " & [Pages]			
Report Footer						
	=Count(*)					

Customer Report						
Zip	Count	LastName	CustID	FirstName		
52306		Jones	2	Sally		
	1					
97201		Bat	3	Ben		
		Smith	1	Sam		
	2					
	3					

Wednesday, April 25, 2012 Page 1 of 1

Now right-click the screen and select **Report View**. The result is the completed report.

The form wizard generates the report with a default style. Access provides many other styles, which can be chosen from the **Themes** drop-down menu at the top left of the **Design** ribbon.

