

Simple Invoicing Desktop Database with MS Access 2013

© 2013 by David W. Gerbing
School of Business Administration
Portland State University

October 30, 2013

Contents

1	Create a New Database	1
2	Customer and Product Tables	2
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
3	Database Structure	6
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
4	The Invoice	9
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	12
4.2.1	Shorten Width of OrderLine Fields	12
4.2.2	LineTotal Calculation	12
4.2.3	Order Subtotal Calculation	13
4.2.4	Display Subtotal Calculation with the Currency Format	13
5	Enhance the Invoice	13
5.1	Shorten Blank Space on Invoice	13
5.2	Change Customer ID Label	14
5.3	Add Company Name to Top	14

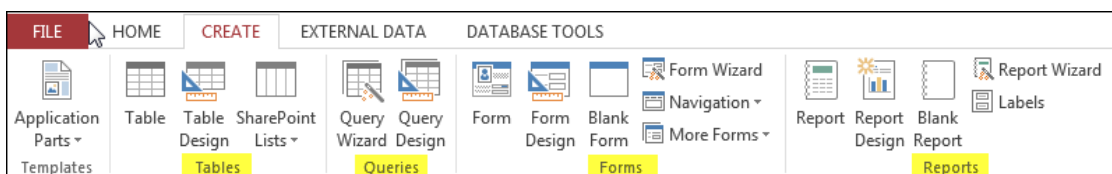
6	Process an Invoice	15
6.1	Display the Invoice for Data Entry	15
6.2	Print the Invoice	15
7	Query the Database	16
8	Database Report	17
8.1	Grouping Variables	18
8.2	Calculate Totals	20

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.

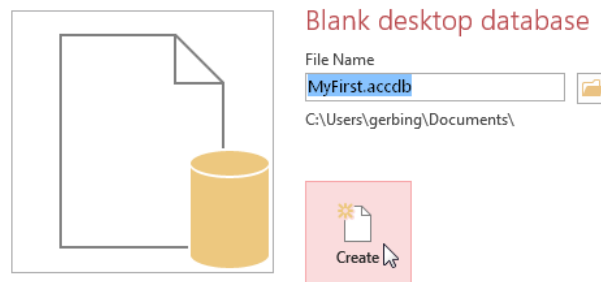
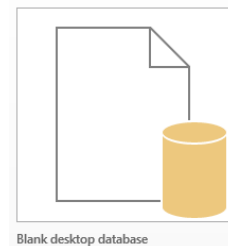


Following are the directions for creating a simple database to generate an invoice for selling products to customers.

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.

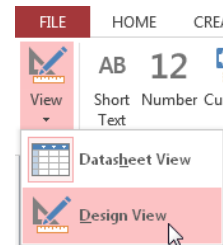


2 Customer and Product Tables

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.

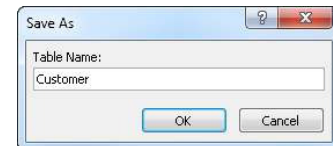
A DBMS requires different views of the database from which to work. Each **View** accomplishes a different task, with its own functions, rules and menu system. The most fully featured editing of the database structure takes place in the **Design View**. To move to this View, click the **View** drop down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



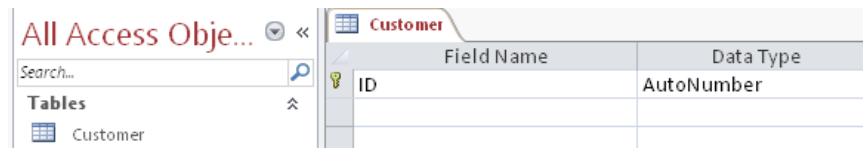
2.1.2 Save the Changes

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.



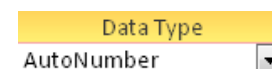
2.1.3 Primary Key

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.



[Click here to download full PDF material](#)