Northeastern University College of Computer and Information Science

CS1100: Access Lab 1 Creating and Querying Databases

To complete this assignment you must submit an electronic copy to Blackboard by the due date. You will be creating your own database, so there is no starter file for this lab. In this lab you are asked to create a new database for inventory management of a small computer reseller, populate the database with sample data and formulate queries to retrieve data.

Creating the Database

The founding of *WizBang Computers, LLC* is the dream of Jason Dunn, a long time computer enthusiast. He finally has enough business customers to make his side venture a full time job. His goal is to build *WizBang* into the premier reseller and servicer of business computers for small to mid-size firms. As his sales of custom-built PCs have taken off, Jason realizes that he needs a better inventory management tool than the Excel spreadsheet he has been using thus far. However, Jason is too busy providing on-site support services for his clients and therefore has hired you to build an Access database for *WizBang* that will allow him to track his inventory, customer contacts, and orders. Eventually, the data will not only help him manage his inventory better, but also provide insight into sales for better sales forecasting and inventory planning.

Planning the Tables

A database development effort starts with an analysis of the data that needs to be stored. Commonly a data model is constructed to assist with this effort. So far, the following tables have been identified as being important:

Table Name	Content
Contacts	Information about customers
Products	Information about products available for sale
Orders	Information about orders placed by customers
ZipCodes	Cities and states and their associated zip code
LineItems	Lists of products and the orders in which they were ordered

- Renk 1
 - 1. Start Microsoft Access and create a blank database by selecting "Blank database" from the list of available templates.
 - 2. Name your database using the following naming rule: Orders_*Lastname*, *e.g.*, Orders_Smith

- 3. Note the location of the database (remember the folder in which it was saved; most likely the Documents folder.) Lastly, click on **Create** to create the database. Note that data is automatically saved as soon as you change something in Access; there is no separate Save mechanism.
- 4. Access will automatically create your first table under the temporary name "*Table1*". Before you can rename the table, we need to first define its columns (*i.e.*, its data content) and then close the table. We will start by creating the *Contacts* table as follows:
 - a. Switch to table design, by clicking on the Design View icon. When prompted, save the table under the name "*Contacts*".

View
-

	Contacts						
4	Field Name	Data Type					
8	ContactID	AutoNumber					
	FirstName	Short Text					
	LastName	Short Text					
	Address	Short Text					
	ZipCode	Short Text					
	PhoneNumber	Short Text					

b. Define the fields of the *Contacts* table as follows:

c. Now create the *Products* table by selecting the **Create** tab and clicking on **Table**.





Once again, switch to the Design View by clicking on the Design View icon. When prompted, save the table under the name "*Products*". Define the table as follows:

	Products						
	Field Name	Data Type					
P	ProductID	AutoNumber					
	ProductName	Short Text					
	UnitPrice	Currency					
	UnitsInStock	Number					
	RetailPrice	Currency					

d. Follow the same process to create the *Orders*, *LineItems*, and *ZipCodes* tables. The definitions of the tables are below:

The second secon						
	Field Name	Data Type				
P	OrderID	AutoNumber				
	ContactID	Number				
	OrderDate	Date/Time				
T	LineItems					
2	Field Name	Data Type				
8	LineItemID	AutoNumber				
	OrderID	Number				
	ProductID	Number				
	QuantityOrdered	Number				
	PriceCharged	Currency				
ZipCodes						
	Field Name	Data Type				
P	ZipCode	Short Text				
	City	Short Text				
	State	Short Text				

e. When done, close all of your tables, by clicking the right mouse button on any of the table tabs and selecting "*Close All*". You will be prompted to save the tables. Choose "*Yes*" for all of them.

Populating the Database

Now that the database has been created, we need to put data into the database. This is often done by loading an Excel table containing data into the database or by writing a nice user interface that users can use to type in the data. In our case, we will populate the data by using the Access table data entry tool.

1. Let's start by adding a few of our contacts' information into the *Contacts* table. Doubleclick on the *Contacts* table in the **Tables** browser. That will bring up the table entry sheet. Here's what it should look like:

All Access Objects 💿 «		Contacts			
Search P	\angle	ContactID 👻	FirstName 🕞	LastName 🚽	Add
Tables	*	(New)			
Contacts					

2. Enter the following data into the table sheet view. Note that you **DO NOT** enter any value for the *ContactID* field. That field is an auto number field that is automatically filled in by Access.

	Contacts								
2		ContactID	*	FirstName 🔹	LastName 🔹	Address 👻	ZipCode 🔸	PhoneNumt -	
	+		1	Peter	Levoy	17 Halpern Ave	02536	508 904 6588	
	+		2	Jane	Wu	68 Trust Cir	09056	910 332 4654	
	+		3	Jim	Wobek	5 Main Ln	02536		

Note that we don't store where the customer lives; we only store the zip code. The actual city and state is stored in the ZipCodes tables which this table references.

As the fourth contact enter your own name, address, and phone number. To protect your privacy, you may add a fictitious address and phone but you <u>must</u> enter your correct name and you must provide some kind of address. (Your zip code should be added to the zip code table as well.)

3. Follow the same procedure to continue with your data entry by adding the following data to the other tables:

	Products								
$\left \right $	ProductID	•	ProductNam	•	UnitPrice 👻	UnitsInStock -	RetailPrice 👻	Click to Add	-
		1	Intel i5 2.7Gh	z	\$89.99	1	\$129.99		
		2	1TB Seagate		\$99.00	:	\$119.99		
		3	Windows 7		\$49.99	1:	L \$99.00		
*	(Ne	w)							
	ZipCodes								
\square	ZipCode	*	City	-	State 👻	Click to Add	•		
	02536		Falmouth		MA				
	09056		APO		AE				
*									
	Orders								
	OrderID	•	ContactID	-	OrderDate 👻	Click to Add	•		
		1		1	4/1/2011				
		2		1	4/19/2011				
		3		2	5/13/2011				
		4		1	6/1/2011				
		5		2	9/1/2011				
		6		2	9/5/2011				
*	(Ne	w)							

This table is important as it establishes which customer placed which order. For example, order number 1 was placed by contact 1 (Peter Levoy) on 4/1/2011. In fact, Peter also placed orders 2 and 4, while orders 3, 5, and 6 were all placed by contact 2 (Jane Wu).

What exactly was ordered as part of each order, how much of it, and how much was actually charged to the customer is stored in the *LineItems* table. We do this so that there is no redundantly stored information in our database. Here is the *LineItems* table:

	Ineltems										
4	LineItemID 👻	OrderID 🚽	ProductID 👻	QuantityOrd 🗸	PriceCharge 👻	Click to Add	Ŧ				
	1	1	1	2	\$129.99						
	2	1	2	1	\$119.99						
	3	2	1	1	\$129.99						
	4	2	3	5	\$94.50						
*	(New)										

Establishing Relationships

In order for our queries to be properly executed by Access, we need to tell Access how we have linked the information in the different tables. For example, *Contacts* and *ZipCodes* are linked. So are *Orders* and *Contacts*, as well as *LineItems* with *Orders* and *Products*.

To define the relationships, we need to close all of our tables in the table view and then switch to the **Database Tools** view.

File Hom	e Create	External Data Database To	ols
Compact and Repair Database Tools	Visual Run Basic Macro Macro	Relationships Object Dependencies Relationships	 Database Docu Analyze Perform Analyze Table Analyze
All Access Obje	ects 💌 «		
Search	<u>م</u>		
Tables	*		
Contacts			
LineItems			
Orders			
Products			
ZipCodes			

Once you click on the **Relationships** tool, you get a list of tables for which you can establish relationships. Select all of them and then click **Add** and then **Close**. You can also add each table one by one.



Now we are ready to create the relationships. To do so, click on the field of one table that points to another table and then drag the line towards the other table and drop if off over the field that is the linked field. For example, to create the relationship between **Contacts** and **ZipCodes**, click on the *ZipCode* field of the **ZipCodes** table and then drop that field off on top of the *ZipCode* field of the **Contacts** table. **It's important that it is done in that order!** You will then see a line between the linked fields.

Establish the following relationships:

- ContactID in Contacts to ContactID in Orders
- ZipCode in ZipCodes to ZipCode in Contacts
- ProductID in Products to ProductID in LineItems
- OrderID in Orders to OrderID in LineItems

Here's the resulting relationship model:



The database is now fully defined and populated with a little bit of data. Now we are ready to do some sample queries.

Querying the Database

To create a query, select the **Create** tab and then pick the **Query Design** tool. Close the resulting dialog without picking any tables.



To create each query, drag the needed tables into the query designer and then drag the appropriate fields into the columns list. Specify any *GROUP BY* or *WHERE* clauses as needed.

Name each query with the pattern: QLastnameX, where Lastname is your last name and X is the question number, *e.g.*, QSmith1.

- 1. Create a query that lists all of the customers' first and last names regardless of whether they placed an order or not. <u>Hint</u>: Your name must appear in the result.
- 2. Create a query that lists all of the customers' first and last names plus their address with the street, city, state, and zip code. The result should contain all customers regardless of whether they placed an order or not.

- 3. Create a query that lists the Order IDs of all of the orders placed by "Jane Wu". Use the information given (first and last name) and not ContactID. Remove any duplicates.
- 4. Create a query that lists the PriceCharged for Windows 7. List the ProductID, ProductName and PriceCharged. (Note that PriceCharged is different from RetailPrice.)
- 5. Create a query that lists what products were ordered on 4/19/2011. List the last name of the customer, the order date, product id and product name. Remove any duplicates.
- 6. Create a query that lists all the first and last names of all customers who placed at least one order. Remove any duplicates.
- 7. Create a query that lists all the products that have a retail price of more than \$100. List the product ID and the product name and the retail price.
- 8. Create a query that retrieves the name (first and last), address, city, state, and zip code of the contact with ID=4 (or whichever contact ID is <u>your own name</u>.)

Grading Rubric

This rubric is intended to guide graders in their evaluation of the students' submissions.

Criterion	Discussion	Grading
Tables not correct	All tables must have the correct names and attributes (columns or fields)	 -5% for each table that is incorrectly named (including spelling mistakes) -5% for each table that contains an attribute with the incorrect name or datatype
Queries produce correct output and are designed correctly	Each query must produce the correct result.	-5% for each query that does not produce correct result or is not designed to produce correct output for all data possibilities
Relationships not correct	All relationships must be defined correctly and connect the right attributes, i.e., primary key/foreign key pairs	-2% for each incorrect relationship
Data not correct	All data must be entered as specified	-1% for each record or field that is not correctly entered
Queries not named correctly	Queries should be named as QLastnameX, where Lastname is the student's last name and X is the query number.	-10%