**ncsu_logo**

**BUS 442: Information Systems Development**

**Project 3 for Teams of 3: Creating an Application to Track and Manage Customer Data**

**Part 1: Design and Build the Database**

*X for Y Solutions* is a small, hypothetical tech startup company who just relocated to the downtown Raleigh, NC market from Philadelphia, PA. The management of the company has established a presence in their leased headquarters at HQ Raleigh but still primarily sells its products and services to clients in the northeastern part of the United States and in Canada. Because the company is finalizing their move to the southeastern part of the U.S., their budget is limited. The company owners want to track information about their customers, but they cannot yet afford expensive CRM software. They have decided to hire a team of college interns to build an app for them to track customer information. The deliverable of this project will be an application to track and manage *X for Y Solutions* customer data. At a minimum, the app should allow the users to view and maintain individual data about each customer, as well as display the cumulative and average sales of all customers.

**Project (Part 1) Requirements**

The project has three parts: a relational database design and implementation, an interface design and implementation, and VB code design and implementation. For part 1, design and create a new relational database (using MS Access), called ***CustomerDatabase***, that stores data about customers of this business. Although a database like this in an industry environment may require many tables, you only need to create and import a few tables for this project. The main table for this project is the customer table: ***CUSTOMER***. You will also create a sales rep table, **SALESREP**, that connects to the customer table. Theirs is a 1:M relationship; that is, one sales rep sells to many customers, but a customer is serviced by one and only one sales rep. Sales reps are assigned to customers based on geographic region.

**Defining the Data and MetaData**

Launch MS Access and create a new customer database. Add two tables to this database: the SALESREP table and the CUSTOMER table. Create a 1:M relationship between the two tables. Additionally, your team may choose to add other tables and/or attributes to your Customer Database that you think would enhance the app.

**Sales Rep Table:**

The SALESREP table should have the following attributes with the metadata defined as follows. All data should be defined at the atomic level (e.g., sales rep name must be defined as sales rep last name and sales rep first name).

**SalesRepID**: Required; 4 integers, **Primary Key**

**SalesRepFirstName**: Required; 30 characters

**SalesRepLastName**: Required; 30 characters

**SalesRepRegion**: Required; 5 characters; Lookup Column constraints of North, East, South, and West

Use a LOOKUP inside Access

**Customer Table:**

The CUSTOMER table should have the following attributes with the metadata defined as follows. All data should be defined at the atomic level.

**CustomerID**: Required; 9 integers, **Primary Key**

**CustomerName**: Required; 30 characters; This is a company name (e.g., ABC, Inc.)

**SalesRepID**: Required; 4 characters; Constraints of valid entries in the SALEREP table

Populate the table first, then use LOOKUP to find the salesrep table then drag the ID column to the…

**Classification**: Required; 15 characters; Lookup Column constraints of *Preferred* or *Not Preferred*

Set up LOOKUP to only allow those 2 options above

**Active Status**: Required; 3 characters; Lookup Column constraints of Yes or No

**Payment Terms Code**: Required; 1 number; Lookup Column constraints of *1, 2, 3, 4,* or *5*. Examples: *1* (Net 30) requires that the customer pay their net balance owed within 30 days; *2* (10 Net 30) allows payment within 30 days with a 10% discount; *3* requires cash in advance; *4* requires cash on delivery; and *5* (5 Net 30) allows for payment within 30 days with a 5% discount.

**Bill To Address 1**: Required; 30 characters

**Bill To Address 2**: Not Required; 30 characters

**Bill To Country**: Required; 3 characters; Use the Lookup Wizard to add column constraints of *CAN* (Canada) and *USA* (United States of America).

**Bill To State/Province Abbreviation:** Required; 2 characters; Only allow the states in the USA and provinces in Canada used currently by *X for Y Solutions*. Each are ISO-standardized to two characters. Use the Lookup Wizard Column constraints to add the appropriate northeastern U.S. states if the Country selected was USA. These include *NY, NJ, VT, MA, RI,* and *DE*. The two-character lookup column constraints for Canadian provinces are NS (*Nova Scotia), QC (Quebec), ON (Ontario), PE (Prince Edward Island),* and *NB (New Brunswick).*

**Bill To City**: Required; 25 characters

**Bill To Postal (Zip) Code** : Required; 9 numbers; Example: 27526-0298. (Note: One option is to only store the postal code in the CUSTOMER table and link to another table in the database to retrieve city and state. This option would be consistent with a relational database normalized to third-normal-form (3NF). However, this particular choice is optional for this app.)

**Email Address**: Required; 20 characters

**Phone Number**: Required; 10 numbers (formatting symbols not required)

**Website Address:** Required; 25 characters

**Notes**: Not required; 50 characters

**YTD Loyalty Points:** Not required; 6 integers

**YTD Sales:** Required; 12 decimal; This attribute will be eventually sourced by the Order and Invoicing application. For purposes of this app, enter a hypothetical number for each customer.

**Discount %:** Not required; 4 decimal; Example: 10% would be entered as .10.

**Adding Constraints in the DBMS**

To insert a validation rule in the Access database to constrain the state to only legitimate countries, perform the following:

In the Validation Rule Property of the **Property Sheet**, type:

*[State] In ("NY","NJ", “VT”, “MA”, “RI”, “DE”) And [Country] Like "USA" Or [State] In ("NS","QC", “ON”, “PE”, “NB”) And [Country] Like "CAN" Or [Country] Is Null*

In the Validation Text property of the Property sheet, type an error message, like:

*Invalid state or province. Please enter a valid state or province.*

**You can copy and paste those validation in Access**

**Populating the Database:**

Once the database is built, populate the SALESREP table first with fictitious sales reps and their information. Assign a unique number to each sales rep as the unique primary key. Then, populate the CUSTOMER table by keying information for at least twenty fictitious customers, again with a unique customer number as the unique key. We will discuss Part 2 of the project in the next class. Please contact me for any information that needs clarification.

* Create your database model 1st (the design)
* Then set up data validation (put as much validation in your actual database in access), everything that keyed in need to be validated.
* Then key the data in