

Microsoft Access – Part I (Database Design Basics)

ShortCourse Handout

July 2004

Copyright © 2004 Heide Mansouri, Technology Support, Texas Tech University. ALL RIGHTS RESERVED. Members of Texas Tech University or Texas Tech Health Sciences Center may print and use this material for their personal use only. No part of this material may be reproduced in any form without written permission from the [author](#).

Introduction

Access - Part I (database design basics) is a non-credit, 2-hour course designed as an introductory course to Microsoft Access database design. This shortcourse is the foundation for Access Part II and Access Part III short courses series. Microsoft Access is a **Relational Database Management System**, which stores and retrieves information according to relationships you define. It can help you to organize your data to make it easier to enter, edit, and retrieve the information. With Access 2002, you can create **tables** to contain your information, **queries** to retrieve information from the tables, and **forms** and **reports** to make the information available to users in various ways. The databases files you create in Access have **.mdb** extensions.

Course Objectives

After completing this shortcourse, you should be able to:

- Explain the purpose of creating an Access database;
- Sketch the design of a database;
- Create a new database using database wizard;
- Work on the two views of a table;
- Explain what a database relationship is;
- Format a datasheet; and
- Explain the differences between Microsoft Excel databases, and Microsoft Access databases.

Access Database

Access Database consists of a **collection of Tables, Queries, Forms, Reports, and other objects** that you build to manage and analyze and present your data. These components are also called **Objects**. You create as many objects as you

need, and Access stores them in the database. These objects are dependent to other objects in the database. That is, a Query, Form, or Report draws information from tables so that changing data in any of these objects changes the data in all these objects.

Definitions:

- **Database** is a structured collection of information pertaining to a specific topic.
- **Relational Database** is a collection of information that's related.
- **Record** is a collection of data about an individual item (person, object, or thing).
- **Field** is a single item of data common to all records.
- **File** is a set of related records.

There are two types of databases:

- **Flat File Databases** are used for simple lists and may have **duplication** of data such as information on a spreadsheet.
- **Relational Databases** are those in which data is held in a number of cross-referenced files in order to reduce duplication. They make it easier to find, analyze, maintain, and protect your data because it is all held in one place.

Electronic database allow the user to:

- Store and manage a large amount of information related to a particular topic.
- Easily input and edit data.
- Automatically update and recalculate data.
- Easily search and select data.
- Format, arrange, and present information in the way that you want it.
- Share the information with other applications programs.

Questions to ask yourself, when you are designing a database:

- What do I want? (What do I want to keep track of?)
- What have I got? (What is the scope of this database?)
- What do I need to get there? (What is the design of my tables, forms, reports, etc.?)

Failure to Plan Is Planning to Fail

Because the objects in a database are related to one another, building a database requires careful planning. A designer's best tools are a pencil and paper. It is important to plan what you are doing. The sooner you touch the computer the sooner you'll make a mistake. If you don't plan, you will often have to start again. Also, document what you are doing.

Before turning on your computer:

- Define the purpose of your project.
- Consider questions you or others who will be using this database and may want to ask in the future.
- Determine the tables (one for each subject)
- Be sure to include at least one common field in related tables.
- Test the design of your database by entering sample data and run a simple query.

Basic Table design rules:

- Make each record unique (do not repeat the same kind of information in the same table).
- Make each field unique.
- Make each field functionally dependent (all the information in a record should be related). If the field is not related to the subject, it should be in another table.
- Don't include calculated fields. For example include gross pay and deductions on a person's salary record, but net pay can be calculated.
- Make sure that data is broken down into its smallest logical parts (Name broken into First name and Last name for example)

Understanding Tables

Tables are building blocks of an Access database. Tables hold all the information in your database. Each **column** in a table represents a **field**. Each **row** represents a **record**. Each **field** must be the **smallest piece of information** it can be (it cannot be broken down further into more fields). Each table's **information** should be **related** in some way. Tables can be **linked** using the **relationships**.

Data type

Data type is the type of value that will be entered into the fields.

Data Types are:

- Text (by default) – **up to 255** characters
- Memo – **up to 65,535** characters
- Number
- AutoNumber
- Date/time
- Yes/No

Relationships

Relationships are ties that are created between tables in the database so that Access can combine data from more than one table.

Types of Relationships

- In **One-to-One relationship**, each record in Table A has only one matching record in Table B.
- In **One-to-Many relationship**, each record in Table A has many matching records in Table B. This is the most common type of relationships.
- In **Many-to-Many relationship**, a record in Table A has many matches in Table B, and a record in Table B has many matches in Table A. This type of relationship is possible by defining a third table (called a **junction table**) whose primary key consists of two fields (the primary keys of both Tables A and B).

Terminologies

- **Relationships** are ties that are created between tables in the database so that Access can combine data from more than one table.
- **Parent table** is the primary table in a relationship.
- **Child table** is a table to which another table is related.
- **Primary key** is the field that uniquely identifies a record, such as Social Security Number. Each table should have a primary key. The primary key will determine how to join your tables.
- **Foreign key** is the field in a child table that is related to the primary key of the parent table.

Types of Primary Keys

- **AutoNumber Primary Keys** can be set to enter a sequential number automatically as each record is added. This is the simplest way to create the primary key.
- **Single-field Primary Keys**
- **Unique values** such as ID Numbers or Part Numbers are set as the primary key.
- **Multiple-field Primary Keys** - In a many-to-many relationship you need to create a junction table. Primary keys from one-to-many tables together will be the primary key for this table.

Object Naming Rules

Access will complain if you don't follow these **Rules**:

- You cannot give a table, a query, a form, or a report the same name.
- The name can be up to 64 characters including spaces. However, the name cannot start with a space.
- Avoid using spaces in **Object names**. Use CustomerID instead of Customer ID
- Avoid using the names that are the same as built-in Access function names or property names. Strange things may happen if you use those names.
- You cannot use the following characters in an Object name
- (.) period
- (!) exclamation point
- (`) accent grave
- [] brackets
- " " double quotation marks

For more information about naming rules, look up the "**Naming Rules**" on the Help menu.

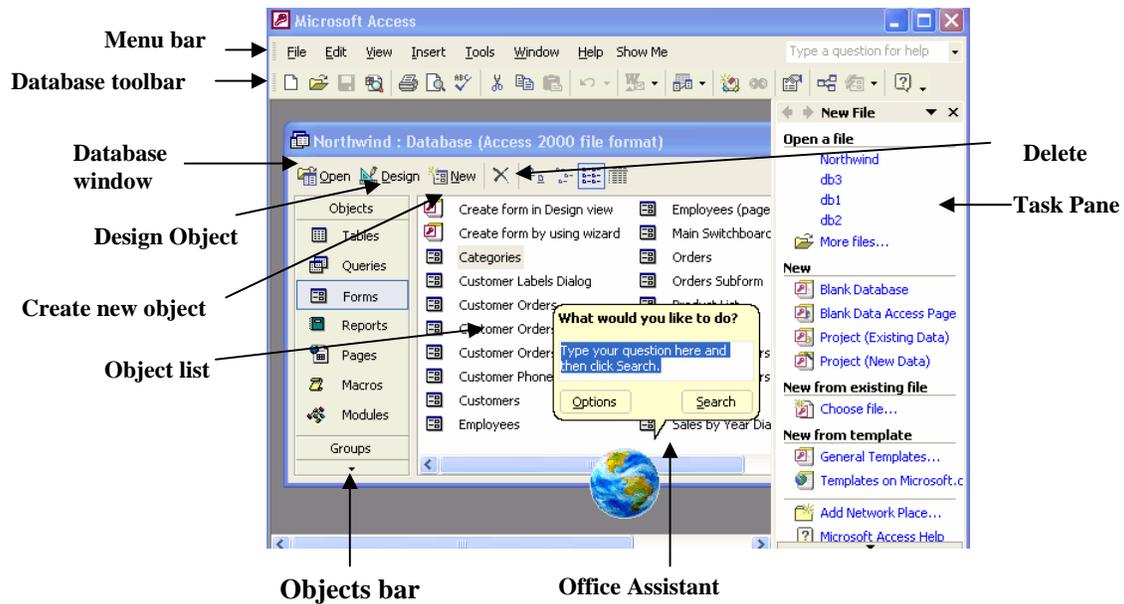
Access Database Objects

Access Database consists of 7 objects:

- **Tables** hold information – Tables are building blocks of database.
- **Queries** let you ask questions about your data in your database or make changes.
- **Forms** are for viewing and editing information.

- **Reports** are for summarizing and printing data.
- **Pages** are HTML files that let you view Access data with Internet Explorer.
- **Macros** perform one or more database functions automatically.
- **Modules** are another type of Access object that you may or may not work with. A module is a program you write in VB (Visual Basic) to automate and customize database functions.

These Object buttons in the Database window provide direct access to every object in your database.



To view relationships:

While on the database window, from the standard toolbar,

- Click on the Relationship button, Or
- Choose Tools > Relationships

Creating relationships allows you to pull together related data from different tables.

They are also useful when you are creating forms, queries, and reports.

To create a relationship, you should have a field that is common in both tables.

Referential integrity is a system of rules that Microsoft Access uses to make sure that relationship between records in related tables are valid and that you don't accidentally delete or change related data.

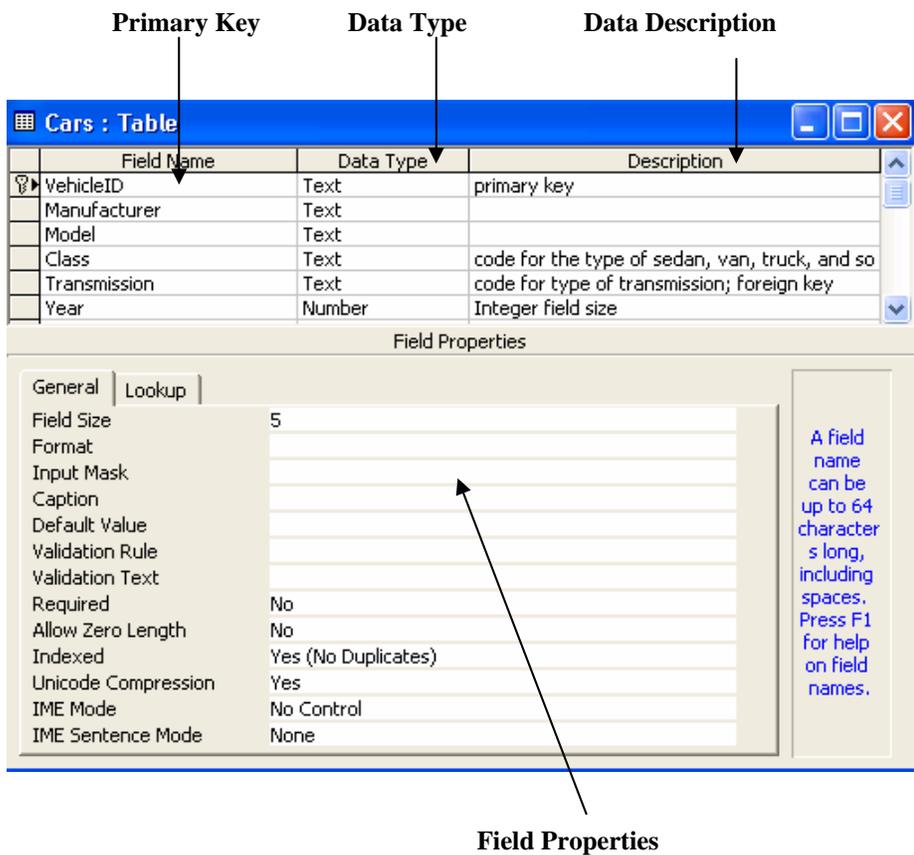
To create a relationship:

- On the database window click Tools.

- Click Relationships.
- Add Tables
- Check-mark the Enforce Referential Integrity.
- Click Create.

To delete a field:

- Open the table in the Design view.
- Select the row.
- Press Delete row button.



Field Properties

All Microsoft Access objects have properties that you can set to determine how the object appears or works.

- **Field Name Property** - Use it to name a field
- **Data Type property** – Use it to select the type of data that will be kept in the field.
- **Description property**- Use it to enter a description of the field.

In Design View (View > Design)

- Click in a field name to add fields to a table.
- Select a field to delete it from the table.
- Insert a row to add a field.
- Add field properties.

In Datasheet view (View > Datasheet)

- Use the **Down-arrow** mouse Pointer to select a column/row.
- Click within any record to deselect the selected column.
- Use the **Horizontal Double-arrow** mouse pointer to resize a column/row.

To format a datasheet:

- From the Format menu
 - Choose Datasheet
 - Cell Effect
 - Border
 - Gridline color
 - Background color

Sorting Data

You sort data to organize it according to the way you want to view the data.

Note: You need to have the smallest piece of information possible in each field. For example you can't have a customer's first and last name in one field, if you will be sorting by last name.

New Database

You can create a new database using General Templates or using Database Wizard.

A **database Wizard** creates a database for you automatically.

- From File menu choose New.
- Select the Access database wizards.
- Select the Database tab.
- Click Contact Management.
- Click OK.

Note: A primary goal of good database design is to make sure your data can be easily maintained over time.

Use MS Excel if:

- You want to crunch numbers and perform automatic calculations.
- You want to track a simple list of data.
- You want to create charts and graphs of your data easily.

Use MS Access if:

- You have a large amount of information that is related to a particular subject.
- You want to maintain records for ongoing use.
- The information is subject to many changes (address, pricing, etc.).
- You want to generate reports based on the information.

Useful Web Site addresses:

- Database Normalization at <http://www.webopedia.com/TERM/N/normalization.html>
- Systems and Database Design at <http://cbit.ttu.edu>

Where to Get Help

If you need help with your project, you may contact Help Central Office at 742-HELP.

If you need help from your instructor, you may e-mail heide.mansouri@ttu.edu.

Exercise

To Create the Sample Table:

- Double-click Create a Table in Design View.
- Create the following fields:

ID, Employee Name, Address, Phone, Training Date, Class Taken, Credit Hours, Passed?

Using the Data Type, change the following fields data type:

- | <u>Field</u> | <u>Change the Data Type to:</u> |
|-----------------|--|
| • ID | Number and make it a <u>Primary Key</u> |
| • Training Date | Date/Time |
| • Credit Hours | Number |
| • Passed? | Yes/No |

Using Input Mask, create an Input Mask for the Phone field:

- Click on Phone field.
- Click on Input Mask in Field properties.
- Click the  button to start the **Input Mask Wizard**.
- Choose a mask.
- Follow the Steps. Make sure to select the **Without the symbols in the mask** button to store the data.
- Click Finish.

Using Validation Rules, create a Validation Rule for the Phone field:

- Click on Phone field.
- Click in Validation Rule box.
- Type **Like "806*"** or **Like "425*"** and press Enter.
- In the validation text box, type the Area Code; must be 806 or 425.
- Click in the caption box and type Phone number.
- Save and return to datasheet view to enter the data from the **Sample Table**.
- Format your table.
- Save.

Sample Table

ID	Employee Name	Address	Phone	Training Date	Class Taken	Credit Hours	Passed?
1	Phil Sharp	34 Ave A	(806)555-1234	5/5/2001	Learning Skill	3	Yes
2	Becky Smith	12 Ave H	(806)555-2786	5/5/2001	Customer Services	2	Yes
3	Nick Cook	59 Ave Q	(425)555-8976	6/15/2001	Public Speaking	9	No
4	Adam Good	89 Ave U	(806)555-0102	6/15/2001	Learning Skill	3	Yes

Please e-mail your comments or suggestions to: heide.mansouri@ttu.edu