



HILLSBOROUGH
Community College

Microsoft Access 2003
Module 1

<http://pds.hccfl.edu/pds>

Microsoft Access 2003: Module 1

June 2005

The material contained in this training material is copyrighted ©2006 Hillsborough Community College Department of Professional Development Services and may not be reproduced without express, written permission. Other trademarks, trade names, logos, designs, brand names, and product services mentioned in this publication may be trademarks or registered trademarks of third parties.

Table of Contents

Database Structure	5
Database Terminology.	6
Naming Conventions.	6
Relationships	5
One-to-many	12
Many-to-many	13
One-to-one.	13
Create a New Database.	14
Create a Table in Design View	14
Save a Table	14
Use Input Mask Wizard	14
Insert Data Type: Yes/No	14
Insert Data Type: Lookup Wizard	14
Insert Data Type: Yes/No	14
Insert and Delete Field	14
Primary Key.	14

Objectives

At the end of this training session you should be able to:

1. Understand the three types of relationships: one-to-many, many-to-many and one-to-one;
2. Create a new database;
3. Create tables for data entry in design view;
4. Use, understand, and set data types: Text, Memo, Number, Currency, Date/Time, AutoNumber, Yes/No, Lookup wizard.
5. Create and delete a primary key;
6. Insert and delete a row.

Before a database is developed, plan the structure from the beginning to the end. Flow-charting or blueprinting in advance will eliminate restructuring and/or starting from scratch.

Questions to Ask Before Designing a Database:

1. What does the database need to do?
2. What functions need to be achieved?
3. Which objects (such as tables, forms) depend on each other?
4. What items are needed for the database?
5. Who will use the database?
6. How will the output data (report) be generated?
7. How will the database be organized?

Access Database Terminology

It is necessary to understand and be familiar with the basics of a database and its objects to create or generate information.

Table- The central framework of a database that stores data in fields (columns) and records (rows).

Query- Allows for table inquiries. A query can change, delete, add, arrange data in tables. Also aids gathering information for forms and reports.

Form- Displays and enters data in a fitted format. Forms can also contain other nested forms (subforms).

Report- Allows for the printing and print preview of information such as labels, lists, form letters, invoices, summaries, display charts, etc. The user can personalize reports by adding a logo or picture, organizing headers, details, footers, and sorting columns.

Page- Also known as Data Access Page. Allows the publication of a web page and web access to a database. A page can be viewed and edited, and the information can be altered.

Macro- Allows for automating simple and common tasks such as opening and closing a form, exporting data, printing data in a report, and saving data.

Module- Modules are a collection of Visual Basic declarations and procedures that allow for the automation and customization of Access, giving the user more explicit control over actions.

Naming Conventions

The Leszynski Naming Convention (LNC), originally created by Stan Leszynski and Greg Reddick, developed guidelines called tags to assist the user to open, edit, and troubleshoot without deciphering what is contained within the object itself.

File names can be more than one word; however, do not use underscores or spaces. Instead, capitalize the first letter of each word.

e.g. **tblEmpInfo**

Recommended Naming Conventions

Table	tbl
Query	qry
Form	frm
Report	rpt
Macro	mcr
Module	mdl
Database	dbf

When organizing and creating a database, the question, "How will the objects (tables) relate?" must be asked.

There are three types of relationships in Microsoft Access:

- One-to-many
- Many-to-many
- One-to-one

One-to-many relationship

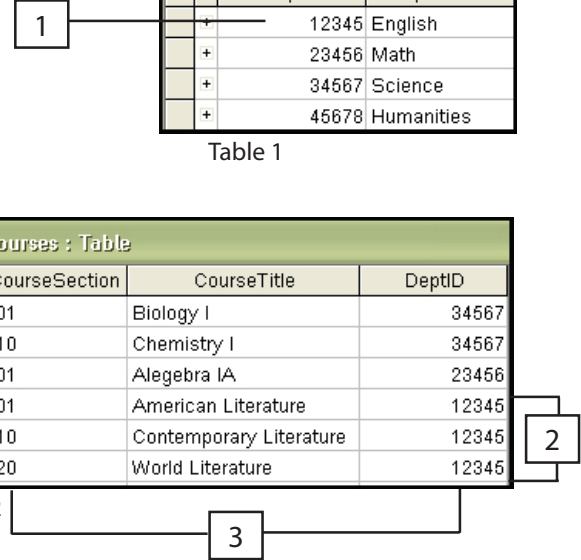
The most common type of relationship. A record from one table (Table 1) can have matching records in another table (Table 2); however, Table 2 has only one matching record in Table 1.

tblDepartment : Table		
	DeptID	Department
	12345	English
+	23456	Math
+	34567	Science
+	45678	Humanities

Table 1

tblCourses : Table			
	CourseSection	CourseTitle	DeptID
+	201	Biology I	34567
+	210	Chemistry I	34567
+	501	Algebra IA	23456
+	901	American Literature	12345
+	910	Contemporary Literature	12345
+	920	World Literature	12345

Table 2



Let's look at the example.

- 1 One Department
- 2 has more than one Course,
- 3 yet each Course has one Department.

Many-to-many relationship

Not recommended due to problems enforcing referential integrity. A record from one table (Table 1) can have many matching records in another table (Table 2) Also, a record in Table 2 can have many matching records in Table 1. This is possible through the use of a third table called a junction table. The junction table has a primary key that consists of two fields from Tables 1 and 2.

Let's look at the example.

- 1 Primary key from Inventory Details table (tblInventDetails)
- 2 Primary key from the equipment table (tblEquipment)
- 3 One department can have several types of equipment,
- 4 And each type of equipment can appear in several departments.

tblCheckOuts : Table		
	CheckOutID	Department
+	1000	English
+	2000	Math
+	3000	Science
+	4000	Humanities

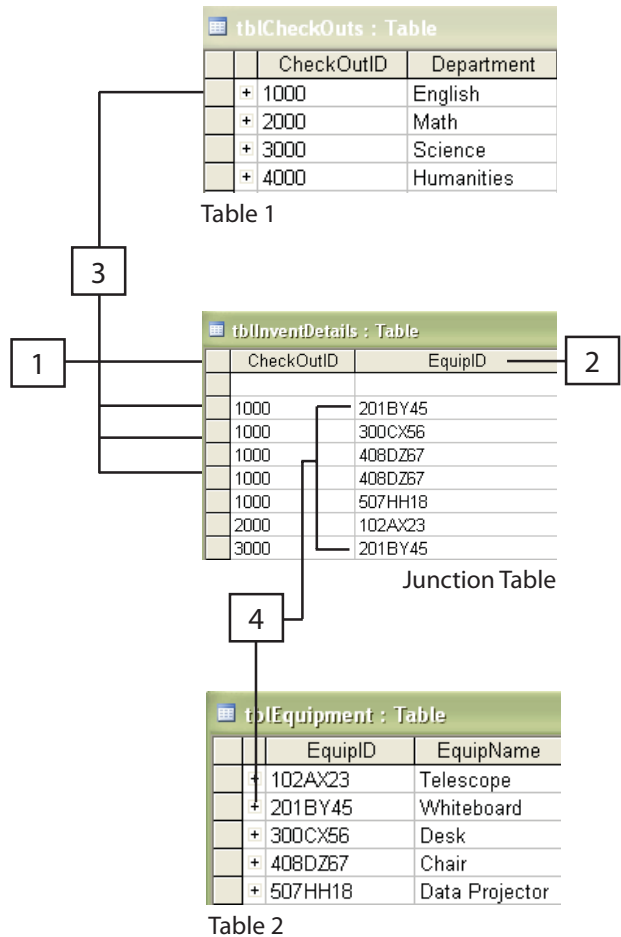
Table 1

tblInventDetails : Table		
	CheckOutID	EquipID
	1000	201BY45
	1000	300CX56
	1000	408DZ67
	1000	408DZ67
	1000	507HH18
	2000	102AX23
	3000	201BY45

Junction Table

tblEquipment : Table		
	EquipID	EquipName
+	102AX23	Telescope
+	201BY45	Whiteboard
+	300CX56	Desk
+	408DZ67	Chair
+	507HH18	Data Projector

Table 2

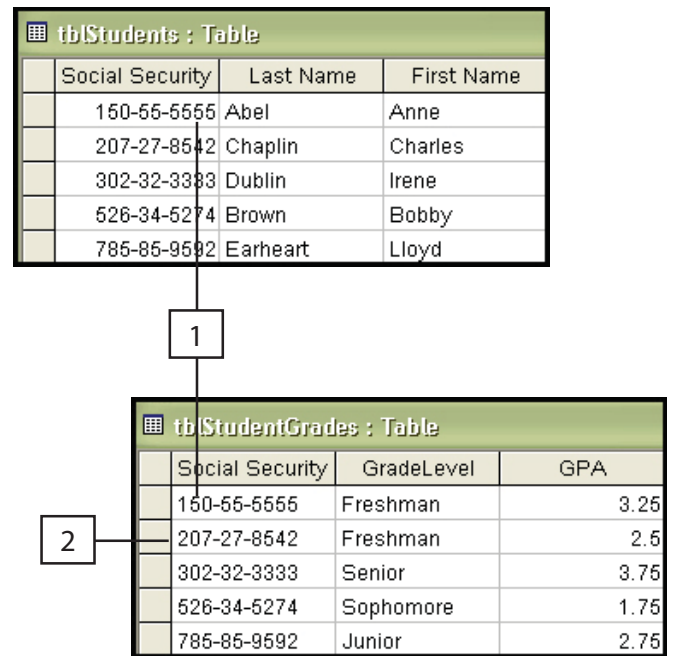


One-to-one relationship

Least common type of relationship. However, a one-to-one relationship is useful if there are records that must be kept confidential and secure. A record from one table (Table 1) can have one single matching record in another table (Table 2), and Table 2 can have only one single matching record in Table 1.

Let's look at the example.

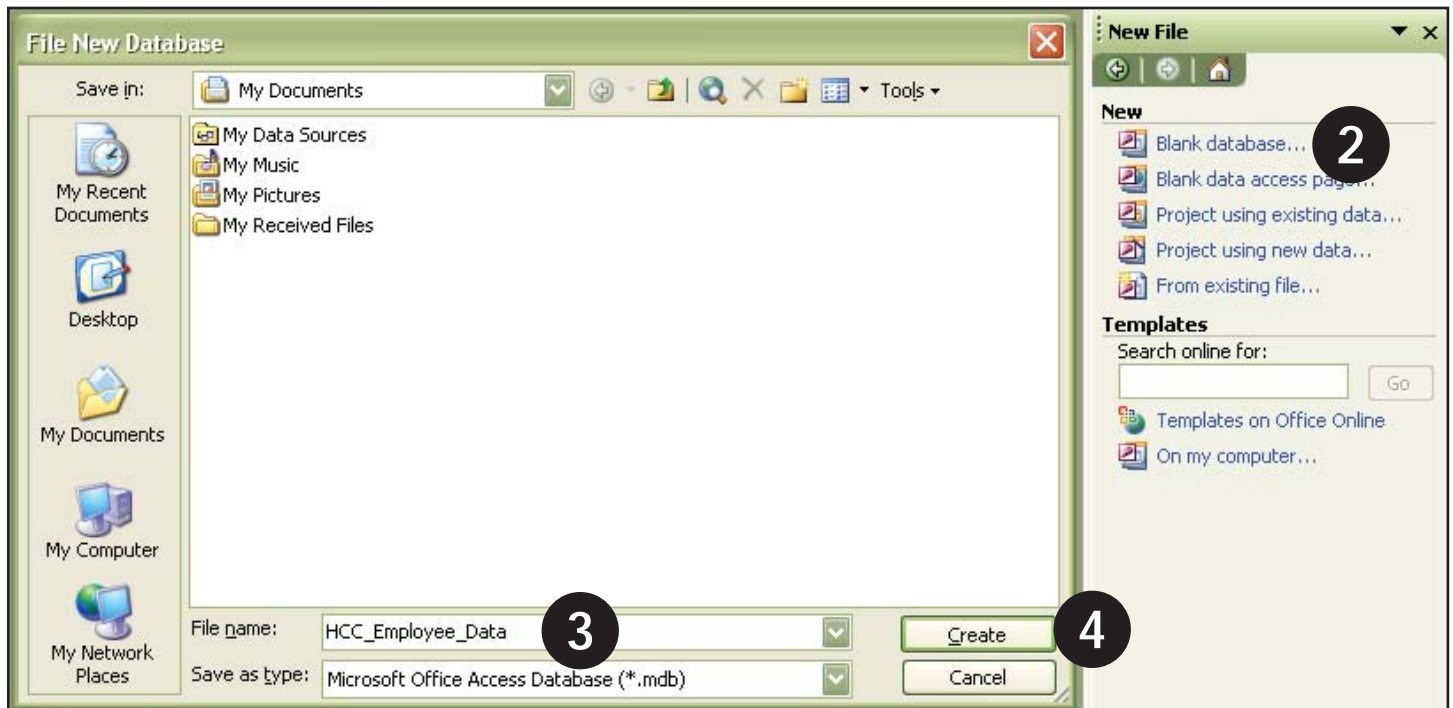
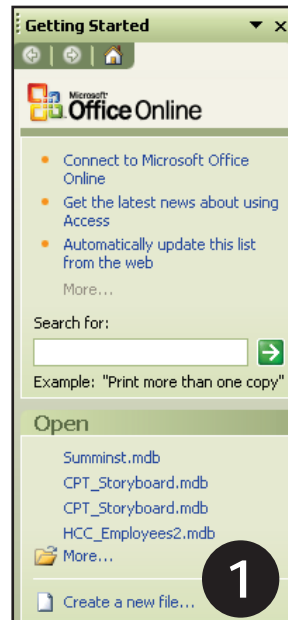
- 1 All students' have one matching record in students table (tblStudents)
- 2 The values are a subset of the social security field and the students' table (tblStudents).



Create a New Database

Before creating tables and forms, creating a general database is recommended.


1. Under the Getting Started task pane, click on **Create a new file** link.
2. Click on **Blank Database** from the menu bar.
3. After the **Fill New Database** appears, type the file name of the database.
e.g. **HCC_Employee_Data**
4. Click **Create** or press **Enter** on your keyboard.



Create a Table Using Design View

Creating a table in Design View gives the user freedom to construct a table with specifications for data collection.

1. Under **Objects**, click on **Tables**.
2. Click on **Create table in Design view**.
3. Type in the **Field Name**.
e.g. **LastName**

 When entering field names, do not use spaces or punctuation.


4. Under **Data Type**, click on the down arrow and select **Text**.

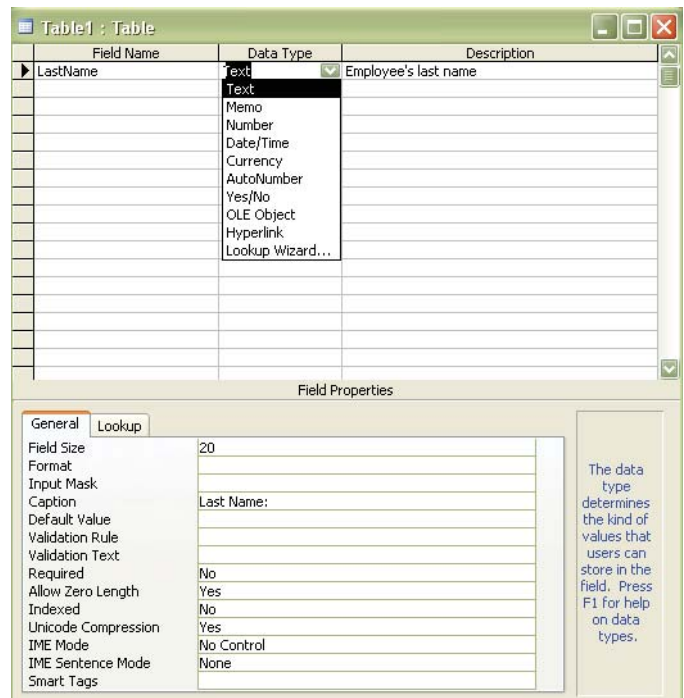
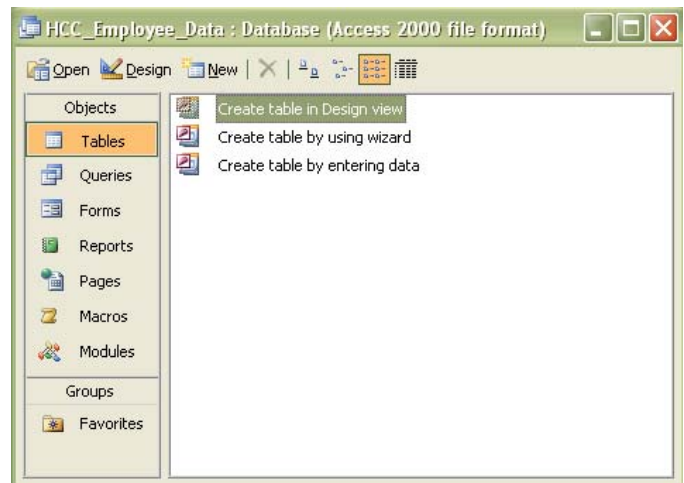
Look at page 9 for **Data Type** descriptions or hit **F1** on your keyboard for more details and help on data types.

5. Locate the **General** tab and type 20 for the Field Size.

Typing in 20 for the field size indicates that the last name can be no more than 20 characters long.

6. Type a label for the **Caption**.
e.g. **Last Name**:

 Spaces and punctuation can be used for captions. Whatever is typed in the caption will appear in both forms and reports. If no caption is entered, the field name will appear instead.



Activity

Add the following fields to the table, FirstName, MI (middle initial), Street, City, State, Zip, Exempt (Yes/No) and Campus (Lookup Wizard). Also add the Field Size and Caption for each field name.

Save a Table

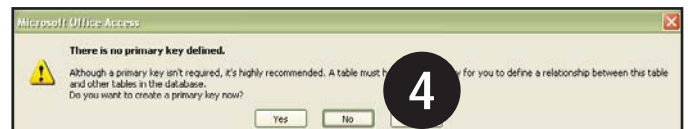
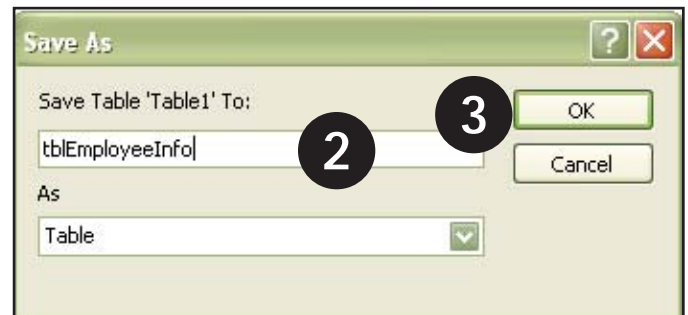
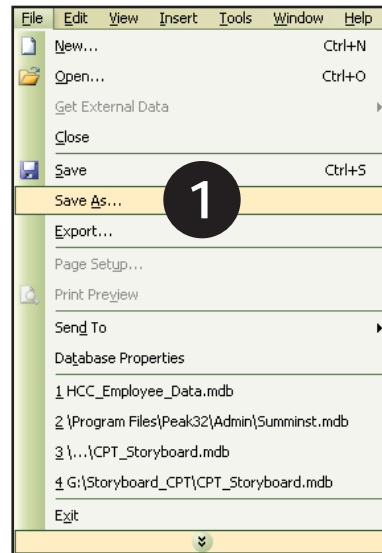
Saving a table with a specific title will help when organizing information, creating queries, forms, and pages, and connecting to other tables.

In this example, we are saving the table, tblEmployeeInfo.

1. Click on **File>Save As**. A **Save As** dialog box appears.
2. Type the name of the table.
e.g. **tblEmployeeInfo**
3. Click on **OK**.
4. A dialog box appears stating that no primary key is defined. In this example, select **No**.



Primary keys are not required; however, select a primary key when you need to create relationships between tables in a database. Primary keys cannot be duplicated; therefore, never use a last name, address, or a field where information is likely to be repeated. Ideal primary keys are employee numbers, social security numbers, product numbers, etc.



Data Types

It is important to assign data types to a field since they specifies what type of data can be entered into a field.


Data Type	Description
Text	Add text entries and number entries that do not require calculations such as addresses, social security numbers, dates, and phone numbers. Maximum number of characters, including spaces, is 255 characters.
Memo	Add long text or number entries. Maximum number of characters is 64,000. Use when more than 255 characters are necessary.
Number	Add positive/negative numbers for calculations. Not appropriate for currency or items that require exact calculations. 15 digit maximum.
Currency	Use for calculations that do not need to be rounded. 15 digit maximum.
Date/Time	Add date and time. 8 character default.
AutoNumber	Use when sequential, random, or replication ID numbers are needed. 9 digit maximum.
Yes/No	Use when Yes/No, True/False, or On/Off are necessary.
OLE Object	Add objects such as Excel workbooks or Word documents that are linked or embedded. 1 gigabyte character limit.
Hyperlink	Link to files, objects, or web links. 2,048 character maximum.
Lookup Wizard	Use when choosing data such as tables or a list of values.

Insert Input Mask Wizard

The Input Mask Wizard allows the designer to define the format for entering data.

In this example, a format is set for entering a zip code.

To use the input mask wizard the table must be in **Design View**.

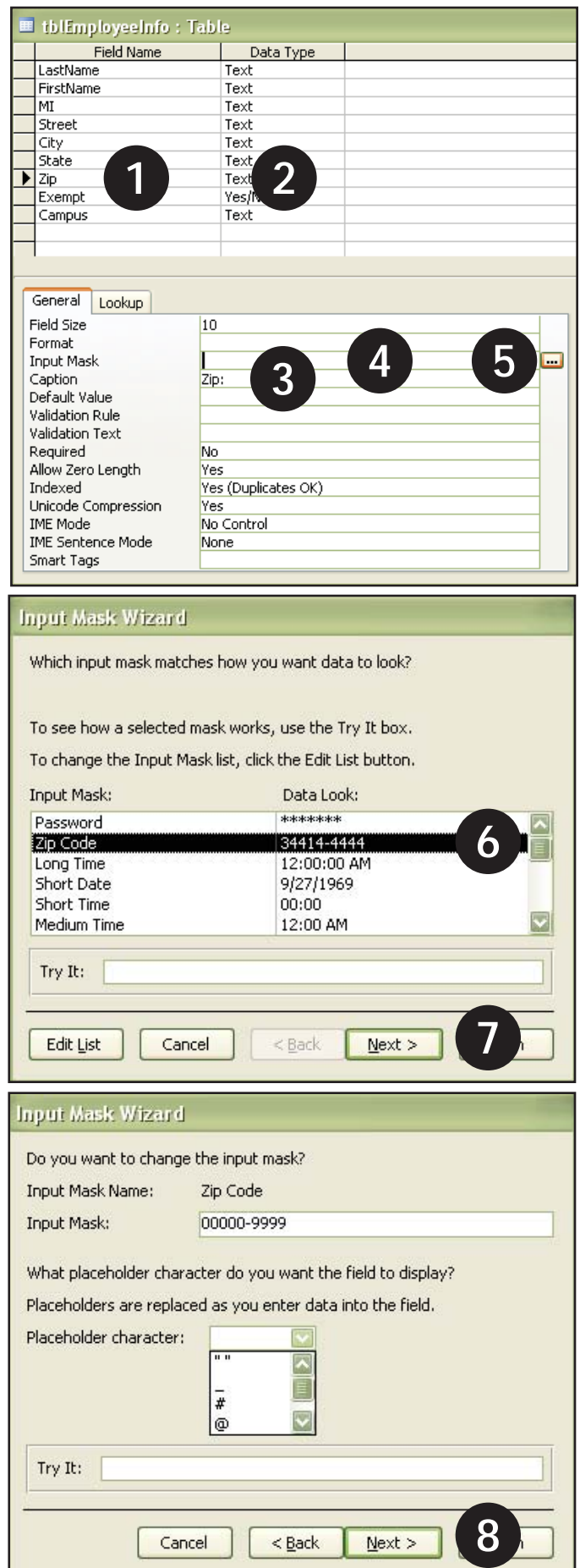
1. Type **Zip** under the **Field Name**.
2. Select **Text** for **Data Type**.
3. Type **Zip:** for the **Caption**.
4. Click in the **Input Mask** field.
5. Click on the  **build** button.

An Input Mask Wizard box appears. In this example, the Zip Code format is selected.

6. Select **Zip Code**.
7. Select a placeholder and click on **Next**.

The underscore **_** is the default placeholder character.

8. Click on **Next**.



The screenshot shows the Microsoft Access interface for configuring an input mask for a field in a table named **tblEmployeeInfo**. The table structure is shown at the top, with fields: LastName, FirstName, MI, Street, City, State, Zip, Exempt, and Campus. The **Zip** field is selected, and its properties are shown in the **General** tab. The **Field Name** is **Zip**, the **Data Type** is **Text**, and the **Caption** is **Zip:**. The **Input Mask** field is empty, and the **build** button (three dots) is highlighted.

The **Input Mask Wizard** dialog box is shown in two stages. In the first stage, the wizard asks "Which input mask matches how you want data to look?". The **Zip Code** option is selected, and the **Next >** button is highlighted.

In the second stage, the wizard asks "Do you want to change the input mask?". The **Input Mask Name** is **Zip Code** and the **Input Mask** is **00000-9999**. The wizard also asks "What placeholder character do you want the field to display?". The **Placeholder character** is **_** (underscore), and the **Next >** button is highlighted.

Create a Custom Input Mask


Creating a customized Input Mask ensures formatting for entering data. If the information is not entered appropriately, the data will not be accepted.

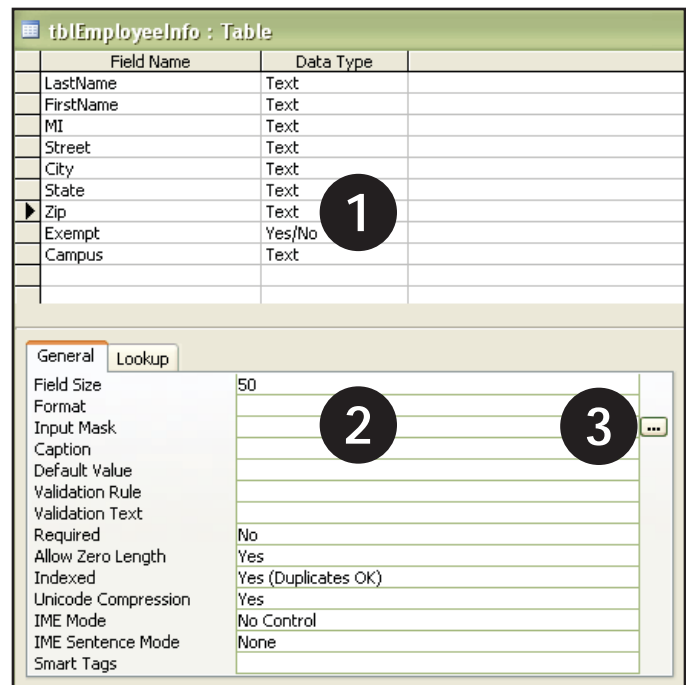
In this example, a zip code input mask is created.

To create a custom input mask, the table must be in design view.

1. Select the **Field Name** to create the **Input Mask**. In this example, use **Zip**.
2. Under **Field Properties**, click in the **Input Mask** field.

A **build button** appears to the right of the **Input Mask** field.

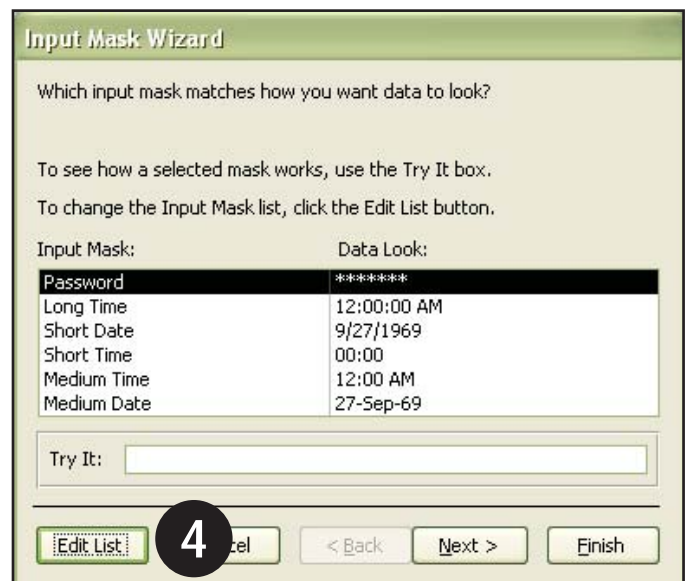
3. Click on the  build button.
4. After the **Input Mask Wizard** dialog box appears, click on **Edit List**.
5. Enter the **Description**, **Input Mask**, **Placeholder**, **Sample Data**, and select **Text/Unbound** for **Mask Type**.
6. Afterwards, click on the **Close** button to save the Zip Code input mask.



Field Name	Data Type
LastName	Text
FirstName	Text
MI	Text
Street	Text
City	Text
State	Text
Zip	Text
Exempt	Yes/No
Campus	Text

Field Properties for Zip:

Field Size	50
Format	
Input Mask	
Build	...
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (Duplicates OK)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	



Input Mask Wizard

Which input mask matches how you want data to look?

To see how a selected mask works, use the Try It box.
To change the Input Mask list, click the Edit List button.

Input Mask:	Data Look:
Password	*****
Long Time	12:00:00 AM
Short Date	9/27/1969
Short Time	00:00
Medium Time	12:00 AM
Medium Date	27-Sep-69

Try It:

Edit List (4) Cancel < Back Next > Finish



Customize Input Mask Wizard

Do you want to edit or add input masks for the Input Mask Wizard to display?

Description: Zip Code Help

Input Mask: 00000-9999 (5)

Placeholder: | (6) Close

Sample Data: 34414-4444

Mask Type: Text/Unbound



Record: 1 of 1

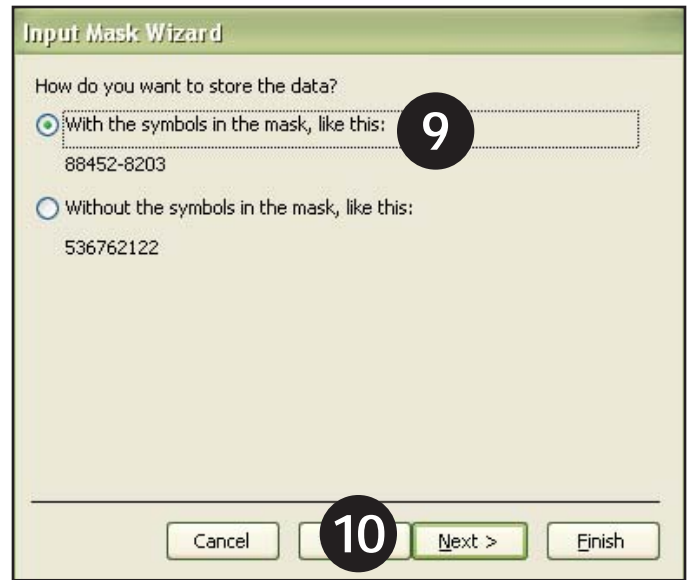
- Click on the **With the symbols in mask** radio button.

Here the data can be stored with symbols or without symbols in the table. Storing with symbols allows for easy viewing of the data.

- Click on **Next**.
- Click on **Finish**.

Notice that the Zip Code Input Mask is added under the **Field Properties>Input Mask>0000-9999;;**.

 To see the formatting in the table, click on the  Datasheet View button.



Input Mask Wizard

How do you want to store the data?

With the symbols in the mask, like this: **9**
88452-8203

Without the symbols in the mask, like this:
536762122

Cancel **10** Next > Finish

tblEmployeeInfo : Table

Field Name	Data Type
LastName	Text
FirstName	Text
MI	Text
Street	Text
City	Text
State	Text
Zip	Text
Exempt	Yes/No
Campus	Text

General	Lookup
Field Size	10
Format	
Input Mask	0000-9999;;
Caption	Zip:
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (Duplicates OK)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

Data Type: Yes/No

As mentioned on page 9, data types are used to specify what type of data can be entered and stored into a field.

In this example, the Yes/No data type is used for Exempt field name. Using the Yes/No format is helpful for simple, clear-cut answers.

To use the data type **Yes/No** the table must be in **design view**.

1. Type in **Field Name**.
e.g. **Exempt**
2. Under **Data Type**, click on the down arrow and select **Yes/No**.

Yes is represented by a check and **No** is not checked in the **Datasheet View**.



Yes/No also represents the following: True/False, 1/0, and On/Off.

tblEmployeeInfo : Table	
Field Name	Data Type
LastName	Text
FirstName	Text
MI	Text
Street	Text
City	Text
State	Text
Zip	Text
▶ Exempt	Text
Campus	Text
	Memo
	Number
	Date/Time
	Currency
	AutoNumber
	Yes/No
	OLE Object
	Hyperlink
	Lookup Wizard...

Data Type: Lookup Wizard

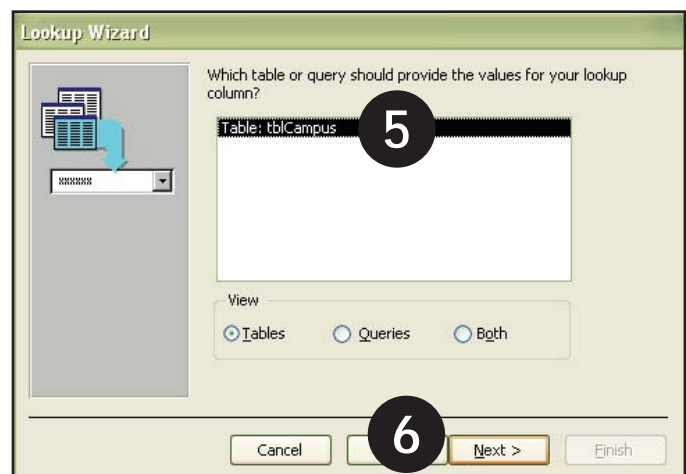
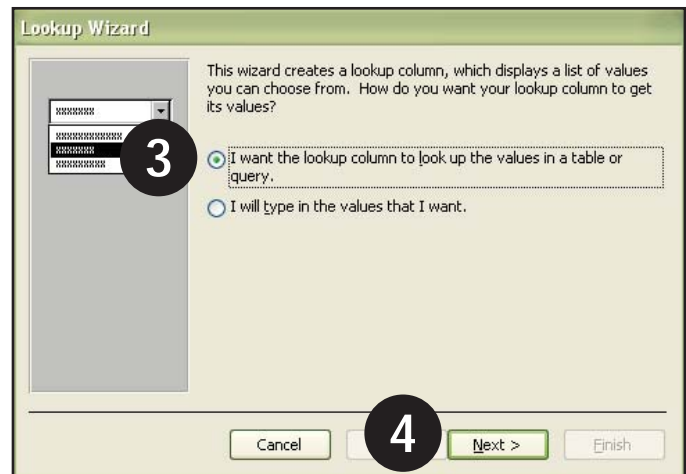
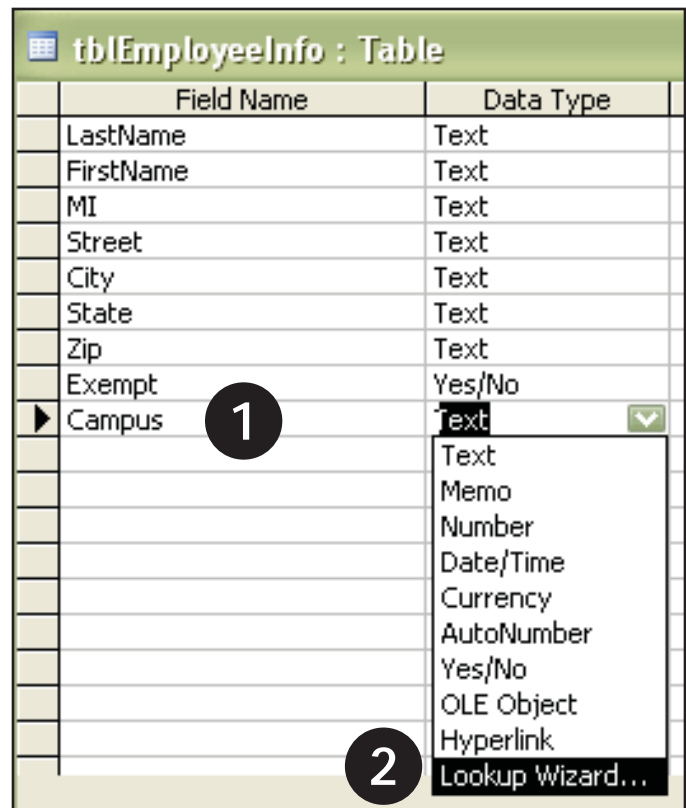
The purpose of the Lookup Wizard is to connect one field from one table to another table or a list of values. Using the Lookup Wizard is useful when choices need to be limited or the same.

For instance, Hillsborough Community College has the campus location of District Administrative Offices. If members of the HCC community were to type District Administrative Offices in a table, they could type GK, DAO, District Administrative Offices, or some other spelling or misspelling. Therefore, restricting the location to DAO ensures the same information is entered.

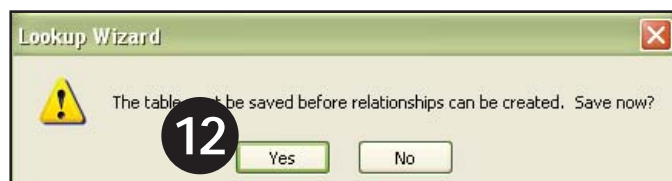
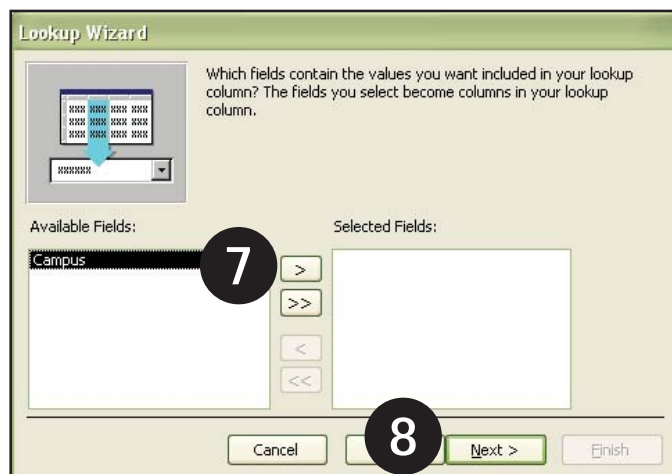
In this example, a drop-down menu is created by using the **Lookup Wizard**, which will connect to tblCampus.



To use the Lookup Wizard the table must be in **design view**.


1. Type in the **Field Name**.
e.g. **Campus**
2. Under **Data Type**, click on the down arrow and select **Lookup Wizard**.
3. A **Lookup Wizard** dialog box appears. Click on the **I want the lookup column** radio button.
4. Click on **Next**.
5. Select the table for the lookup.
e.g. **Table:tblCampus**
6. Click on **Next**.



7. Select the available field to include by clicking on the single-headed **next** arrow.
e.g. **Campus**
8. Click on **Next**.
9. Set sort order as needed and click on **Next**.
10. Adjust the width of column and click on **Next**.
11. After entering the label name for the lookup column, click on **Finish**.
e.g. **Campus**
12. Click **Yes** to save table and create relationships.



 To view the down-drop menu for the different campuses, click on the  **Datasheet View** button.

 To manually add a drop down menu, click on the **Lookup** tab under **Field Properties**. Next set the **Display Control** to **Combo Box** and the **Row Source** to the required table or query. In this example, use **tblCampus**.

Activity

Use the **Lookup Wizard** to connect the field name **State** to the table **tblState**.

Or challenge yourself and manually add a lookup.

Insert Field to a Table in Datasheet View

After a table is created, additional Field Names may be needed.

In this example, we are adding an EmployeeID (employee number) field.

To insert the field the table must be in **design view**.

1. Click on the row below where the field is to appear.

In this example, click on the field name **LastName**.

2. Click on **Insert>Insert Row** or the  **Insert Row** on the toolbar.

3. Type in **Field Name** and **Data Type**

e.g. **Field Name: EmployeeID**

Data Type: Text

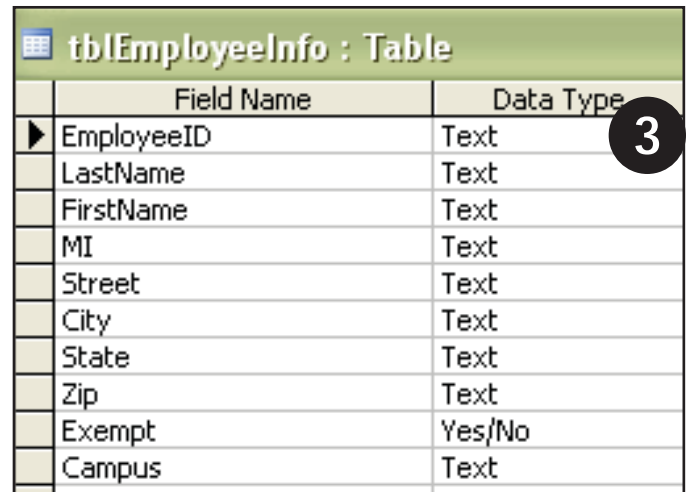
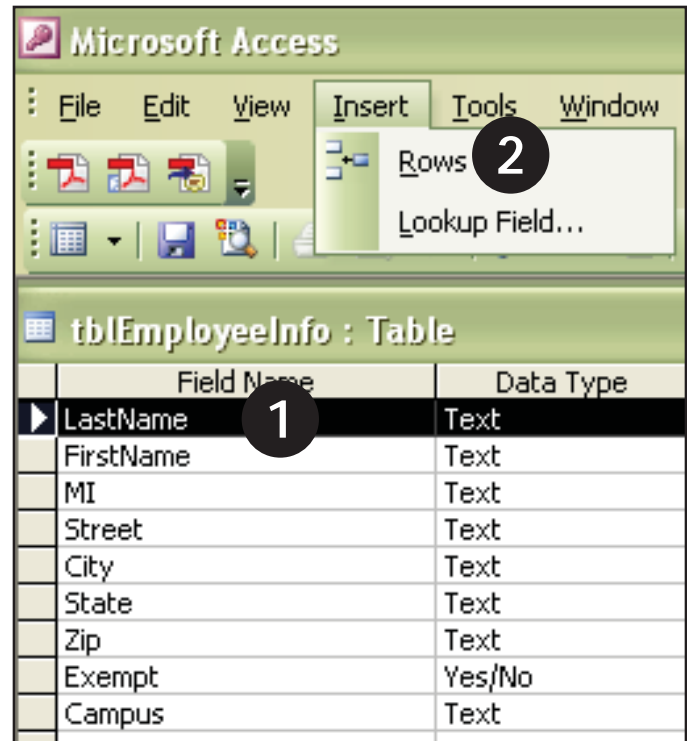
Remember to enter the **Field Properties** details under the **General Tab**. **Field Size: 7** and **Caption: Employee ID:**.



The data type is set to Text because this data doesn't require any calculation.

To delete a Field Name, select the Field Name row selector (small boxes located to the left of Field Names) and press the

Delete key or the  **Delete Row** on the toolbar or **Edit>Delete** or **Delete**.



Activity

Insert a **Field Name** named **AutoNum** with a **Data Type** of **AutoNumber**. Enter the **Caption** as **Entry Number:**.

Set a Primary Key

A primary key or primary keys uniquely identify each record in a table. An advantage of a primary key is that it does not allow duplication of information. Access does not require that a primary key, and multiple primary keys can be set.

In this example, a primary key is set for the Field Name: EmployeeID.

To set a primary key, the table must be in **design view**.

1. Click on the row selector for the field to be set as a primary key.
e.g. **EmployeeID**.
2. Click on the **Primary Key** button.

If there is more than one primary key, press and hold down the **Ctrl** key, and click on the row selector for other fields.



To remove a primary key, click on the row selector of the field set as a primary key and click on the primary key button.

Field Name	Data Type
AutoNum	AutoNumber
EmployeeID	Text
LastName	Text
FirstName	Text
MI	Text
Street	Text
City	Text
State	Text
Zip	Text
Exempt	Yes/No
Campus	Text

Activity

Set a primary key for AutoNum and remove the primary key for Employee ID.