

# Access Tutorial 2: Tables

## 2.1 Introduction: The importance of good table design

Tables are where data in a database is stored; consequently, tables form the core of any database application. In addition to basic data, Access permits a large amount of domain knowledge (such as captions, default values, constraints, etc.) to be stored at the table level.



Extra time spent thinking about table design can result in enormous time savings during later stages of the project. Non-trivial changes to tables and relationships become increasingly difficult as the application grows in size and complexity.

## 2.2 Learning objectives

- How do I enter and edit data in the datasheet view of a table?
- How do I create a new table?
- How do I set the primary key for a table?
- How do I specify field properties such as the input mask and caption?
- Why won't an autonumber field restart counting at one?
- What are the different types of keys?

## 2.3 Tutorial exercises

In this tutorial, you will learn to interact with existing tables and design new tables.

### 2.3.1 Datasheet basics

- If you have not already done so, open the `univ0_vx.mdb` database file from [Tutorial 1](#).
- Open the `Departments` table. The important elements of the **datasheet** view are shown in [Figure 2.1](#).
- Use the field selectors to adjust the width of the `DeptName` field as shown in [Figure 2.1](#).
- Add the Biology department (BIOL) to the table, as shown in [Figure 2.2](#).
- Delete the “Basket Weaving” record by clicking on its **record selector** and pressing the *Delete* key.

### 2.3.2 Creating a new table

In this section you will create and save a very basic skeleton for table called `Employees`. This table could be used to keep track of university employees

such as lecturers, department heads, departmental secretaries, and so on.

- Return to the database window and create a new table as shown in [Figure 2.3](#).
- In the **table design window** shown in [Figure 2.4](#), type in the following information:

| Field name | Data type | Description (optional) |
|------------|-----------|------------------------|
| EmployeeID | Text      | use employee S.I.N.    |
| FName      | Text      | First name             |
| LName      | Text      | Last name              |
| Phone      | Text      |                        |
| Salary     | Currency  |                        |

- Select *File > Save* from the main menu (or press *Control-S*) and save the table under the name `Employees`.

**FIGURE 2.1:** The datasheet view of the Departments table.

The field names are shown in the “field selectors” across the top of the columns.

? You can temporarily sort the records in a particular order by right-clicking any of the field selectors.

The records are shown as rows.

The black triangle indicates the “current record”.

The grey boxes are “record selectors”.

The “navigation buttons” at the bottom of the window indicate the current record number and allow you to go directly to the first, previous, next, last, or new record.

**a** Resize the DeptName column by clicking near the column border and dragging the border to the right.

| DeptCode | DeptName         | Building |
|----------|------------------|----------|
| BSKW     | Basket Weaving   | ANGU     |
| COMM     | Commerce and     | ANGU     |
| CRWR     | Creative Writing | BUCH     |
| EDUC     | Education        | SCRF     |
| ENGL     | English          | BUCH     |
| MATH     | Math             | MATH     |
| MUSC     | Music            |          |
| *        |                  |          |

The asterisk (\*) indicates a place holder for a new record.



FIGURE 2.2: Adding and saving a record to the table.

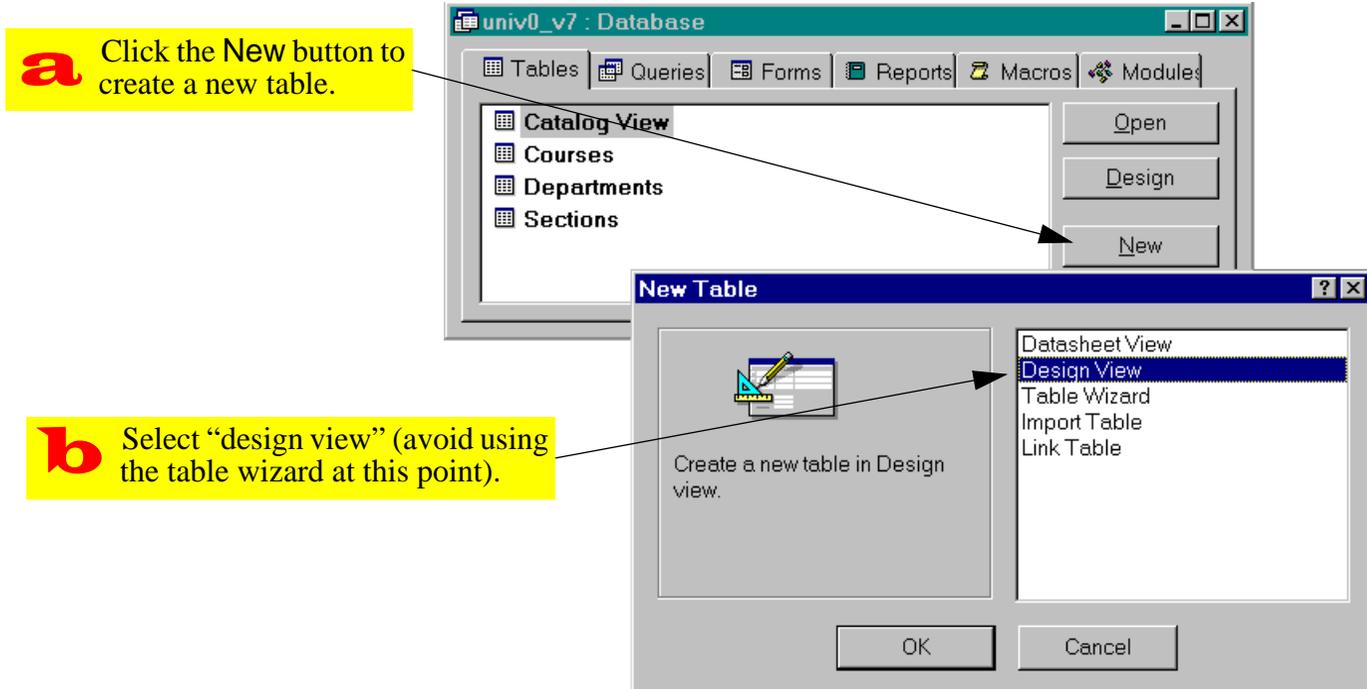
**a** Add a new record by clicking in the DeptCode field of the “new record” field (marked by the asterisk).

| Departments : Table |          |          |                                      |
|---------------------|----------|----------|--------------------------------------|
|                     | DeptCode | DeptName | Building                             |
|                     | BSKW     |          |                                      |
|                     | COMM     |          |                                      |
|                     | CRWR     | BSKW     | Basket Weaving                       |
|                     | EDUC     | COMM     | Commerce and Business Administration |
|                     | ENGL     | CRWR     | Creative Writing                     |
|                     | MATH     | EDUC     | Education                            |
|                     | MUSC     | ENGL     | English                              |
|                     | BIOL     | MATH     | Math                                 |
|                     | *        | MUSC     | Music                                |
|                     |          | BIOL     | Biology                              |

**?** It is seldom necessary to explicitly save new records (or changes to existing records) since Access automatically saves whenever you move to another record, close the table, quit Access, etc.

**b** To permanently save the change to the data, click on the record selector (note the icon changes from a pencil to a triangle).

FIGURE 2.3: Create a new table.



**FIGURE 2.4:** Use the table design window to enter the field properties for the `Employees` table.

**a** Enter the field names and data types for the five fields.

The “description” column allows you to enter a short comment about the field (this information is not processed in any way by Access).

The “field properties” section allows you to enter information about the field and constraints on the values for the field.

| Field Name | Data Type | Description         |
|------------|-----------|---------------------|
| EmployeeID | Text      | use employee S.I.N. |
| FName      | Text      |                     |
| LName      | Text      |                     |
| Phone      | Text      |                     |
| Salary     | Currency  |                     |

Field Properties

General | Lookup

Field Size: 50

Format: \_\_\_\_\_

Input Mask: \_\_\_\_\_

Caption: \_\_\_\_\_

Default Value: \_\_\_\_\_

Validation Rule: \_\_\_\_\_

Validation Text: \_\_\_\_\_

Required: No

Allow Zero Length: No

Indexed: Yes (Duplicates OK)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

### 2.3.3 Specifying the primary key

Tables normally have a primary key that uniquely identifies the records in the table. When you designate a field as the primary key, Access will not allow you to enter duplicate values into the field.

- Follow the steps in [Figure 2.5](#) to set the primary key of the table to `EmployeeID`.

### 2.3.4 Setting field properties

In this section, you will specify a number of field properties for the `EmployeeID` field, as shown in [Figure 2.6](#).

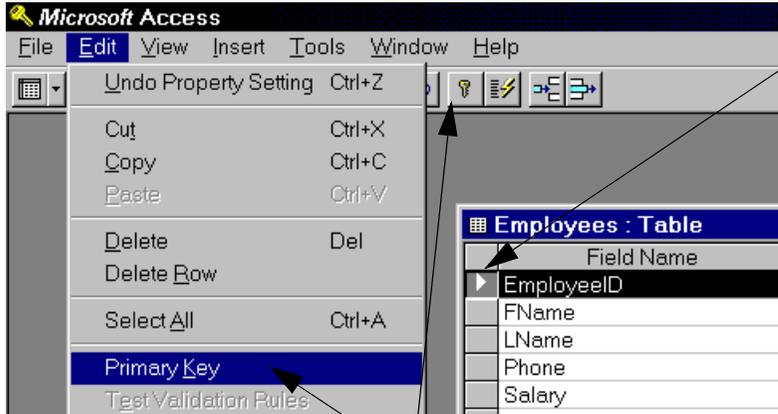
- Since we are going to use the employees' Social Insurance Number (S.I.N.) to uniquely identify them, set the *Field Size* property to 11 characters (9 for numbers and 2 for separating spaces)
- Set the *Input Mask* property to the following:  
000\ 000\ 000;0
- Set the *Caption* property to `Employee ID`

**FIGURE 2.6:** Set the field properties for the `EmployeeID` field.

| Employees : Table |            |           |                     |
|-------------------|------------|-----------|---------------------|
|                   | Field Name | Data Type |                     |
| ?                 | EmployeeID | Text      | use employee S.I.N. |
|                   | FName      | Text      | first name          |
|                   | LName      | Text      | last name           |
|                   | Phone      | Text      |                     |
|                   | Salary     | Currency  |                     |

| General           | Lookup              |
|-------------------|---------------------|
| Field Size        | 11                  |
| Format            |                     |
| Input Mask        | 000\ 000\ 000;0     |
| Caption           | Employee ID         |
| Default Value     |                     |
| Validation Rule   |                     |
| Validation Text   |                     |
| Required          | No                  |
| Allow Zero Length | No                  |
| Indexed           | Yes (No Duplicates) |

**FIGURE 2.5:** Set the primary key for the `Employees` table.

**a** Click on the grey box beside the field (or fields) that form the primary key.

**?** To select more than one field for use as the primary key, hold down the **Control** key while clicking on the grey boxes.

**b** Either click the key-shaped icon in the tool bar or select **Edit > Primary Key** from the menu.

- Select *View > Datasheet* from the main menu to switch to datasheet mode as shown in [Figure 2.7](#). Enter your own S.I.N. and observe the effect of the input mask and caption on the `EmployeeID` field.
- Select *View > Table Design* from the main menu to return to design mode.
- Set the field properties for `FName` and `LName` (note that *Length* and *Caption* are the only two properties that are relevant for these two fields)

### 2.3.5 Using the input mask wizard

In this section, you will use the input mask wizard to create a complex input mask for a standard field type. You will also use the help system to learn more about the meaning of the symbols used to create input masks.

- Select the `Phone` field, move the cursor to the input mask property, and click the button with

three small dots (  ) to invoke the input mask wizard.

- Follow the instructions provided by the wizard as shown in [Figure 2.8](#).
- Press *F1* while the cursor is still in the input mask property. Scroll down the help window to find the meaning of the “0”, “9”, “>” and “L” input mask symbols.

## 2.4 Discussion

### 2.4.1 Key terminology

A key is one or more fields that uniquely determine the identity of the real-world object that the record is meant to represent. For example, there is a record in the student information system that contains information about you as a student. To ensure that the record is associated with you and only you, it con-

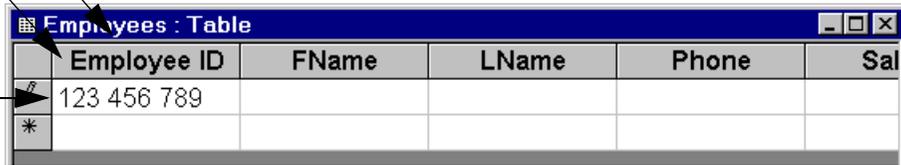
**FIGURE 2.7:** Observe the effect of the input mask and caption properties on the behavior of the EmployeeID field during data entry

**a** Try entering various characters and numbers into the EmployeeID field.

If a caption is specified, it replaces the field name in the field selector.

Note that the input mask will not let you type any characters other than numbers from 0-9. In addition, the spaces between the groups of numbers are added automatically.

**b** Press the Escape key when you are done to clear the changes to the record.



| Employee ID | FName | LName | Phone | Sal |
|-------------|-------|-------|-------|-----|
| 123 456 789 |       |       |       |     |
| *           |       |       |       |     |

**?** Input masks provide a relatively easy way to avoid certain basic data input errors without having to write complex error checking programs. Note, however, that it is possible to over-constrain a field so that users are unable to enter legitimate values.

**FIGURE 2.8:** Use the input mask wizard to create an input mask.

**a** Select “phone number” from the list of commonly-used field types.

**?** The items in this list depend on the “international settings” specified for Windows (e.g., “Zip Code” may show instead of “Postal Code”).

**b** In Step 2, you may edit the input mask (e.g., remove the area code section).

**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:     |
|-------------------------|----------------|
| Phone Number            | (206) 555-1212 |
| Social Insurance Number | 555 333 555    |
| Postal Code             |                |
| Password                |                |
| Medium Date             |                |
| Short Date              |                |

**Input Mask Wizard**

Do you want to change the input mask?

Input Mask Name: Phone Number

Input Mask: !(999) 000-0000

What placeholder character do you want?

Placeholders are replaced as you enter data.

Placeholder character: [v]

**Input Mask Wizard**

How do you want to store the data?

With the symbols in the mask, like this:  
(206) 555-1212

Without the symbols in the mask, like this:  
2065551212

**?** Since the input mask controls how the information in the field looks, it is possible to save some disk space by storing the data without the extras symbols, spaces, etc. For the size of system we are building, however, this savings is negligible.

Cancel < Back Next >

tains a field called “student number” that is guaranteed to be unique.

The advantage of using student number as a key instead of some other field—like “student name”—is that there may be more than one person with the same first and last name. The combination of student name and address is probably unique (it is improbable that two people with the same name will at the same address) but using these two fields as a key would be cumbersome.

Since the terminology of keys can be confusing, the important terms are summarized below.

1. **Primary key** — The terms “key” and “primary key” are often used interchangeably. Since there may be more than one **candidate** key for an application, the designer has to select one: this is the primary key.
2. **Concatenated key:** The verb “concatenate” means to join together in a series. A concate-

nated key is made by joining together two or more fields. Course numbers at UBC provide a good example of a concatenated key made by joining together two fields: `DeptCode` and `CrsNum`. For example, department alone cannot be the primary key since there are many courses in each department (e.g., COMM 335, COMM 391). Similarly, course number cannot be used as a key since there are many courses with the same number in different departments (e.g., COMM 335, HIST 335, MATH 335). However, department and course number together form a concatenated key (there is only one COMM 335).

3. **Foreign key:** In a one-to-many relationship, a foreign key is a field (or fields) in the “child” record that uniquely identifies the correct “parent” record. For example, `DeptCode` and `CrsNum` in the `Sections` table are foreign keys since these two keys taken together are the primary key of

the `Courses` table. Foreign keys are identified in Access by creating relationships (see [Tutorial 3](#)).

### 2.4.2 Fields and field properties

#### 2.4.2.1 Field names

Access places relatively few restrictions on field names and thus it is possible to create long, descriptive names for your fields. The problem is that you have to type these field names when building queries, macros, and programs. As such, a balance should be struck between readability and ease of typing. You are advised to use short-but-descriptive field names with no spaces.

For example, in [Section 2.3.2](#) you created a field with name `FName`. However, you can use the caption property to provide a longer, more descriptive label such as `First name`. The net result is a field name that is easy to type when programming and a field caption that is easy to read when the data is viewed.

In addition, you can use the comment field in the table design window to document the meaning of field names.



It is strongly recommended that you avoid all non-alphanumeric characters whenever you name a field or database object. Although Access will permit you to use names such as `Customer#`, non-alphanumeric characters (such as `#`, `/`, `$`, `%`, `~`, `@`, etc.) may cause undocumented problems later on.

#### 2.4.2.2 Data types

The field's data type tells Access how to handle the information in the field. For instance, if the data type is date/time, then Access can perform date/time arithmetic on information stored in the field. If the same date is stored as text, however, Access treats it just like any other string of characters. Normally,

the choice of data type is straightforward. However, the following guidelines should be kept in mind:

1. Do not use a **numeric** data type unless you are going to treat the field as a number (i.e., perform mathematical operations on it). For instance, you might be tempted to store a person's student number as an integer. However, if the student number starts with a zero, then the first digit is dropped and you have to coerce Access into displaying it. Similarly, a UBC course number (e.g., 335) might be considered a number; however, since courses like 439B have to be accommodated, a numeric data type for the course number field is clearly inappropriate.
2. Access provides a special data type called **Auto Number (Counter** in version 2.0). An autonumber/counter is really a number of type **Long Integer** that gets incremented by Access every time a new record is added. As such, it is convenient

for use as a primary key when no other key is provided or is immediately obvious.



Since an autonumber is really Long Integer and since relationships can only be created between fields with the same data type, it is important to remember that if an autonumber is used on the “one” side of a relationship, a long integer must be used for the “many” side.

### 2.4.2.3 “Disappearing” numbers in autonumber fields

If, during the process of testing your application, you add and delete records from a table with an autonumber key, you will notice that the deleted keys are not reclaimed.

For instance, if you add records to your `Customer` table (assuming that `CustID` is an autonumber), you will have a series of `CustID` values: 1, 2, 3... If you

later delete customer 1 and 2, you will notice that your list of customers now starts at 3.

Clearly, it would be impossible for Access to renumber all the customers so the list started at 1. What would happen, for instance, to all the printed invoices with `CustID = 2` on them? Would they refer to the original customer 2 or the newly renumbered customer 2?



The bottom line is this: once a key is assigned, it should never be reused, even if the entity to which it is assigned is subsequently deleted. Thus, as far as you are concerned, there is no way to get your customers table to renumber from `CustID = 1`.

Of course, there is a long and complicated way to do it, but since used an autonumber in the first place, you do not care about the actual value of the key—you just want it to be unique. In short, it makes abso-

lutely no difference whether the first customer in your customers table is `CustID = 1` or 534.

### 2.4.2.4 Input masks

An input mask is a means of restricting what the user can type into the field. It provides a “template” which tells Access what kind of information should be in each space. For example, the input mask `>LLLL` consists of two parts:

1. The right brace `>` ensures that every character the user types is converted into upper case. Thus, if the user types `comm`, it is automatically converted to `COMM`.
2. The characters `LLLL` are place holders for letters from A to Z with blank spaces not allowed. What this means is that the user has to type in exactly four letters. If she types in fewer than four or types a character that is not within the A to Z scope (e.g., `&`, `7`, `%`), Access will display an error message.

There are a large number of special symbols used for the input mask templates. Since the meaning of many of the symbols is not immediately obvious, there is no requirement to remember the character codes. Instead, simply place the cursor on the input mask property and press *F1* to get help. In addition, the wizard can be used to provide a basic input mask which can later be modified.

### 2.4.2.5 Input masks and literal values

To have the input mask automatically insert a character (such as a space or a dash) in a field, use a slash to indicate that the character following it is a literal.

For example, to create an input mask for local telephone numbers (e.g., 822-6109), you would use the following template: `000\ -0000 ; 0` (the dash is a literal value and appears automatically as the user enters the telephone number).

The semicolon and zero at the end of this input mask are important because, as the on-line help system points out, an input mask value actually consists of three parts (or “arguments”), each separated by a semicolon:

- the actual template (e.g., `000\ -0000`),
- a value (0 or 1) that tells Access how to deal with literal characters, and
- the character to use as a place holder (showing the user how many characters to enter).

When you use a literal character in an input mask, the second argument determines whether the literal value is simply displayed or displayed *and* stored in the table as part of the data.

For example, if you use the input mask `000\ -0000 ; 1`, Access will not store the dash with the telephone number. Thus, although the input mask will always display the number as “822-6109”, the number is actually stored as “8226109”. By using the

input mask `000\ -0000 ; 0`, however, you are telling Access to store the dash with the rest of the data.



If you use the wizard to create an input mask, it asks you a simple question about storing literal values (as shown in [Figure 2.8](#)) and fills in the second argument accordingly. However, if you create the input mask manually, you should be aware that by default, Access does not store literal values. In other words, the input mask `000\ -0000` is identical to the input mask `000\ -0000 ; 1`. This has important consequences if the field in question is subject to referential integrity constraints (the value “822-6109” is not the same as “8226109”).

## 2.5 Application to the assignment

You now have the skills necessary to implement your tables.

- Create all the tables required for the assignment.
- Use the autonumber data type (counter in version 2.0) for your primary keys where appropriate.
- Specify field properties such as captions, input mask, and defaults where appropriate.



If you create an input mask for `ProductID`, ensure you understand the implications of [Section 2.4.2.5](#).

- Set the *Default* property of the `OrderDate` field so that the current date is automatically inserted into the field when a new order is created (hint: see the `Date()` function in the on-line help system).

## 2. Tables

- Do not forget to modify your `Products` table (the data types, lengths, and field properties of imported tables normally need to be fine tuned)
- Populate (enter data into) your master tables. Do not populate your transaction tables.



For the purpose of the assignment, the term “transaction” tables refers to tables that contain information about individual transactions (e.g., `Orders`, `OrderDetails`, `Shipments`, `ShipmentDetails`). “Master” tables, in contrast, are tables that either do not contain information about transactions (e.g., `Customers`) or contain only summary or status information about transactions (e.g., `BackOrders`).