

## Chapter 2: Working with Tables: Design, Properties, Views and Wizards

### Lab Objectives:

After finishing the lab, students will learn:

- how to design a database table in Microsoft access database.
- three different approaches for creating table.
- about the possible field data types and properties available within access table design screen and the purpose of each.
- how to set the primary key for a table.
- the importance of data validation and how it is implemented in Microsoft access.

### Lab Practice Problem

Creating a database table for a university.

#### Problem Statement

You are supposed to create a database table for maintaining student records for a university with the following specifications.

- Student personal information including name, address, and telephone must be maintained.
- Student academic details like major, number of credit hours completed etc. must also be maintained.
- Student status about the financial aid must be maintained.

### Steps in Table Creation

#### Step I: Deciding about the fields and their data types:

By following the guidelines provided from page# 50 to 52, the following list of fields with their data types can be thought of.

<b>Field Name</b>	<b>Type</b>
StudentID	Text
FirstName	Text
LastName	Text
Address	Text
City	Text
State	Text
PostalCode	Text
PhoneNumber	Text
Major	Text
BirthDate	Date/Time
Financial Aid	Yes/No
Gender	Text
Credits	Number
QualityPoints	Number
International	Yes/No

In brief, the guideline says:

- All necessary data should be included which is required by the university (by interviewing with the university officials)
- Avoid the saving of composite data. For example, instead of saving full name in one field, there should be three different parts of the name, named last name, first name and middle name.
- Avoid the fields which can be calculated from some existing fields. For example, if date of birth is being saved, there is no need to save the age as a separate field as it can always be calculated using the current date and the date of birth values.

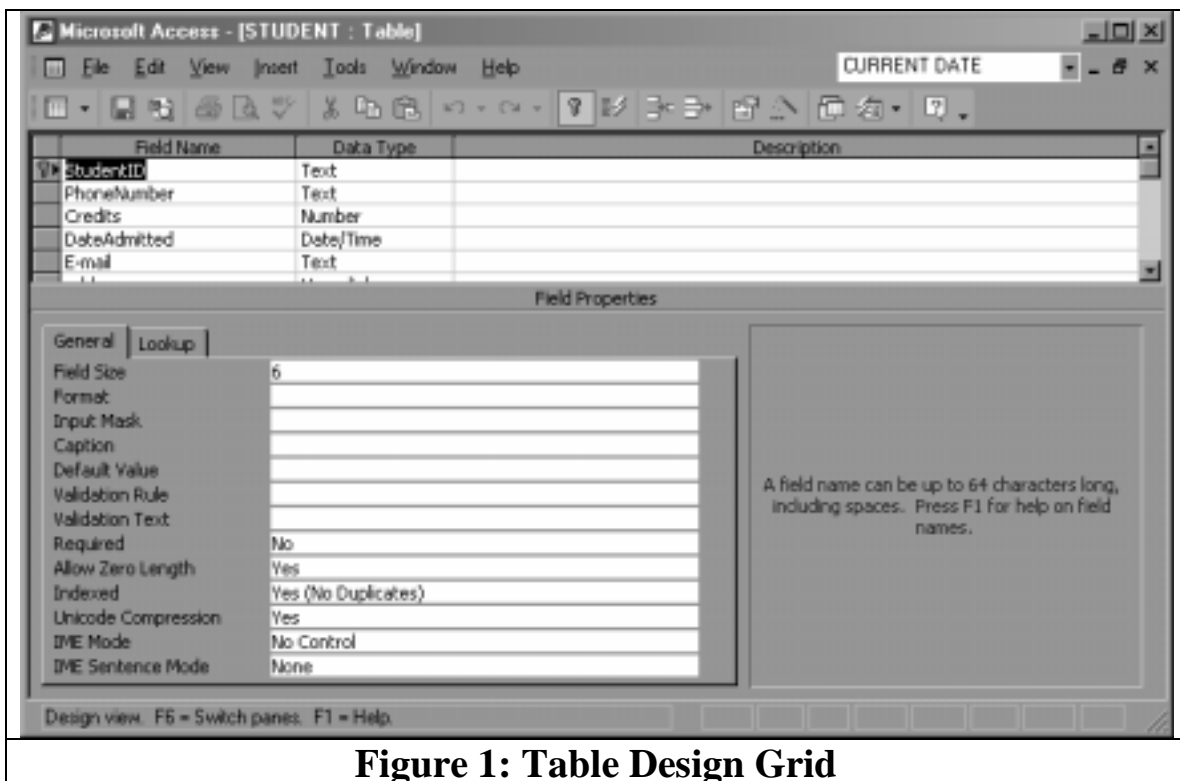
## Step II: Selecting the approach for creating table

Microsoft Access supports three approaches for creating tables:

1. Create table in Design View
2. Create table by using wizard
3. Create table by entering data

### 1. Create table in Design View

- The most basic and flexible way of creating a table.
- Requires the user to think about the whole table layout before its creation.
- A design grid will be provided to define fields, their data types and the set of properties for each field.



**Figure 1: Table Design Grid**

### **Hands On: 8.1: Creating a table using Design View**

1. From the database window, press the button **New**.
2. On the resulting window, select the option '**Design View**'.
3. The design grid will allow you to enter the field name, the data type and a description for more elaborations.
4. For every field, the properties matrix can also be entered.

### 2. Create table by using Wizard

The wizard option allows the user to create a new table with some pre-built fields.

The tables which can be created are only for some generalized type of situations such as students, employees, etc.

In case the type of table you are looking for, does not exist in the list of the tables created, the design view approach should be used.

### **Hands On: 8.2: Creating a table by using Wizard**

Follow step3 on page 58. Named the table as STUDENT-W.

### 3. Create table by entering data

Microsoft Access allows the user to start entering data without defining any table.

The table automatically starts with a large number of columns (named as Field1, Field2, and so on).

Only those fields will be included in the finalized table whose values will be entered by the user.

The data types will be chosen automatically depending upon the values entered by the user in every field. For example, a field with letters only will be assigned as Text, a field with numeric values will be assigned as Number and so on .

### **Hands On: 8.3: Creating a table by entering data**

1. On the database window, select Table tab. Select the button New to create new table.
2. On the resulting dialogue, select the option named 'Create table by entering data'.
3. A datasheet view with a large number of columns will be started.
4. Start entering the data in the fields.
5. After you finished with the data, change the view to 'Design View'.
6. In the table save dialogue, use the name 'Student-data'
7. Change the field labels as you need and also make the data type more accurately assigned and select the table's primary key.

### Step III: Finding out the property matrix for each field

For each field in the table, a list of property values is needed to be set. The properties will define different aspects of the field including the following:

1. Determining the nature of data- whether needed in calculations or not. [**Data type:** Text or Number or anything else]
2. Will all the values be unique or not? [**Primary Key**]
3. Maximum number of characters needed for each character field. [**Field Size**]
4. Estimating the range (smallest and the largest possible field values) of values for numeric fields. [**Field Size**]
5. Should the field be displayed with some specified format. For example, \$ symbol with currency fields. [**Format**]
6. Should the user be provided with some help in the data entry for the field values using a mask. [**Input mask**]
7. Should the field be labeled with a different more expressive caption, not by the field name itself? [**Caption**]
8. Is there any value which frequently appears as the field value? For example, Gender field most contain the value 'Male'. [**Default Value**]
9. Should the field be checked for validity of the values entered? If yes, what conditions should be used? [**Validation Rule**]
10. In case, the user enter an invalid data, should a customized message is needed to be displayed? [**Validation Text**]
11. Should the users be allowed to leave the value blank? [**Required**]
12. For text and memo field, should the user be allowed to enter strings with no characters. [**Allow Zero length**]
13. Will the field be using for searching purposes frequently? [**Indexed**]

#### **Hands On 8.4: Determining field properties**

Each field will be judged thoroughly with a perspective of finding the property values needed. Following tables will provide details about each field in detail.

**Field Name:** StudentID

**Field Property Matrix:**

<b>Data Type</b>	Text
<b>Primary Key</b>	Yes
<b>Field Size</b>	6
<b>Format</b>	#
<b>Input Mask</b>	000000
<b>Caption</b>	ID Number
<b>Default Value</b>	
<b>Validation Rule</b>	>"000000"
<b>Validation Text</b>	Invalid value of ID entered.
<b>Required</b>	Yes
<b>Allow Zero Length</b>	No
<b>Indexed</b>	Yes (No Duplicates)

**Note:** Format and Input Mask properties should not be defined at the same time. Otherwise, the Format property takes precedence when the data is displayed. This means that even if you've saved an input mask, the input mask is ignored when data is formatted and displayed.

**Field Name:** FirstName, LastName, Address, City, State

**Field Property Matrix:**

<b>Data Type</b>	Text
<b>Primary Key</b>	No
<b>Field Size</b>	25, 25, 25, 10, 2
<b>Format</b>	
<b>Input Mask</b>	
<b>Caption</b>	First Name, .....
<b>Default Value</b>	
<b>Validation Rule</b>	
<b>Validation Text</b>	
<b>Required</b>	Y
<b>Allow Zero Length</b>	N
<b>Indexed</b>	N

**Field Name:** PhoneNumber

**Property Matrix:**

<b>Data Type</b>	Text
<b>Primary Key</b>	No
<b>Field Size</b>	11
<b>Format</b>	
<b>Input Mask</b>	\(000") "000\ -0000;1
<b>Caption</b>	Telephone
<b>Default Value</b>	
<b>Validation Rule</b>	
<b>Validation Text</b>	
<b>Required</b>	No
<b>Allow Zero Length</b>	Yes
<b>Indexed</b>	No

**Field Name:** Credits

**Property Matrix:**

<b>Data Type</b>	Number [0 – 128]
<b>Primary Key</b>	No
<b>Field Size</b>	Byte
<b>Format</b>	
<b>Input Mask</b>	999
<b>Caption</b>	Credit Earned
<b>Default Value</b>	0
<b>Validation Rule</b>	>0 And <=128
<b>Validation Text</b>	Invalid value for credits earn.
<b>Required</b>	Yes
<b>Allow Zero Length</b>	No
<b>Indexed</b>	Yes (Duplicates OK)

**Field Name:** QualityPoints

**Property Matrix:**

<b>Data Type</b>	Number
<b>Primary Key</b>	No
<b>Field Size</b>	Byte
<b>Format</b>	
<b>Input Mask</b>	999
<b>Caption</b>	Quality Points
<b>Default Value</b>	0
<b>Validation Rule</b>	
<b>Validation Text</b>	
<b>Required</b>	Yes
<b>Allow Zero Length</b>	No
<b>Indexed</b>	No

**Field Name:** DateAdmitted

**Property Matrix:**

<b>Data Type</b>	Date/Time
<b>Primary Key</b>	No
<b>Field Size</b>	NA
<b>Format</b>	
<b>Input Mask</b>	90\90\0000
<b>Caption</b>	Date Admitted
<b>Default Value</b>	Date()
<b>Validation Rule</b>	
<b>Validation Text</b>	
<b>Required</b>	
<b>Allow Zero Length</b>	
<b>Indexed</b>	

**Field Name:** Gender

**Property Matrix:**

<b>Data Type</b>	Text
<b>Primary Key</b>	No
<b>Field Size</b>	1
<b>Format</b>	
<b>Input Mask</b>	A
<b>Caption</b>	
<b>Default Value</b>	“M”
<b>Validation Rule</b>	“M” or “F”
<b>Validation Text</b>	Invalid value for gender.
<b>Required</b>	
<b>Allow Zero Length</b>	
<b>Indexed</b>	

**Field Name:** E-mail

**Property Matrix:**

<b>Data Type</b>	Text
<b>Primary Key</b>	No
<b>Field Size</b>	
<b>Format</b>	
<b>Input Mask</b>	
<b>Caption</b>	
<b>Default Value</b>	
<b>Validation Rule</b>	
<b>Validation Text</b>	
<b>Required</b>	
<b>Allow Zero Length</b>	
<b>Indexed</b>	

**Field Name:** HomePage

**Property Matrix:**

<b>Data Type</b>	Hyperlink
<b>Primary Key</b>	No
<b>Field Size</b>	
<b>Format</b>	
<b>Input Mask</b>	
<b>Caption</b>	
<b>Default Value</b>	
<b>Validation Rule</b>	
<b>Validation Text</b>	
<b>Required</b>	
<b>Allow Zero Length</b>	
<b>Indexed</b>	

## Chapter1: Creating Relationships in a Relational database & its Significance (pg 30)

### Defining a Relationship

A relationship creates a link between two related table. A common field is essential to be included in the tables involved in a relationship.

### Example of a relational database

- On page 30, the database named LookingAhead.mdb, contains three tables, named: Employees, Locations & Titles.
- The tables are related to each other with the following relationships:

#### Locations : Employees

- Shows a one-to-many relationship [1 location appears in many employees records]
- Parent table= Locations
- Child table = Employees

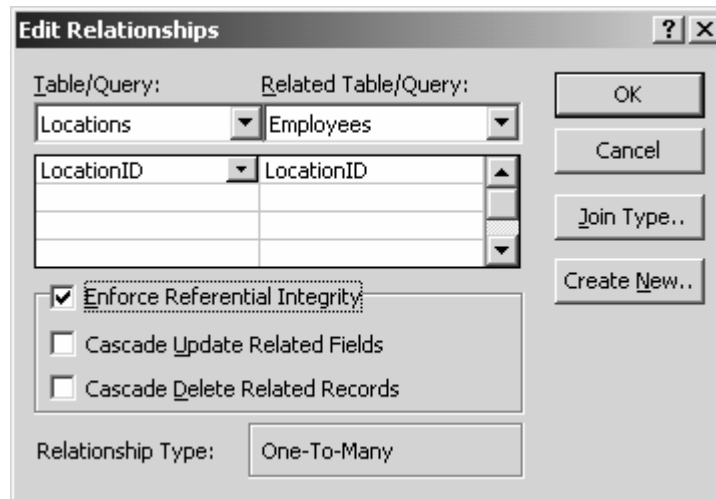
#### Titles : Employees

- Shows a one-to-many relationship [1 title appears in many employees records]
- Parent table= Titles
- Child table = Employees

### Creating Relationships in Microsoft Access

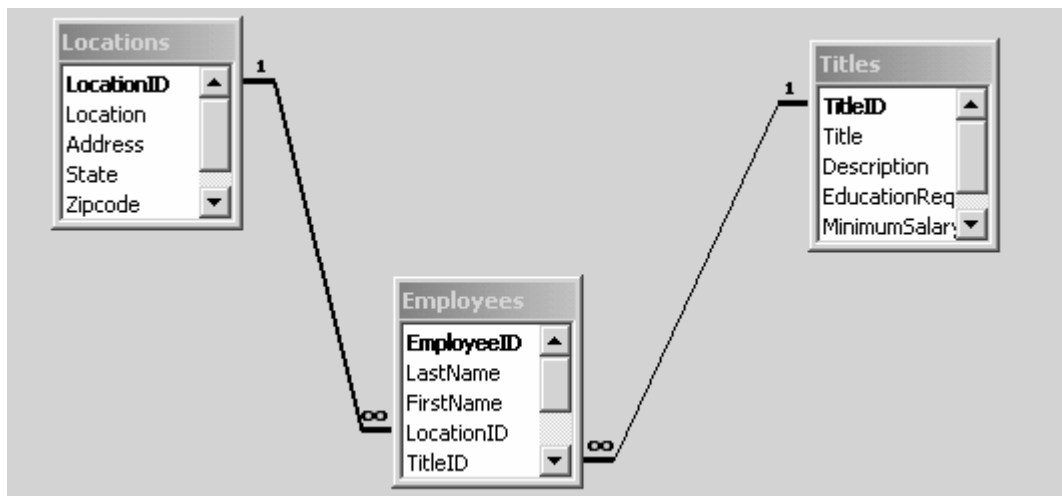
1. From database window, open relationship window.
2. Show the tables among which the relationship needs to be defined.
3. Drag the common field from one table to the other.
4. A relationship window will be opened as follows:





**Figure 2: Relationship Definition Window**

After creating the relationships, the table relationships will look like as follows:



### **Benefits of Creating Relationship**

#### 1. Retrieving Query Results

Microsoft Access will easily find the complete information about some entity by traveling through defined relationships from table to table.

#### 2. Automatically reject invalid data (referential integrity)

The data which is invalid due to a non-existing parent record will automatically be checked and rejected.

#### 3. Easy data entry

Microsoft Access lets the user to enter data into related table from one common point (parent table datasheet view).

While using Forms to enter data, only valid values will be presented in lists to enter the accurate data only.

#### 4. Presenting data in Report Formats

The data can be arranged more effectively on reports if the relationships are defined properly.

#### **Hands On 8.5: Creating Relationships between tables**

1. Open the database LookAhead.mdb.
2. Check the referential integrity feature by entering a wrong record in the child table 'Employee' for a wrong location 'L44' which is not available in the 'Locations' table. The error will go on unchecked.
3. Open the relationship window and create relationship between tables as instructed on page 33.
4. Check the referential integrity feature again (as instructed on page 34).

#### **Hands On 8.6: Changing data in related tables from one common point**

Change the records of location table with the related employees as instructed on page 37.

#### **Hands On 8.7: Reports showing parent-child arrangement of data**

See the reports already done as 'Employees by Title' (page 38). See the arrangement of data there.