



NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

TRACKING OF PERSONAL FINANCES

**Graduation Project
COM-400**

Student: Mehmet Ali Yıldız

Supervisor: Assist. Prof. Erdal Onurhan

Nicosia - 2003

ACKNOWLEDGEMENT

"First, My sincerest thank to my supervisor Mr. Erdal Onurhan and lab assistant Jan Jua Awais for their sugestion over the Graduation Project and for their peerless helps.

Second, I wish to thank Near East University because of giving a chance for education and for made the work possible

Third, I want to thank my family and my friends for their constant encouragement and support during the preparation of this project.

Finally, I wish to thankevery individuals who have shared their helps."

ABSTRACT

Tracking of Personal Finance program is controled and calculated revenues and expences of any person. This program makes easy to track financial balance. It contains a main form and two subforms. Main form takes essential information about person. Subform takes essential information about expences and incomes.

The aim of this project is the grouping and calculation of revenues and expences. For example, fixed incomes (electric, water, rent etc.) and variable incomes(cost of variable material (cost of individual)). Program divides revenues and expences into two groups that are mounthly and yearly.

There are also extra incomes and expences. So complexity of tracking finance are increased. Because they are variable cost so when any values of extras are calculated we can not be appointed their values to standard type. But program has a part that name is extra name. It gives Name of Extra and takes its name as a fixed incomes or expences. This standardization is minimized the complexity of tracking.

The based idea of this program is to increase sensivity and to make easy the calculation.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	<i>i</i>
ABSTRACT	<i>ii</i>
TABLE OF CONTENTS	<i>iii</i>
INTRODUCTION	1
CHAPTER ONE : INTRODUCTION TO ACCESS	2
1.1 A Few Terms	2
1.2 Getting Started	3
1.3 Blank Access database	3
1.4 Access database wizards, pages, and projects	4
1.5 Open an existing database	4
1.6 Converting to Access 2000	4
CHAPTER TWO: SCREEN LAYOUTS	5
2.1 Database Window	5
2.2 Design View	6
2.3 Datasheet View	7
CHAPTER THREE: TABLES	9
3.1 Introduction to Tables	9
3.2 Create a Table in Design View	10
3.2.1 Field Name	10
3.3 Field Properties	11
3.4 Primary Key	14

3.5 Indexes	15
3.6 Field Validation Rules	15
3.7 Input	15
CHAPTER FOUR: DATASHEET RECORDS	17
4.1 Adding Records Masks	17
4.2 Editing Records	17
4.3 Deleting Records	18
4.4 Adding and Deleting Columns	18
4.5 Resizing Rows and Columns	18
4.6 Freezing Columns	18
4.7 Hiding Columns	19
4.8 Finding Data in a Table	20
4.9 Replace	21
4.10 Check Spelling and AutoCorrect	22
4.11 Print a Datasheet	22
CHAPTER FIVE: TABLE RELATIONSHIPS	23
5.1 Table Relationships	23
CHAPTER SIX: QUERY	26
6.1 Introduction to Queries	26
6.2 Create A Query in Design View	26
6.3 Query Wizard	29
6.4 Find Duplicates Query	30
6.5 Delete a Query	33

CHAPTER SEVEN: FORM	34
7.1 Create Form in Design View	34
7.2 Adding Records Using A Form	35
7.3 Editing Form	36
CHAPTER EIGHT: SUBFORM	38
8.1 What Is A Subform?	38
8.2 Create a Form and Subform at Once	38
8.3 Subform Wizard	42
8.4 Drag-and-Drop Method	43
CHAPTER NINE: REPORTS	45
9.1 Reports	45
9.2 Using the Wizard	45
9.3 Create in Design View	48
CHAPTER TEN: INTRODUCTION TO PERSONAL FINANCE	49
10.1 Personal Information Table	49
10.2 Desing of Personal Finance Form	50
10.3 Personal ID	51
10.4 Name, Surname, Phone, Address	51
10.5 Expenses Button	52
10.5.1 Microsoft Visual Basic Codes For Expences Button	52
10.6 Revenue Buton	53
10.6.1 Microsoft Visual Basic Codes For Revenue Button	53

CHAPTER ELEVEN: EXPENCES FORM	54
11.1 Expenses Table	54
11.2 Desing of Expenses form	55
11.3 Expences In a mount	56
11.4 Extra Name And Amount In A Mount	56
11.5 Expenses in A Year	57
11.6 Extra Amount And Name In A Year	57
11.7 Add Button	57
11.7.1 Microsoft Visual Basic Codes For Add Button	57
11.8 Delete Button	58
11.8.1 Microsoft Visual Basic Codes For Delete Button	58
11.9 Search Button	58
11.9.1 Microsoft Visual Basic Codes For Search Form	59
11.9.2 Search Button of Search Form	59
11.9.2.1 Microsoft Visual Basic Codes For Search Button of Search Form	59
11.9 Total Button	60
CHAPTER TWELVE: REVENUE FORM	61
12.1 Revenue Table	61
12.2Design of Revenue Form	62
12.3 Revenue In A Mounth	63
12.4 Extra Name And Amount In A Mounth	63
12.5 Expences In A Year	64
12.6 Extra Amount And Name In A Year	64

12.7 Add Button	64	
12.7.1 Microsoft Visual Basic Codes For Add Button	64	
12.8 Delete Button	65	
12.8.1 Microsoft Visual Basic Codes For Delete Button	65	
12.9 Search Button	65	
12.9.1 Microsoft Visual Basic Codes For Search Form	66	
12.9.2 Search Button of Search Form	66	
12.9.2.1 Microsoft Visual Basic Codes For Search Button of Search Form		67
12.10 Total Button	67	
REFERENCES	68	
CONCULUTION	69	

INTRODUCTION

Access is usually used for creating a database. It is easiest way to create relationship between the related informations. It is minimized the complexity of relation

In this project we are considered that how can we create an access project and what is the Tracking of Personal Finances program. The project consist of introduction 12 chapter.

Chapter one describes introduction to access and some terms about access.

Chapter two describes component of screen layouts and how can we used it. Screen layouts describes database window, design view and datasheet view.

Chapter three describes introduction to tables and creating a tables with design view.

Chapter four describes datasheet records and its functions. These functions are adding record, deleting record, and editing record etc..

Chapter five describes relationship tables and how can we create a relationship between two or more tables.

Chapter six describes introduction to query and creating a query in design view.

Chapter seven explains form and creating a form in design view.

Chapter eight explains what is a subform and how can be it design in form wizard.

Chapter nine explains report and how it can be it design in design view and report wizard.

Chapter ten describes introduction to tracking of personal finances and what is it.

Chapter eleven describes expenses form of personal finances and its components.

Chapter twelve describes revenue form of personal finances and its components.

CHAPTER ONE

INTRODUCTION TO ACCESS

1.1 A Few Terms

These words are used often in Access so you will want to become familiar with them before using the program and this tutorial.

1. A **database** is a collection of related information.
2. An **object** is a competition in the database such as a table, query, form, or macro.
3. A **table** is a grouping of related data organized in fields (columns) and records (rows) on a datasheet. By using a common field in two tables, the data can be combined. Many tables can be stored in a single database.
4. A **field** is a column on a datasheet and defines a data type for a set of values in a table. For a mailing list table might include fields for first name, last name, address, city, state, zip code, and telephone number.
5. A **record** in a row on a datasheet and is a set of values defined by fields. In a mailing list table, each record would contain the data for one person as specified by the intersecting fields.
6. **Design View** provides the tools for creating fields in a table.
7. **Datasheet View** allows you to update, edit, and delete information from a table.

1.2 Getting Started

After opening Access, you will be presented with the window shown below. Select one of the first two options if you are creating a new database, or the third if you want to edit an existing database. All three choices are explained in **figure 1.1**

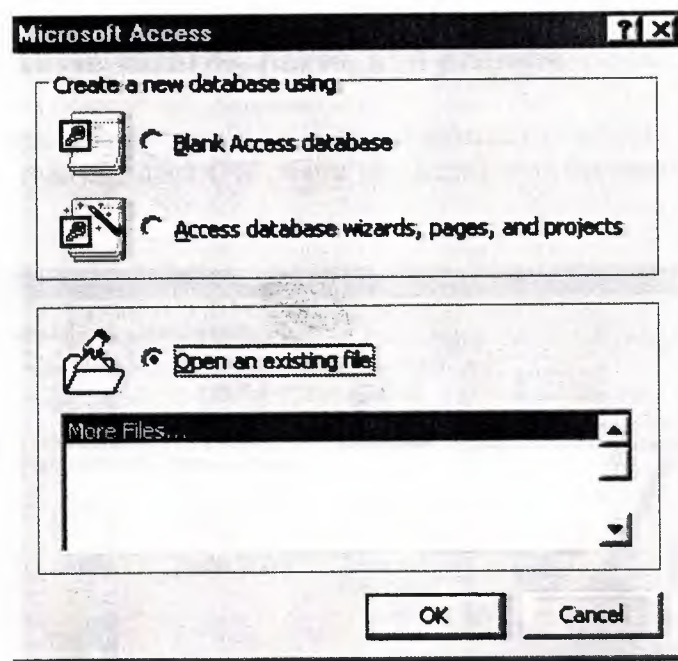


Figure 1.1. Microsoft Access

1.3 Blank Access database

1. Unlike Word documents, Excel worksheets, and Power Point presentations, you must save an Access database before you start working on it. After selecting "Blank Access database" (**Figure1.2.**), you will first be prompted to specify a location and name for the database.

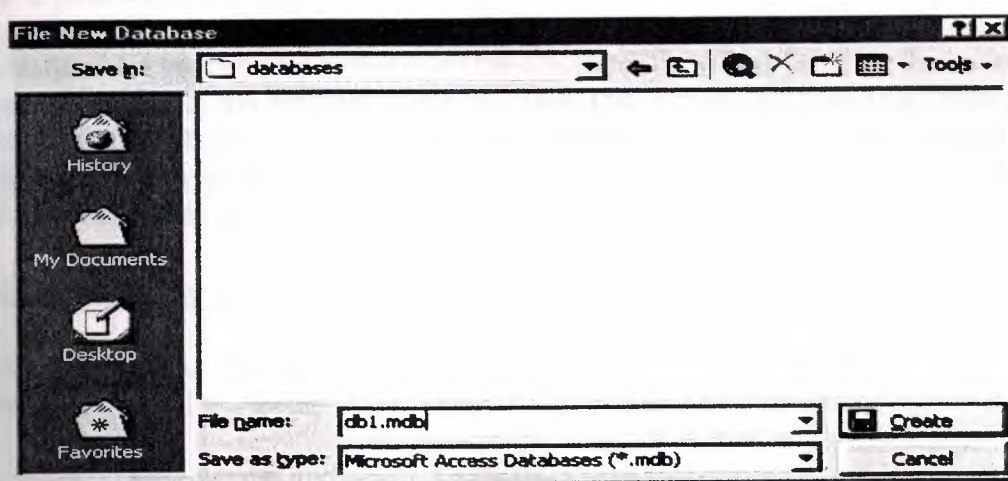


Figure 1.2.

2. Find the folder where the database should reside in the **Save in** drop-down menu.
3. Type the name of the database in the **File name** line and click the **Create** button.

1.4 Access database wizards, pages, and projects

Access' wizards and layout are existing database structures that only need data input. Select a database type and click **OK**. Name the database on the next screen (Figure-1.3.).

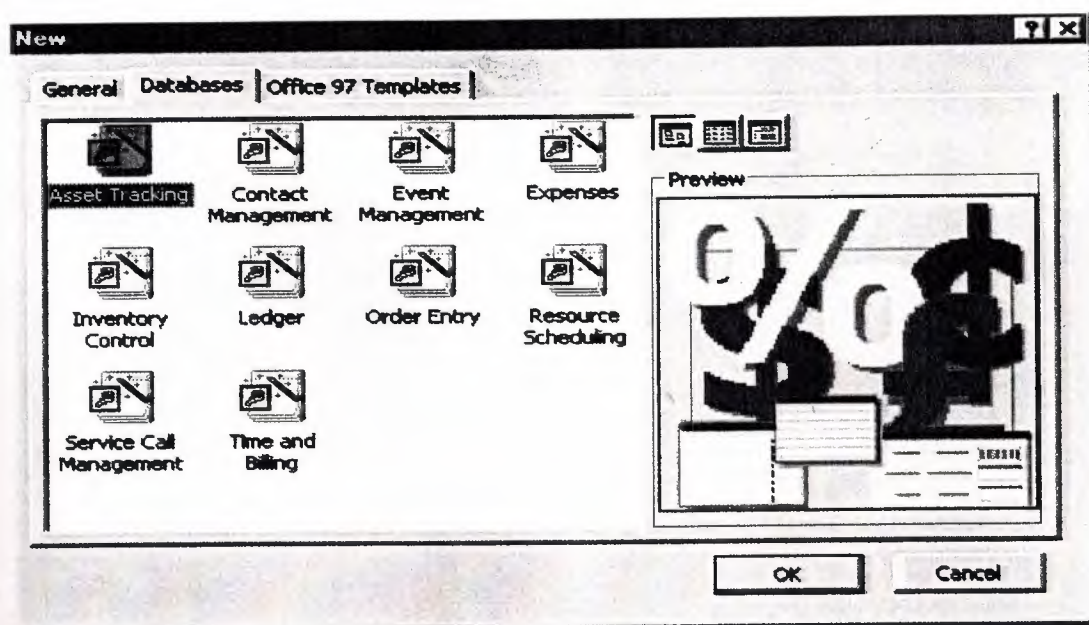


Figure 1.3.

1.5 Open an existing database

If the database was opened recently on the computer, it will be listed on the main window. Highlight the database name and click **OK**. Otherwise, highlight "More Files..." in the list and click **OK**. From the subsequent window, click the "Look In:" drop-down menu to find the folder where the database is located, highlight the database name in the listing and click **OK**.

1.6 Converting to Access 2000

Before opening an existing file that was created in a previous version of Access, it must first be converted to Access 2000 format. Convert a database by following these steps:

1. Open Access and select **Tools|Database Utilities|Convert Database|To Current Access Database Version** from the menu bar.

2. Select the database that should be converted and click the **Convert** button.
 3. The new version will be a completely separate database and the old one will remain intact so you must then name the new version of the database.
- Adding Records



Figure 2.1

CHAPTER TWO

SCREEN LAYOUTS

2.1 Database Window

The Database Window organizes all of the objects in the database. The default tables listing provides links for creating tables and will list all of the tables in the database when they have been added.

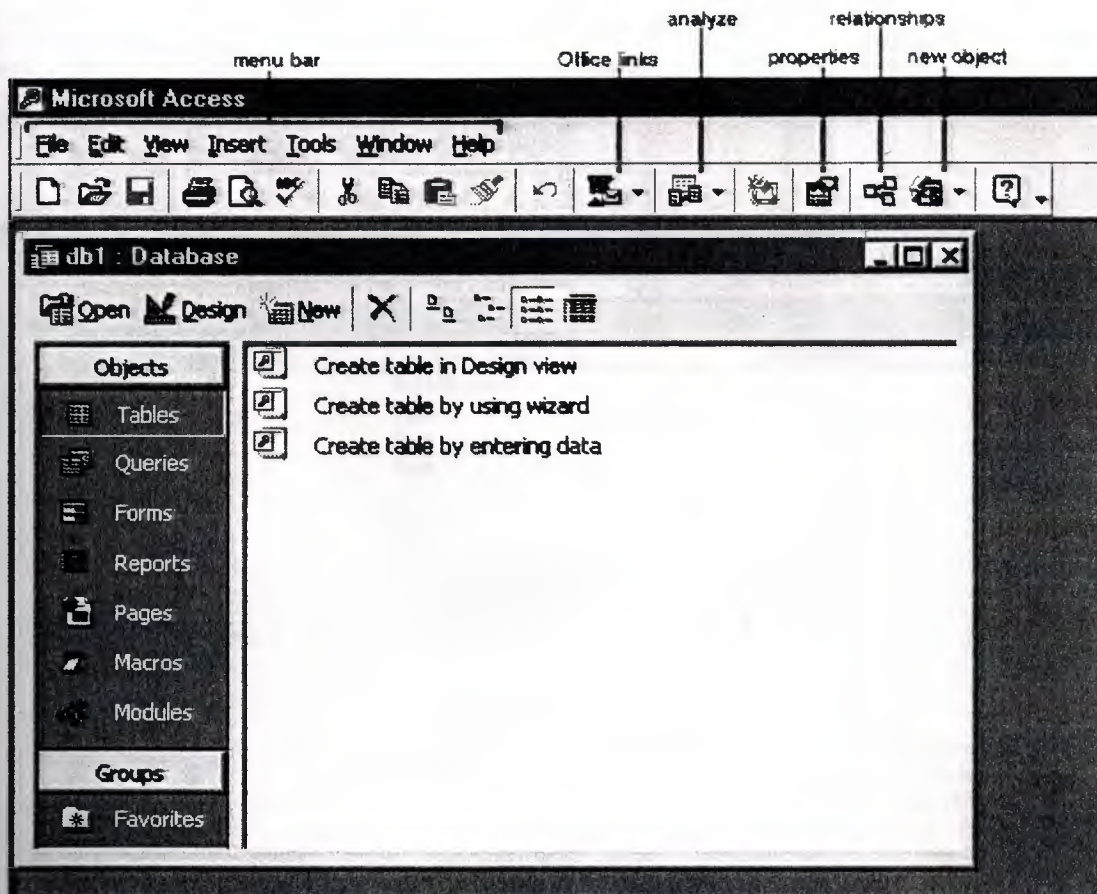


Figure 2.1.

2.2 Design View

Design View customizes the fields in the database so that data can be entered.

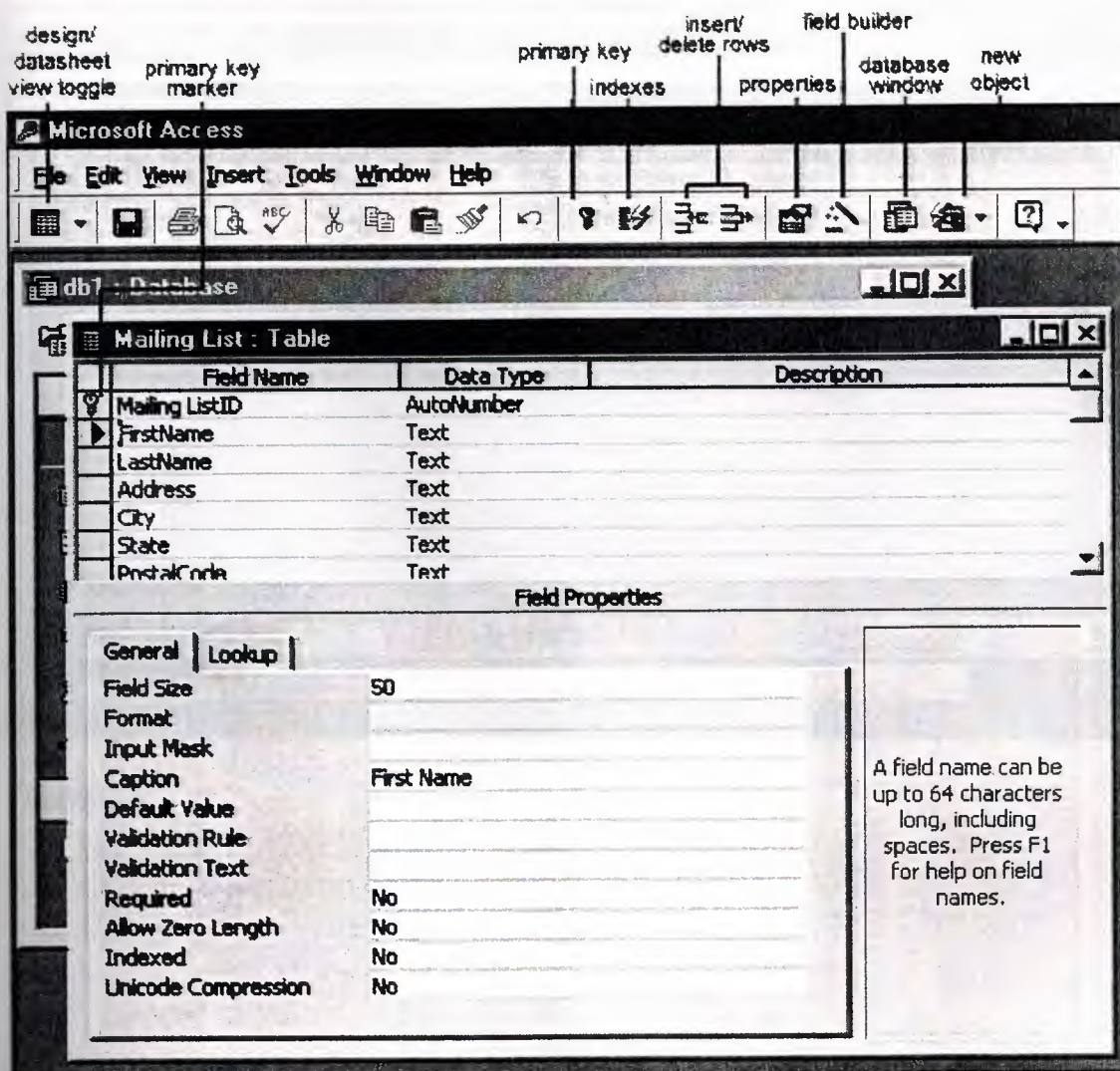


Figure 2.2

2.3 Datasheet View

The datasheet allows you to enter data into the database

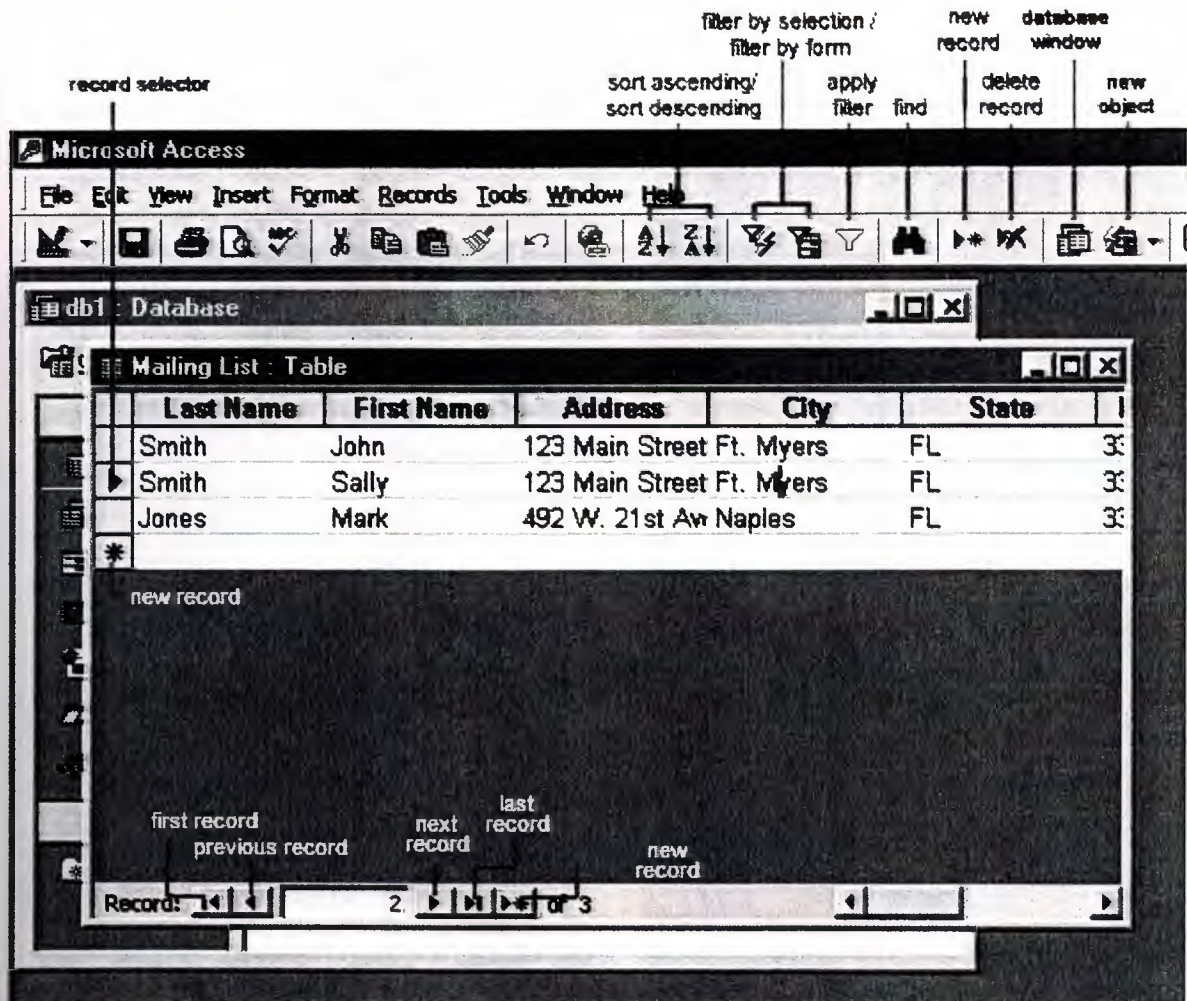


Figure 2.3.

CHAPTER THREE

TABLES

3.1 Introduction to Tables

Tables are grids that store information in a database similar to the way an Excel worksheet stores information in a workbook. Access provides three ways to create a table for which there are icons in the Database Window. Double-click on the icons to create a table.

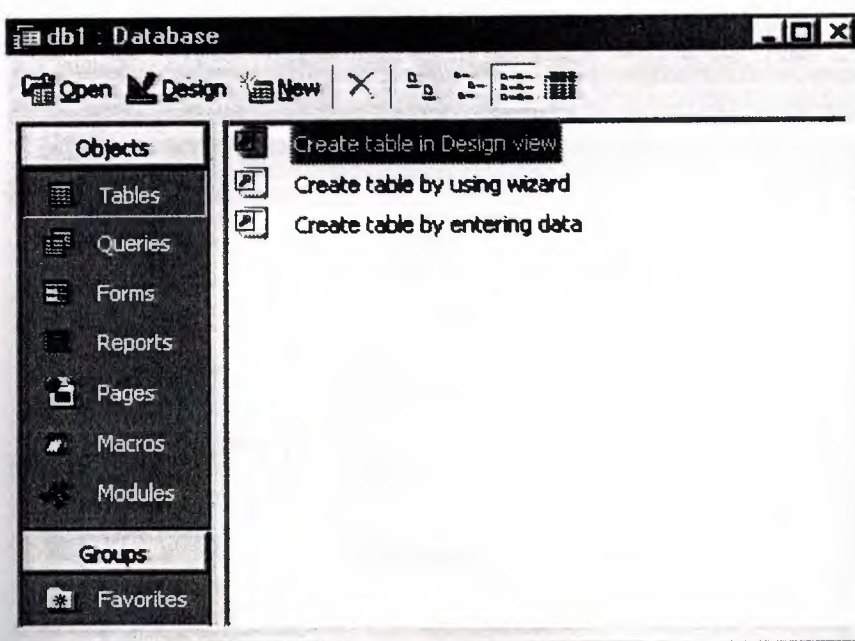


Figure 3.1. The Database Window

1. **Create table in Design view** will allow you to create the fields of the table. This is the most common way of creating a table and is explained in detail below.
2. **Create table using wizard** will step you through the creation of a table.
3. **Create table by entering data** will give you a blank datasheet with unlabelled columns that looks much like an Excel worksheet. Enter data into the cells and click the **Save** button. You will be prompted to add a **primary key** field. After the table is saved, the empty cells of the datasheet are trimmed. The fields are given generic names such as "Field1", "Field2", etc. To rename them with more descriptive titles that reflect the content of the fields, select **Format|Rename Column** from the menu bar or highlight the column, right-click on it with the mouse, and select **Rename Column** from the shortcut menu.

3.2 Create a Table in Design View

Design View will allow you to define the fields in the table before adding any data to the datasheet. The window is divided into two parts: a top pane for entering the field name, data type, and an option description of the field, and a bottom pane for specifying field properties.

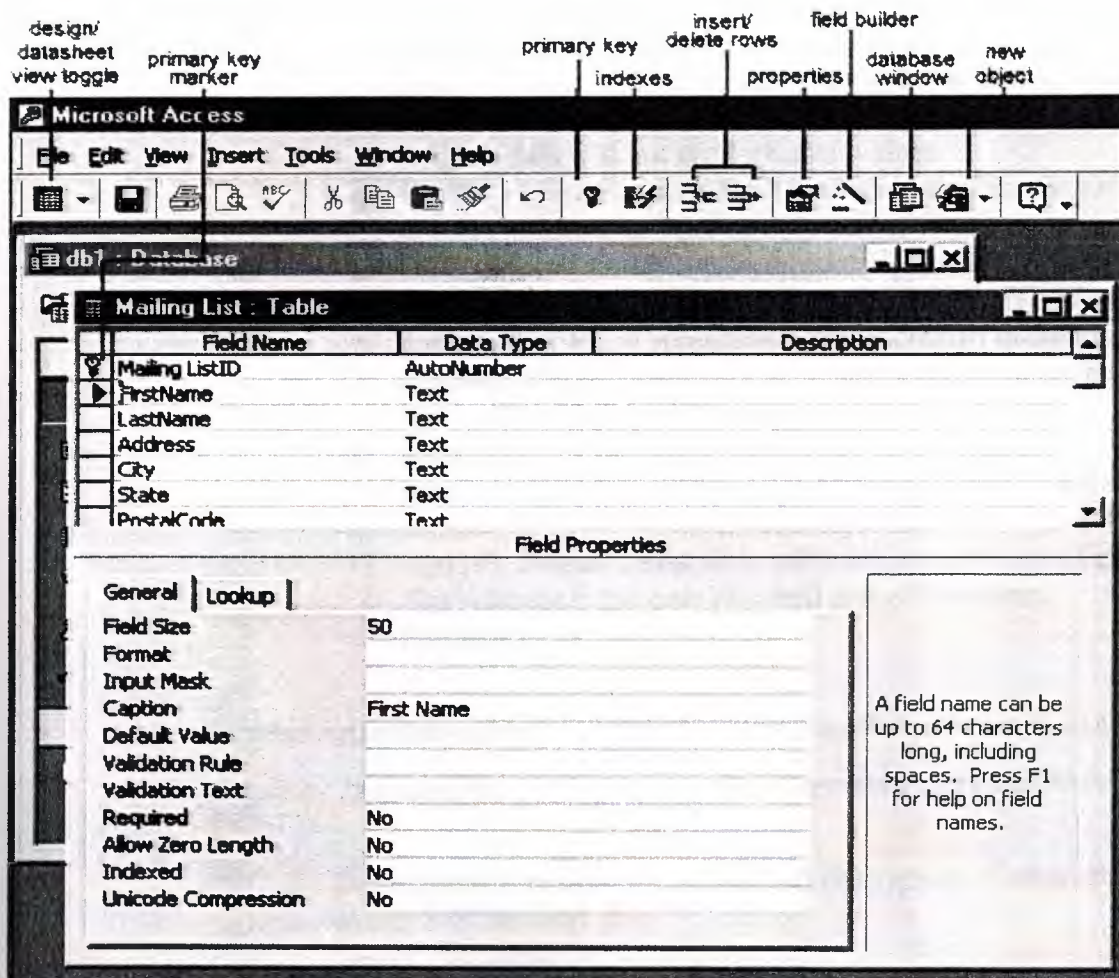


Figure 3.2.

3.2.1 Field Name

This is the name of the field and should represent the contents of the field such as "Name", "Address", "Final Grade", etc. The name can not exceed 64 characters in length and may include spaces.

1. **Data Type** is the type of value that will be entered into the fields.
 - a. **Text** - The default type, text type allows any combination of letters and numbers up to a maximum of 255 characters per field record.
 - b. **Memo** - A text type that stores up to 64,000 characters.
 - c. **Number** - Any number can be stored.

- d. **Date/Time** - A date, time, or combination of both.
- e. **Currency** - Monetary values that can be set up to automatically include a dollar sign (\$) and correct decimal and comma positions.
- f. **AutoNumber** - When a new record is created, Access will automatically assign a unique integer to the record in this field. From the General options, select Increment if the numbers should be assigned in order or random if any random number should be chosen. Since every record in a datasheet must include at least one field that distinguishes it from all others, this is a useful data type to use if the existing data will not produce such values.
- g. **Yes/No** - Use this option for True/False, Yes/No, On/Off, or other values that must be only one of two.
- h. **OLE Object** - An OLE (Object Linking and Embedding) object is a sound, picture, or other object such as a Word document or Excel spreadsheet that is created in another program. Use this data type to embed an OLE object or link to the object in the database.
- i. **Hyperlink** - A hyperlink will link to an Internet or Intranet site, or another location in the database. The data consists of up to four parts each separated by the pound sign (#): DisplayText#Address#SubAddress#ScreenTip. The Address is the only required part of the string.
Examples:

Internet hyperlink example: FGCU Home Page#http://www.fgcu.edu#

Database link example: #c:\My Documents\database.mdb#MyTable

- 2. **Description (optional)** - Enter a brief description of what the contents of the field are.

- 3. **Field Properties** - Select any pertinent properties for the field from the bottom pane.

3.3 Field Properties

Properties for each field are set from the bottom pane of the Design View window.

- 1. **Field Size** is used to set the number of characters needed in a text or number field. The default field size for the text type is 50 characters. If the records in the field will only have two or three characters, you can change the size of the field to save disk space or prevent entry errors by limiting the number of characters allowed. Likewise, if the field will require more than 50 characters, enter a number up to 255. The field size is set in exact characters for Text type, but options are give for numbers:

2. **Byte** - Positive integers between 1 and 255
3. **Integer** - Positive and negative integers between -32,768 and 32,768
4. **Long Integer (default)** - Larger positive and negative integers between -2 billion and 2 billion.
5. **Single** - Single-precision floating-point number
6. **Double** - Double-precision floating-point number
7. **Decimal** - Allows for Precision and Scale property control
8. **Format** conforms the data in the field to the same format when it is entered into the datasheet. For text and memo fields, this property has two parts that are separated by a semicolon. The first part of the property is used to apply to the field and the second applies to empty fields.
9. **Text and memo format.**

Table 3.1 Table Of Text Format.

Text Format			
Format	Datasheet Entry	Display	Explanation
@@@-@@@@	1234567	123-4567	@ indicates a required character or space
@@@-@@@@&	123456	123-456	& indicates an optional character or space
<	HELLO	hello	< converts characters to lowercase
>	hello	HELLO	> converts

			characters to uppercase
@!	Hello	Hello!	\ adds characters to the end
@;"No Data Entered"	Hello	Hello	
@;"No Data Entered"	(blank)	No Data Entered	

10. Number format. Select one of the preset options from the drop down menu or construct a custom format using symbols explained below:

Table 3.2. Table Of Number Format

Number Format			
Format	Datasheet Entry	Display	Explanation
###,##0.00	123456.78	123,456.78	0 is a placeholder that displays a digit or 0 if there is none.
\$###,##0.00	0	\$0.00	# is a placeholder that displays a digit or nothing if there is none.
###.00%	.123	12.3%	% multiplies the number by 100 and added a percent sign

11. Currency format. This formatting consists of four parts separated by semicolons:
format for positive numbers; format for negative numbers; format for zero values; format for Null values.

Table 3.3. Table Of Currency Format.

Currency Format	
Format	Explanation
\$##0.00;(\$##0.00)[Red];\$0.00;"none"	Positive values will be normal currency format, negative numbers will be red in parentheses, zero is entered for zero values, and "none" will be written for Null values.

12. Date format. In the table below, the value "1/1/01" is entered into the datasheet, and the following values are displayed as a result of the different assigned formats.

Table 3.4. Table Of Date Format.

Date Format		
Format	Display	Explanation
dddd","mmm d","yyyy	Monday, January 1, 2001	dddd, mmmm, and yyyy print the full day name,

		month name, and year
ddd", "mmm ". " d", "yy	Mon, Jan. 1, '01	ddd, mmm, and yy print the first three day letters, first three month letters, and last two year digits
"Today is " dddd	Today is Monday	
h:n:s: AM/PM	12:00:00 AM	"n" is used for minutes to avoid confusion with months

13. **Yes/No** fields are displayed as check boxes by default on the datasheet. To change the formatting of these fields, first click the Lookup tab and change the Display Control to a text box. Go back to the General tab choices to make formatting changes. The formatting is designated in three sections separated by semicolons. The first section does not contain anything but the semicolon must be included. The second section specifies formatting for Yes values and the third for No values.

Table 3.5. Table Of Yes/No Format

Yes/No Format	
Format	Explanation
;"Yes"[green];"No"[red]	Prints "Yes" in green or "No" in red

14. **Default Value** - There may be cases where the value of a field will usually be the same for all records. In this case, a changeable default value can be set to prevent typing the same thing numerous times. Set the Default Value property.

3.4 Primary Key

Every record in a table must have a primary key that differentiates it from every other record in the table. In some cases, it is only necessary to designate an existing field as the primary key if you are certain that every record in the table will have a different value for that particular field. A social security number is an example of a record whose values will only appear once in a database table.

Designate the primary key field by right-clicking on the record and selection **Primary Key** from the shortcut menu or select **Edit|Primary Key** from the menu bar. The primary key field will be noted with a key image to the left. To remove a primary key, repeat one of these steps.

If none of the existing fields in the table will produce unique values for every record, a separate field must be added. Access will prompt you to create this type of field at the

beginning of the table the first time you save the table and a primary key field has not been assigned. The field is named "ID" and the data type is "autonumber". Since this extra field serves no purpose to you as the user, the autonumber type automatically updates whenever a record is added so there is no extra work on your part. You may also choose to hide this column in the datasheet as explained on a later page in this tutorial.

3.5 Indexes

Creating indexes allows Access to query and sort records faster. To set an indexed field, select a field that is commonly searched and change the Indexed property to **Yes (Duplicates OK)** if multiple entries of the same data value are allowed or **Yes (No Duplicates)** to prevent duplicates.

3.6 Field Validation Rules

Validation Rules specify requirements (change word) for the data entered in the worksheet. A customized message can be displayed to the user when data that violates the rule setting is entered. Click the expression builder ("...") button at the end of the Validation Rule box to write the validation rule. Examples of field validation rules include ≥ 0 to not allow zero values in the record, and ??? to only all data strings three characters in length.

3.7 Input Masks

An input mask controls the value of a record and sets it in a specific format. They are similar to the Format property, but instead display the format on the datasheet before the data is entered. For example, a telephone number field can be formatted with an input mask to accept ten digits that are automatically formatted as "(555) 123-4567". The blank field would look like () - . An input mask to a field by following these steps:

1. In design view, place the cursor in the field that the input mask will be applied to.
2. Click in the white space following **Input Mask** under the **General** tab.
3. Click the "..." button to use the wizard or enter the mask, (@@@) @@@-@@@, into the field provided. The following symbols can be used to create an input mask from scratch:

Table 3.6. Table Of Input Mask Symbols

Input Mask Symbols	
Symbol	Explanation

A	Letter or digit
0	A digit 0 through 9 without a + or - sign and with blanks displayed as zeros
9	Same as 0 with blanks displayed as spaces
#	Same as 9 with +/- signs
?	Letter
L	Letter A through Z
C or &	Character or space
<	Convert letters to lower case
>	Convert letters to upper case

CHAPTER FOUR

DATASHEET RECORDS

4.1 Adding Records

Add new records to the table in datasheet view by typing in the record beside the asterisk (*) that marks the new record. You can also click the new record button at the bottom of the datasheet to skip to the last empty record.

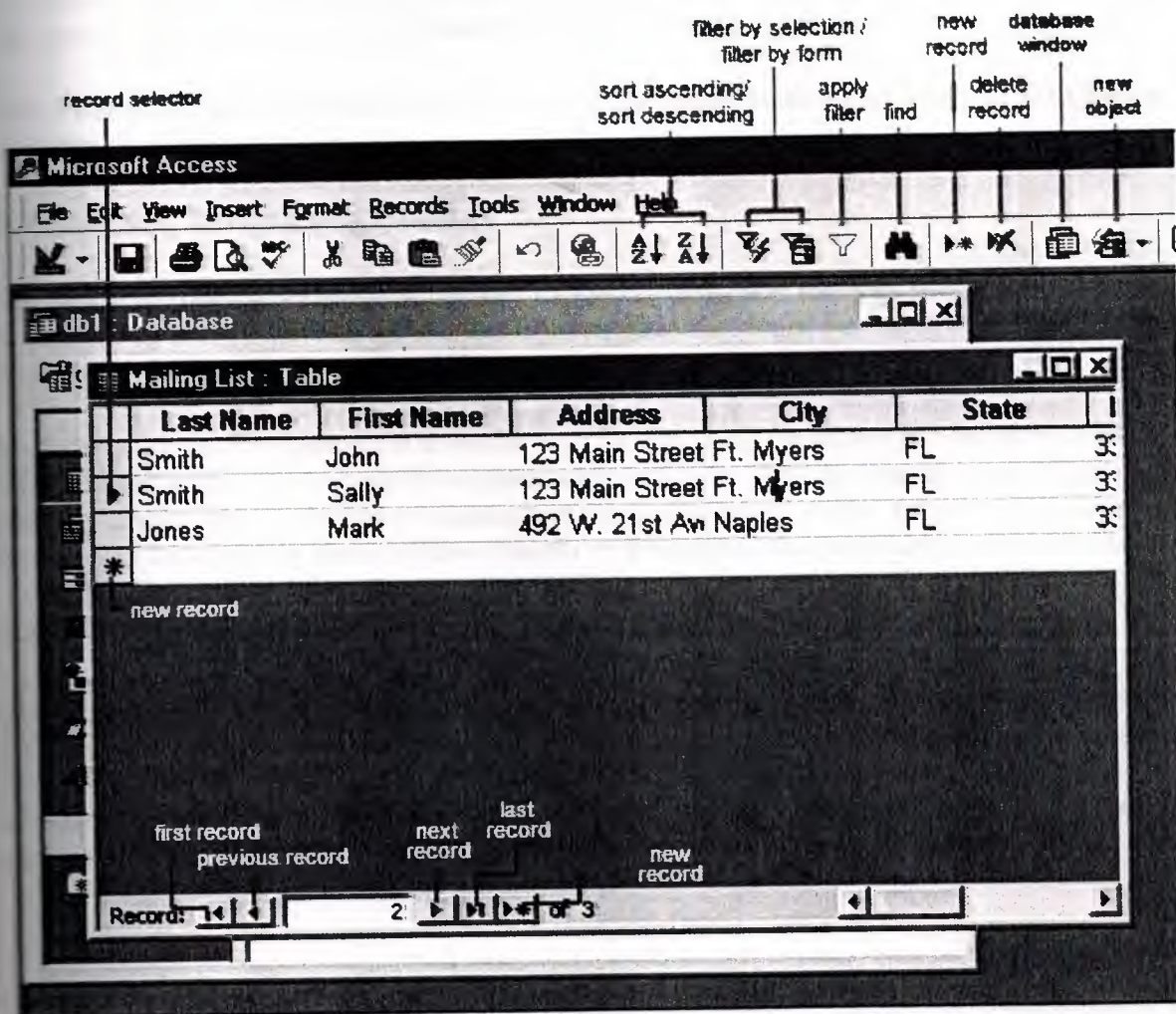


Figure 4.1.

4.2 Editing Records

To edit records, simply place the cursor in the record that is to be edited and make the necessary changes. Use the arrow keys to move through the record grid. The previous, next, first, and last record buttons at the bottom of the datasheet are helpful in maneuvering through the datasheet.

4.3 Deleting Records

Delete a record on a datasheet by placing the cursor in any field of the record row and select **Edit|Delete Record** from the menu bar or click the **Delete Record** button on the datasheet toolbar.

4.4 Adding and Deleting Columns

Although it is best to add new fields (displayed as columns in the datasheet) in design view because more options are available, they can also be quickly added in datasheet view. Highlight the column that the new column should appear to the left of by clicking its label at the top of the datasheet and select **Insert|Column** from the menu bar.

Entire columns can be deleted by placing the cursor in the column and selecting **Edit|Delete Column** from the menu bar.

4.5 Resizing Rows and Columns

The height of rows on a datasheet can be changed by dragging the gray sizing line between row labels up and down with the mouse. By changing the height on one row, the height of all rows in the datasheet will be changed to the new value.

Column width can be changed in a similar way by dragging the sizing line between columns. Double click on the line to have the column automatically fit to the longest value of the column. Unlike rows, columns on a datasheet can be different widths. More exact values can be assigned by selecting **Format|Row Height** or **Format|Column Width** from the menu bar.

4.6 Freezing Columns

Similar to freezing panes in Excel, columns on an Access table can be frozen. This is helpful if the datasheet has many columns and relevant data would otherwise not appear on the screen at the same time. Freeze a column by placing the cursor in any record in the column and select **Format|Freeze Columns** from the menu bar. Select the same option to unfreeze a single column or select **Format|Unfreeze All Columns**.

ID	First Name	Last Name	Home Phone	Work Phone
1	John	Smith	(941) 555-1234	(941) 555-1111
2	Sally	Smith	(941) 555-1234	(941) 555-4321
3	Mark	Jones	(941) 555-9942	(941) 555-2301

freeze columns

Record: 1 of 3

Figure 4.1. Table Of Mailing list Windows

4.7 Hiding Columns

Columns can also be hidden from view on the datasheet although they will not be deleted from the database. To hide a column, place the cursor in any record in the column or highlight multiple adjacent columns by clicking and dragging the mouse along the column headers, and select **Format|Hide Columns** from the menu bar.

To show columns that have been hidden, select **Format|Unhide Columns** from the menu bar. A window displaying all of the fields in the table will be listed with check boxes beside each field name. Check the boxes beside all fields that should be visible on the data table and click the **Close** button.

Unhide Columns

Column:

☒ ID

☐ First Name

☐ Last Name

☒ Address

☒ City

☒ State

☒ Postal Code

☒ Home Phone

☒ Work Phone

Close

Figure 4.2. Unhide Columns Windows.

4.8 Finding Data in a Table

Data in a datasheet can be quickly located by using the **Find** command.

1. Open the table in datasheet view.
2. Place the cursor in any record in the field that you want to search and select **Edit|Find...** from the menu bar.
3. Enter the value criteria in the **Find What:** box.
4. From the **Look In:** drop-down menu, define the area of the search by selecting the entire table or just the field in the table you placed your cursor in during step 2.
5. Select the matching criteria from **Match:** to and click the **More >>** button for additional search parameters.
6. When all of the search criteria is set, click the **Find Next** button. If more than one record meets the criteria, keep clicking **Find Next** until you reach the correct record.

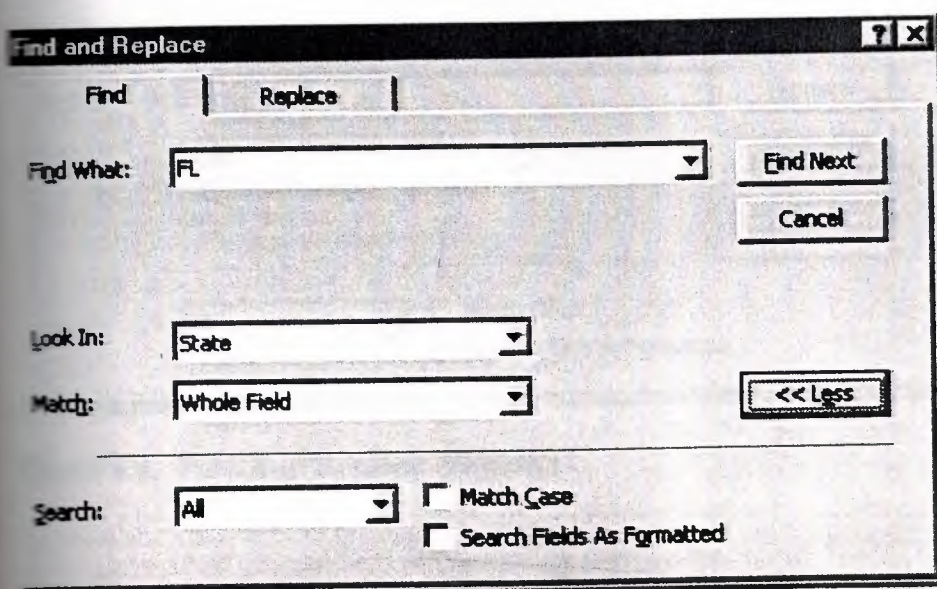


Figure 4.3. Find And Replace Windows

4.9 Replace

The replace function allows you to quickly replace a single occurrence of data with a new value or to replace all occurrences in the entire table.

1. Select **Edit|Replace...** from the menu bar (or click the **Replace** tab if the Find window is already open).
2. Follow the steps described in the Find procedure for searching for the data that should be replaced and type the new value of the data in the **Replace With:** box.
3. Click the **Find Next** button to step through occurrences of the data in the table and
4. click the **Replace** button to make single replacements. Click **Replace All** to change all occurrences of the data in one step.

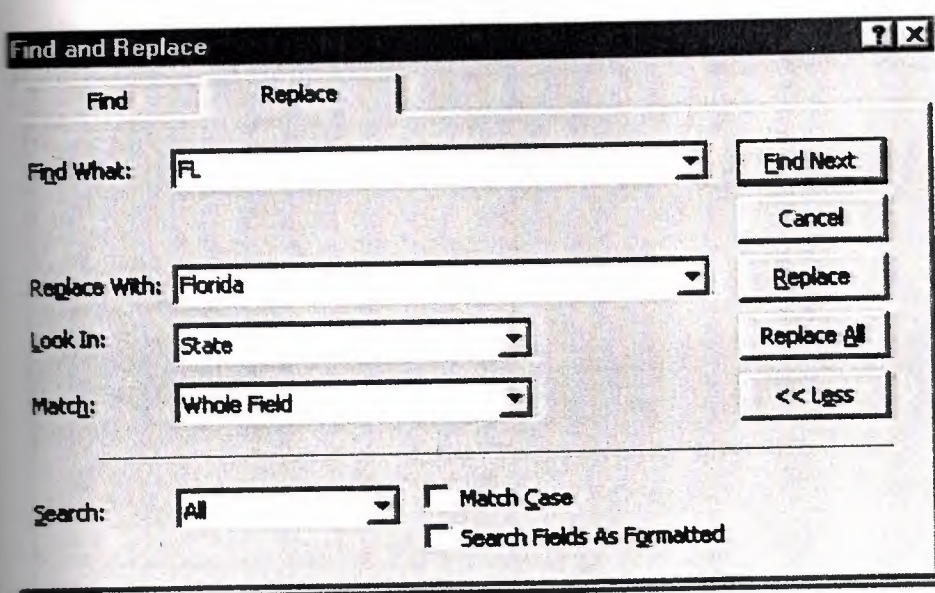


Figure 4.4. Find And Replace Windows

4.10 Check Spelling and AutoCorrect

The spell checker can be used to flag spelling errors in text and menu fields in a datasheet. Select **Tools|Spelling** from the menu bar to activate the spell checker and make corrections just as you would using Word or Excel. The AutoCorrect feature can automatically correct common spelling errors such as two INitial Capitals, capitalizing the first letter of the first word of a sentence, and anything you define. Select **Tools|AutoCorrect** to set these features.

4.11 Print a Datasheet


Datasheets can be printed by clicking the **Print** button on the toolbar or select **File|Print** to set more printing options.

CHAPTER FIVE

TABLE RELATIONSHIPS

5.1 Table Relationships

To prevent the duplication of information in a database by repeating fields in more than one table, table relationships can be established to link fields of tables together. Follow the steps below to set up a relational database:

1. Click the **Relationships** button on the toolbar. 
2. From the **Show Table** window (click the **Show Table** button on the toolbar to make it appear), double click on the names of the tables you would like to include in the relationships. When you have finished adding tables, click **Close**.

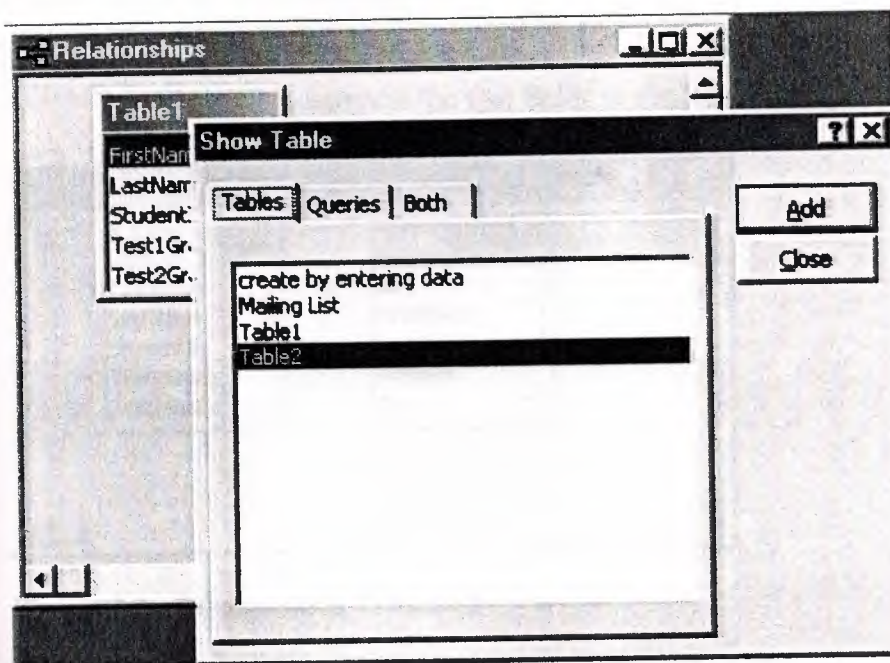


Figure 5.1. Adding Tables Into Relationship Windows

3. To link fields in two different tables, click and drag a field from one table to the corresponding field on the other table and release the mouse button. The **Edit Relationships** window will appear. From this window, select different fields if necessary and select an option from **Enforce Referential Integrity** if necessary. These options give Access permission to automatically make changes to referential tables if key records in one of the tables is deleted. Check the **Enforce Referential Integrity** box to ensure that the relationships are valid and that the data is not accidentally deleted when data is added, edited, or deleted. Click **Create** to create the link.

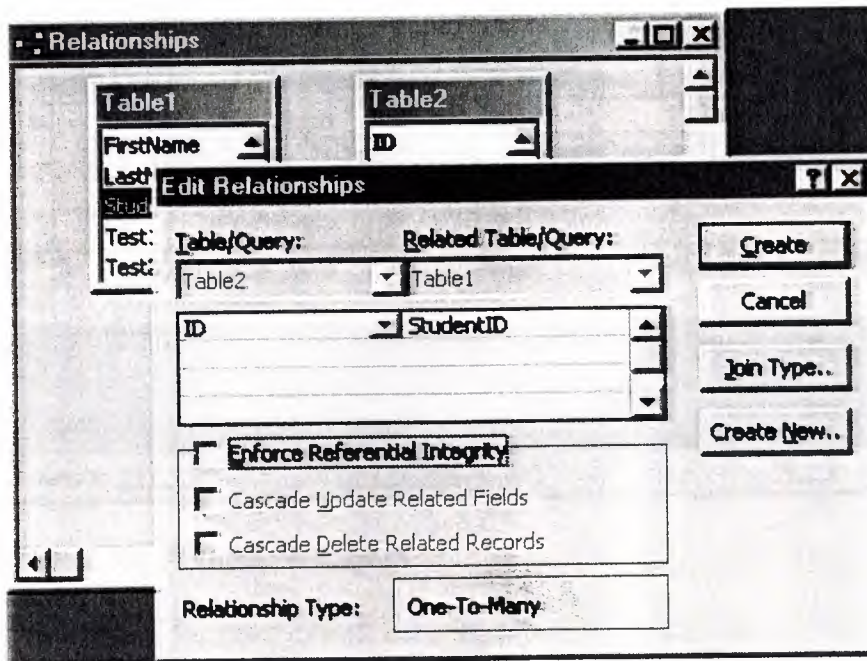


Figure 5.2. Edit Relationship Windows

1. A line now connects the two fields in the Relationships window.

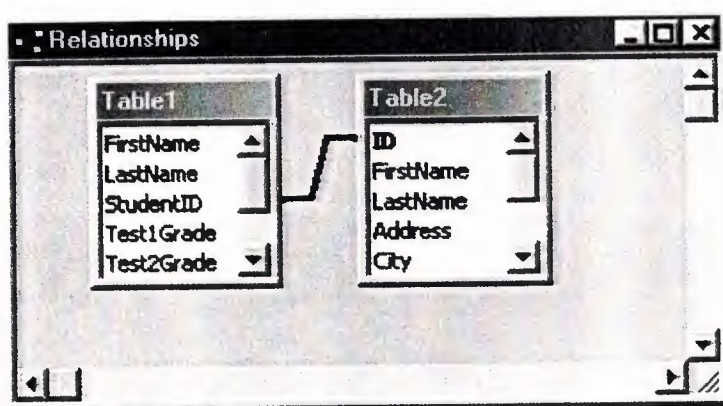


Figure 5.3. Relationship Windows

2. The datasheet of a relational table will provide expand and collapse indicators to view subdatasheets containing matching information from the other table. In the example below, the student address database and student grade database were related and the two can be shown simultaneously using the expand feature. To expand or collapse all subdatasheets at once, select **Format|Subdatasheet|Expand All** or **Collapse All** from the toolbar

CHAPTER SIX

QUERY

6.1 Introduction to Queries

Queries select records from one or more tables in a database so they can be viewed, analyzed, and sorted on a common datasheet. The resulting collection of records, called a **dynaset** (short for dynamic subset), is saved as a database object and can therefore be easily used in the future. The query will be updated whenever the original tables are updated. Types of queries are *select queries* that extract data from tables based on specified values, *find duplicate* queries that display records with duplicate values for one or more of the specified fields, and *find unmatched* queries display records from one table that do not have corresponding values in a second table.

6.2 Create A Query in Design View

Follow these steps to create a new query in Design View:

From the Queries page on the Database Window, click the **New** button.

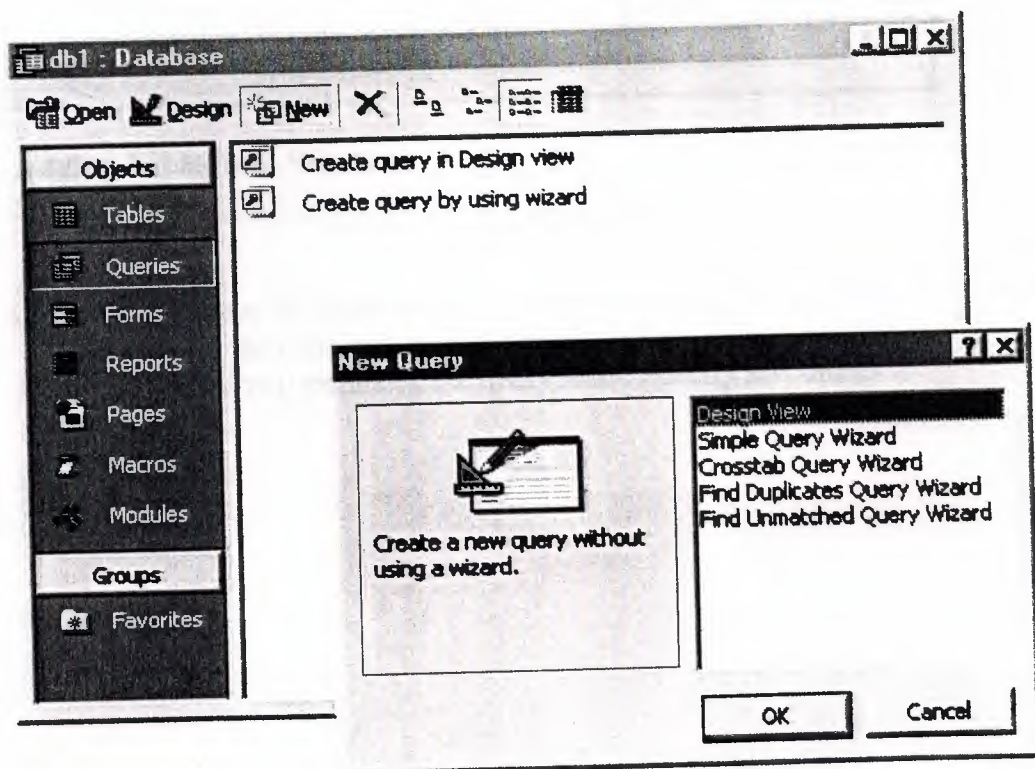


Figure 6.1. Database Windows

1. Select Design View and click OK.

2. Select tables and existing queries from the **Tables** and **Queries** tabs and click the **Add** button to add each one to the new query.
3. Click **Close** when all of the tables and queries have been selected.

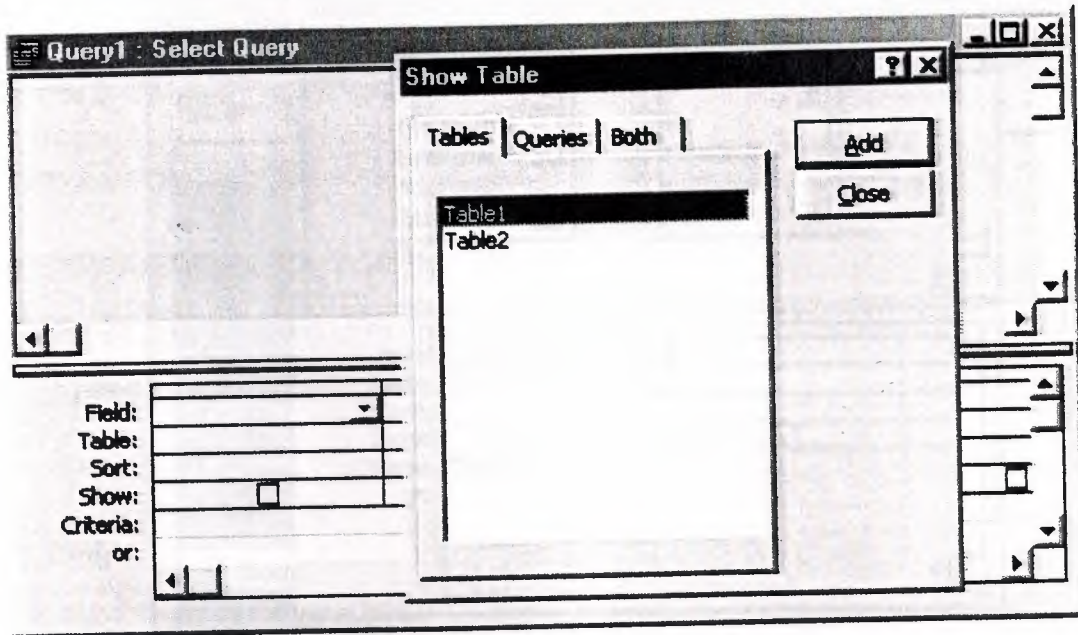


Figure 6.2. Adding Tables Into Windows

4. Add fields from the tables to the new query by double-clicking the field name in the table boxes or selecting the field from the **Field:** and **Table:** drop-down menus on the query form. Specify sort orders if necessary.

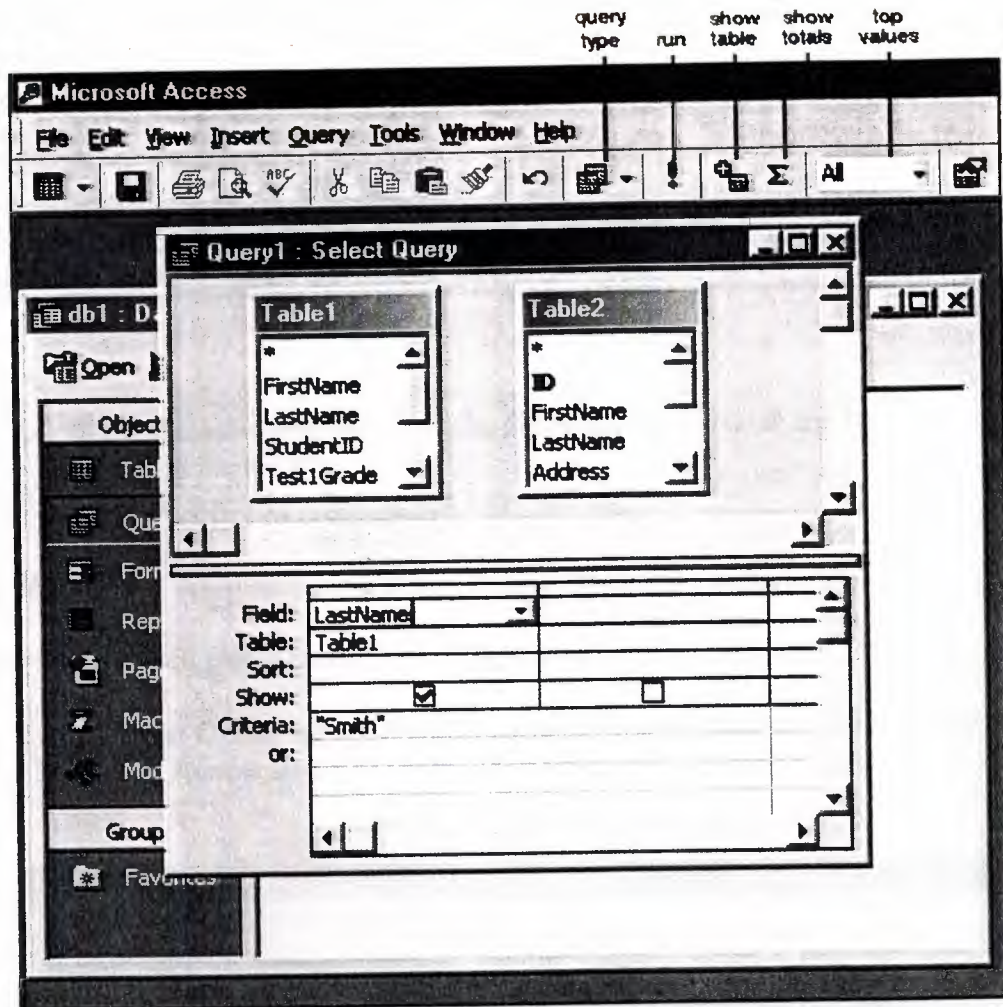


Figure 6.3. Select Query Windows


5. Enter the criteria for the query in the **Criteria:** field. The following table provides examples for some of the wildcard symbols and arithmetic operators that may be used. The **Expression Builder**  can also be used to assist in writing the expressions.

Table 6.1. Query Wildcards and Expression Operators

Query Wildcards and Expression Operators	
Wildcard / Operator	Explanation
? Street	The question mark is a wildcard that takes the place of a single letter.
43th *	The asterisk is the wildcard that represents a number of characters.
<100	Value less than 100
>=1	Value greater than or equal to 1

<>"FL"	Not equal to (all states besides Florida)
Between 1 and 10	Numbers between 1 and 10
Is Null	Finds records with no value
Is Not Null	or all records that have a value
Like "a*"	All words beginning with "a"
>0 And <=10	All numbers greater than 0 and less than 10
"Bob" Or "Jane"	Values are Bob or Jane

6. After you have selected all of the fields and tables, click the **Run** button on the toolbar.
7. Save the query by clicking the **Save** button.

6.3 Query Wizard

Access' Query Wizard will easily assist you to begin creating a select query.

1. Click the **Create query by using wizard** icon in the database window to have Access step you through the process of creating a query.

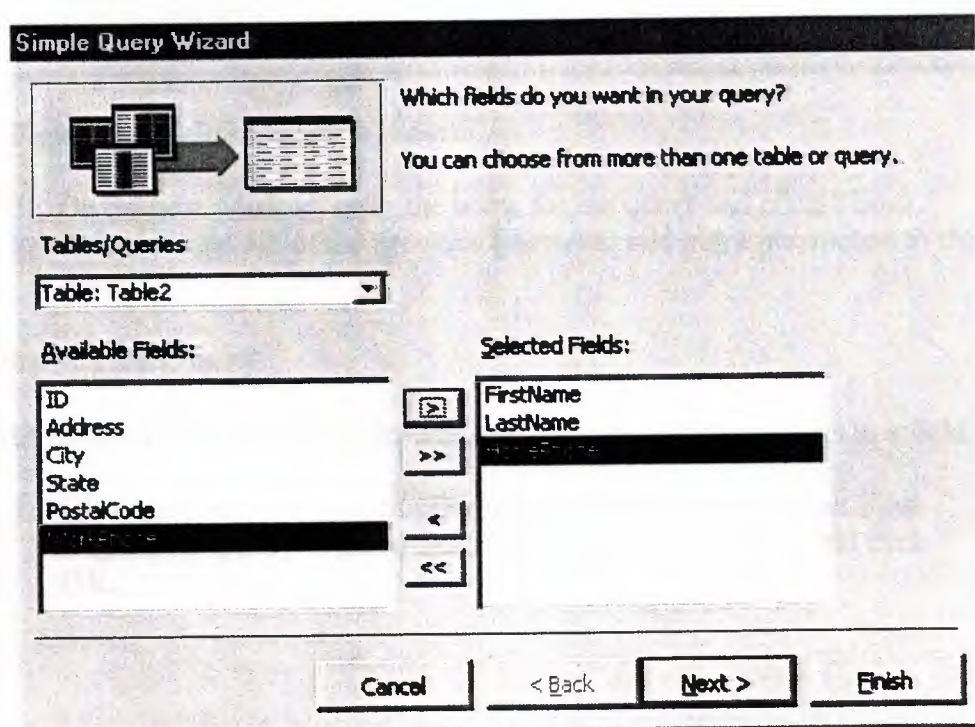


Figure 6.4. Simple Query Wizard

2. From the first window, select fields that will be included in the query by first selecting the table from the drop-down **Tables/Queries** menu. Select the fields by clicking the **>** button to move the field from the Available Fields list to Selected Fields. Click the double arrow button

>> to move all of the fields to Selected Fields. Select another table or query to choose from more fields and repeat the process of moving them to the Selected Fields box. Click **Next >** when all of the fields have been selected.

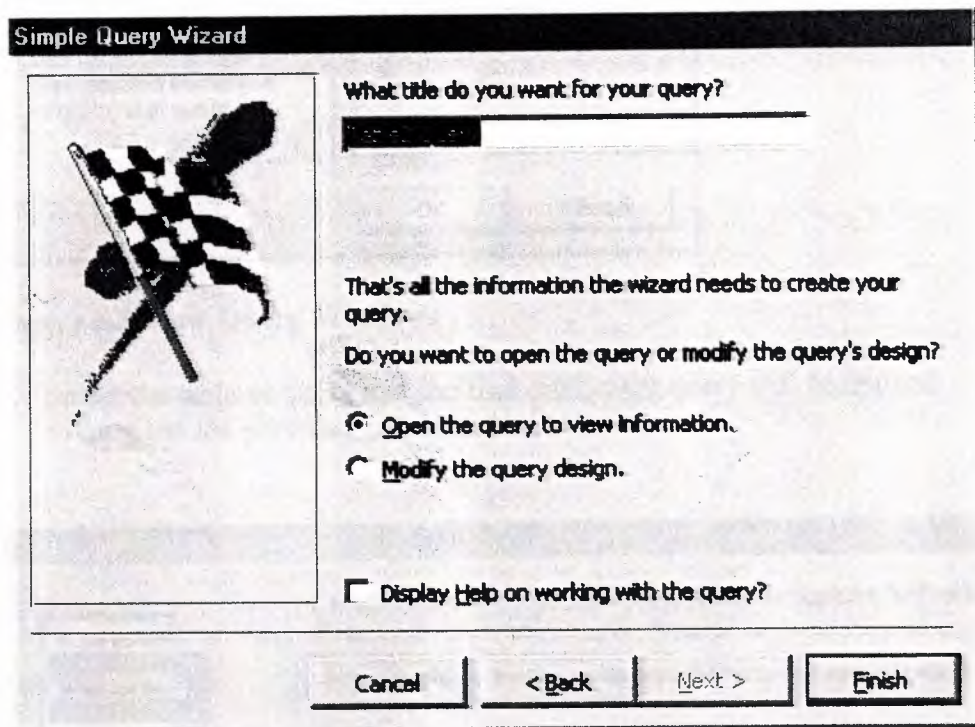


Figure 6.4. Select Query Windows

3. On the next window, enter the name for the query and click **Finish**.
4. Refer to steps 5-8 of the previous tutorial to add more parameters to the query.

6.4 Find Duplicates Query

This query will filter out records in a single table that contain duplicate values in a field.

1. Click the **New** button on the Queries database window, select **Find Duplicates Query Wizard** from the **New Query** window and click **OK**.

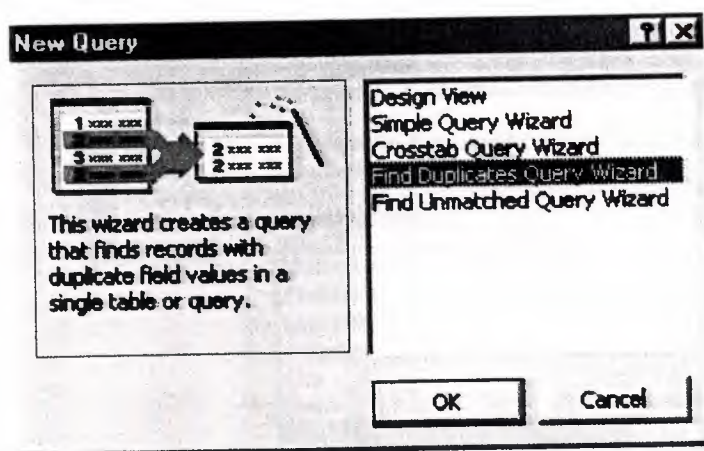


Figure 6.5. New Query Windows

2. Select the table or query that the find duplicates query will be applied to from the list provided and click **Next >**.

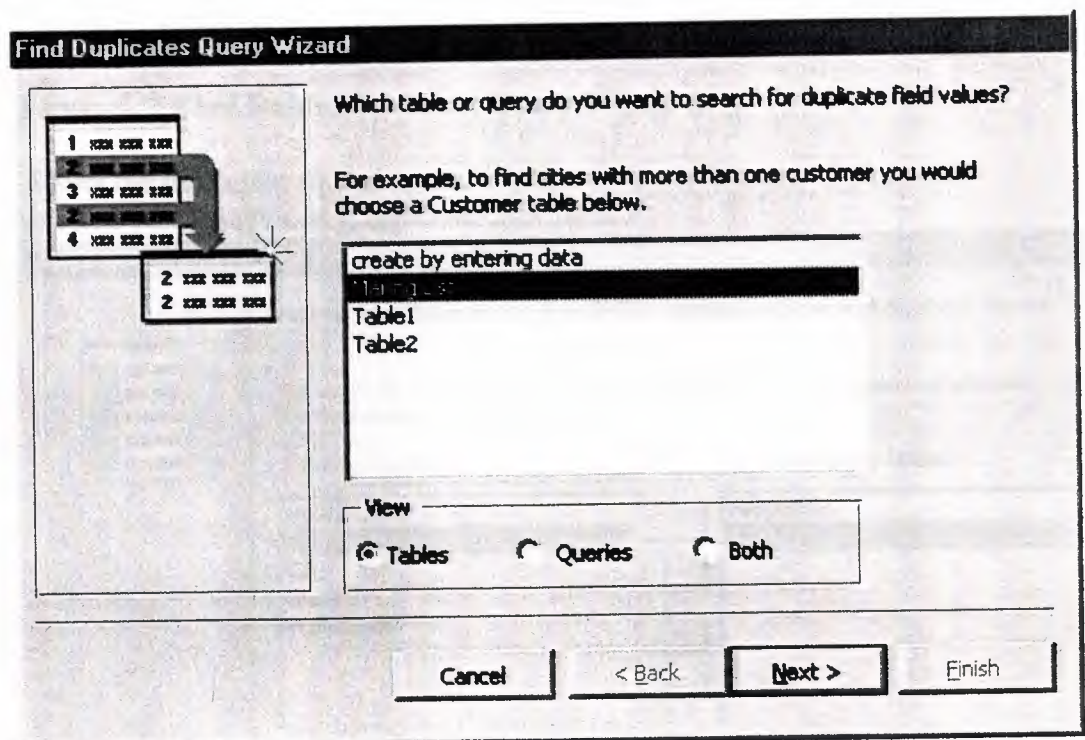


Figure 6.6. Find Duplicates Select Query Windows

3. Select the fields that may contain duplicate values by highlighting the names in the Available fields list and clicking the > button to individually move the fields to the Duplicate-value fields list or >> to move all of the fields. Click **Next >** when all fields have been selected.

Find Duplicates Query Wizard

Which fields might contain duplicate information?

For example, if you are looking for cities with more than one customer, you would choose City and Region fields here.

Available fields:

Mailing ListID	>
FirstName	>>
LastName	>>
Address	<
City	<<
State	
PostalCode	

Duplicate-value fields:

HomePhone

Cancel < Back **Next >** Finish

Figure 6.7. Find Duplicates Select Query Windows

Select the fields that should appear in the new query along with the fields selected on the previous screen and click **Next >**.

Find Duplicates Query Wizard

Do you want the query to show fields in addition to those with duplicate values?

For example, if you chose to look for duplicate City values, you could choose CustomerName and Address here.

Available fields:

Mailing ListID	>
Address	>>
City	<
State	<<
PostalCode	

Additional query fields:

FirstName
CustomerName

Cancel < Back **Next >** Finish

Figure 6.8. Find Duplicates Select Query Windows

4. Name the new query and click **Finish**.

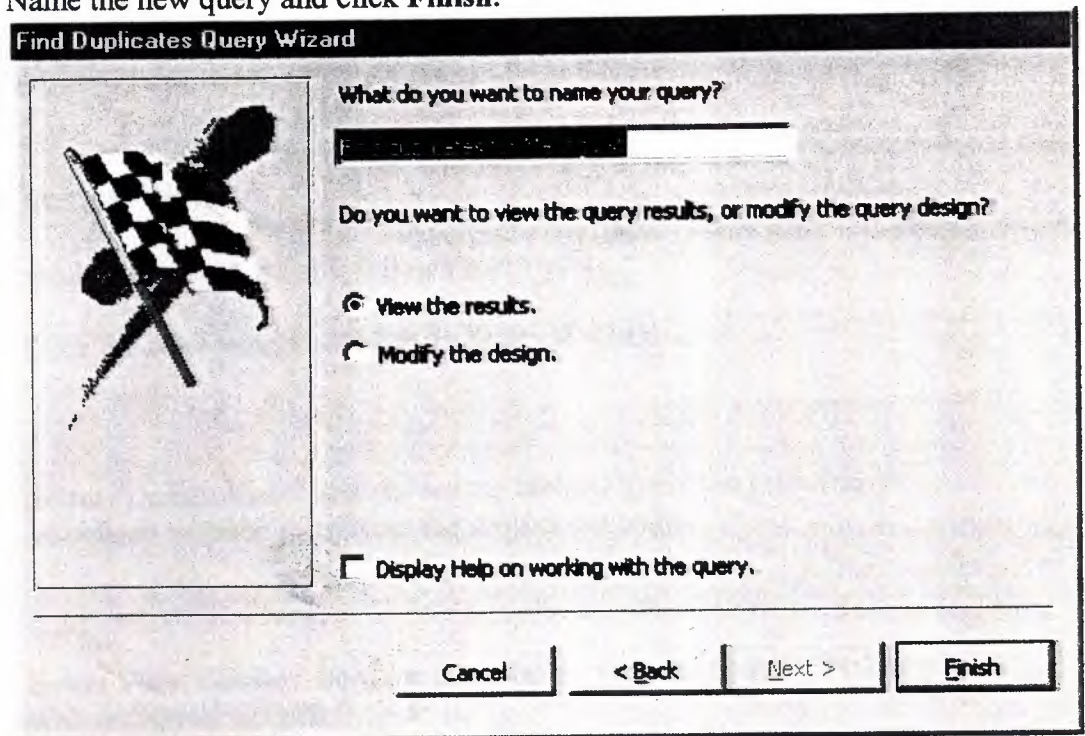


Figure 6.9. Find Duplicates Select Query Windows

6.5 Delete a Query

To delete a table from the query, click the table's title bar and press the **Delete** key on the keyboard.

CHAPTER SEVEN

FORM

7.1 Create Form in Design View

To create a form from scratch without the wizard, follow these steps:

1. Click the **New** button on the form database window.
2. Select "Design View" and choose the table or query the form will be associated with the form from the drop-down menu.
3. Select **View|Toolbox** from the menu bar to view the floating toolbar with additional options.

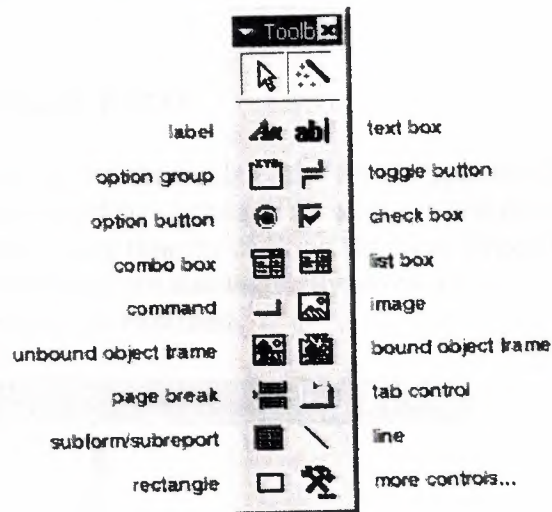


Figure 7.1. ToolBox Windows

4. Add controls to the form by clicking and dragging the field names from the Field List floating window. Access creates a text box for the value and label for the field name when this action is accomplished. To add controls for all of the fields in the Field List, double-click the Field List window's title bar and drag all of the highlighted fields to the form.

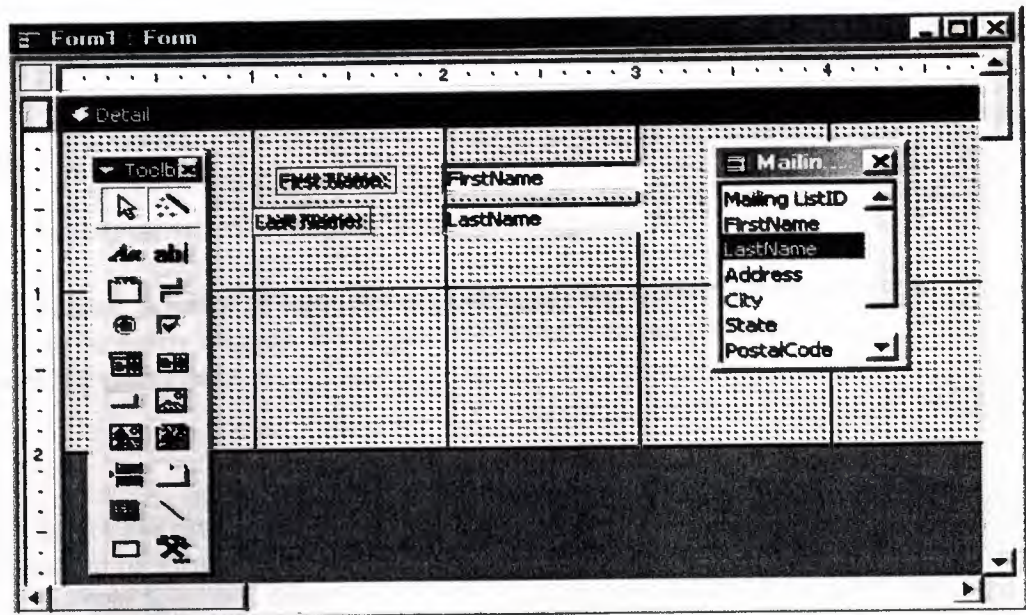



Figure 7.2. Form Design Windows

7.2 Adding Records Using A Form

Input data into the table by filling out the fields of the form. Press the **Tab** key to move from field to field and create a new record by clicking **Tab** after the last field of the last record. A new record can also be created at any time by clicking the **New Record** button  at the bottom of the form window. Records are automatically saved as they are entered so no additional manual saving needs to be executed.

StudentID	
Test1Grade	80
Test2Grade	95
Test3Grade	90
CourseAverage	0
ExtraCredit	No

Record: 1 of 3

Figure 7.3 Find Duplicates Select Query Windows

7.3 Editing Forms

The follow points may be helpful when modifying forms in Design View.

1. **Grid lines** - By default, a series of lines and dots underlay the form in Design View so form elements can be easily aligned. To toggle this feature on and off select **View|Grid** from the menu bar.
2. **Snap to Grid** - Select **Format|Snap to Grid** to align form objects with the grid to allow easy alignment of form objects or uncheck this feature to allow objects to float freely between the grid lines and dots.
3. **Resizing Objects** - Form objects can be resized by clicking and dragging the handles on the edges and corners of the element with the mouse.
4. **Change form object type** - To easily change the type of form object without having to create a new one, right click on the object with the mouse and select **Change To** and select an available object type from the list.
5. **Label/object alignment** - Each form object and its corresponding label are bounded and will move together when either one is moved with the mouse. However, to change the position of the object and label in relation to each other (to move the label closer to a text box, for example), click and drag the large handle at the top, left corner of the object or label.
6. **Tab order** - Alter the tab order of the objects on the form by selecting **View|Tab Order...** from the menu bar. Click the gray box before the row you would like to change in the tab order, drag it to a new location, and release the mouse button.

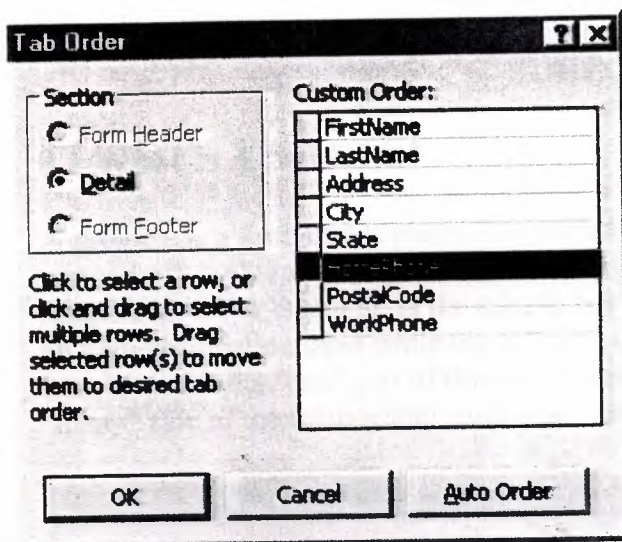


Figure 7.5. Tab Order Windows

7. **Form Appearance** - Change the background color of the form by clicking the **Fill/Back Color** button on the formatting toolbar and click one of the color swatches on the palette. Change the color of individual form objects by highlighting one and selecting a color from the **Font/Fore Color** palette on the formatting toolbar. The font and size, font effect, font alignment, border around each object, the border width, and a special effect can also be modified using the formatting toolbar:

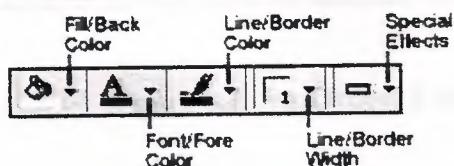


Figure 7.6.

8. **Page Header and Footer** - Headers and footers added to a form will only appear when it is printed. Access these sections by selecting **View|Page Header/Footer** on the menu bar. Page numbers can also be added to these sections by selecting **Insert|Page Numbers**. A date and time can be added from **Insert|Date and Time....** Select **View|Page Header/Footer** again to hide these sections from view in Design View.

CHAPTER EIGHT

SUBFORM

8.1 What Is A Subform?

A subform is a form that is placed in a parent form, called the main form. Subforms are particularly useful to display data from tables and queries that have one-to-many relationships. For example, in the sample below, data on the main form is drawn from an item information table while the subform contains all of the orders for that item. The item record is the "one" part of this one-to-many relationship while the orders are the "many" side of the relationship since many orders can be placed for the one item.

The screenshot shows a window titled 'frmOrdersTestSub'. It contains a main form with three fields: 'ItemNo', 'Description', and 'UnitPrice'. The 'ItemNo' field contains 'AP653', 'Description' contains 'Pencil #2', and 'UnitPrice' contains '\$5.00'. Below these fields is a subform with a table structure. The table has four columns: 'OrderNo', 'ItemNo', 'Quantity', and 'Total'. The first row shows '00001', 'AP653', '10', and '\$50.00'. The second row shows '00002', 'AP653', '8', and '\$40.00'. Below the table is a record navigation bar with 'Record: 1 of 2'. At the bottom of the window is another record navigation bar with 'Record: 1 of 3'.

ItemNo	Description	UnitPrice
AP653	Pencil #2	\$5.00

OrderNo	ItemNo	Quantity	Total
00001	AP653	10	\$50.00
00002	AP653	8	\$40.00

Record: 1 of 2

Record: 1 of 3

Figure 8.1. frmOrdersTestSub Windows

The remainder of this page explains three methods for creating subforms and they assume that the data tables and/or queries have already been created.

8.2 Create a Form and Subform at Once

Use this method if neither form has already been created. A main form and subform can be created automatically using the form wizard if **table relationships** are set properly or if a query involving multiple tables is selected. For example, a relationship can be set between a table containing customer information and one listing customer orders so the orders for each customer are displayed together using a main form and subform. Follow these steps to create a subform within a form:

1. Double-click **Create form by using wizard** on the database window.

2. From the **Tables/Queries** drop-down menu, select the first table or query from which the main form will display its data. Select the fields that should appear on the form by highlighting the field names in the **Available Fields** list on the left and clicking the single arrow > button or click the double arrows >> to choose all of the fields.

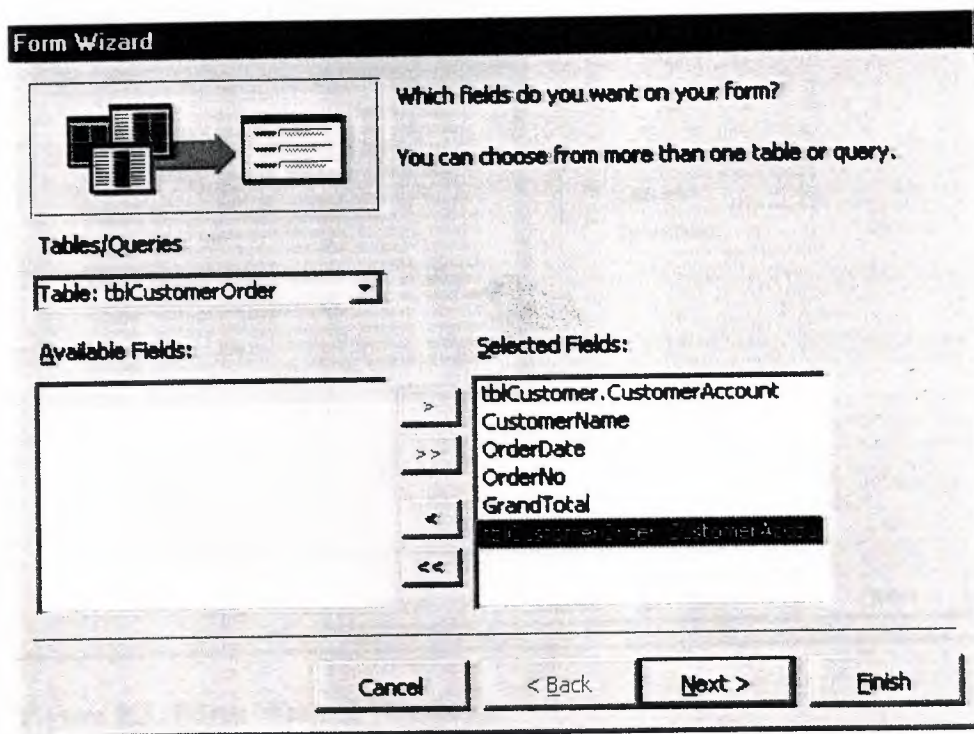


Figure 8.2. Form Wizard Windows

3. From the same window, select another table or query from the **Tables/Queries** drop-down menu and choose the fields that should appear on the form. Click **Next** to continue after all fields have been selected.
4. Choose an arrangement for the forms by selecting **form with subform(s)** if the forms should appear on the same page or **Linked forms** if there are many controls on the main form and a subform will not fit. Click **Next** to proceed to the next page of options.

5. Select a tabular or datasheet layout for the form and click **Next**.

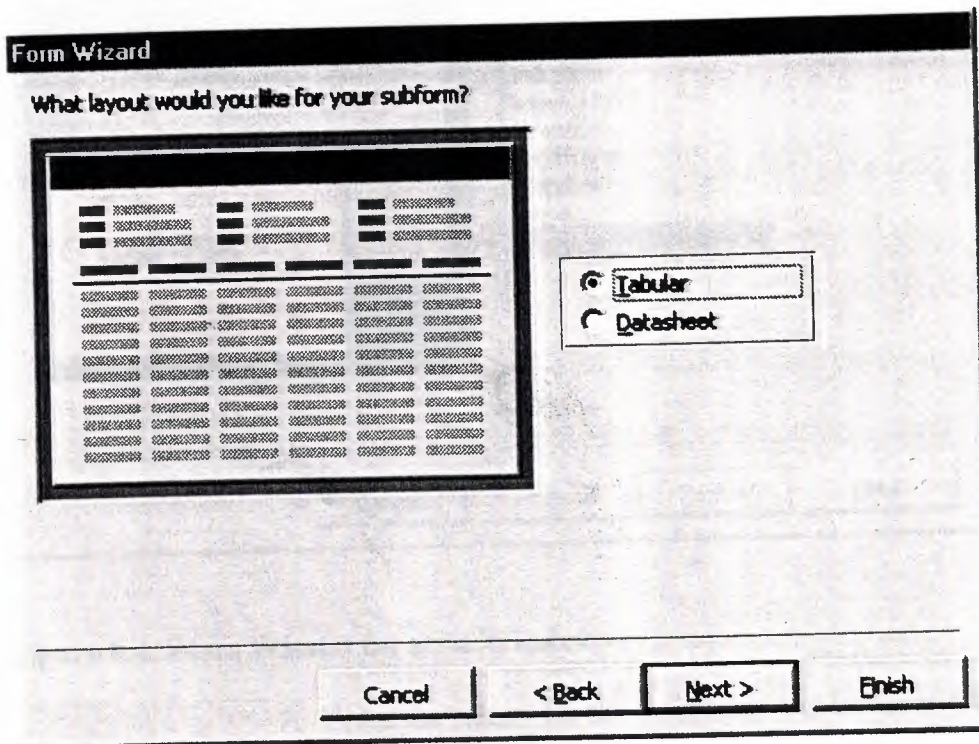


Figure 8.3. Form Wizard Windows

6. Select a style for the form and click **Next**.

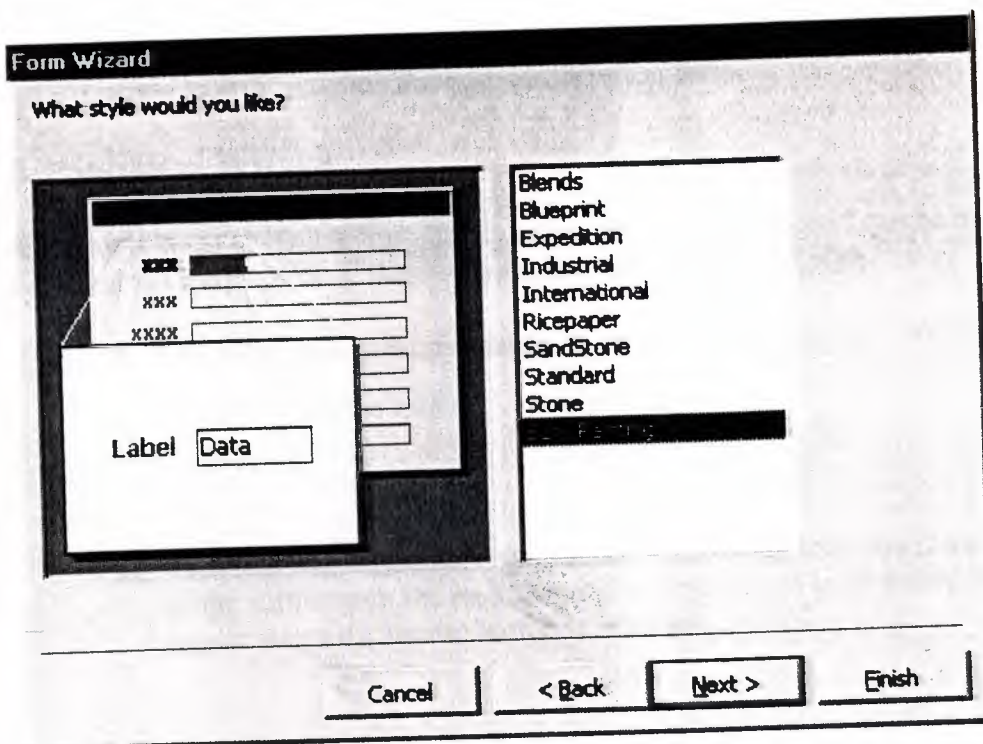


Figure 8.4. Form Wizard for style Windows

7. Enter the names for the main form and subform. Click **Finish** to create the forms.

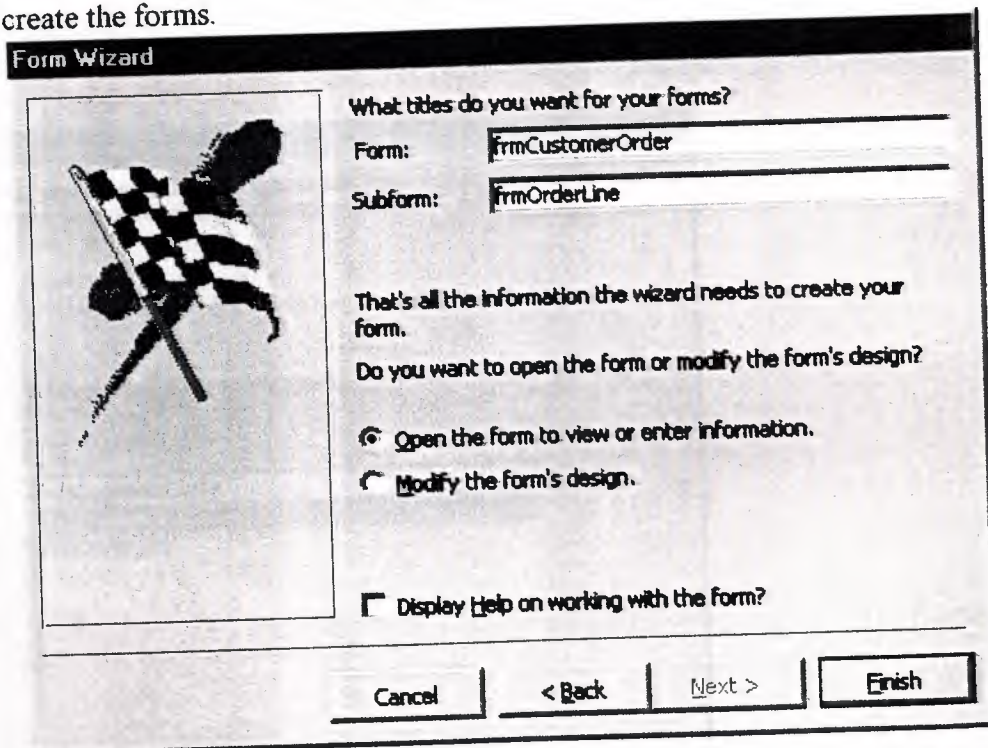




Figure 8.5. Form Wizard Windows

8. New records can be added to both tables or queries at once by using the new combination form.

8.3 Subform Wizard

If the main form or both forms already exist, the Subform Wizard can be used to combine the forms. Follow these steps to use the Subform Wizard:

1. Open the main form in **Design View** and make sure the **Control Wizard** button  on the toolbox is pressed in.
2. Click the **Subform/Subreport** icon  on the toolbox and draw the outline of the subform on the main form. The Subform Wizard dialog box will appear when the mouse button is released.
3. If the subform has not been created yet, select "Use existing Tables and Queries". Otherwise, select the existing form that will become the subform. Click **Next** to continue

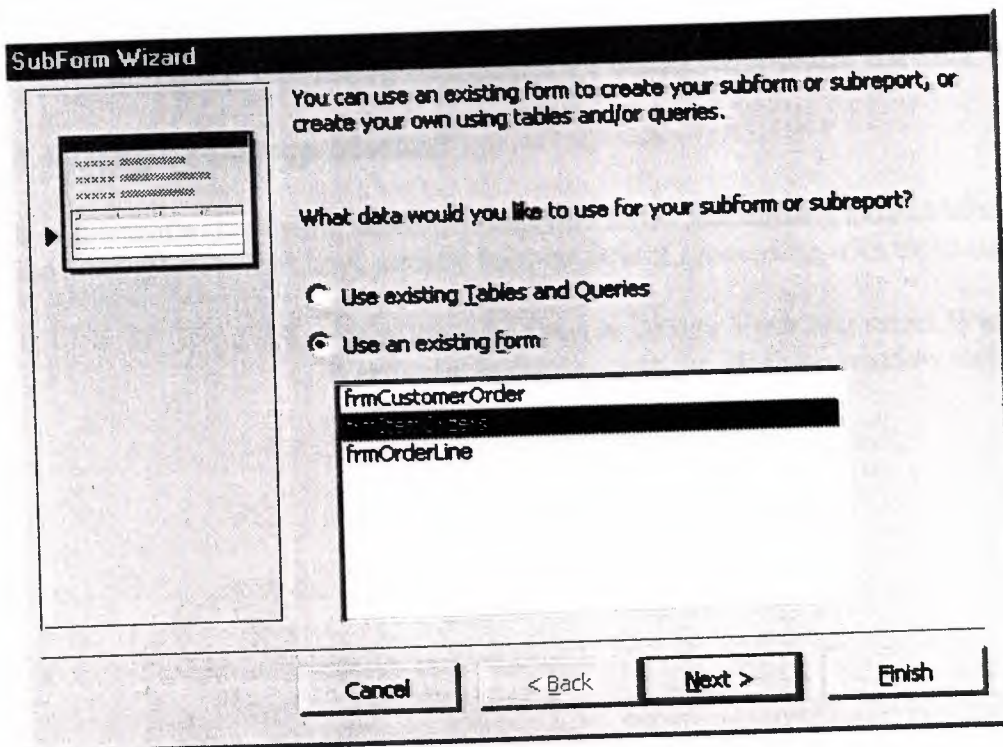


Figure 8.6. SubForm Wizard Windows

4. The next dialog window will display table relationships assumed by Access. Select one of these relationships or define your own and click **Next**

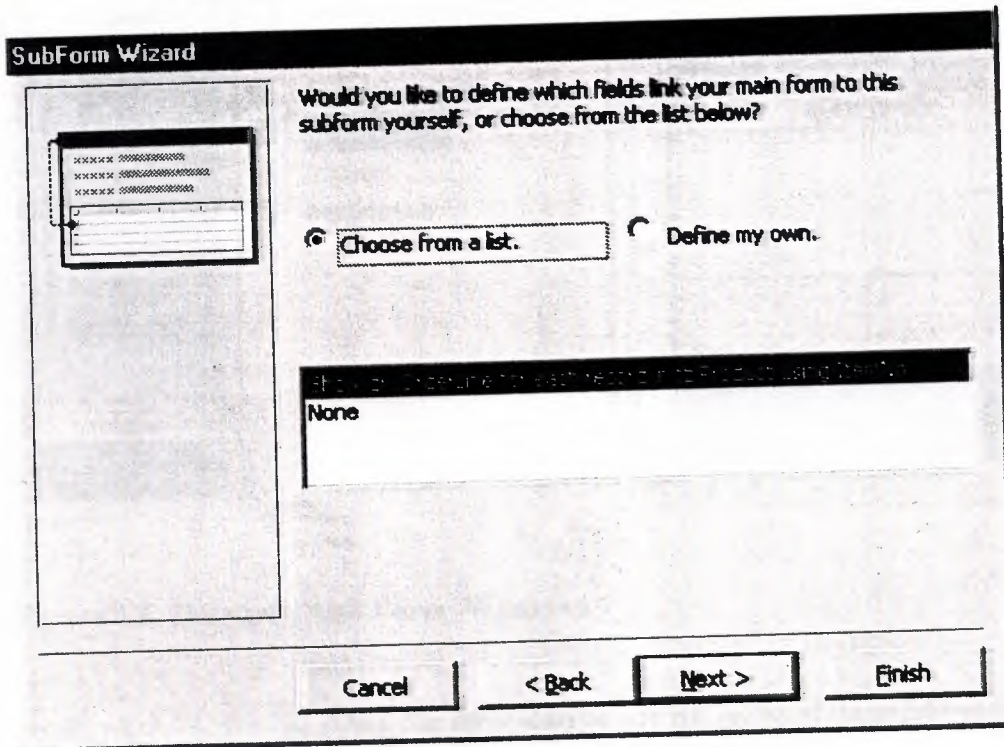


Figure 8.7. SubForm Wizard Windows

5. On the final dialog box, enter the name of the subform and click **Finish**.

8.4 Drag-and-Drop Method

Use this method to create subforms from two forms that already exist. Make sure that the table relationships have already been set before proceeding with these steps.

1. Open the main form in **Design View** and select **Window|Tile Vertically** to display both the database window and the form side-by-side.

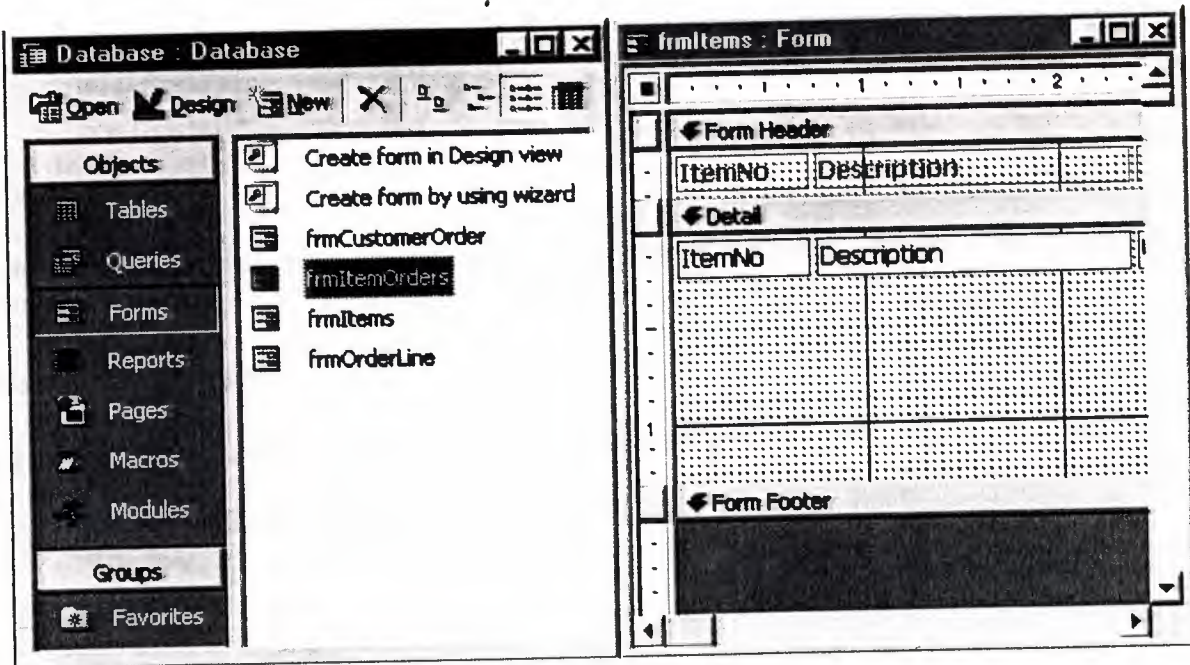


Figure 8.8. Database And Form Windows

2. Drag the form icon beside the name of the subform onto the detail section of the main form design

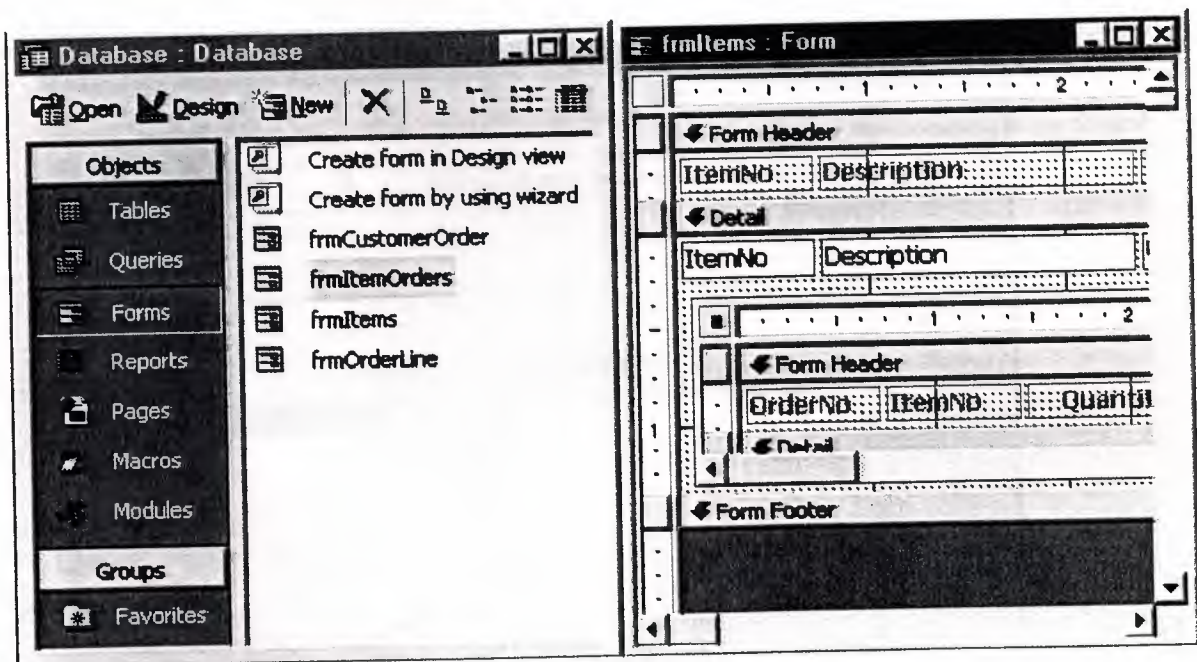


Figure 8.9. Database And Form Windows

CHAPTER NINE

REPORTS

9.1 Reports

Reports will organize and group the information in a table or query and provide a way to print the data in a database.

9.2 Using the Wizard

Create a report using Access' wizard by following these steps:

1. Double-click the "Create report by using wizard" option on the Reports Database Window.
2. Select the information source for the report by selecting a table or query from the **Tables/Queries** drop-down menu. Then, select the fields that should be displayed in the report by transferring them from the **Available Fields** menu to the **Selected Fields** window using the single right arrow button > to move fields one at a time or the double arrow button >> to move all of the fields at once. Click the **Next >** button to move to the next screen.

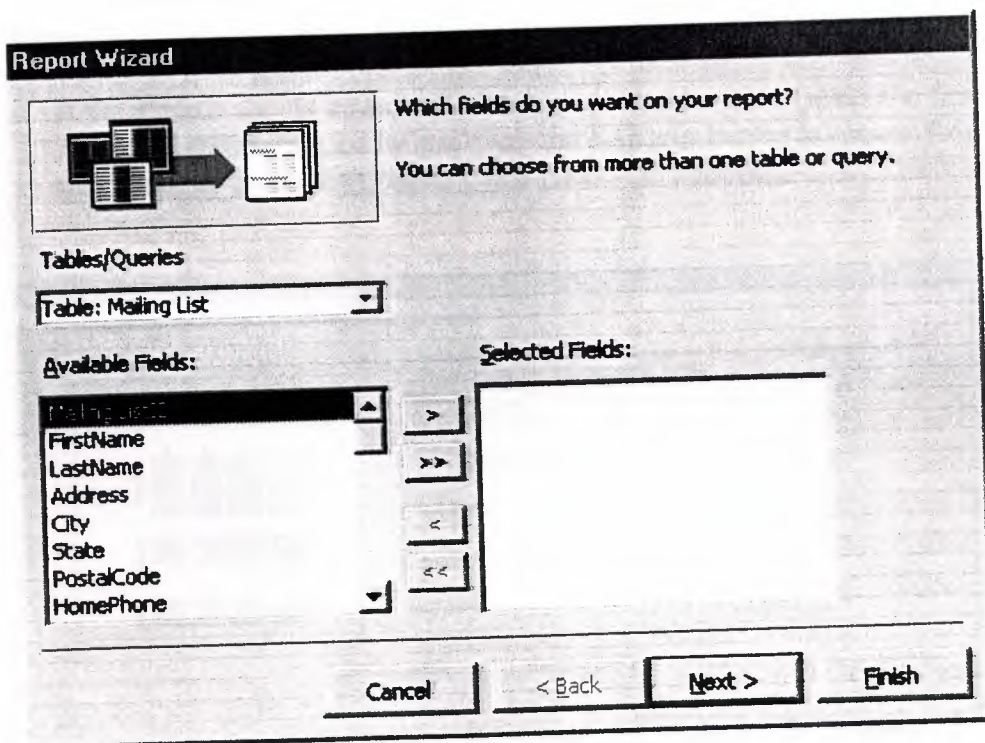


Figure 9.1. Report Wizard Windows

3. Select fields from the list that the records should be grouped by and click the right arrow button > to add those fields to the diagram. Use the **Priority** buttons to change the order of the grouped fields if more than one field is selected. Click **Next >** to continue.

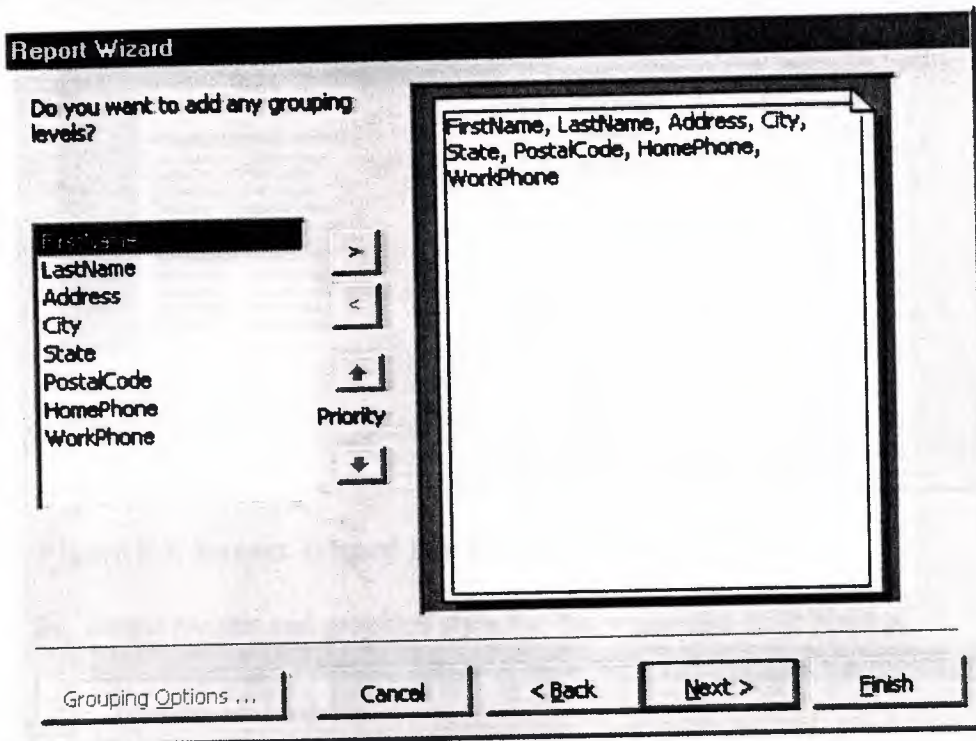


Figure 9.2. Report Wizard For Grouping Levels

4. If the records should be sorted, identify a sort order here. Select the first field that records should be sorted by and click the A-Z sort button to choose from ascending or descending order. Click **Next >** to continue.

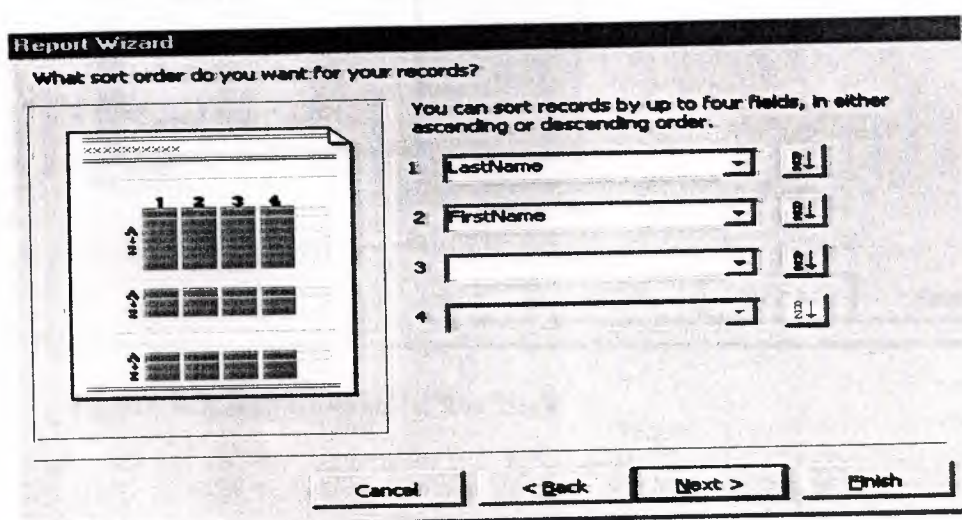


Figure 9.3. Report Wizard For Records

5. Select a layout and page orientation for the report and click **Next >**.

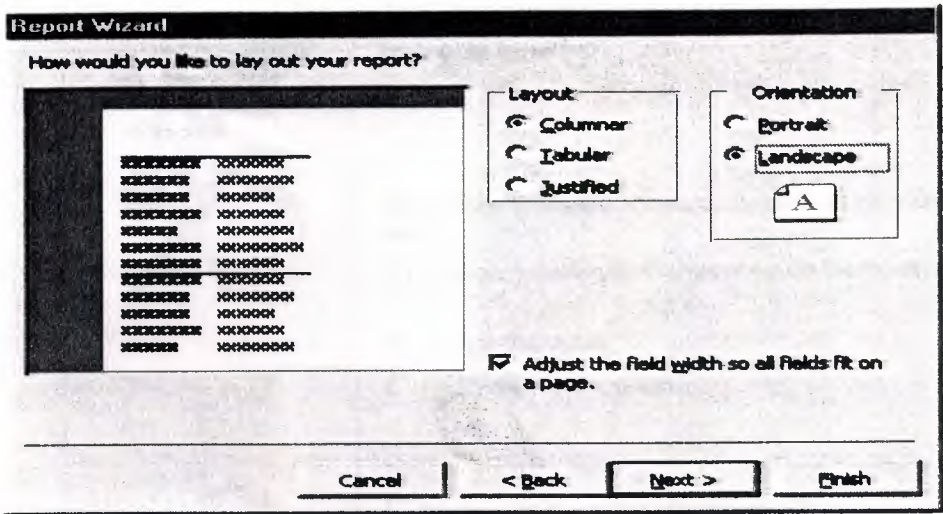


Figure 9.4. Report Wizard For Layout

6. Select a color and graphics style for the report and click **Next >**.

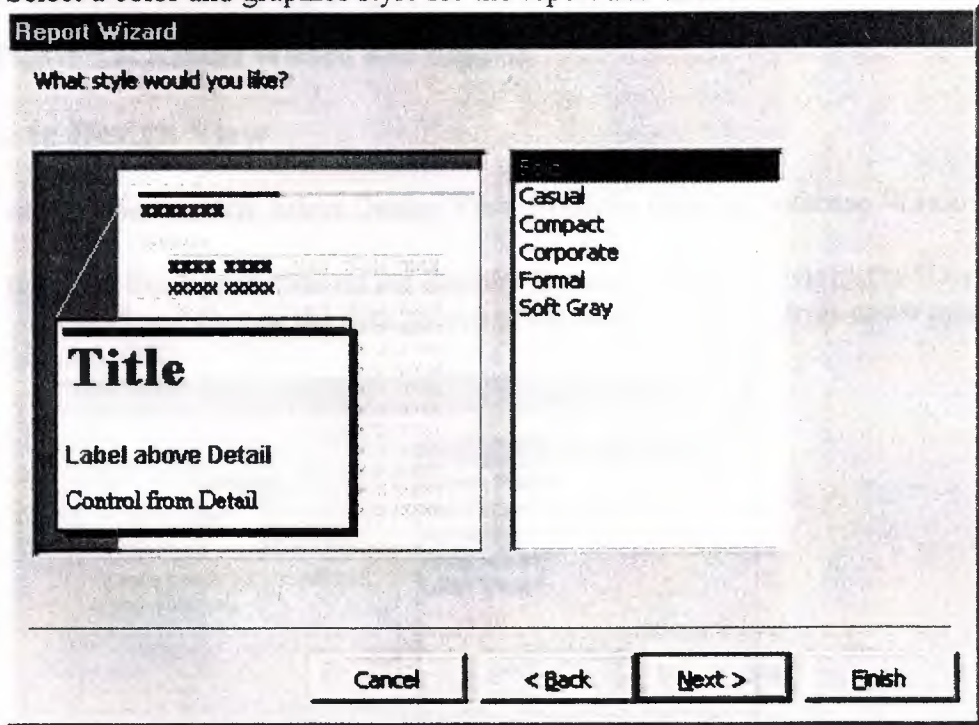


Figure 9.5. Report Wizard For Style

7. On the final screen, name the report and select to open it in either Print Preview or Design View mode. Click the **Finish** button to create the report.

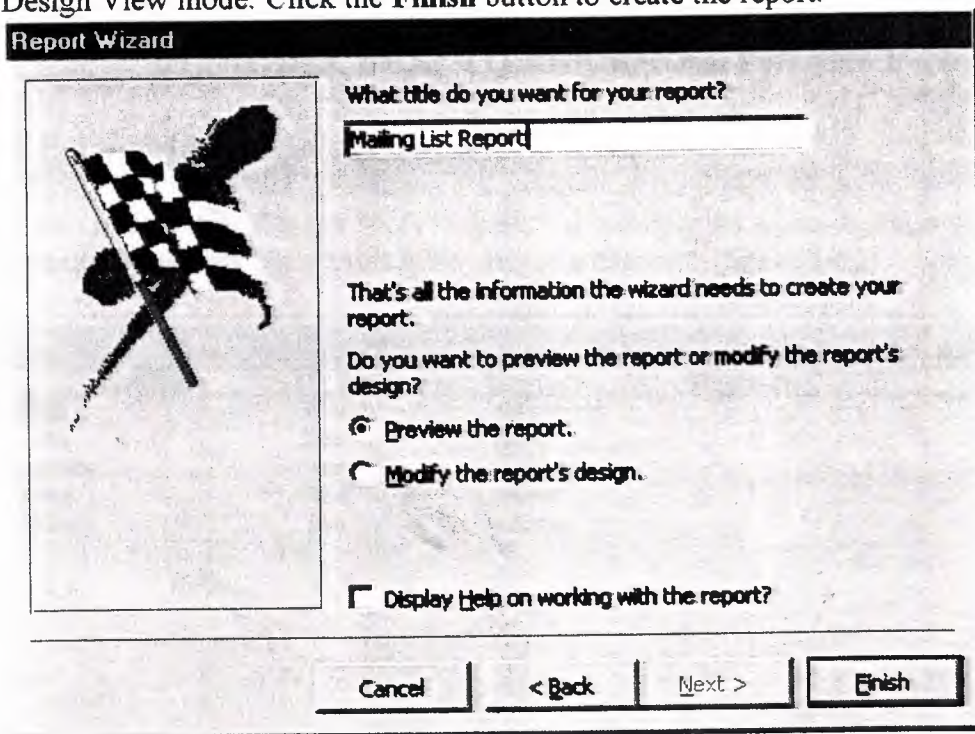


Figure 9.6. Report Wizard For Reports

9.3 Create in Design View

To create a report from scratch, select Design View from the Reports Database Window.

1. Click the **New** button on the Reports Database Window. Highlight "Design View" and choose the data source of the report from the drop-down menu and click **OK**.

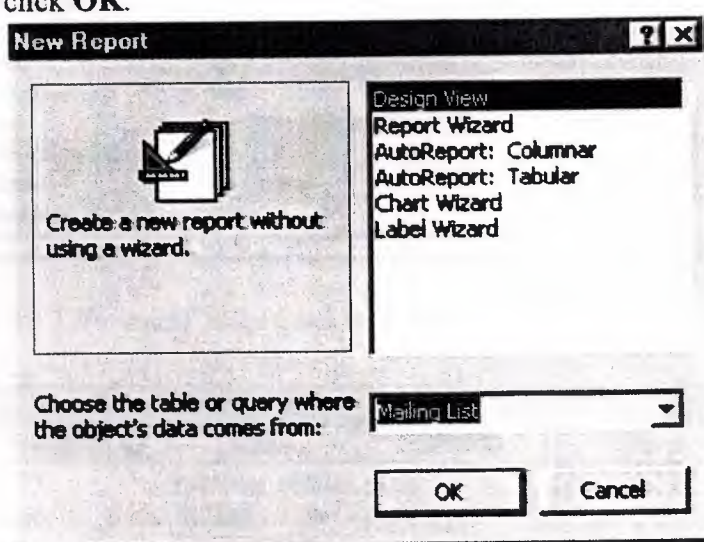


Figure 9.7. New Report Windows

CHAPTER TEN

INTRODUCTION TO PERSONAL FINANCE

10.1 Personal Information Table

Here we create a table (figure 10.1) for personal information when we enter some information about any person this table creates a database (figure 10.2)

Field Name	Data Type	Description
personal id	Text	personal id
name	Text	Name
surname	Text	Surname
phone	Text	phone
address	Text	address

Field Properties	
General Lookup	
Field Size	10
Format	
Input Mask	
Caption	Personal ID
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None

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

Figure 10.1 Personal Information Table

Personal ID	name	Surname	Phone	Address
+ 11	mehmet alitt6t6	yildiz	(212) 212 21 22	
+ 2354	ISLAM	TURFENT		GFGF
+ 323	hakan	yilmaz		dsdag
+ 566666	5565	656		

Record: 2 of 4

Figure 10.2 Personal Information Database

10.2 Desing of Personal Finance Form

From Database (Access 2000 file format) windows we can desing a new form. After entering desing view we can prepeare form (**figure 10.3**)

The screenshot shows the Microsoft Access 2000 Form Designer interface. The title bar reads 'personInfTBL : Form'. The form is designed on a grid. At the top, there are sections for 'Form Header' and 'Detail'. Below these, the form is divided into two main sections: 'PERSONAL INFORMATION' and 'FINANCE INFORMATION'. The 'PERSONAL INFORMATION' section contains five text boxes with labels: 'Personal ID' (with value 'pend'), 'name', 'Surname', 'Phone', and 'Address'. The 'FINANCE INFORMATION' section contains two text boxes labeled 'EXPENCES' and 'REVENUE'. A 'Toolbox' is open on the right side of the form, showing various controls like text boxes, labels, and buttons. The form is currently in Design View.

Figure 10.3 Desing of Personal Information Form

After desing we will have following form (figure 10.4).

personInfTBL

PERSONAL INFORMATION

Personal ID: 11

name: mehmet ali

Surname: yildiz

Phone: [212] 212 21 22

Address:

FINANCE INFORMATION

EXPENCES REVENUE

figure 10.4. user information window

10.3 Personal ID

Every user have to enter a personal number for taking a place or for having his/her own information.

10.4 Name, Surname, Phone,Address

Information about user Name, Surname, Phone and Address can be entered in this section.

After entering information, user will enter informatin about his/her expences or revenues with these buttons:

10.5 Expences Button

With this button you can see expences form (figure 10.5).

EXPENCES

EXPENCES IN A MOUNTH

PERSONAL ID

INSURANCE NO

ELECTRIC

WATER

RENT

INSURANCE

TAX

EXTRA NAME IN ONE MOUNTH

EXTRA AMOUNT IN ONE MOUNTH

EXPENCES IN A YEAR

TAX IN YEAR

EXTRA NAME IN ONE YEAR

EXTRA AMOUNT IN ONE YEAR

INSURANCE IN YEAR

DATE

TOTAL

ADD **DELETE** **SEARCH**

Figure 10.5 Expences Windows.

10.5.1 Microsoft Visual Basic Codes For Expences Button

Option Compare Database

Private Sub Command18_Click()

DoCmd.OpenForm "EXPENCES"

End Sub

10.6 Revenue Buton

With this button you can see revenue form (figure 10.6).

REVENUES

REVENUES IN ONE MOUNTH REVENUES IN ONE YEAR

PERSON ID:

RENTS

SOLARY

EXTRA NAME

EXTRA AMOUNT

EXTRA NAME

EXTRA AMOUNT

DATE

TOTAL

ADD DELETE SEARCH

Figure 10.6 Revenue Form.

10.6.1 Microsoft Visual Basic Codes For Revenue Button

Option Compare Database

Private Sub Command20_Click()

DoCmd.OpenForm "REVENUETBL"

End Sub

CHAPTER ELEVEN

INTRODUCTION TO EXPENCES

11.1. Expences Table

Here we create a table (**figure 11.1**) for personal expences information when we enter some information about any person's expences this table creates a database (**figure 11.2**)

EXPENCES : Table			
	Field Name	Data Type	Description
<input checked="" type="checkbox"/>	INSURANCE NO	Text	INSURANCE NO
<input type="checkbox"/>	Elec	Number	ELECTRIC IN A MOUNTH
<input type="checkbox"/>	Water	Number	WATER IN A MOUNTH
<input type="checkbox"/>	Rent	Number	RENT IN A MOUNTH
<input type="checkbox"/>	Ins	Number	INSURANCE IN A MOUNTH
<input type="checkbox"/>	Tax	Number	TAX IN A MOUNTH
<input type="checkbox"/>	ExtMoNAME	Text	NAME OF EXTRA IN A MOUNTH
<input type="checkbox"/>	ExtMoAmount	Number	AMOUNT OF EXTRA IN A MOUNTH
<input type="checkbox"/>	ExtYeNAME	Text	NAME OF EXTRA IN YEAR
<input type="checkbox"/>	ExtYeAmount	Number	AMOUNT OF EXTRA IN YEAR
<input type="checkbox"/>	InsYE	Number	INSURANCE IN YEAR
<input type="checkbox"/>	TaxYE	Number	TAX IN YEAR
<input type="checkbox"/>	DATE	Date/Time	DATE
<input type="checkbox"/>	ID	Number	

Field Properties	
General Lookup	
Field Size	50
Format	
Input Mask	
Caption	INSURANCE NO
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None

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

Figure 11.1 Personal Expences Information Table

EXPENCES : Table						
	INSURANCE N	ELECTRIC	WATER	RENT	INSURANCE	TAX
1		0	0	0	0	0
111		0	0	0	0	0
11222		0	0	0	0	0
233534		3230	32760	320	230	320
433		0	0	0	0	0
*		0	0	0	0	0

Figure 11.2 Personal Expences Information database

11.2 Design of Expences Form

From Database (Access 2000 file format) windows we can desing a new form. After entering desing view we can prepeare form (figure 11.1)

Figure 11.3 Design of Expences Form.

After desing we will have following form (figure 11.4).

EXPENCES	
EXPENCES IN A MOUNTH	EXPENCES IN A YEAR
PERSONAL ID <input type="text"/>	TAX IN YEAR <input type="text" value="0"/>
INSURANCE NO <input type="text" value="1"/>	EXTRA NAME IN ONE YEAR <input type="text"/>
ELECTRIC <input type="text" value="0"/>	EXTRA AMOUNT IN ONE YEAR <input type="text"/>
WATER <input type="text" value="0"/>	<input type="text" value="0"/>
RENT <input type="text" value="0"/>	INSURANCE IN YEAR <input type="text"/>
INSURANCE <input type="text" value="0"/>	<input type="text" value="0"/>
TAX <input type="text" value="0"/>	DATE <input type="text" value="26.06.2003"/>
EXTRA NAME IN ONE MOUNTH <input type="text"/>	
EXTRA AMOUNT IN ONE MOUNTH <input type="text" value="0"/>	
TOTAL <input type="text"/>	
<input type="button" value="ADD"/> <input type="button" value="DELETE"/> <input type="button" value="SEARCH"/>	

Figure 11.4 Expences form.

Here there are sections for information of cost.

11.3 Expences In A Mount

In **Expences In A Mounth** side user can be entered his/her expences in a mounth. For example cost of electric and water.

11.4 Extra Name And Amount In A Mount

In here user can enter extra expences and its name in a mount. These expences are not dependent with fixed cost. For example you did some expences for your healt. If you want to see your expences you have to enter name of that extra and amount of that extra. Because this program could not be attached every kind of expences.

11.5 Expences In A Year

In **Expences In A Year** side user can be entered his/her expences in a year. But these expences are divided to 12 months and result is added with expences of a month.

11.6 Extra Amount And Name In A Year

User wants to arrange his/her variable expences that has not fixed or standard name so he/she have to enter a name of extra and amount of extra.

There are three buttons for deleting, adding, and searching some information

11.7 Add Button

With this button user can be entered new information about new user.

11.7.1 Microsoft Visual Basic Codes For Add Button

```
Private Sub Command46_Click()
```

```
Dim db As DAO.Database
```

```
Dim rs As DAO.Recordset
```

```
Dim s As String
```

```
Set db = CurrentDb()
```

```
s = "SELECT * FROM EXPENCES"
```

```
Set rs = db.OpenRecordset(s)
```

```
If rs.EOF And rs.BOF Then
```

```
rs.AddNew
```

```
rs.Fields(1).Value = Me.PerID
```

```
rs.Fields(2).Value = Me.Elec
```

```
rs.Fields(3).Value = Me.ExtMoAmount
```

```
rs.Fields(4).Value = Me.ExtMoNAME
```

```
rs.Fields(5).Value = Me.ExtYeAmount
```

```
rs.Fields(6).Value = Me.ExtYeNAME
```

```
rs.Fields(7).Value = Me.Water
```

```
rs.Fields(8).Value = Me.Rent
```

```
rs.Fields(9).Value = Me.Tax
```

```
rs.Fields(10).Value = Me.Ins
```

```
rs.Fields(11).Value = Me.InsNO
```

```
rs.Fields(12).Value = Me.TaxYE
```

```
rs.Fields(13).Value = Me.InsYE
```

```
rs.Update
```

```
End If
```

11.8 Delete Button

Delete button is used for deleting any information.

11.8.1 Microsoft Visual Basic Codes For Delete Button

```
Private Sub Command47_Click()
```

```
Dim db As DAO.Database
```

```
Dim rs As DAO.Recordset
```

```
Dim s As String
```

```
Set db = CurrentDb()
```

```
s = "SELECT * FROM EXPENCES WHERE PerID='" & Me.PerID & "'"
```

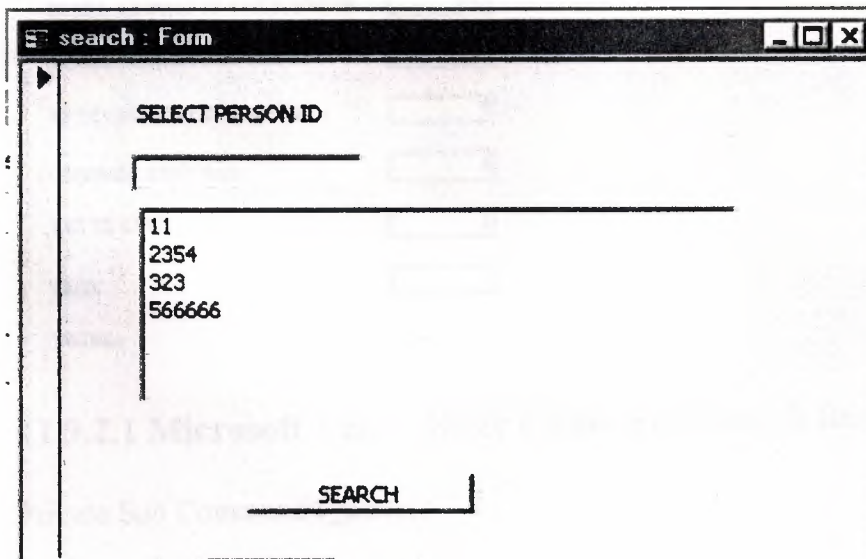
```
Set rs = db.OpenRecordset(s)
```

```
rs.Fields.Delete
```

```
End Sub
```

11.9 Search Button

This button is used for searching any informatin. After pressing this button we will see following form (figure 11.3)



The screenshot shows a Windows-style window titled "search : Form". Inside the window, there is a label "SELECT PERSON ID" positioned above a list box. The list box contains four entries: "11", "2354", "323", and "566666". Below the list box, there is a button with the text "SEARCH".

Figure 11.5 Search Form

11.9.1 Microsoft Visual Basic Codes For Search Form

```
Private Sub Command50_Click()
```

```
DoCmd.OpenForm "SEARCHEXP"
```

```
End Sub
```

11.9.2 Search Button of Search Form

After selecting person id with search button of search form we can see information of that person (figure 11.4)

INSURANCE NO	211134
PERSON ID	123
<hr/>	
ELECTRIC	3720
WATER	3760
RENT	370
INSURANCE	230
TAX	370
EXTRA NAME IN ONE MONTH	0000
EXTRA AMOUNT IN ONE MONTH	7370
EXTRA NAME IN ONE YEAR	000
EXTRA AMOUNT IN ONE YEAR	0
INSURANCE IN YEAR	0
TAX IN YEAR	0
DATE	
TOTAL:	44130

11.9.2.1 Microsoft Visual Basic Codes For Search Button of Search Form

```
Private Sub Command4_Click()
```

```
DoCmd.OpenReport "EXPENCES"
```

```
End Sub
```


11.10 Total Button

This button is for calculation of average expences in a month. It divides extra amount of one year and it adds this amount to that month .

11.11 Microsoft Visual Basic Codes For Total Button

```
Private Sub Command33_Click()
```

```
Me.Text34 = Me.Elec + Me.Water + Me.Rent + Me.Ins + Me.Tax + Me.ExtMoAmount +  
(Me.ExtYeAmount + Me.InsYE + Me.TaxYE) / 12
```

```
End Sub
```



Figure 12.1 Personal Budget

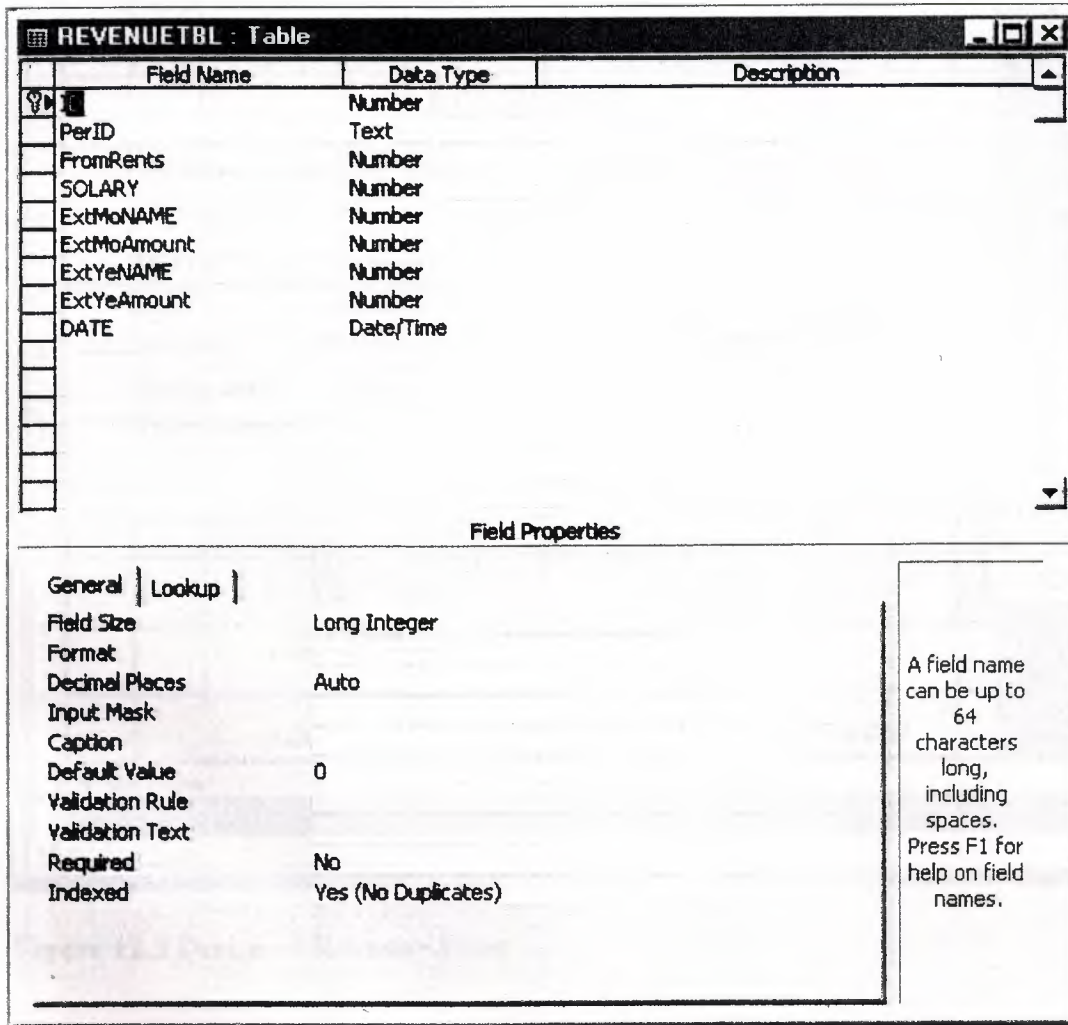


CHAPTER TWELVE

REVENUE FORM

12.1 Revenue Table

Here we create a table (**figure 12.1**) for personal expences information when we enter some information about any person's expences this table creates a database (**figure 12.2**)

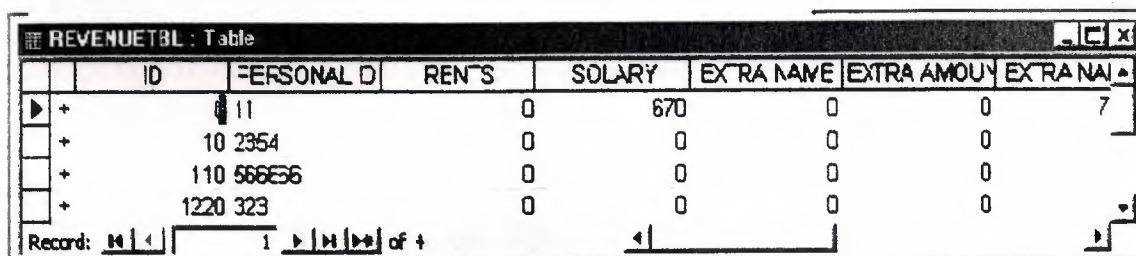


Field Name	Data Type	Description
PerID	Number	
FromRents	Text	
SOLARY	Number	
ExtMoNAME	Number	
ExtMoAmount	Number	
ExtYeNAME	Number	
ExtYeAmount	Number	
DATE	Date/Time	

Field Properties	
General	Lookup
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)

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

Figure 12.1 Personal Revenues Information Table



ID	PERSONAL ID	REN'S	SOLARY	EXTRA NAME	EXTRA AMOUNT	EXTRA NAME
11	11	0	670	0	0	7
10	2354	0	0	0	0	0
110	566E36	0	0	0	0	0
1220	323	0	0	0	0	0

Record: 1 of 4

Figure 12.2 Personal Revenues Information Database

12.2 Design of Revenue Form

From Database (Access 2000 file format) windows we can desing a new form. After entering design view we can prepeare form (figure 12.3)

The screenshot shows the Microsoft Access 2000 design view of a form titled 'REVENUETBL'. The form is structured with a 'Form Header' section at the top, followed by a 'Detail' section. The 'Form Header' section contains a title 'REVENUES' and two sub-sections: 'REVENUES IN ONE MOUNTH' and 'REVENUES IN ONE YEAR'. The 'REVENUES IN ONE MOUNTH' section contains five fields: 'PERSON ID' (Unbound), 'RENTS' (Unbound), 'SOLARY' (Unbound), 'EXTRA NAME' (Unbound), and 'EXTRA AMOUNT' (Unbound). The 'REVENUES IN ONE YEAR' section contains three fields: 'EXTRA NAME' (Unbound), 'EXTRA AMOUNT' (Unbound), and 'DATE' (Unbound, with a date picker icon). Below these sections is a 'TOTAL' label with a calculation formula: '[FromRents]+[SOL]'. At the bottom of the form are three buttons: 'ADD', 'DELETE', and 'SEARCH'. The form is divided into 'Form Header' and 'Form Footer' sections. A 'Toolbox' is visible on the right side of the design view.

Figure 12.3 Design of Revenue Form

After desing we will have following form (figure 12.4).

REVENUES	
REVENUES IN ONE MOUNTH	REVENUES IN ONE YEAR
PERSON ID: <input type="text"/>	
RENTS <input type="text"/>	EXTRA NAME <input type="text"/>
SOLARY <input type="text"/>	EXTRA AMOUNT <input type="text"/>
EXTRA NAME <input type="text"/>	
EXTRA AMOUNT <input type="text"/>	
	DATE <input type="text" value="13.06.2003"/>
TOTAL <input type="text"/>	
ADD NEW <input type="button"/> DELETE <input type="button"/> SEARCH <input type="button"/>	

Figure 12.4. Revenue Windows

Here there are sections for information of incomes

12.3 Revenues In A Mount

In **Revenues In A Mounth** side user can be entered his/her revenues in a mounth. For example incomes from rents and solary.

12.4 Extra Name And Amount In A Mount

In here user can enter extra revenues and its name in a mounth. These revenues are not dependent with fixed incomes. For example you taked some revenues from anywhere like lotto. If you want to see your revenues you have to enter name of that extra and amount of that extra. Because this program could not be attached every kind of revnues.

12.5 Revenues In A Year

In **Revenues In A Year** side user can be entered his/her revenues in a year. But these revenues are divided to 12 months and result is added with revenues of a month.

12.6 Extra Amount And Name In A Year

User wants to arrange his/her variable revenues that has not fixed or standard name so he/she have to enter a name of extra and amount of extra.

There are three buttons for deleting, adding, and searching some information

12.7 Add Button

With this button user can be entered new information about new user.

12.7.1 Microsoft Visual Basic Codes For Add Button

```
Private Sub Command28_Click()
```

```
Dim db As DAO.Database
```

```
Dim rs As DAO.Recordset
```

```
Dim s As String
```

```
Set db = CurrentDb()
```

```
s = "SELECT * FROM REVENUETBL"
```

```
Set rs = db.OpenRecordset(s)
```

```
If rs.EOF And rs.BOF Then
```

```
rs.AddNew
```

```
rs.Fields(1).Value = Me.PerID
```

```
rs.Fields(2).Value = Me.FromRents
```

```
rs.Fields(3).Value = Me.SOLARY
```

```
rs.Fields(4).Value = Me.ExtMoNAME
```

```
rs.Fields(5).Value = Me.ExtMoAmount
```

```
rs.Fields(6).Value = Me.ExtYeNAME
```

```
rs.Fields(7).Value = Me.ExtYeAmount
```

```
rs.Update
```

```
End If
```

```
End Sub
```

12.8 Delete Button

Delete button is used for deleting any information.

12.8.1 Microsoft Visual Basic Codes For Delete Button

```
Private Sub Command39_Click()
```

```
Dim db As DAO.Database  
Dim rs As DAO.Recordset  
Dim s As String
```

```
Set db = CurrentDb()
```

```
s = "SELECT * FROM REVENUETBL WHERE PerID=" & Me.PerID & ""
```

```
Set rs = db.OpenRecordset(s)
```

```
rs.Fields.Delete
```

```
End Sub
```

12.9 Search Button

This button is used for searching any informatin. After pressing this button we will see following form (**figure 11.3**)

The screenshot shows a Windows-style window titled "search : Form". Inside the window, there is a label "SELECT PERSON ID" above a list box. The list box contains four items: "11", "2354", "323", and "566666". Below the list box, there is a button labeled "SEARCH".

Figure 12.5 Search Form

12.9.1 Microsoft Visual Basic Codes For Search Form

```
Private Sub Command30_Click()
```

```
DoCmd.OpenForm "search"
```

```
End Sub
```

12.9.2 Search Button of Search Form

After selectting person id with search button of search form we can see information of that person (figure 11.4)

ID	<input type="text" value=""/>
PERSONAL ID	<input type="text" value="1"/>
<hr/>	
RENTS	<input type="text" value="0"/>
SOLARY	<input type="text" value="670"/>
EXTRA NAME IN ONE MONT	<input type="text" value="0"/>
EXTRA AMOUNT IN ONE MOUNT	<input type="text" value="0"/>
EXTRA NAME IN YEA	<input type="text" value="7670"/>
EXTRA AMOUNT IN ONE YEA	<input type="text" value="0"/>
TOTAL:	<hr/> <input type="text" value="670"/>
ID	<input type="text" value="1770"/>
PERSONAL ID	<input type="text" value="323"/>
<hr/>	
RENTS	<input type="text" value="0"/>
SOLARY	<input type="text" value="0"/>
EXTRA NAME IN ONE MONT	<input type="text" value="0"/>
EXTRA AMOUNT IN ONE MOUNT	<input type="text" value="0"/>
EXTRA NAME IN YEA	<input type="text" value="0"/>
EXTRA AMOUNT IN ONE YEA	<input type="text" value="0"/>
TOTAL:	<hr/> <input type="text" value="0"/>

Figure 12.6. Searching results

12.9.2.1 Microsoft Visual Basic Codes For Search Button of Search Form

```
Private Sub Command4_Click()
```

```
DoCmd.OpenReport "REVENUETBL"
```

```
End Sub
```

12.10 Total Button

This button is for calculation of avarege expences in a mount. It divides extra amount of one year and it adds this amount to that mounth .

12.11 Microsoft Visual Basic Codes For Total Button

```
Private Sub Command16_Click()
```

```
Me.Text18= Me.FromRents+ Me.SOLARY+ Me.ExtMoAmount+( Me.ExtYeAmount)/12
```

```
End Sub
```

CONCLUSION

Every access project have one or more table. This is general necessity of the access. In this project tables are used for creating a database. Sometimes that tables are related with each other. So, relationship section of access is very important to create relationship between tables of the this project . After creating tables and relationships, there must be a user interface that is used for presenting of informations as a form format.

In this project every steps of access are applied for track a personal finance. This project generates an easy way to tracking of personal finances. Every users can store his/her own financial information and they can calculate or considere balance of their financial information. The main problem of tracking of personal finances is to store or calculate variable cost and variable incomes. But in this project, this problem does not occur because for every variable values has an extra section. User can enter these variable values as an extra form.

The realization of tracking of personal finances is performed using access. In the result of using the tracking of personal finances program , effective of complexity of calculation and track are minimized.

REFERENCES

Refence to book:

- [1] Microsoft Access 97, microsoft

References to Electronic Source- Online sources from Web:

- [1] Access 2000 Tutorial, " <http://www.fgcu.edu/support/office2000/access> "
- [2] Tony's Main Microsoft Access Page, "<http://www.granite.ab.ca/accsmsr.htm> "