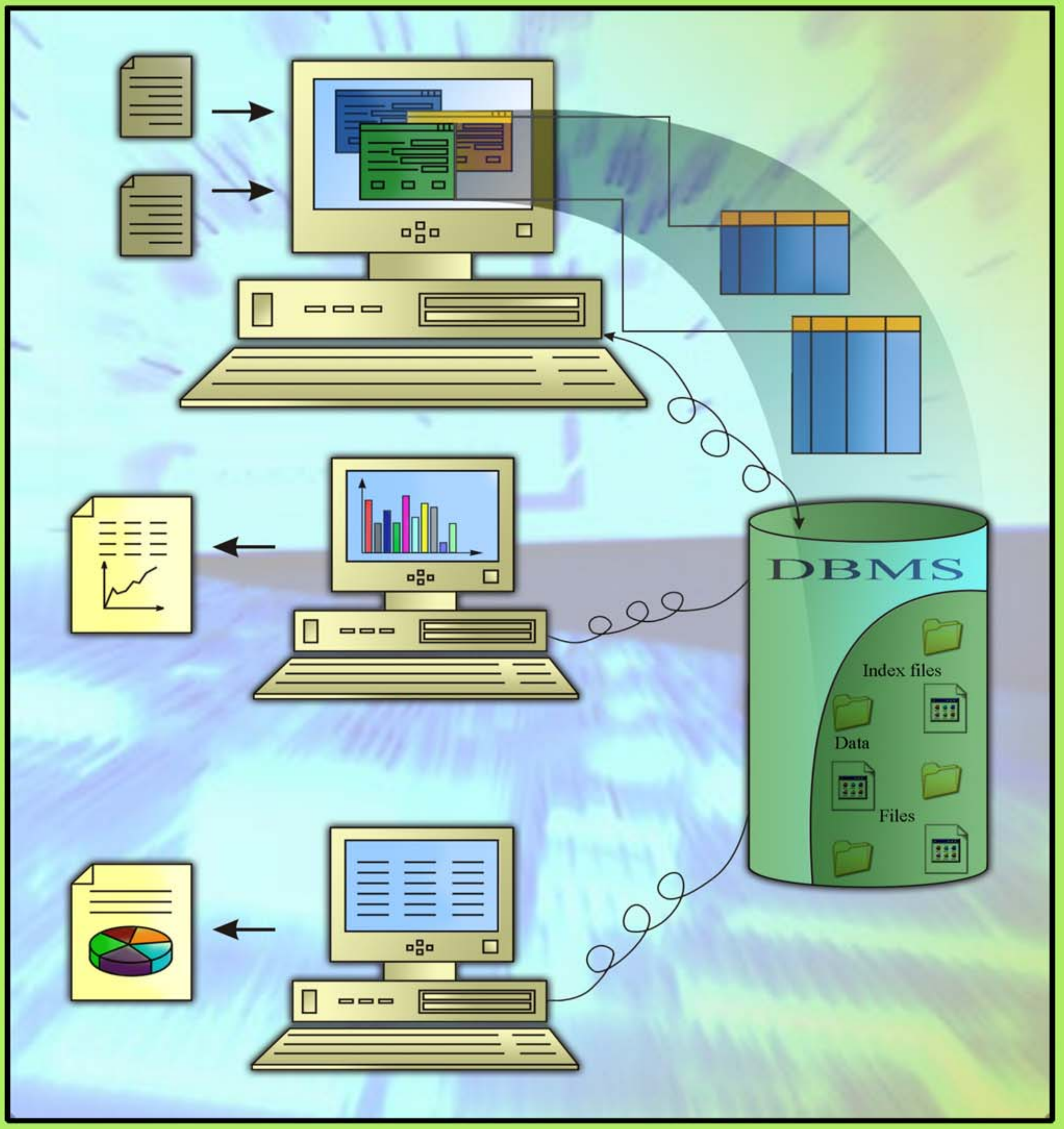
Indira Gandhi National Open University
School of Computer and Information Sciences

MCS - 023
INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS



CASE STUDY

3

Indira Gandhi
National Open University
School of Computer &
Information Sciences

MCS-023
INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS

Block

3

APPLICATION DEVELOPMENT: DEVELOPMENT OF A HOSPITAL MANAGEMENT SYSTEM (An HMS)

Introduction 3

Need to Develop the Hospital Management System (An HMS) 4

Creating a Database for HMS 5

Developing Front End Forms 11

Reports 18

Using Queries and Record set 35

Summary 47

Programme / Course Design Committee

Prof. Sanjeev K. Aggarwal, IIT, Kanpur
Prof. M. Balakrishnan, IIT, Delhi
Prof. Harish Karnick, IIT, Kanpur
Prof. C. Pandurangan, IIT, Madras
Dr Om Vikas, Sr. Director,
Ministry of CIT, Delhi
Prof. P. S. Grover, Sr. Consultant,
SOCIS, IGNOU

Faculty of School of Computer and Information Sciences
Shri Shashi Bhushan
Shri Akshay Kumar
Prof. Manohar Lal
Shri V.V. Subrahmanyam
Shri P.Venkata Suresh

Block Preparation Team

Mr. Milind Mahajani (Content Editor)
Software Consultant
New Delhi

Ms. Ranjana Sharma
Ansal Institute of Technology
Gurgaon

Prof. (Retd) A.K.Verma (Language Editor)
New Delhi

Course Coordinator: Shri Akshay Kumar

Block Production Team

Shri T.R. Manoj, Section Officer (Pub.) and Shri H.K. Som, Consultant
CRC prepared by Shri A N Kispotta

Acknowledgement

To Prof. Parvin Sinclair for help in the coordination of this block.

May, 2005

ISBN —

Further information on the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068.

Printed and published on behalf of the Indira Gandhi National Open University, New Delhi by the Director, SOCIS.

DEVELOPMENT OF A HOSPITAL MANAGEMENT SYSTEM (An HMS)

Section No.	Title	Page Nos.
1	Introduction	3
2	Need to Develop the Hospital Management System (An HMS)	
2.1	Advantages of Hospital Management System	
2.2	ER diagram for HMS	
2.3	The Software Development Tools required	
3	Creating a Database for HMS	5
3.1	Creating a Database using Ms-Access	
3.2	Table Relationships	
4	Developing Front End Forms	11
5	Reports	18
5.1	Creating a Report using the Wizard	
5.2	Creating Mail Merge Labels using a Wizard	
5.3	Creating a Report using the Access Report Wizard	
5.4	Creating Report using Design View	
6	Using Queries and Record set	35
6.1	Using the Simple Query Wizard	
6.2	Creating a Query in Design View	
6.3	Opening a Query	
6.4	Adding Features in Query	
6.5	Conditional Queries	
6.6	Advanced Query Features	
7.	Summary	47

1.0 INTRODUCTION

In the two previous blocks, we have provided the details of the concepts of Relational Database Management. This block provides an example of the use of a Database Management System and user interface development tool to create a Hospital Management System (HMS). The focus in this block is not on developing the Hospital Management System but on providing practical oriented details about how to work with the tools available in MS Access and Visual Basic to create the basic elements of any application that are the tables, forms, reports and queries. Thus, in this block, we will just show the steps about how to create forms, create tables and create reports and queries. You, in turn, have to develop this system by practicing the steps for creating various sample forms, reports, tables and queries.

Please note that although this application in the block is expected to be developed using Microsoft Access and Visual Basic, you may like to use open source tools **MYSQL** and **PHP.**

Each hospital is unique in its requirements and priorities. Hence the HOSPITAL MANAGEMENT SYSTEM (HMS) covers many hospital activities. It also allows hospitals to choose from the various modules to match their specific needs. Please note that this is just a very simple prototype and should not be used for commercial deployment. Also, you must discuss the specification and design errors of this application for better description of the domain of Hospital administration.

This block pays attention to various steps involved in using the tools to evolve the system. Less emphasis is given to analysis and design of the system, which is dealt in more detail in Block 4.

Objectives

After going through this block you should be able to:

- brief the very basic needs for an Information System in a Hospital;
- use the tools needed to develop tables, forms, reports and queries;
- work with various tools to create a sample system, and
- identify the problems encountered during the implementation of an Information System.

2.0 NEED TO DEVELOP THE HOSPITAL MANAGEMENT SYSTEM (An HMS)

Every hospital big or small keeps the records of its patients including the registration details of the patient and the fee payments. The entry of patients is determined whether s/he has arrived in emergency, OPD or for a routine check-up. The patient who gets admitted is provided with a room according to his/her choice. The patient is allotted a doctor according to his illness. The doctor may refer the patient to another doctor with expertise of the illness. On discharge, the patient is required to settle the bills sent by the accounts department of the hospital.

The hospital also keeps the record of the doctors visiting the hospital, plus the permanent employees of the hospital. Each doctor has a few days associated with his/her visit to the hospital and also the timings when s/he is available in the hospital. The employees draw their salary from the accounts department.

The hospital maintains the record of the inventory of the hospital including the equipment and the medicines, blood bank, etc. A limit for inventory is maintained for every item. When the limit is reached, an order for the purchase of the concerned item is placed. The database is updated after the supplier supplies the product.

2.1 Advantages of Hospital Management System

HMS provides manifold benefits to hospitals whether large or medium sized. Streamlined Operations:

- Minimized documentation and no duplication of records.
- Reduced paper work.

Improved Patient Care:

- Procedures for timely and effective patient care
- Faster information flow between various departments
- Easy access to reference records.

Better Administration Control:

- Availability of timely and accurate information
- Access to updated management information.

Smart Revenue Management:

- Optimized bed occupancy checks
- Effective billing of various services
- Exact stock information.

2.2 ER Diagram of HMS

The following is a very simplified ER Diagram of a much mutated toned down version of HMS that does not fulfill all the requirements above. You must enhance it once you are through with the system.

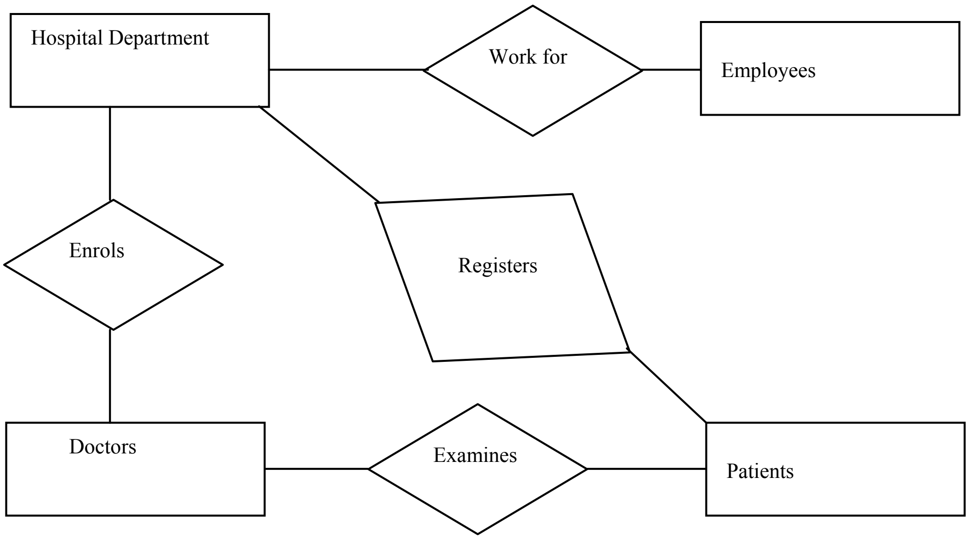


Figure 1: Entity relationship diagram for HMS

2.3 The Software Development tools required

We have selected the combination of MS-Access and VISUAL Basic although you can also use MS-Access forms instead of Visual Basic. The selection of Visual Basic is governed by the reason that this is one of the most simple front end tools. As this application is the first application in any database management system, it may be a good starting point to learn a bit about it. However, you may like to use MySQL and PHP also, a very popular combination of tools for such type of system development. Let us now do some basic development.

Exercise 1: Find out the requirements that can be met considering the Data Model that is being reflected by the ER diagram.

3.0 CREATING A DATABASE FOR HMS

The database has to be created for the Hospital Management System. This is done by creating tables and relations in MS-Access. A Database is a collection of related data describing one or more related activities. A database Management System is a collection of programs that enable the user to create and maintain the database. The following relations are created in MS-Access.

Docdetails – Gives the details of the doctors affiliated to the Hospital.

Patient Registration – Provides the details of a patient as well as arrival status such as OPD or emergency and calculates charges incurred. If a patient is already registered, updation of previous records is made.

Employee Details – This is used to provide the details of employees working in the hospital irrespective of the designation and the department.

Department Details – This provides the details of any department in the hospital including the department heads.

Exercise 2 : Identify the limitations of such a table design. Propose a modified table design for the above.

3.1 Creating a database using MS-Access

1. Open MS-Access
2. Dialog Box – Create a new database using
   Blank database
   Access database wizards, pages and projects
3. Select Create a new database using blank database
4. db1 – create

Now the database db1 has been created. The following screen will appear. You can now create tables, forms, reports, queries etc. in the database db1.

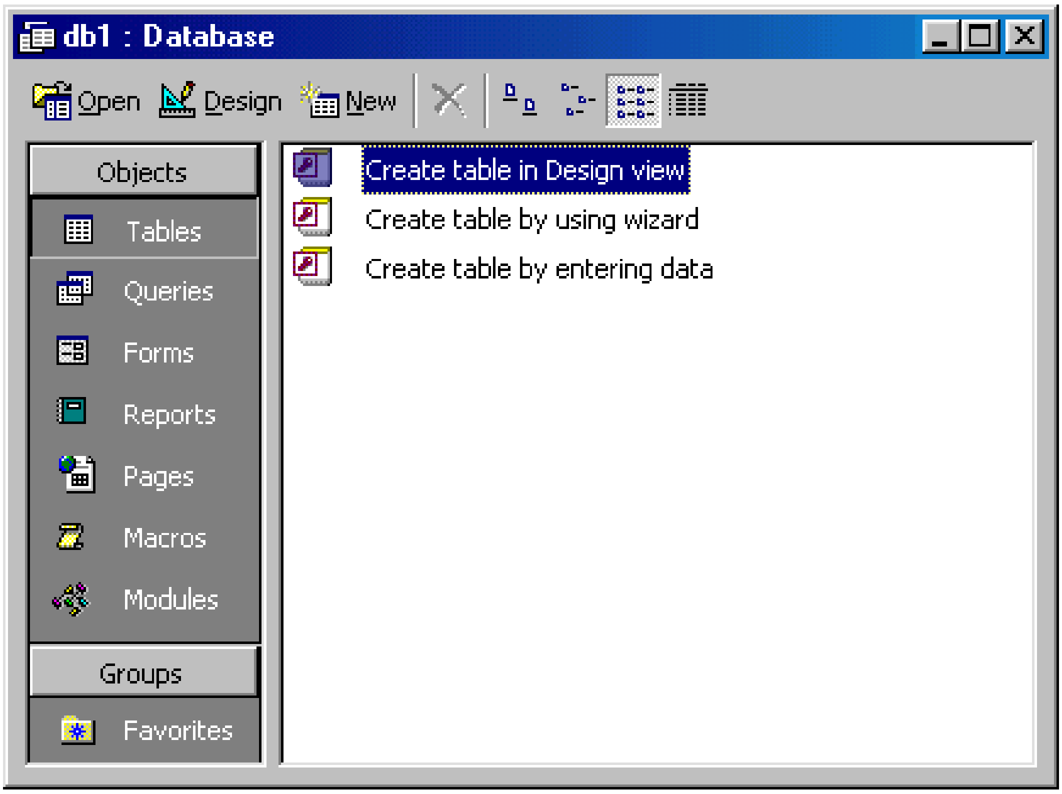
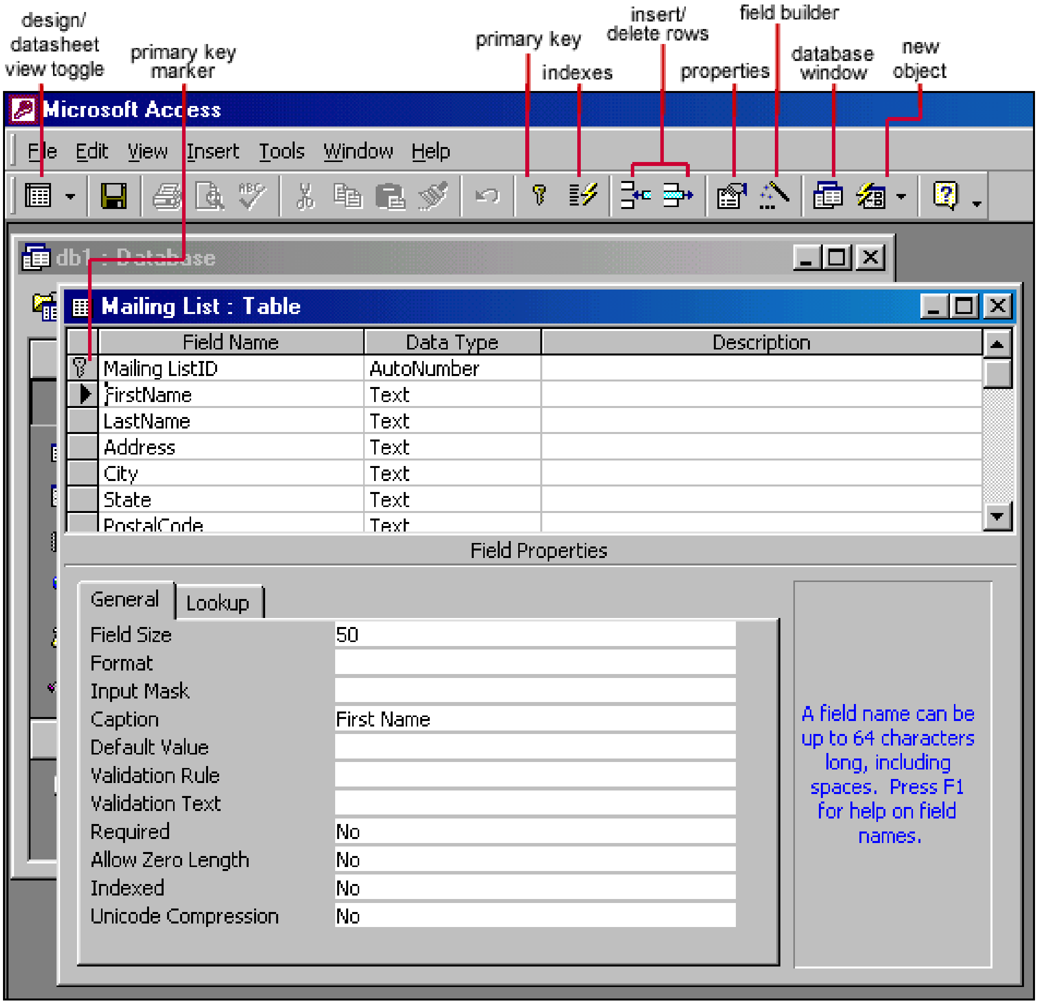


Figure 2: Creating database tables using MS-Access

- **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 :
- **Create table using wizard** will step you through the creation of a table.
- **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. Hence create a table “Docdetails” with fields such as “name”, “address”, “qualifications” 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.

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 2*).



Field Name	Data Type	Description
Mailing ListID	AutoNumber	
FirstName	Text	
LastName	Text	
Address	Text	
City	Text	
State	Text	
PostalCode	Text	

Field Size	50
Format	
Input Mask	
Caption	First Name
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	No
Unicode Compression	No

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

Figure 3: A sample table creation screen

- **Field Name:** This is the name of the field and should represent the contents of the field such as "Name", "Address", "Qualifications", etc. The name cannot exceed 64 characters in length and may include spaces.
- **Data Type** is the type of value that will be entered into the fields.

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. An enrolment number is an example of a field that can be used as primary key to table student.

Designate the primary key field by first selecting them (you may select a composite primary key also) and then by right clicking on it and selecting **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.

Hence in HMS the table Docdetails can be created using the above method. The following tables should be created by you.

DocDetails

DocCode	Text	PK
DeptCode	Text	FK
DocName	Text	
Qualifications	Text	
Designation	Text	
Address	Text	

Patients' Registration

Patient Regn No.	Text	PK
Date of Regn.	Text	
Name	Text	
Father's/husband's name	Text	
Age	Number	
Address	Text	
DocCode	Text	FK
Arrived in	Text	
Charges	Number	

Employee Details

EmpCode	Text	PK
Ename	Text	
Address	Text	
Gender	Text	
Date of Joining	Date/Time	
Salary	Number	
DeptCode	Text	FK

DeptDetails

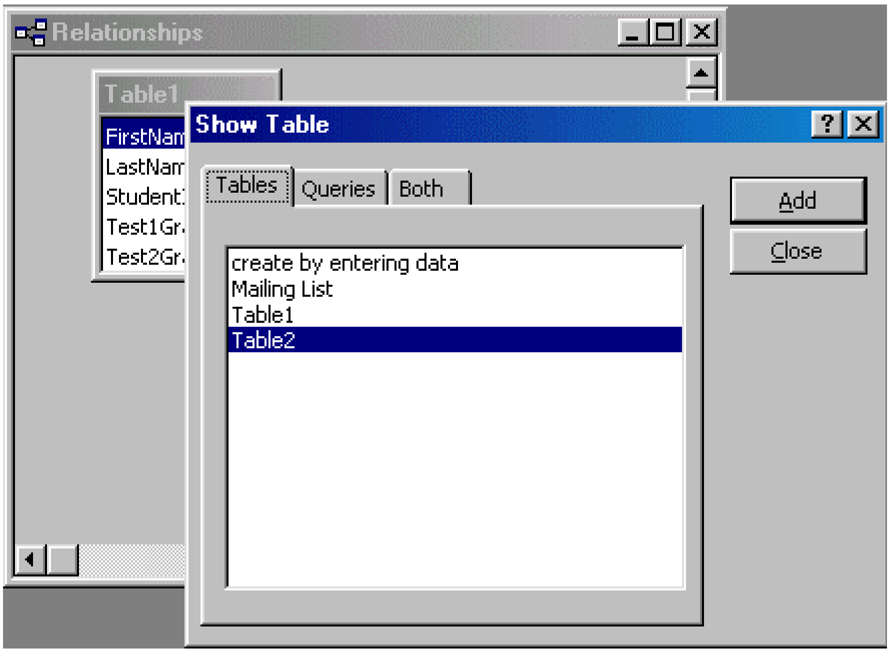
DeptCode	Text	PK
DeptName	Text	
HOD	Text	
HOD Code	Text	

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.

3.2 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:

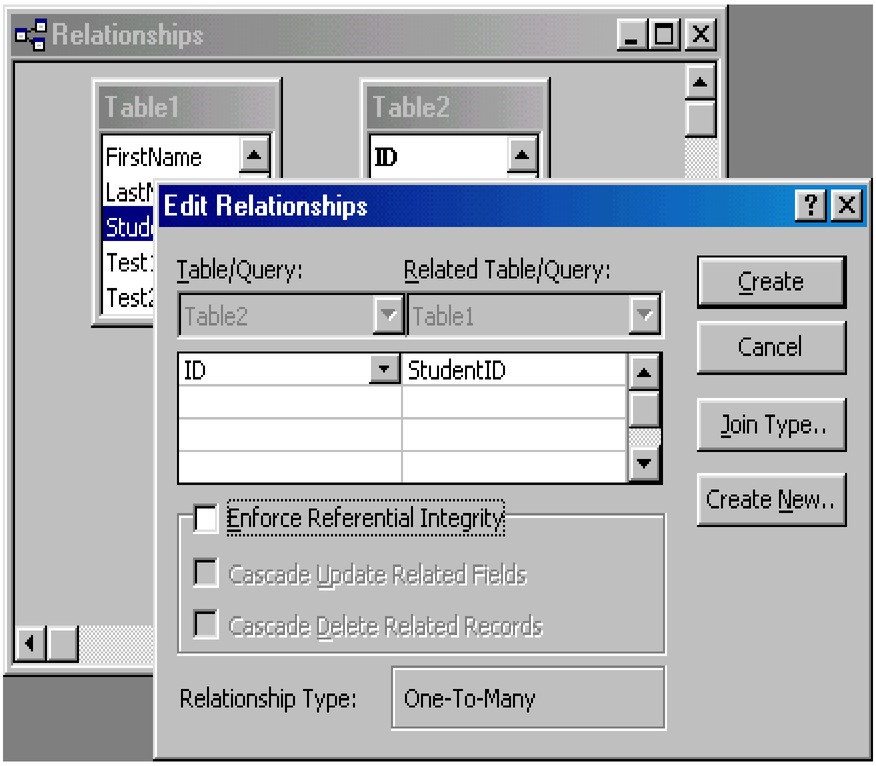
- Click the **Relationships** button on the toolbar.
- From the **Show Table** window (click the **Show Table** button on the toolbar to make it appear), double click on the names of the tables you would like to include in the relationships. When you have finished adding tables, click **Close**.



create by entering data
Mailing List
Table1
Table2

Figure 4: A sample screen for relationships

- 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.



Table/Query: Table2

Related Table/Query: Table1

ID	StudentID

□ Enforce Referential Integrity

□ Cascade Update Related Fields

□ Cascade Delete Related Records

Relationship Type: One-To-Many

Figure 5: Editing a sample relationship

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

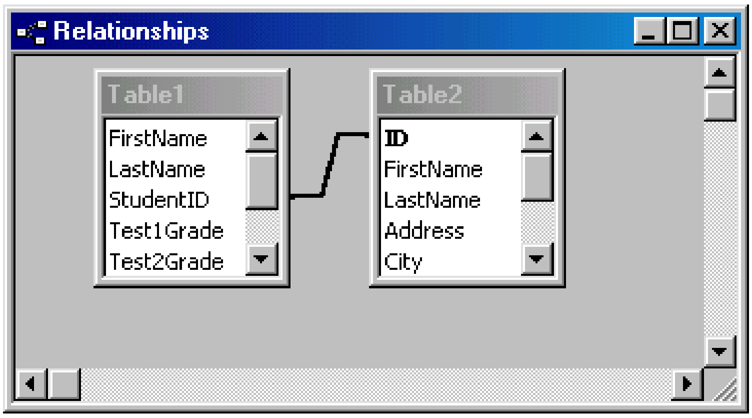


Figure 6: A sample relationship window

- 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, the student address database and student grade database are related.

Hence the relationship for the above is created using the process above. Now, use the tables created by you about HMS and develop the following relationships.

DocDetails

DocCode	Text	PK
DeptCode	Text	FK
DocName	Text	
Qualifications	Text	
Designation	Text	
Address	Text	

Patients' Registration

Patient Regn No.	Text	PK
Date of Regn.	Text	
Name	Text	
Father's/husband's name	Text	
Age	Number	
Address	Text	
DocCode	Text	FK
Arrived in	Text	
Charges	Number	

Employee Details

EmpCode	Text	PK
Ename	Text	
Address	Text	
Gender	Text	
Date of Joining	Date/Time	
Salary	Number	
DeptCode	Text	FK

DeptDetails

DeptCode	Text	PK
DeptName	Text	
HOD	Text	
HOD Code	Text	

4.0 DEVELOPING FRONT END FORMS

Forms can be developed using Microsoft Access form wizard and interfaces. But we have selected Visual Basic here as it is one of the useful tools for form design. You are however free to develop these forms using any other platform. We are providing a very brief account of Visual Basic (VB) here, for more details you may refer for help to VB or further readings.

Using Visual Basic -A brief description

VISUAL BASIC is a high level programming language evolved from the earlier DOS version called BASIC. BASIC means **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. It is a fairly easy programming language to learn. The codes look a bit like English Language. Different software companies produced different versions of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASIC and so on.

VISUAL BASIC is a VISUAL and events driven Programming Language. These are the main divergences from the old BASIC. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In VISUAL BASIC, programming is done in a graphical environment. Because users may click on a certain object randomly, so each object has to be programmed independently to be able to respond to those actions (events). Therefore, a VISUAL BASIC Program is made up of many subprograms, each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another.

The Visual Basic Environment

On start up, Visual Basic 6.0 will display the following dialog box as shown in *Figure7*. Choose to start a new project, open an existing project or select a list of recently opened programs. A project is a collection of files that makes up an application. There are various types of applications we could create. However, we shall concentrate on creating Standard EXE programs (EXE means executable program).

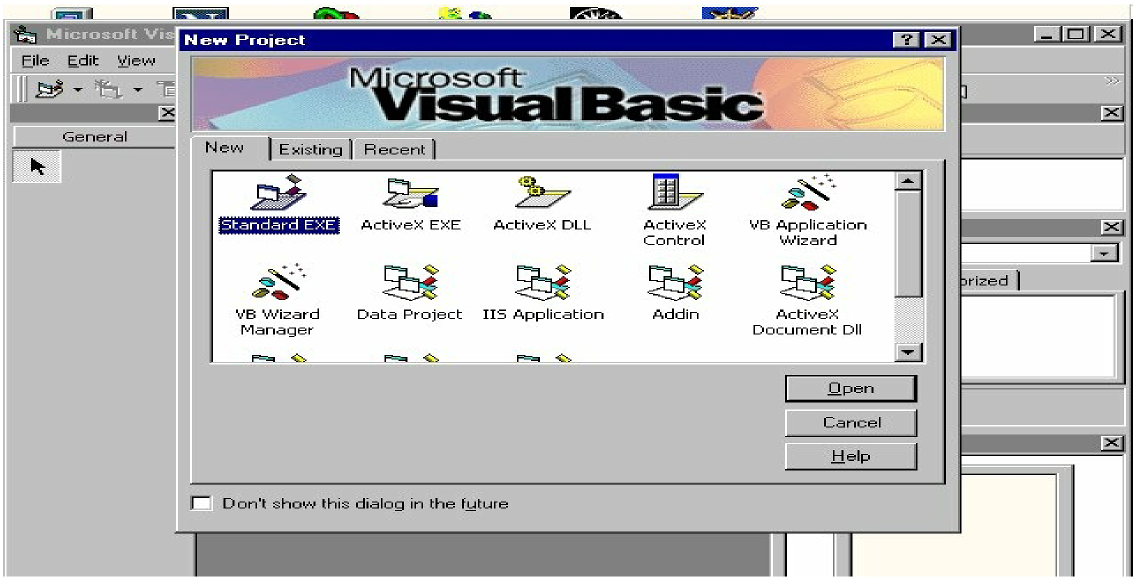


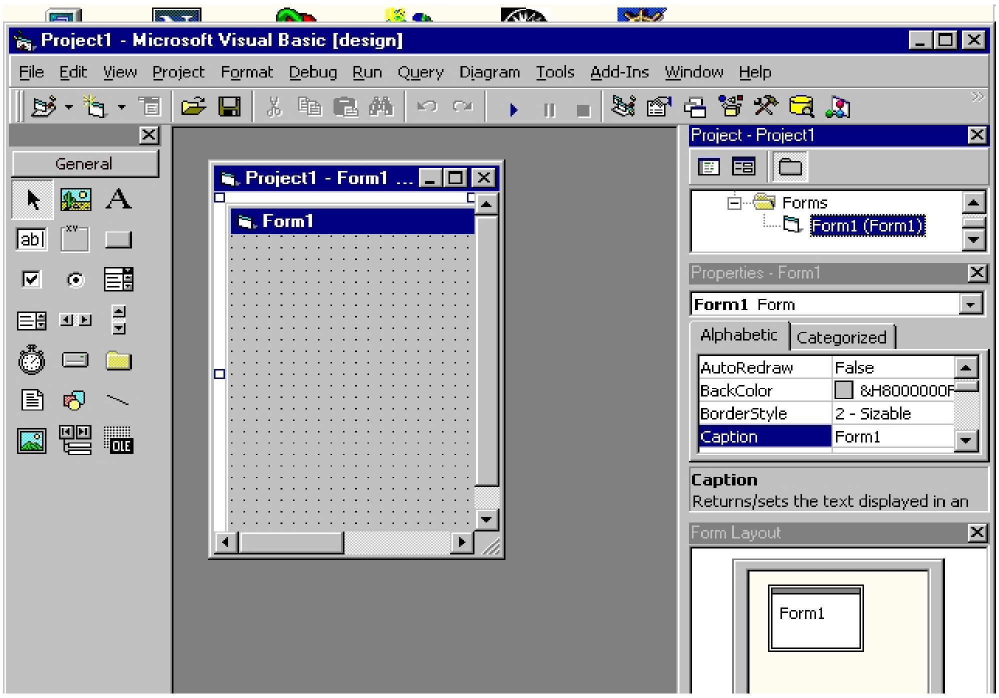
Figure 7: The visual basic start-up dialog box

Now, click on the standard EXE icon to go into the actual VB programming environment.

In *Figure 7*, the Visual Basic Environment consists of –

- The **Blank Form** window which you can design your application's interface.
- The **Project** window displays the files that are created in the application.
- The **Properties** window which displays the properties of various controls and objects that are created in your applications.

It also includes a **Toolbox** that consists of all the controls essential for developing a VB Application. Controls are tools such as boxes, buttons, labels and other objects drawn on a form to get **input** or **display output**. They also add visual appeal.



AutoRedraw	False
BackColor	□ &H8000000F
BorderStyle	2 - Sizable
Caption	Form1

Caption
Returns/sets the text displayed in an

Figure 8: The visual basic environment

To the left of the screen there are a lot of little icons under the heading 'General'. These are called components - ActiveX controls. Basically, these can be added to your program (technical description, that) - such as a button or picture. Let's see one in action.

- Double-click on the component represented by a black 'A'

An item called 'Label1' should appear in the centre of Form1.

To the bottom-right of the screen there is a box entitled "Properties - Label1". If it doesn't say 'Label1', click on the label in the middle of your form.

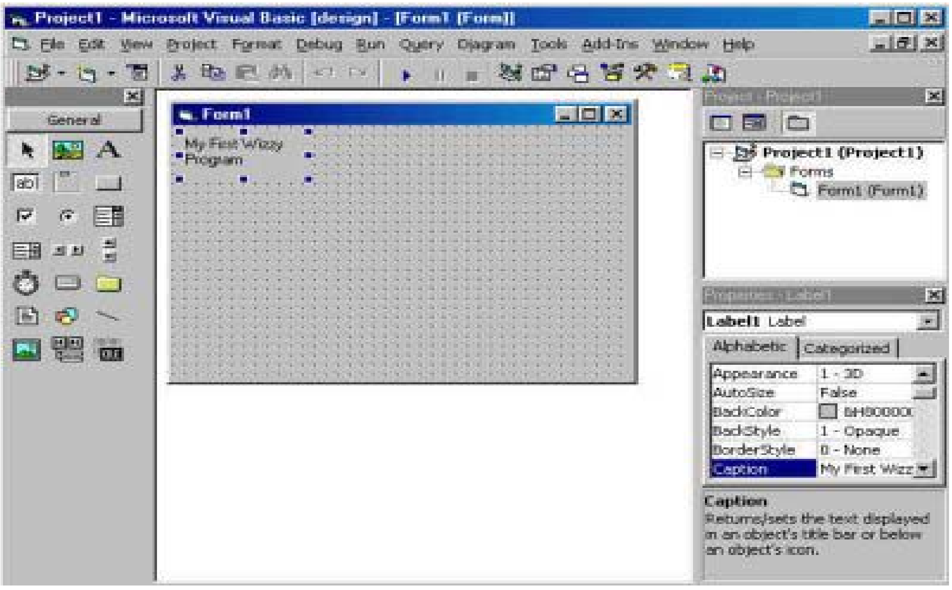
These are the "properties" of the label. They tell it how to work - what text to display, what colour it should be, whether it should have a border, and so on. These can be easily changed by typing in or selecting new values from the list.

- Click on the box to the right of the 'Caption' property.
- Replace the existing "Label1" value with "My First Program".
- Press Enter.

To move the label –

- Click on the label and hold your mouse down.
- Drag your mouse to the upper left hand corner of "Form1" and release the mouse button.

The screen looks something like this right now –


Project1 - Microsoft Visual Basic [design] - [Form1 (Form)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
General
Form1
My First Wizzy Program
Project - Project1
Project1 (Project1)
Forms
Form1 (Form1)
Properties - Label1
Alphabetic
Categorized

Label1 Label

Appearance	1 - 3D
AutoSize	False
BackColor	&H8000000
BackStyle	1 - Opaque
BorderStyle	0 - None
Caption	My First Wizz

Caption
Returns/sets the text displayed in an object's title bar or below an object's icon.

Figure 9: The text label

Double click on the button in the General toolbar. This is a Command Button – you'll find them in virtually all Windows programs, usually in the form of "OK", "Apply" or "Cancel" buttons.

To change the property of Command Button –

- Change its Caption property to "Click Me!"
- Move down in the Properties list and change the Font.
- Resize the Command Button using the eight blue squares surrounding it.

The Form looks like this right now –

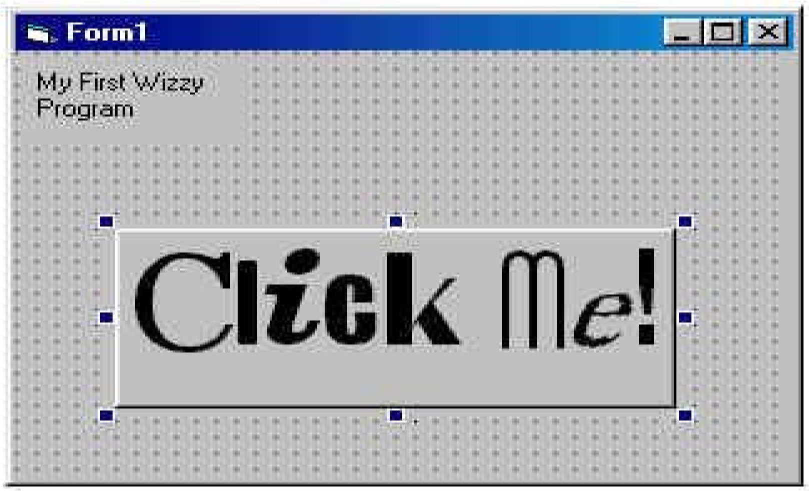


Figure 10: A sample form

Finally, let us add our first line of code: For this double click on the Command Button.

The screen should look like this...

```
Private Sub Command1_Click()

<Cursor flashing here>

End Sub
```

Now type in.

```
MsgBox "Welcome to my First program!"
```

The "Private Sub Command1_Click ()" tells Visual Basic to run this line of code when someone **clicks** on the Command Button called **Command1**. When they do, the line of code runs - telling Visual Basic to display the designated sentence in a message box (**MsgBox**). The "End Sub" merely tells VB to stop running the code.

Control: A control is just an item that can be added to the Form.

We could then change its properties - perhaps alter its Caption to "Click Me You Annoying Person!" or its Font property to Arial, Bold, Size 18. Open Visual Basic and create a new Standard Exe. To do that:

- Click on the Standard Exe icon.
- Click on the Open button

To the left of your screen should be the control toolbox. Right now it's displaying a standard set of Visual Basic controls.

☑ This is the Check Box control. Such boxes are checked and unchecked to set various options. Functionality can be added to the applications. The Check Box properties can be changed in the properties window. For instance, alter its Caption, Forecolor or Font property with a few clicks of the mouse! Some of the properties are hardly ever used or only required in very specific circumstances.

Using more built in VB controls

Image - - This control allows us to display a picture on the Form.

Label - A - This control is often used to display information to the user, such as an explanation of what they should do.

TextBox - ab| - The TextBox control is a little like McDonalds; common, usually trouble-free and definitely overused. You often utilise this control to collect information from the user or display data from a database.

CommandButton - - This is a pretty boring yet very common control. It simply creates a button the user may click to run a slot of code.

DriveListBox - - Whilst this control doesn't have many properties you can fiddle with, it's great to observe. You've seen this famous drop-down list many times before and now it can be incorporated into your own cool program!

Shape - - This control draws a shape on your Form.

To run an application:

- ▶ Press F5 (or click the button)

Every single control you add to a Form has a Name property – it should appear at the very top of the Properties window. Every time a control is added, an automatic name is assigned – such as Command1. It can be renamed to give it a meaningful name.

The Visual Basic community also has their own “naming conventions” where, for instance, Text Box names are preceded by “txt”. Therefore if a Text Box is added that will hold a date, it is advisable to change its name to “txtDate”. Other prefixes include:

ComboBox - cbo (for example, cboProductCodes)

CommandButton - cmd (e.g., cmdOK)

Form - frm (e.g., frmOrders)

CheckBox - chk (eg chkExtraCheese)

OptionButton - opt (e.g., optPlanTwo)

It is advisable to keep names pretty unique.

Code Writing

Writing creative code is easy! Firstly, programmers often refer to “design-time” and “runtime”. Design-time is the period when designing of Form or writing code is done – basically any time your program isn’t running. And runtime is exactly the opposite – the time when your program *is* running.

The properties of controls can be changed whilst you are in design-time. The properties can also be whilst the program is running. For instance, perhaps you want the Caption of your Form to change depending on which button the user clicks.

- Open Visual Basic and create a Standard Exe
- Change the Name of Form1 to “frmTest” and the Caption to “My Test App!”
- Add two Command Buttons and change the Caption of the first to “Hayley” and the second to “Roy”.

Any control can be added to the Form, including Command Buttons. You can easily change the properties of such controls by altering values in the properties window.

How can variables be used in your Visual Basic applications?

A variable can be used in VB. This is explained with the help of the following example.

Example: In order to display a number and increment it, every time the user clicks a Command Button variables can be used.

- Launch Visual Basic and create a new Standard Exe.
- Add a Command Button to your Form.
- Change the button’s Name property to “cmdVariableCount” and its Caption property to “Click Me Quick!”
- Double click on a blank area of the Form to open the code window.

Near the top of the code window, there are two drop–down boxes. The first specifies an *object*, the second a *procedure*. In English, these are defined as:

Object the thing you want to work with, say cmdVariableCount.

Procedure denotes the 'event' for which the code is written. For instance, we may write a code in the Click event to have it respond when the user clicks on a cmdVariableCount. The Procedure drop-down can also display user-defined Subs and Functions.

Click on the object drop-down box, the first one.

You should see a list of objects – probably Form, cmdVariableCount and (General). The first two are objects – and you can add code to respond to their events. Select the (General) section from the drop-down list.

The right-hand combo box should change to (Declarations).

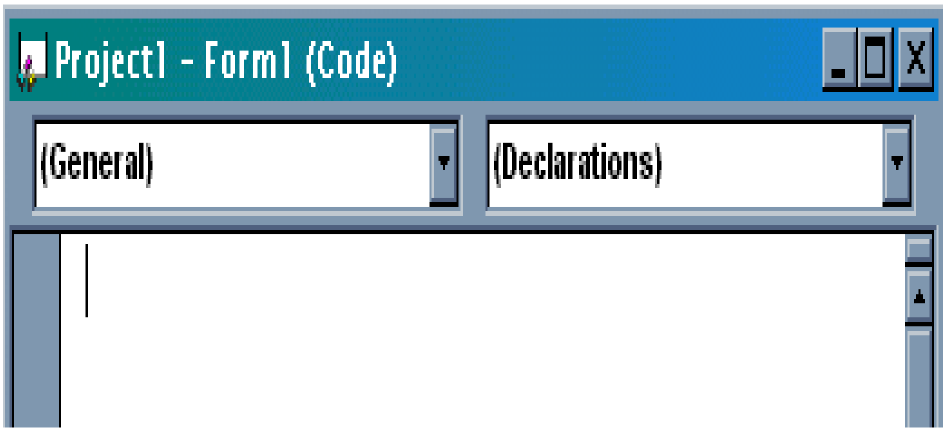


Figure 11: The declarations

You will notice that your cursor is not surrounded by any code as with usual event procedures, such as "Public Sub cmdVariableCount_Click()". That is because you have reached a special place in your code window–the top–where you can create, or "declare" variables.

To declare a variable you use the syntax:

Dim MyVariable as DataType

...where MyVariable is the name of your variable and DataType is the type of information it should hold. Here is a short list of the most common DataType values:

String - If you are going to store text in the variable.

Integer - Used when storing a number.

Boolean - Used when storing a True/False value.

Date - Used when story a date value.

Decoding declarations...

A variable is just something in which we can store a certain type of information.

Loops

This Loop can be used to display a message box for every single doctor in HMS. Two ways can be used – either manually code every single message box or use a loop that says, in English, "display a message box for every Doc".

Creating a loop is simple. Let us look at a sample piece of code.

For i = 1 to 5

```
Msgbox  "I have looped round  " & i & " times"

Next i
```

The first line tells Visual Basic how many times it should loop around. It is saying – loop around this bit of code five times and store the number of times it has looped around in “i”. The second line is just a simple message box that shows “i” – i.e., just the number of times it has looped around. The third line, “Next I” just tells it to go back to the beginning until it is gone from one to five.

Do Until

“Do Until” loop is used when you may want to keep asking a user for a student name until they type in "END" or something.

```
Dim Ename as String

Do Until Ename = "END"

Ename = InputBox ("Enter the Employee name, or type END to finish:")

Msgbox "New employee is entered - " & Ename

Loop
```

The first line merely declares a variable called “Ename” to hold a string (that is, plain text). The second line tells Visual Basic to loop around until the “Ename” variable is equal to “END”.

The third line introduces another function called the InputBox and is used to get information from the user. On running this, a little box will appear for the user to enter some information–in this case, an Employee name. Whatever they enter will be assigned to “Ename”.

And the fourth line just says – loop back to the start again. And at the start, Visual Basic asks the question once more – does “Ename” equal “END”? If it does, then looping stops. If not – looping goes on!

Subs

What are subs, and how can they help VB coders simplify their source?

A sub is basically a piece of code separate from the rest. It is often used when you have a piece of common code that you do not want to rewrite each time you use it.

You can get around these using subs. You can create a singular, generic chunk of code that displays the message and then ends your program.

The ability to create such subs can be a real time saver! It avoids unnecessary code and therefore cuts down on the size of your final program. It also saves on boring maintenance time.

Exercise 3: Create forms for HMS. With this basic information you should try to work with some of the components of Visual Basic by adding various components and create the necessary forms. If you are unable to do so then you may either use MS-Access forms or read through further readings. The forms for HMS that should be designed by you can include:

- Form for data entry information of Doctors, patients, employees and departments.
- Forms for display of detailed information.
- Forms for editing this information.
- Any other form that may be required for the modified HMS by you.

5.0 REPORTS

A report is an effective way to present data in a printed format. Because you have control over the size and appearance of everything on a report, the information can be displayed the way you want to see it. We are demonstrating report creation through MS-Access report facility.

5.1 Creating a Report using the Wizard

As with the Form, a report can be created using the Wizard, unless you are an advanced user.

1. Switch to the Database Window. This can also be done by pressing F11 on the keyboard.
2. Click on the **Reports** button under **Objects** on the left side of screen.
3. Double click on **Create Report Using Wizard.**
4. On the next screen select the fields you want to view on your form. All of them could be selected.
5. Click Next
6. Select if you would like to group your files. Keep repeating this step for as many groupings as you would like.
7. Click Next
8. Select the layout and the paper orientation desired
9. Click Next
10. Select the style desired.
11. Click Next
12. Give a name to the report, and select **Preview the Report.**
13. Select **Finish**
14. The report is displayed. To adjust the design of report, press the design button and adjust the report accordingly.

5.2 Creating Mail Merge Labels using a Wizard

Mailing Labels can be created using Microsoft Access for the database. To do this:

1. Switch to the Database Window. This can be done by pressing F11 on the keyboard.
2. Click on the **Reports** button under **Objects** on the left side of screen.
3. Click on **New.**

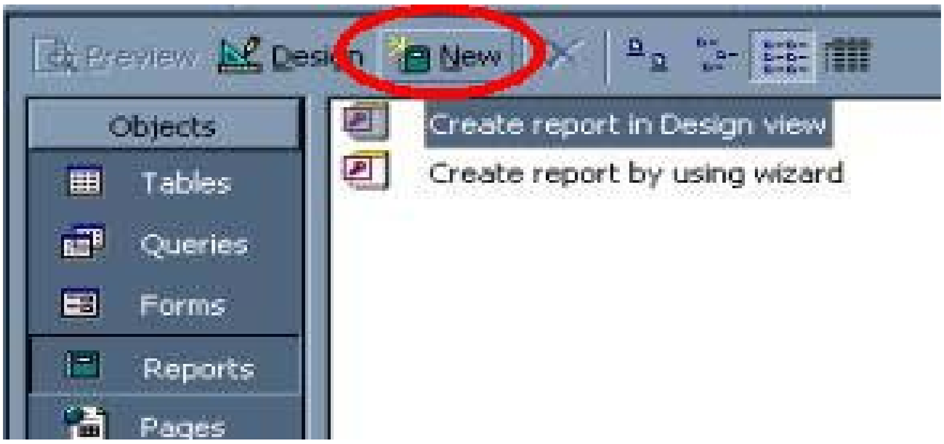
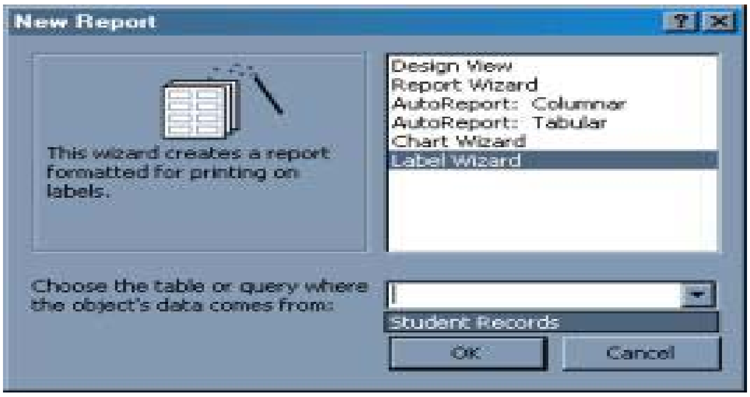


Figure 12: Creating reports

Select **Label Wizard** and the table from which the information is taken.



New Report

This wizard creates a report formatted for printing on labels.

Choose the table or query where the object's data comes from:

Student Records

Figure 13 : Creating report using label wizard

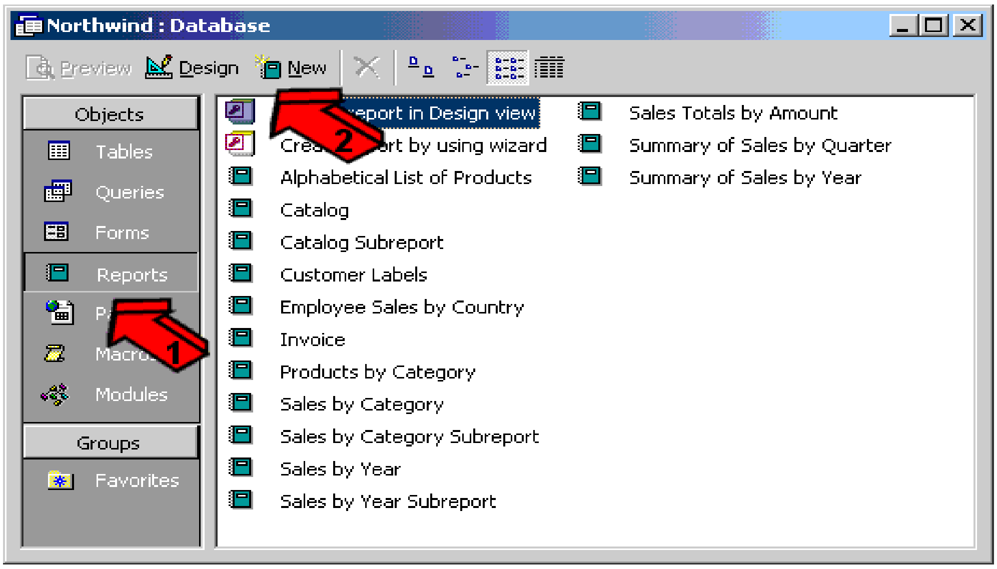
Click OK
Select the layout of the labels
Click Next
Select the font size and required color on each label
Click Next
Select the layout of the label
Click Next
Select how you want your labels sorted
Name the label report and preview it

5.3 Creating a Report using the Access Report Wizard

Microsoft Access provides many built-in wizards which allow one to create database objects, such as reports, quickly and easily. The following steps demonstrate the process of creating a very simple report.

1. **Choose the Reports Menu:** Once a database is opened, the main database menu as shown below is displayed. Click on the "Reports" selection and a list of the various reports Microsoft included in the sample database is displayed. Double-click on a few of these and get a feel for what reports look like and the various types of information that they contain.

2. **Create a new Report:** Click on the "New" button and the process of creating a report from scratch will start.



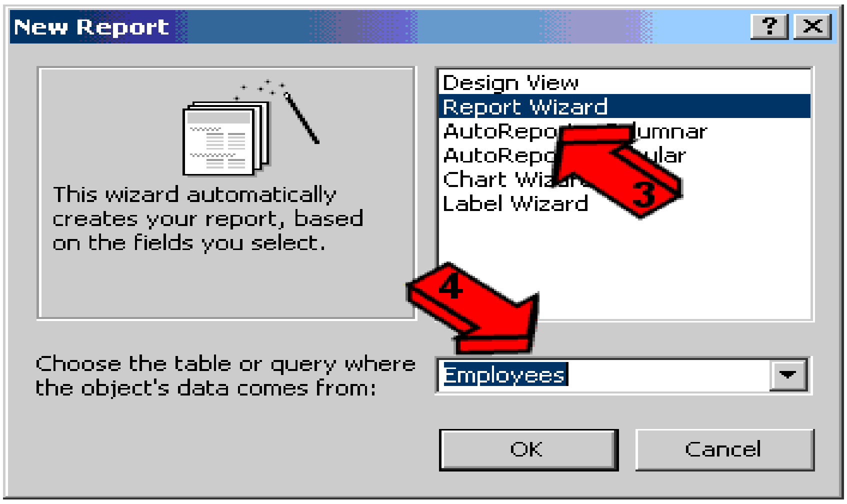
Northwind : Database

eport in Design view
Crea rt by using wizard
Alphabetical List of Products
Catalog
Catalog Subreport
Customer Labels
Employee Sales by Country
Invoice
Products by Category
Sales by Category
Sales by Category Subreport
Sales by Year
Sales by Year Subreport
Sales Totals by Amount
Summary of Sales by Quarter
Summary of Sales by Year

Figure 14: Creating a new report

3. **Select the Report Wizard:** The next screen that appears will require the selection of the method to be used to create the report. Select the Report Wizard, which will help through the creation process step-by-step.

4. **Choose a Table or query:** Choose the source of data for the report. If the information is to be retrieved from a single table, select it from the drop-down box below. Alternatively, for more complex reports, the report can be based on the output of a query previously designed. For example, all of the data we need is contained within the Employees table, so choose this table and click on OK.



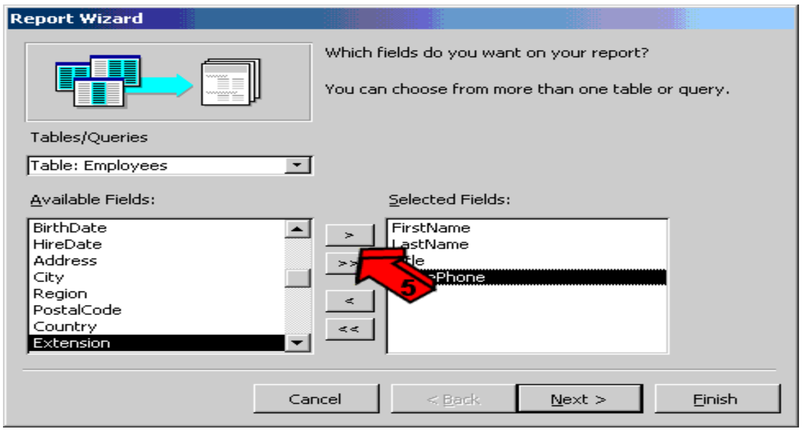
This wizard automatically creates your report, based on the fields you select.

Choose the table or query where the object's data comes from: Employees

Figure 15: Selecting creation method

5. **Select the fields to include:** Use the '>' button to move over the desired fields. Note that the order in which the fields are placed in the right column determines the default order they will appear in the report. For creating a simple report on home phones of Hospital Employees, even the information contained in it is kept simple -- the first and last name of each employee, the title and the home telephone number. Select the fields and click the Next button.

5. **Select the grouping levels:** At this stage, you can select one or more grouping levels to refine the order in which our report data is presented. For example, we may wish to list home phone numbers of the employees by department so that all of the members of each department are listed separately. Click on the Next button to bypass this step and return here later.



Which fields do you want on your report?

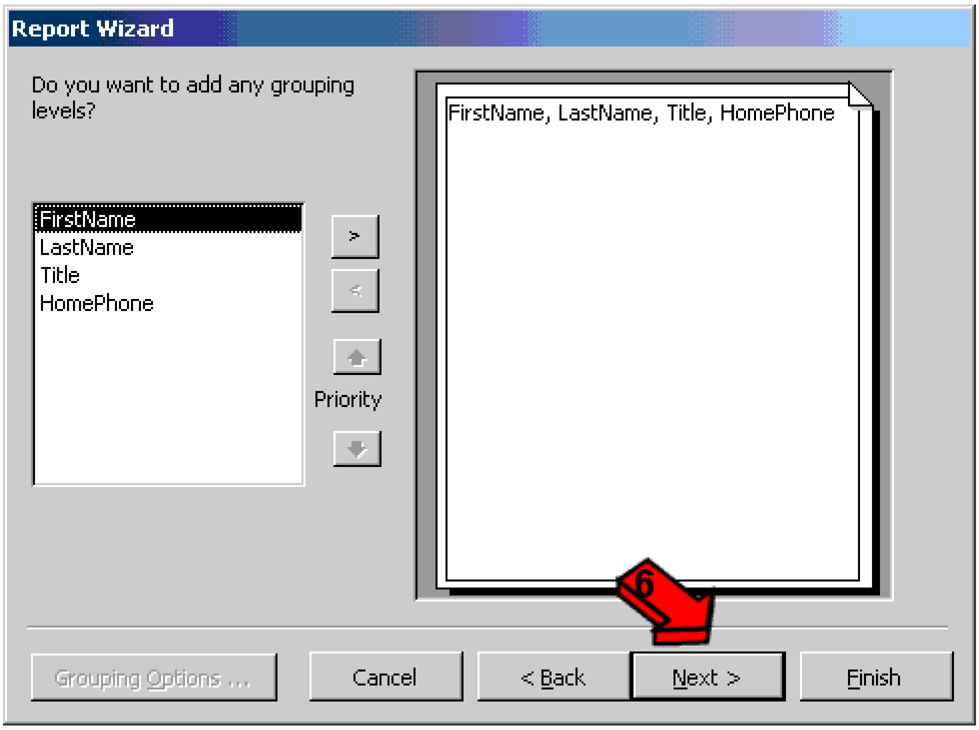
You can choose from more than one table or query.

Tables/Queries: Table: Employees

Available Fields: BirthDate, HireDate, Address, City, Region, PostalCode, Country, Extension

Selected Fields: FirstName, LastName, Title, HomePhone

Figure 16: Selecting the fields to include



Report Wizard

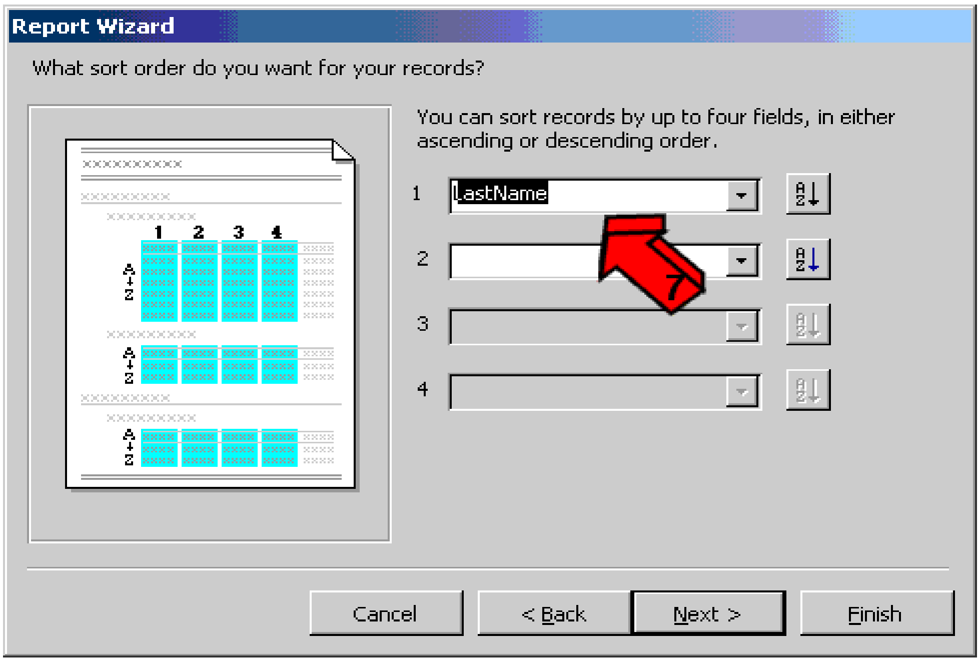
Do you want to add any grouping levels?

FirstName
LastName
Title
HomePhone

FirstName, LastName, Title, HomePhone

Figure 17: Choosing the grouping levels

7. **Choose the sorting options:** In order to make reports useful, results can be sorted by one or more attributes. A list of Doctors or employees can be sorted by the last name of each employee. Select this attribute from the first drop-down box and then click the Next button to continue.



Report Wizard

What sort order do you want for your records?

You can sort records by up to four fields, in either ascending or descending order.

1 LastName

2

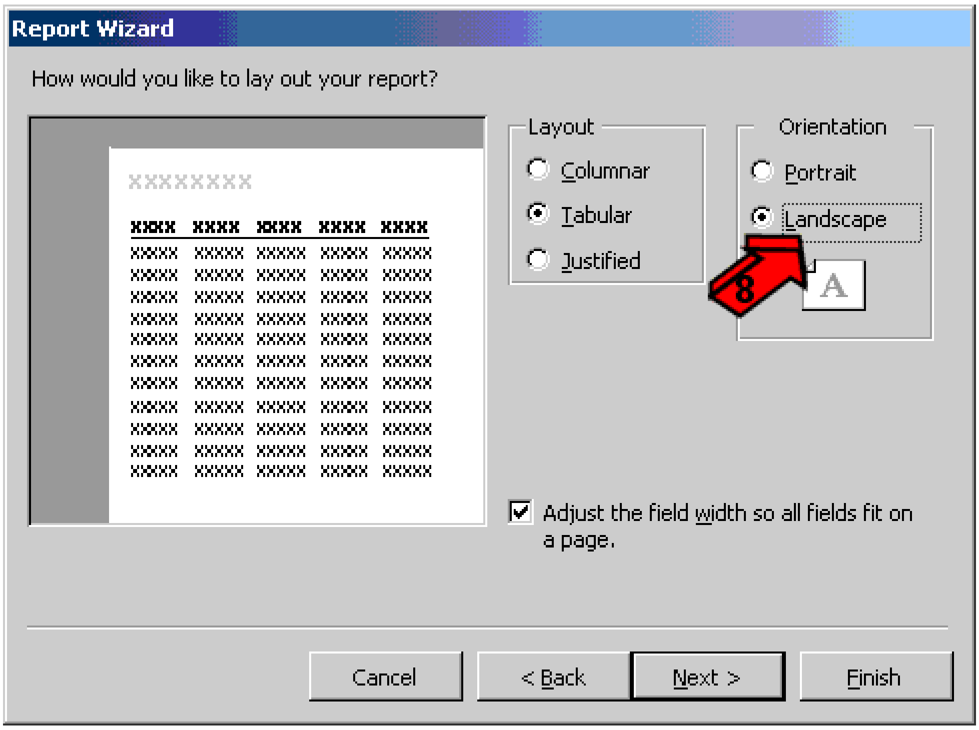
3

4

Figure 18: Choosing the sorting options

8. **Choose the formatting options:** In the next screen, some formatting options are given. The default tabular layout can be accepted but can be changed according to needs. Click the Next button to continue.

9. **Select a report style:** The next screen is to select a style for the report. Click on the various options and a preview of the report in that style can be seen in the left portion of the screen. Select the option and then click the Next button to go to the next screen as given below.



Report Wizard

How would you like to lay out your report?

Layout

○ Columnar

● Tabular

○ Justified

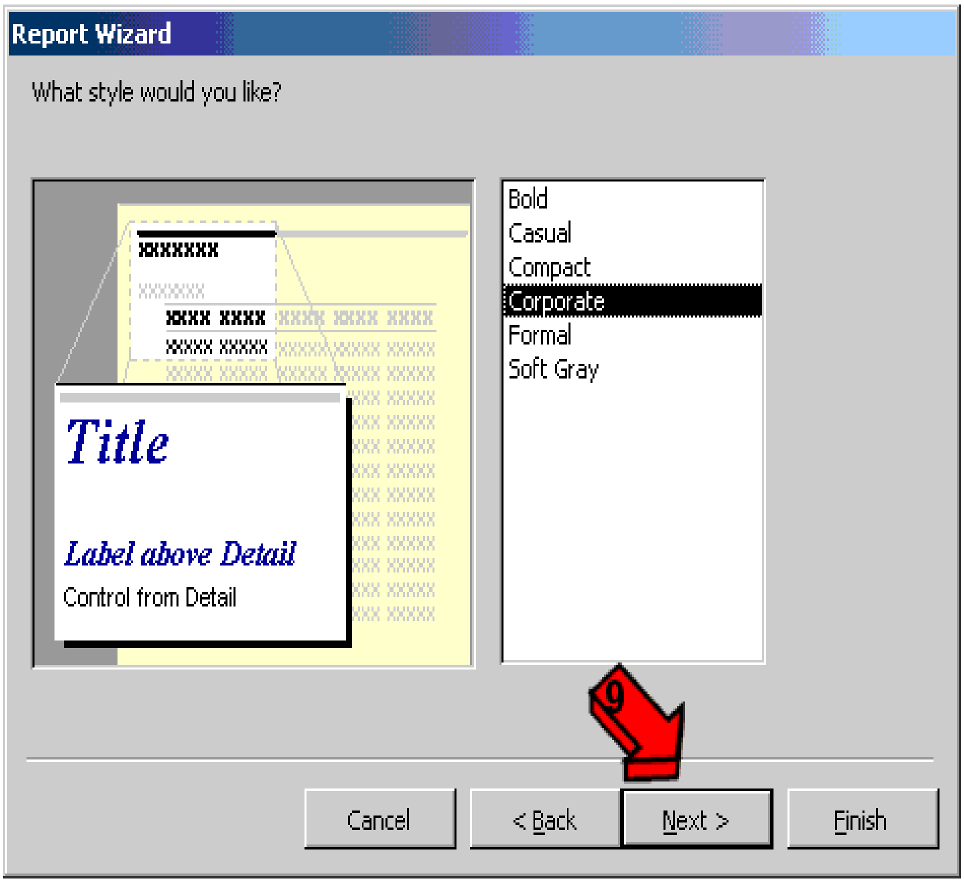
Orientation

○ Portrait

● Landscape

☑ Adjust the field width so all fields fit on a page.

Figure 19: Choosing formatting options



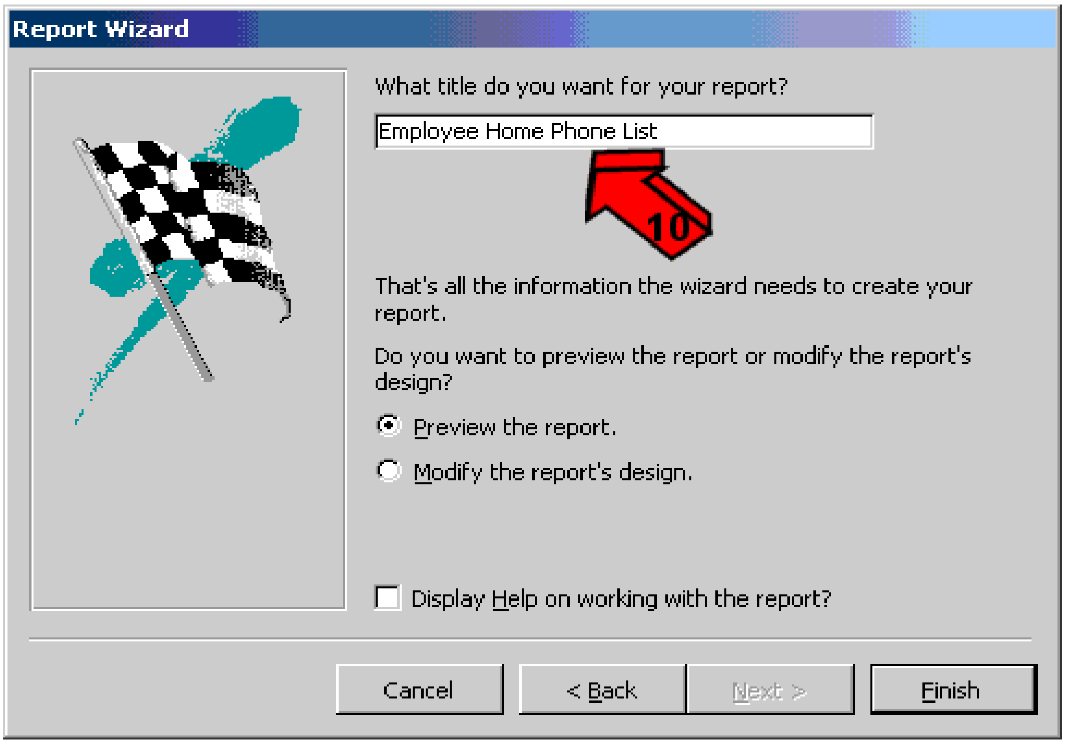
Report Wizard

What style would you like?

Bold
Casual
Compact
Corporate
Formal
Soft Gray

Figure 20: Selecting a report style

10. **Add the title:** Finally, a title is to be given to the report. Access will automatically provide a nicely formatted title at the top of the screen, with the appearance shown in the report style selected during the previous step. Enter a title for the report, select the "Preview the report" option and then click on Finish to see the report.



Report Wizard

What title do you want for your report?

Employee Home Phone List

That's all the information the wizard needs to create your report.

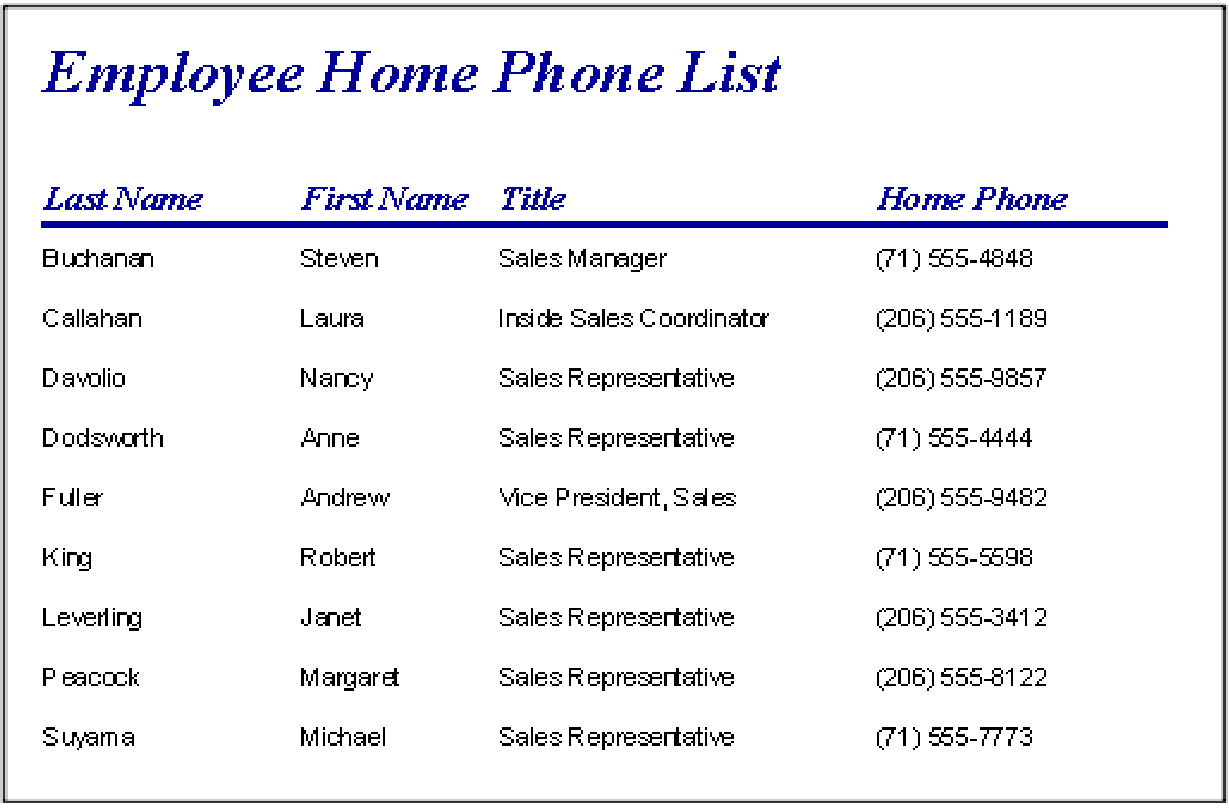
Do you want to preview the report or modify the report's design?

● Preview the report.

○ Modify the report's design.

□ Display Help on working with the report?

Figure 21: Adding a title



Employee Home Phone List

Last Name	First Name	Title	Home Phone
Buchanan	Steven	Sales Manager	(71) 555-4848
Callahan	Laura	Inside Sales Coordinator	(206) 555-1189
Davolio	Nancy	Sales Representative	(206) 555-9857
Dodsworth	Anne	Sales Representative	(71) 555-4444
Fuller	Andrew	Vice President, Sales	(206) 555-9482
King	Robert	Sales Representative	(71) 555-5598
Leverling	Janet	Sales Representative	(206) 555-3412
Peacock	Margaret	Sales Representative	(206) 555-8122
Suyama	Michael	Sales Representative	(71) 555-7773

Figure 22: A sample report

The final report should appear similar to the one presented above but having the data you might input.

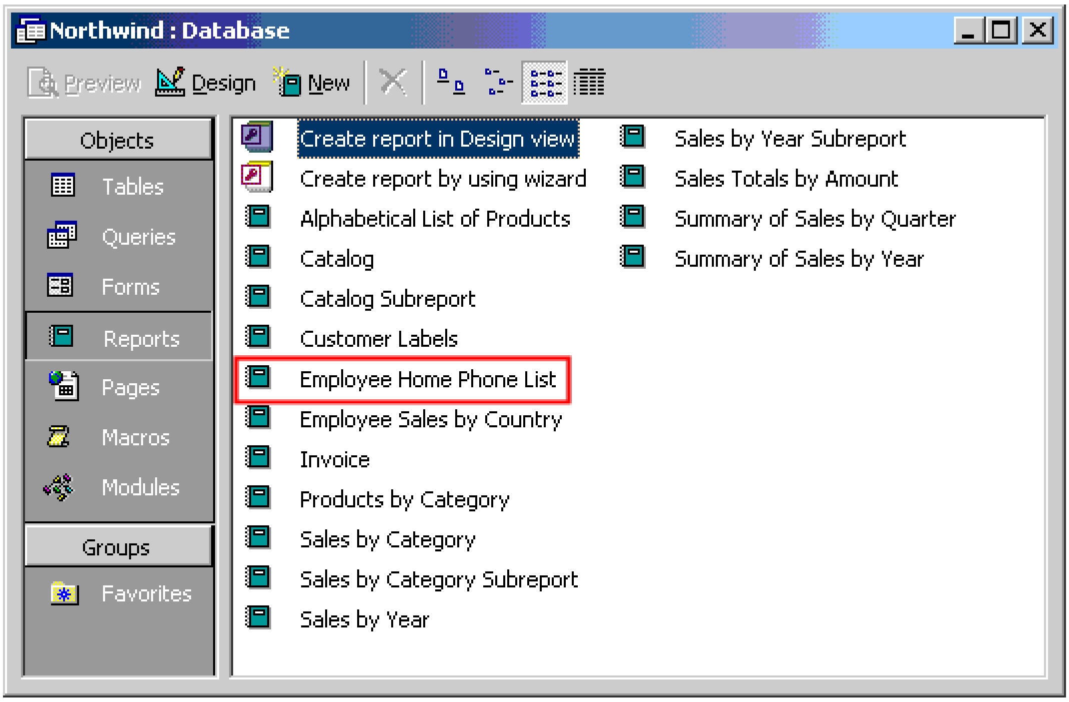
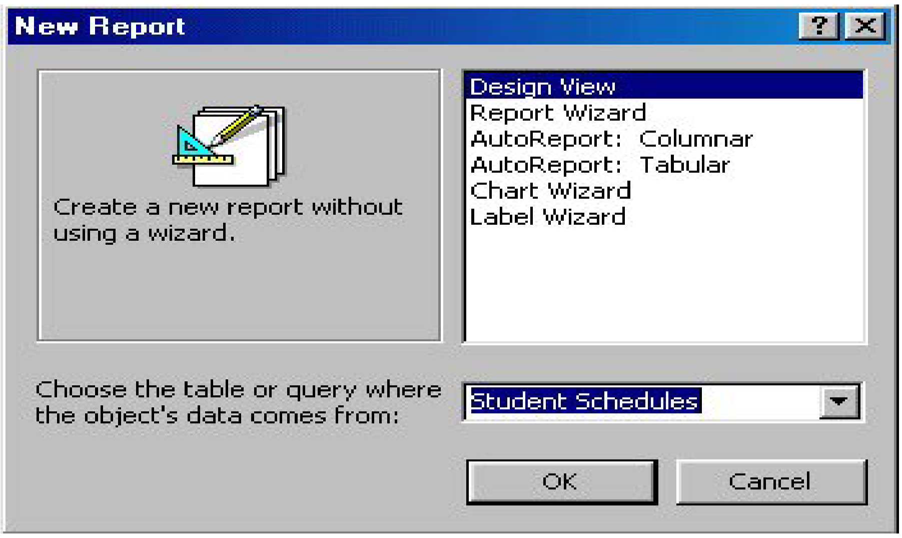


Figure 23: Updated reports menu

5.4 Creating Report using Design view

Access contains wizards that make creating reports very easy. However, sometimes wizards do not create the report the way you want it, or you may need to make modifications. With this in mind, it is helpful to know a little bit about creating a report from scratch.

1. Click on Reports, New, Design View. Click on the list arrow and choose the First Period query. Click on OK.



Create a new report without using a wizard.

Choose the table or query where the object's data comes from: Student Schedules

Figure 24: An initial screen of design view

2. The report will open in Design view. If the Toolbox does not appear, click on the Toolbox icon (wrench and hammer).

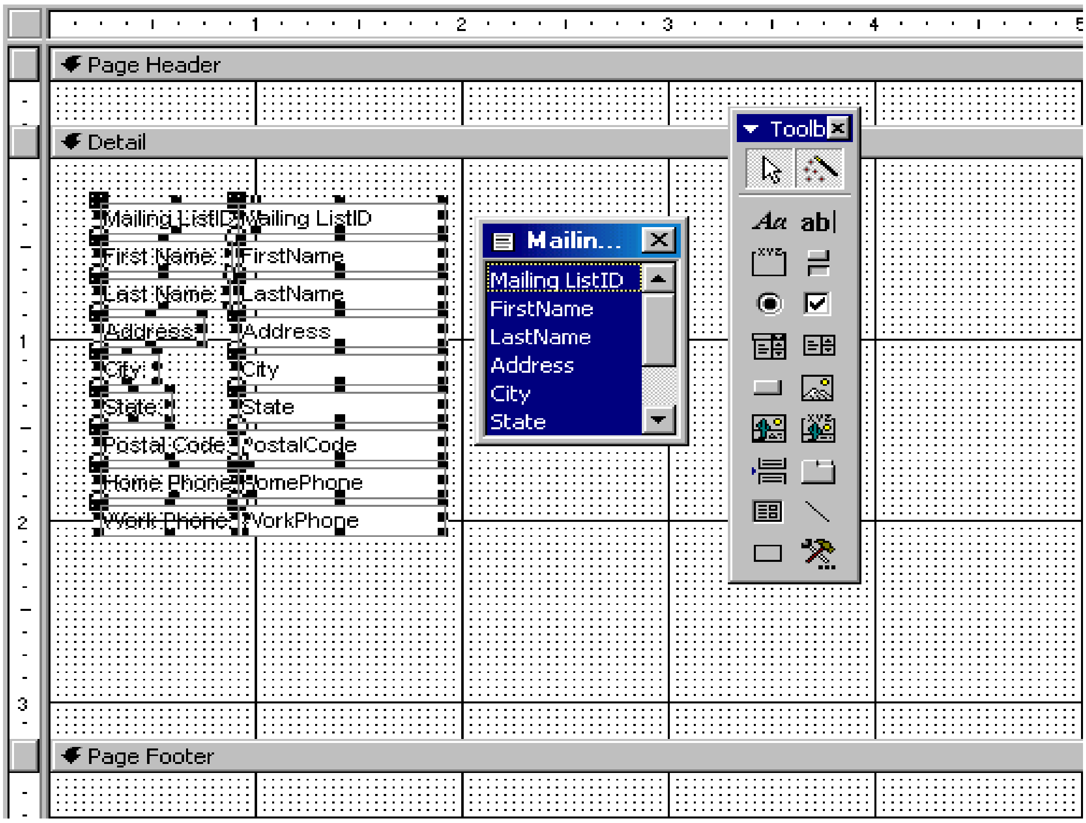


Figure 25: A sample report creation

3. At this point, the report has 3 sections: Page Header, Detail, and Page Footer. Other sections that can be added include Report Header, Report Footer, Grouping Header and Grouping Footer. The characteristics of each section vary:

Report Header

Appears at the top of the ***first*** page of the report. The title of the report usually goes in the report header.

Page Header

Appears at the top of ***every*** page of the report. Column headings usually go in the page header.

Grouping Header

If grouping has been selected, the grouping header may appear at the top of each new group. Grouping headers are optional.

Detail Section

This section contains the field values from the table or query that the report is based on.

Grouping Footer

If grouping has been selected, the grouping footer may appear at the bottom of each new group. Grouping footers are optional. Subtotals might be placed in the grouping footer.

Page Footer

Appear at the bottom of ***every*** page of the report. Page numbers usually go in the page footer section.

Report Footer

Appear at the bottom of the *last* page of the report. A grand total should be placed in the report footer.

Note that the sections at the far right side of this screen capture of a report. Grouping footers and page footers were not used.

4. Click on View in the menu bar, then Report Header/Footer to add these sections.

We will not use the page footer, so move the mouse to the top of the report footer bar; the mouse will change to a double-headed arrow.

Click and drag the report footer bar up until it meets the bottom of the page footer bar. This closes the page footer bar so that it will not take up unnecessary space. Your screen should look like this:

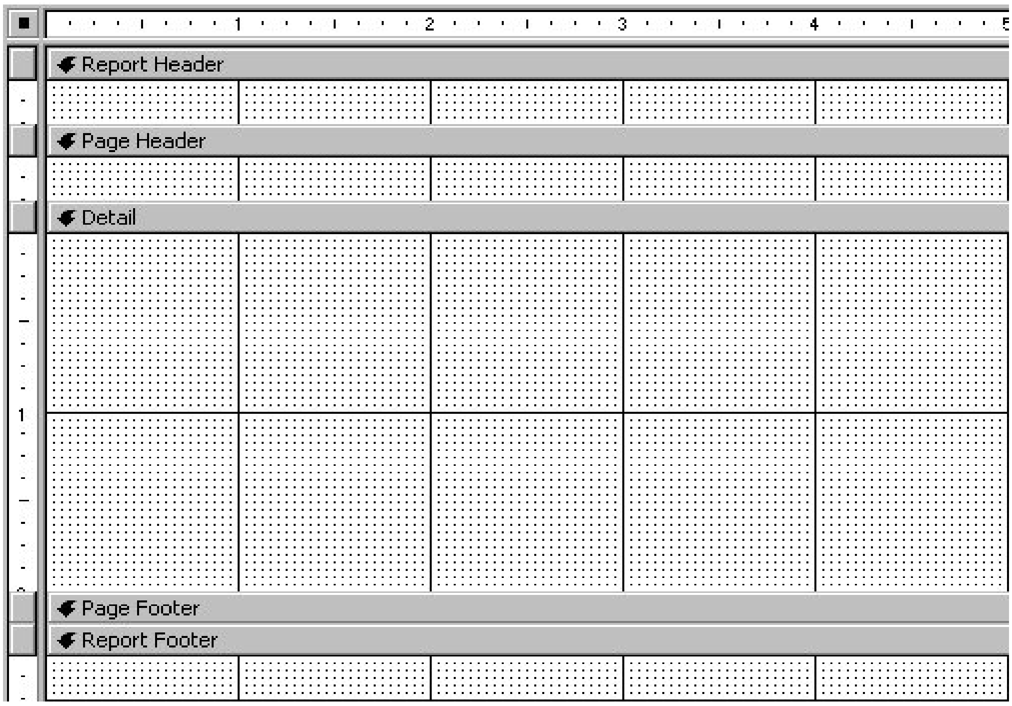
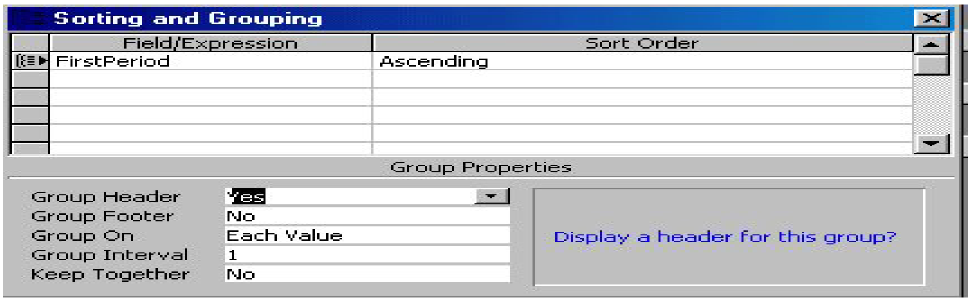


Figure 26: Report with header and footer

5. Next, we need to add the Sorting/Grouping Header. Click on the Sorting/Grouping icon on the toolbar. We want to group on the First Period class, so click on the list arrow at the right side of the field expression box and select First Period. Click in the group Header area, click on the list arrow, and select Yes. This adds the group header.



Field/Expression	Sort Order
FirstPeriod	Ascending

Group Properties

Group Header	Yes
Group Footer	No
Group On	Each Value
Group Interval	1
Keep Together	No

Display a header for this group?

Figure 27: Adding sorting for date in report

6. Close the Sorting and Grouping window. You should now have the following sections in your report: report header, page header, first period header (this is the grouping header - it takes its name from the grouping field we selected), detail, page footer (which we have closed up), and report footer.

7. Now we can begin adding the fields into the report. Click on the Field List icon, and the list of available fields from our First Period query should appear. Click on the first field, SSN, and drag it down into the detail section. Repeat for FirstName, LastName and FirstPeriod. It really doesn't matter where you put them, but try to line them up something like this:

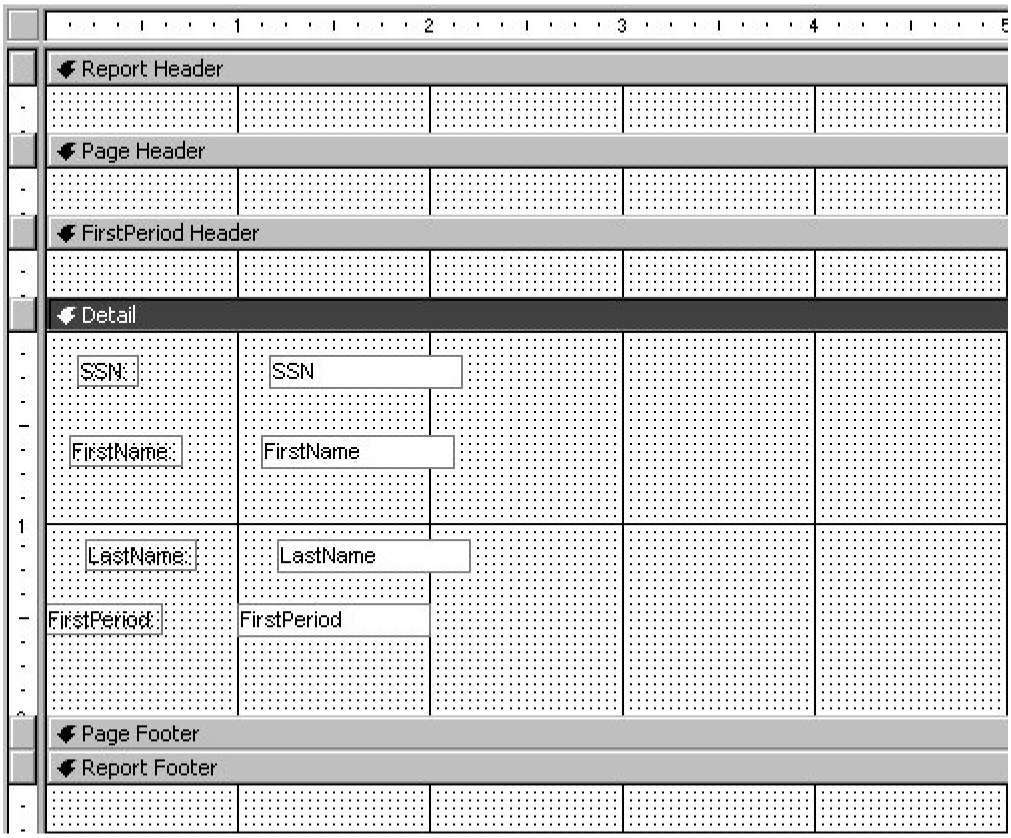


Figure 28: A report after adding fields

8. Close the Field List window. We have all of the fields we need.

9. The items in the detail section are called ***controls***, and in this case, they are ***bound*** controls. This means that they are directly linked to the underlying table or query. Bound controls consist of two parts, the label box, which in our example is on the left, and the text box which is on the right. You can change or delete the label box and the report will still run. However, you cannot change the contents of the text box. If you do this, you have changed the name of the field that the report is searching for in the underlying table or query, and the report will not work.

10. Each set of controls can be moved separately or together. Click on the SSN text box control (the one on the right). A set of handles should appear on that control, and a single handle should appear on its associated label control. Notice that the first item on the left of the formatting toolbar says SSN. This is the object list. Click on the list arrow on the object list, and you'll see all the other objects and their labels. Click on some of the other objects and watch the handles move to select those objects. You can use this object list as a visual clue to determine if you have a text box or a label selected. Text boxes will show the field name in the object list; labels will show label x (where x is some number). Hopefully, this will help to prevent you from making changes to contents of the text box controls.

11. Click back on the SSN text box control, then move your mouse to the top of the text box control. Your mouse pointer should change to a hand. If you click and drag. the hand allows both the text box and the label to move ***together***, keeping their relationship to each other. Now move the mouse to the larger handle located at the top left of either the text box or the label. Your mouse should change to a pointing finger. Clicking and dragging with the finger pointer will move ***only that control***.

All of the labels in our report are at the top of the page in the page header, so we will need to move them there. You can not click and drag across the boundaries of a section, so we will cut and paste instead. You can select more than one control at a time by clicking on the first control, then holding the Shift key down and clicking on the other controls.

12. Click on the SSN label box, hold Shift down, and click on the other label boxes to select them. When you finish, there should be handles on all of the label boxes.

Click on the cut icon.

Click anywhere in the Page Header section, then click on paste. The labels should move to the Page Header section. Your screen should look something like this:

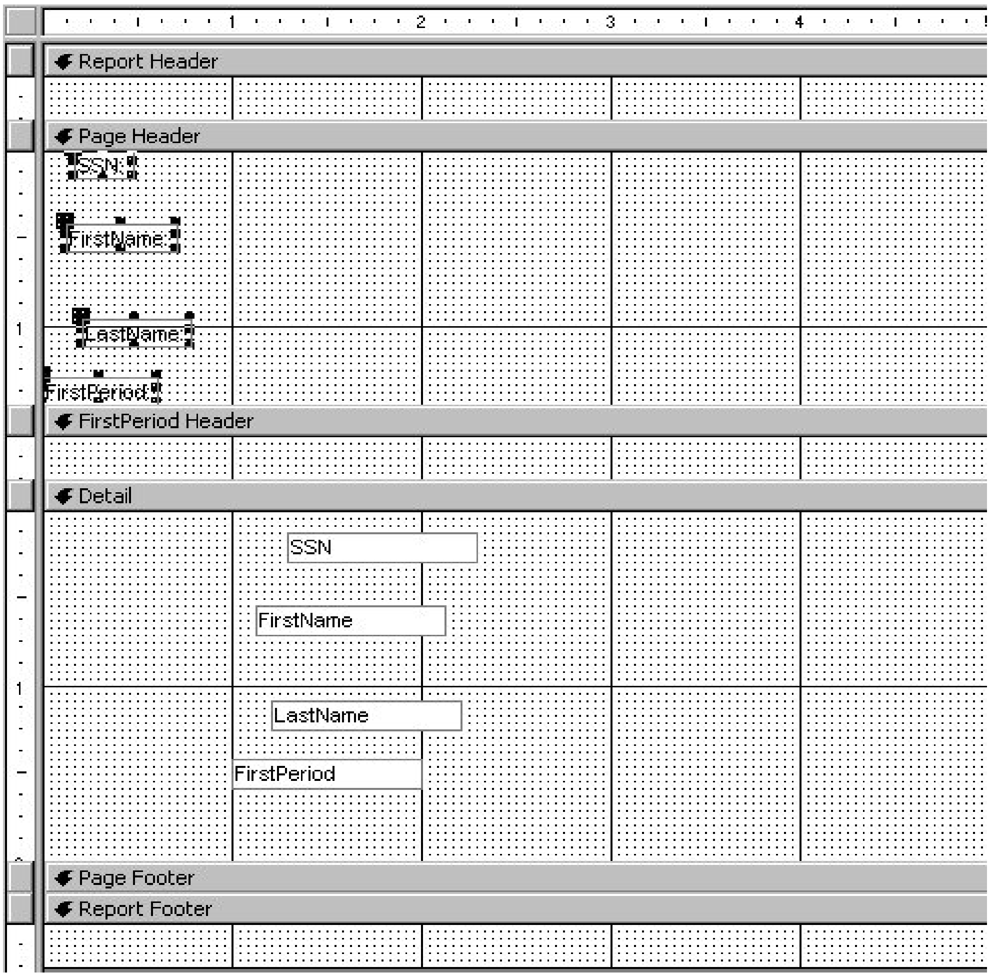


Figure 29: Reports with labels on page header section

Click anywhere in the Page Header section to deselect the labels.

13. By clicking and dragging, begin aligning the labels in the Page Header section with the text boxes in the Detail section. You can check your progress by changing to Print Preview and back to Design View. Use the first icon on the left to change views.

This is a tedious process - sometimes you have to go back and forth many times to get things aligned just right. Using the Grid dots is helpful, but the Align tool can help speed up the process.

Select the SSN label in the Page Header, hold Shift down, then select the SSN text box in the Detail section.

Click on Format, Align, and choose which side to align on. This will align both controls at whatever side you choose.

Try to make your report look like this:

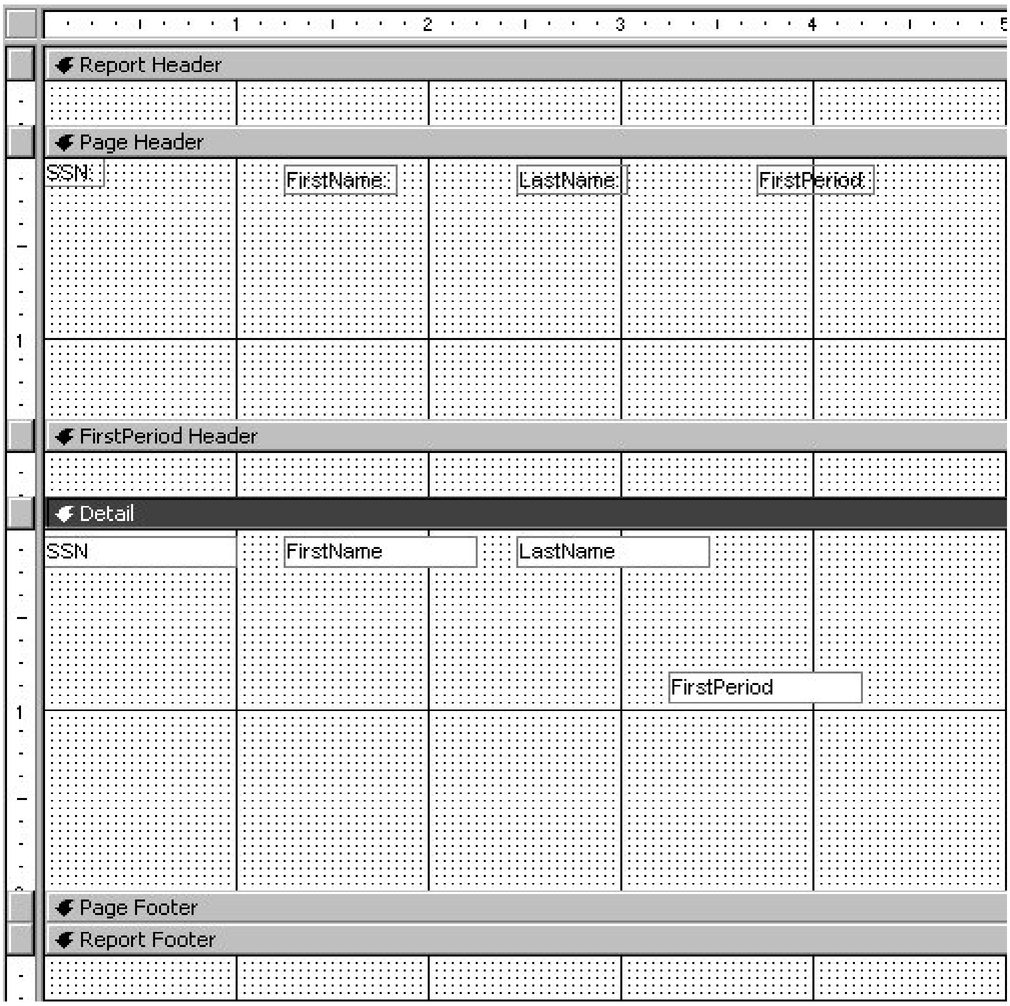


Figure 30: An aligned report

14. You must leave enough space between the controls, especially in the detail section, for the student's entire SSN, first name and last name to show. This may mean that you need to increase the size of the controls, which you can do by clicking and dragging on the handle on the right side of a control. Change to the Print Preview and scroll through the pages of the report. You can change from page to page by clicking on the navigation buttons in the bottom left corner of Print Preview.

Notice that neither of the FirstPeriod controls is aligned. These will be moved into the FirstPeriod Header in the next step.

Select the FirstPeriod label, which is now in the Page Header section. Cut and paste it into the FirstPeriod Header section. Do the same for the FirstPeriod text box that is in the Detail section. Position them at the far left of the grid, using the Align command to adjust them vertically. Chances are they will paste right on top of each other, so try to use the finger pointer to get them separated. Now the Report Design looks something like this:

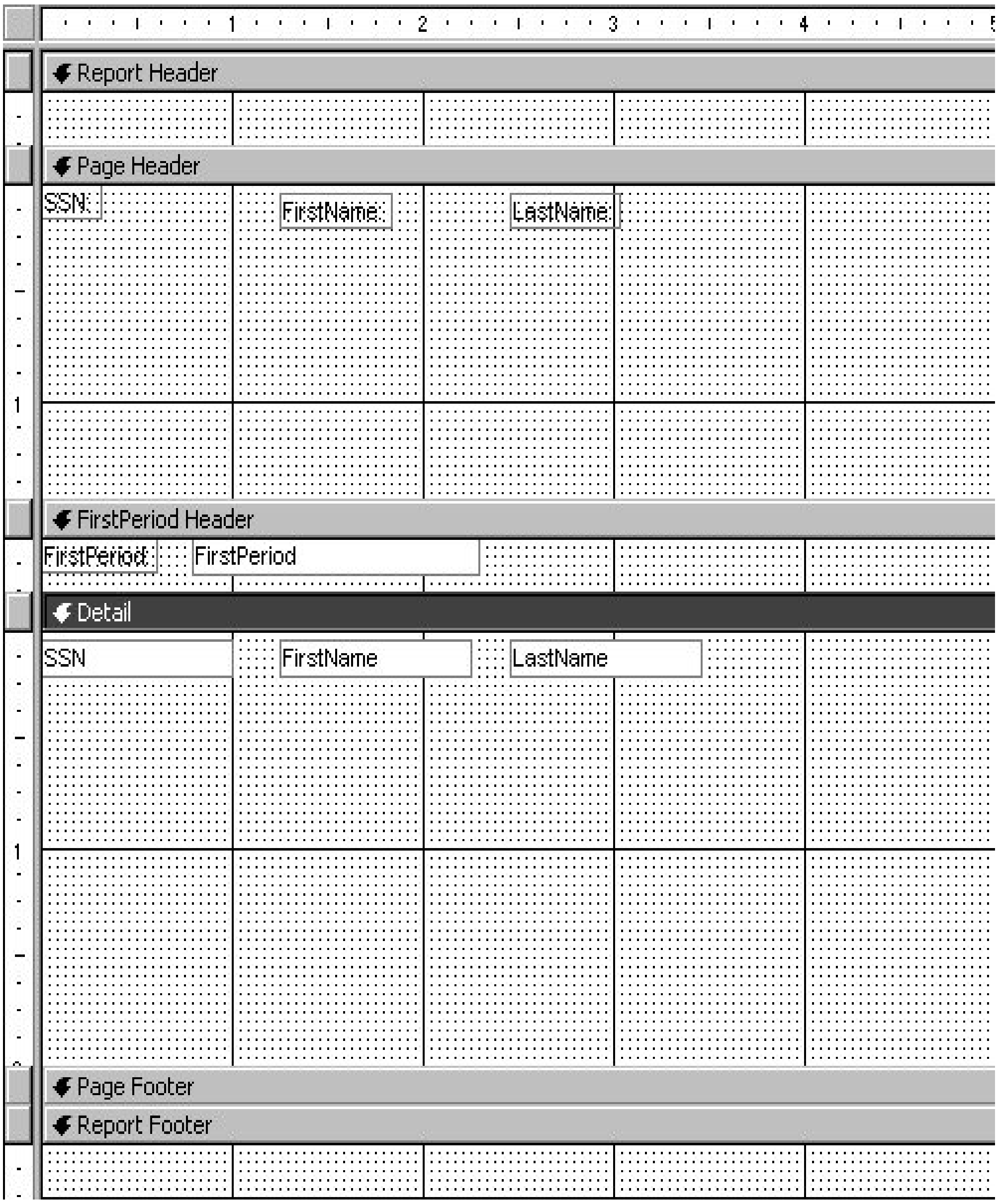


Figure 31: A report having periods

15. If you change to the Print Preview, alignment is probably close, but there is a great deal of blank space in the report. In the next step, we will close up the sections to get rid of the white space in the report.

Move your mouse to the top of the Page Footer bar; it should change to a double-headed arrow. Drag the Page Footer bar up until it almost touches the bottom of the text boxes in the Detail section. Repeat for at the top of the FirstPeriod Header bar - drag it up until it almost touches the bottom of the label boxes in the Page Header section. Your design should look something like as follows.

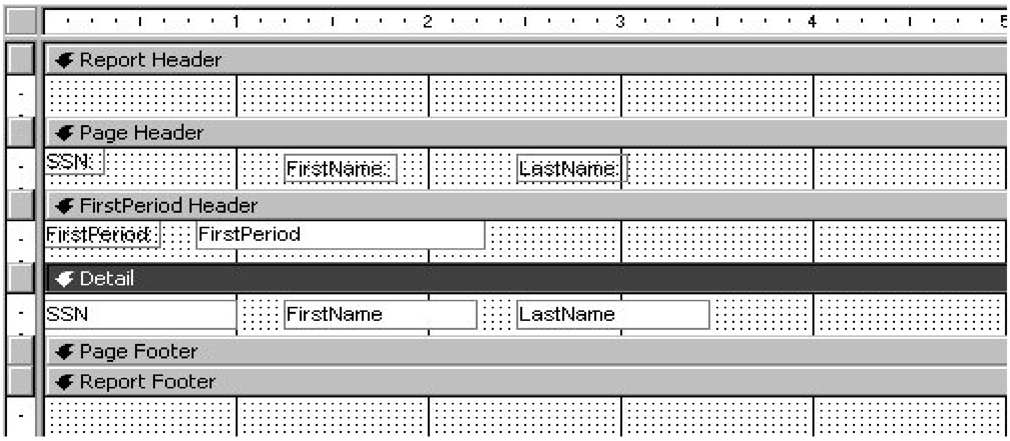


Figure 32: A final report layout

When you change to Print Preview, your report should have shrunk down to one page. However, it is hard to distinguish where you change from one class to another, so in the next step we will add a line and some formatting to make the report easier to read.

16. If your toolbox is not visible on your screen, click on the toolbox icon. The toolbox should appear as a floating toolbar.

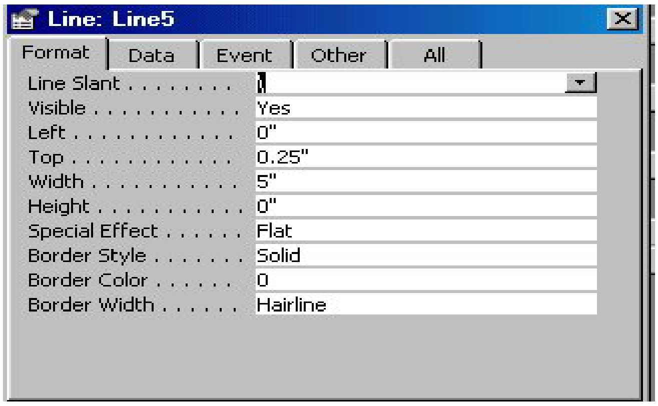
The toolbox contains icons for many things, including some ***unbound*** controls such as lines and labels. Unbound controls are in no way connected to the underlying table or query. You can move them or change them to suit you without worrying.

Click on the line icon on the toolbar. Drag your mouse just under the controls in the FirstPeriod Header to draw the line. If you hold Shift down while you drag, you will be assured of a straight line.

Preview the report. The line helps, but it would be better if it were thicker, if the controls were bold and if there was a space between First and Period. Go back to the Design View to make these changes.

17. Point to the line and click to select it. If this proves too difficult, remember you can use the Object list at the left side of the formatting toolbar. Click on the list arrow and select the line (it should be the only one, so it doesn't matter what number it is). Once the line is selected, the handles should appear on the line.

Each control has ***properties***. Click on the Properties icon on the Report Design toolbar. Since the line was selected, the Line Properties window should appear.



Line Slant . . . . \
Visible . . . . Yes
Left . . . . 0"
Top . . . . 0.25"
Width . . . . 5"
Height . . . . 0"
Special Effect . . . . . . Flat
Border Style . . . . Solid
Border Color 0
Border Width . . . . . . Hairline

Figure 33: Setting properties in report

Click on the Border Width box, then click on the list arrow that appears to the right, and select 2 pt. This will make the line fatter and easier to see. Preview the report.

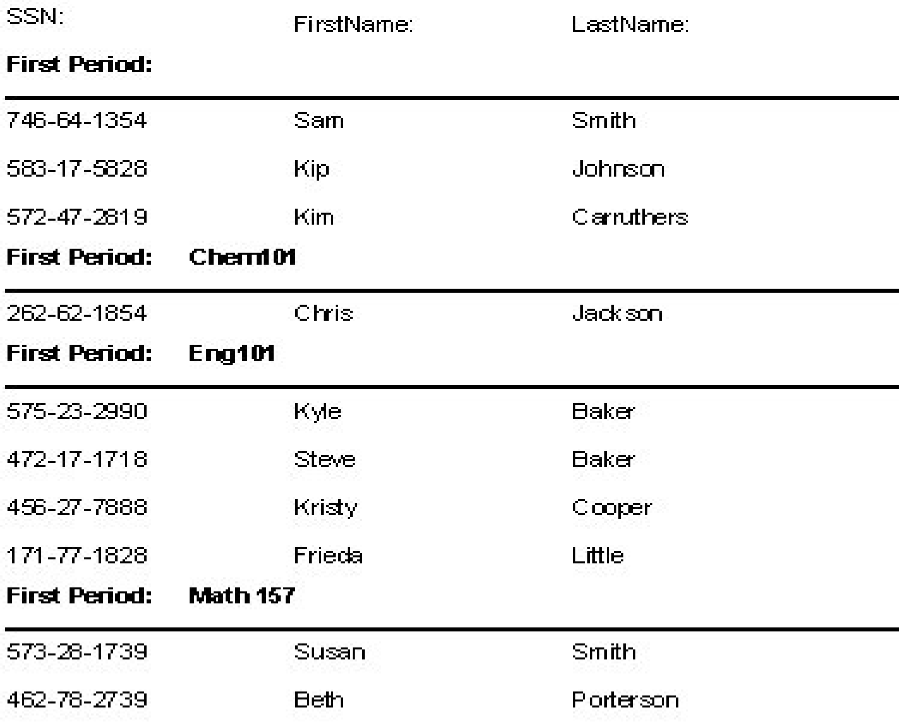
18. Change back to the Design View. Select both the FirstPeriod label and text box (remember to hold shift down), then click on the Bold icon. Preview the report to see the difference.

Change back to the Design View. If both the label and text box are still selected, click somewhere to deselect. Then select only the label box (be sure about this!) We need to put a space between First and Period. There are two ways to do this.
Either select the label box, click on Properties, then change the caption to read First Period, or Click once on the label box, then click again (2 single clicks, not a double-click). This should put you into the Edit mode for the label. Put a space between First and Period.

Either method will work. It is important to remember, though, you are doing this on the label, **NOT** the text box. If you ever should change a text box beyond repair, delete the text box and its corresponding label, click on the Field List icon, and drag the field back into the Report. You will then have to move and edit it back to its proper location and formatting.

A good clue that you have changed a text box is this: when you preview the Report, you get a Parameter message. This is usually what happens when Access cannot find a bound control that is in the report, but not in the underlying table or query.

At this point, the report should look something like this in Print Preview:



SSN:	FirstName:	LastName:
First Period:		
746-64-1354	Sam	Smith
583-17-5828	Kip	Johnson
572-47-2819	Kim	Carruthers
First Period: **Chem101**		
262-62-1854	Chris	Jackson
First Period: **Eng101**		
575-23-2990	Kyle	Baker
472-17-1718	Steve	Baker
456-27-7888	Kristy	Cooper
171-77-1828	Frieda	Little
First Period: **Math 157**		
573-28-1739	Susan	Smith
462-78-2739	Beth	Porterson

If the lack of space between Chem and 101 or Eng and 101 bothers you, you would need to go back to the Student Schedule table and change all the records, adding spaces appropriately.

19. The report would look better with a title. We will put the title in the Report Header, so that it will appear only once, at the beginning of the report.

Click on the Label tool on the Toolbox.

Drag and draw a box in the Report Header section. The size of the box really does not matter – we can adjust it later.

As soon as you leave your finger off the mouse, you will go right into editing the contents of the label. Key in First Period Student Roster, then click some place outside of the box to get out of the editing mode. Preview the report. The title needs to be bigger and bolder. Go back to the Design View.

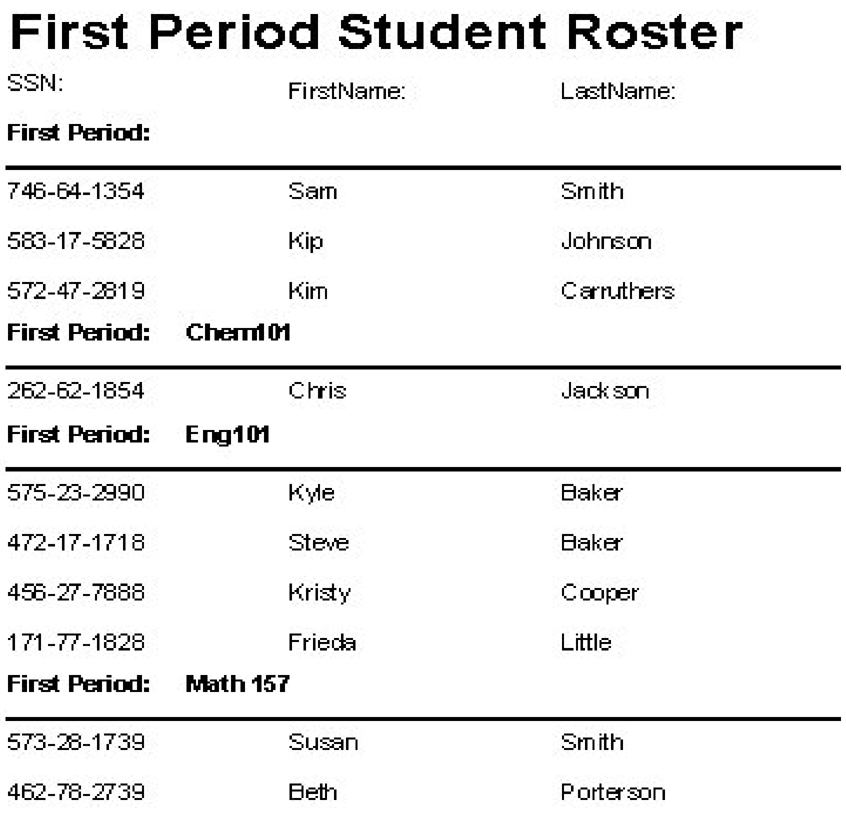
Select the label box; handles should appear around it. Select 18 pts and Bold on the Formatting toolbar. However, as soon as you do this, the text becomes too big for the box.

Click on Format, Size, To fit. This should adjust the box size, although you may have to drag the handle at the far right to increase the length of the box.

While the box is selected, you could also change the font face, colour, fill or back colour and alignment by using other icons on the formatting toolbar.

Move the label so that it aligns at the left side of the report. Check it with Print Preview.

At this point, your report should look something like this:



First Period Student Roster

SSN:	FirstName:	LastName:
First Period:		
746-64-1354	Sam	Smith
583-17-5828	Kip	Johnson
572-47-2819	Kim	Carruthers
First Period: **Chem101**		
262-62-1854	Chris	Jackson
First Period: **Eng101**		
575-23-2990	Kyle	Baker
472-17-1718	Steve	Baker
456-27-7888	Kristy	Cooper
171-77-1828	Frieda	Little
First Period: **Math 157**		
573-28-1739	Susan	Smith
462-78-2739	Beth	Porterson

20. The report fits all on one page and it is readable. Note how the students in each class are grouped by the name of the class they are taking. This is the Grouping Header (FirstPeriod Header) at work.

21. Our example report also had a counter in the Report Footer. This is a ***calculated*** control. Calculated controls display values calculated from one or more fields in the underlying table or query. To create a calculated control, use the text box icon on the Toolbox, then key in the formula.

The section that a calculated control is placed in plays an important role. For instance, if you wanted to find the sum of all invoices for companies in a particular state, where the field is called Invoice, the formula = sum ([Invoice]) would be placed in a text box in the Grouping header for State. Think of this as a subtotal. The **exact same formula** placed in the Report Footer (the **last** thing in the Report) would produce a grand total**.**

Formulas used for calculated controls are very similar to those used in Excel.

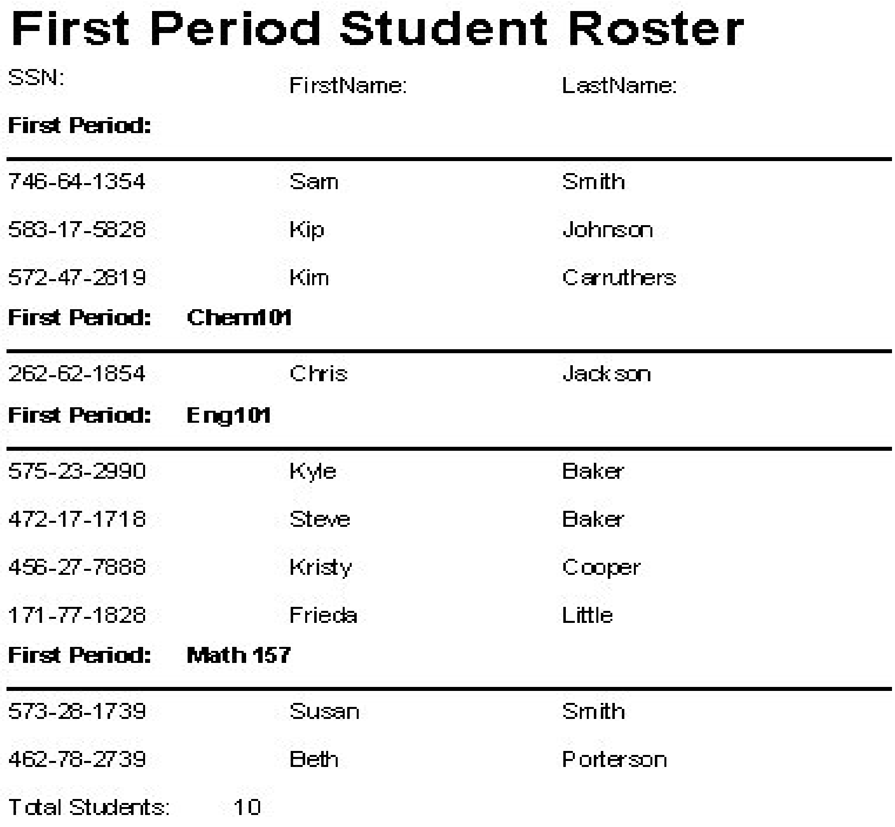
Our example has a calculated control in the Report Footer that counts the number of students in our roster. To create this:

Click on the text box icon on the Toolbox, then click anywhere in the Report Footer section. The text box and its accompanying label will appear. The text box has "Unbound" in it.

Click on Unbound, then key in this formula = count ([LastName]). This will count the number of last names in the detail section. Note the use of both the parentheses and the brackets and their position. Also the name of the field must be spelled identically to the way it is in the underlying table or query.

Select the text box (the one containing the formula), then click on the left align icon on the Formatting toolbar. Numbers align to the right by default. By changing the alignment to left, there will not be as big a gap between the label and the number. Click in the label box, delete the contents, then key in Total Students**.**

Align the boxes at the left side of the report, then switch to Print Preview. The total should be 10 and your finished report should look something like this:



First Period Student Roster

SSN:	FirstName:	LastName:
First Period:		
746-64-1354	Sam	Smith
583-17-5828	Kip	Johnson
572-47-2819	Kim	Carruthers
First Period: **Chem101**		
262-62-1854	Chris	Jackson
First Period: **Eng101**		
575-23-2990	Kyle	Baker
472-17-1718	Steve	Baker
456-27-7888	Kristy	Cooper
171-77-1828	Frieda	Little
First Period: **Math 157**		
573-28-1739	Susan	Smith
462-78-2739	Beth	Porterson

Total Students: 10

Figure 34: A final report printout

Close the report. Save it as First Period Student Roster**.**

22. Things that may cause problems in reports:

If you get a message that some of the area of the report is outside the printable area, take a look at the width of the grid. It can be adjusted by dragging the right side of the grid. You can also change the paper to landscape by clicking on File, Page Setup.

Watch out for typos in field names. Again, a big clue is the Parameter query message. If that happens, go back and start looking at field names in the report.

If you try to open a report and you get a message that tells you the table or query that the report is based on no longer exists or the name may have changed, stop and think. Did you change the name of a table or query lately, or delete a table or query? If that's the case, the report will no longer function.

A lot of times, users see a typo in a table or query name, so they rename it, not thinking about the consequences. If the table or query still exists in the database, try renaming it to the previous name and see if that makes your report work.

23. Take time to further investigate the properties window. Each control has properties, and there are many properties that can help you customize your report.

Exercise 4

Now you must create the reports for HMS. The sample reports that you may create can be:

- Report on Patients of OPD and also a report on other patients
- Report on Doctor's details
- Reports on Departments & employees
- Report giving combination of Doctors and Patients
- Report providing details on Patients attended by a Doctor and so on.
- Reports of total charges obtained.

6.0 USING QUERIES AND RECORD SET

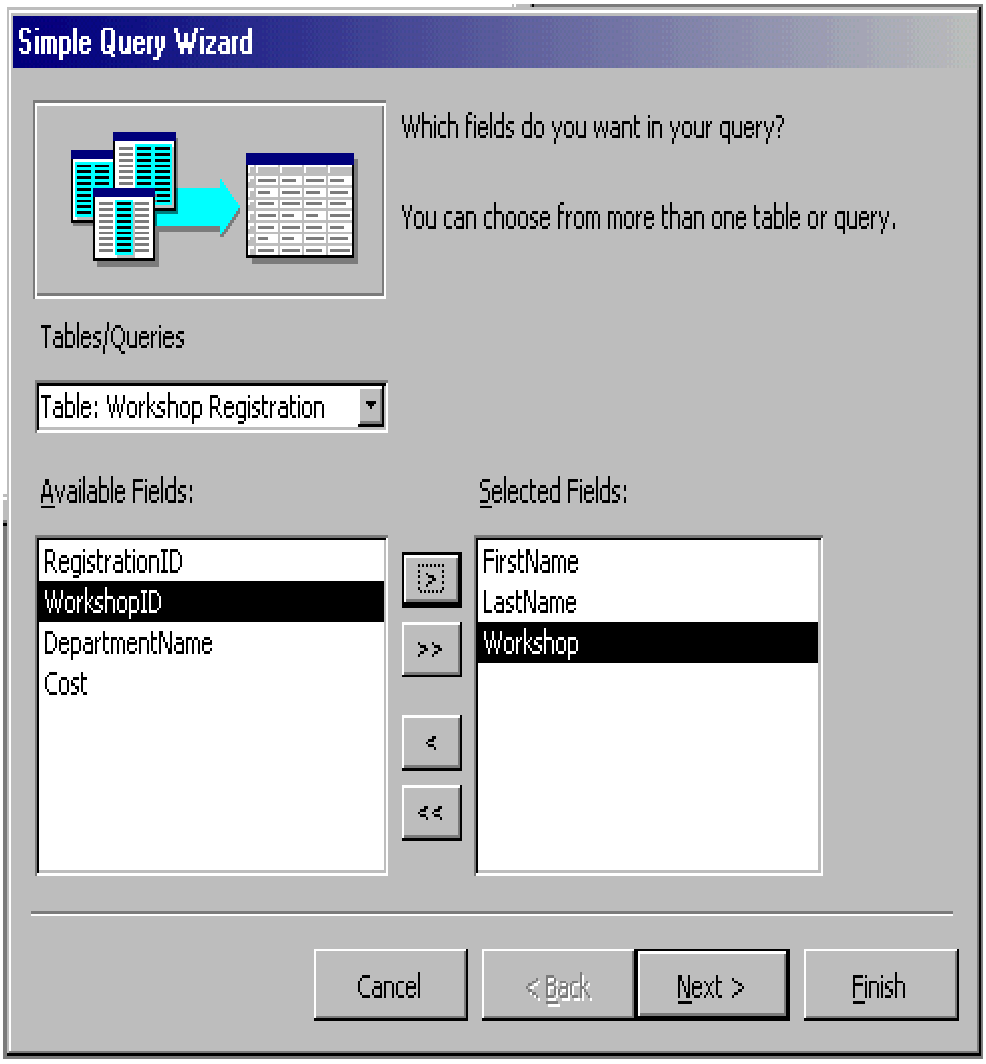
A query is a question about information in a table or tables. Queries can be used to view and analyze data or as a basis for forms and reports. Queries are commonly used to display fields from related tables and enable us to control not only what records display, but also what fields display.

A query does not contain data. Rather, it is a set of instructions. Access uses these instructions to select and display the appropriate records in a table. The query always considers all the data in the table. If the new records meet the conditions of the query, they will be included when the query results appear.

When a query is opened or run, a RecordSet appears. A RecordSet contains all the fields and records that meet the conditions of the query. While the RecordSet is not a table, it can be used under certain conditions to add and edit records in tables.

6.1 Using the Simple Query Wizard

Access provides a Simple Query Wizard that guides through the steps to create a basic select query. When the Simple Query Wizard is used, the table from which the data is to be taken is selected as well as the fields to be displayed in the query. In the last step, the query is named and then whether or not to display the results (the RecordSet) of the query is chosen. To change the design of the query Design View can be used.



Simple Query Wizard

Which fields do you want in your query?

You can choose from more than one table or query.

Tables/Queries: Table: Workshop Registration

Available Fields:
- RegistrationID
- WorkshopID
- DepartmentName
- Cost

Selected Fields:
- FirstName
- LastName
- Workshop

Figure 35: A query form

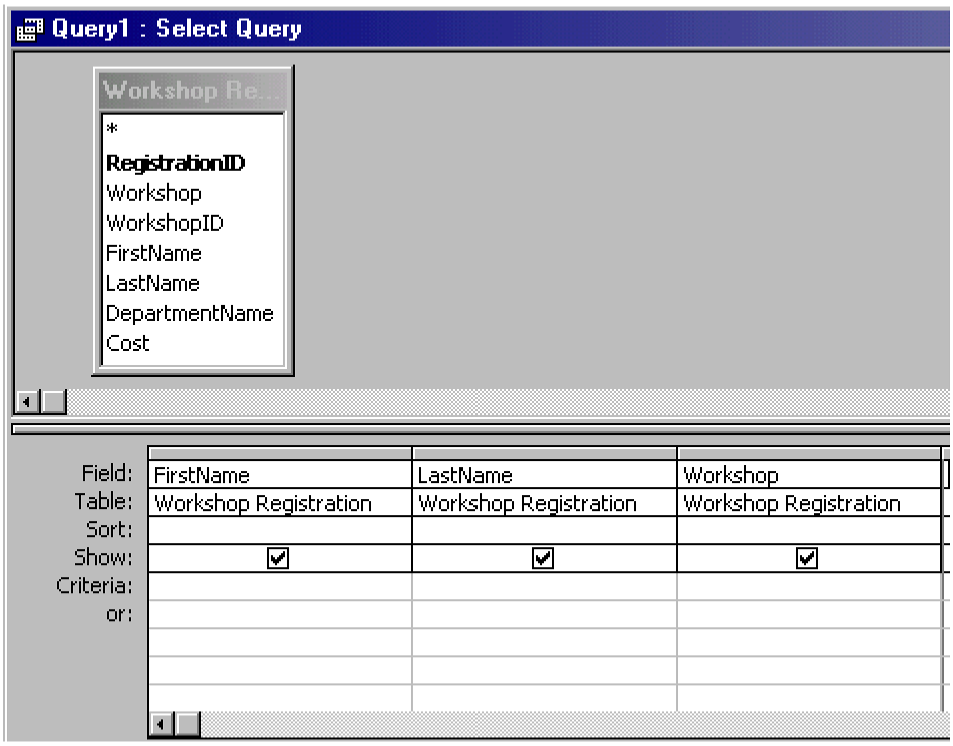
Create a Query that shows the First Name, Last Name, and Workshop for workshop participants.

1. Select the **New** button on the Database window toolbar. Select **Simple Query Wizard**.
2. Select **OK**.
3. Select the **Tables/Queries** list.
4. Select the table or query you want to use as the basis for the query.
5. Use the single right arrow to add the field(s) you want to display in the query from the **Available Fields** list box.
6. Select **Next**.
7. Type a name for the query.
8. Select **Finish**.

6.2 Creating a Query in Design View

A query can be created in Design view. This option gives most flexibility in designing a select query. It allows addition of criteria to select records and sort the resulting RecordSet.

To create a query in Design view, the design grid is used to set up the query. The field list of the required table appears in the top pane of Design view. The required fields can be added to the design grid in the bottom pane of Design view, along with any sort orders or criteria for selecting records.



Workshop Re...

*
RegistrationID
Workshop
WorkshopID
FirstName
LastName
DepartmentName
Cost

Field:	FirstName	LastName	Workshop
Table:	Workshop Registration	Workshop Registration	Workshop Registration
Sort:			
Show:	☑	☑	☑
Criteria:			
or:			

Figure 36: A query making

Create a Query in Design view that shows the First Name, Last Name, and Workshop for workshop participants.

1. Click the **New** button on the Database window toolbar.
2. Select **Design View**.
3. Select **OK**.
4. Select a table and add it to the query.
5. Select **Close** to close the Show Table dialog box.
6. Select the first field you want to add to the query.
7. Drag the field to the desired column in the **Field** row in the design grid.
8. Add other fields to the query as desired.
9. Click the **Save** button.
10. Type a name for the query.
11. Select **OK**.

6.3 Opening a Query

To open a query, Access runs the instructions in the query and displays the resulting RecordSet in **Datasheet** view. If new records have added, since the last execution of a particular query, the new records will appear only if they meet the query criteria.

1. Select the name of the query you want to open.
2. Select the **Open** button on the Database window toolbar.

More than one table can be used in a query. The tables must be joined in order for the query to give accurate results. If they are not joined, a join is created in the top pane of Design view.

On adding more than one table, the field lists appear in the top pane of Design view. If the tables are already related, join lines appear automatically.

Once a table is added to the query, the fields can then be added from the field list to the design grid. The second row in the design grid is the Table row, which indicates from which table the field originates.

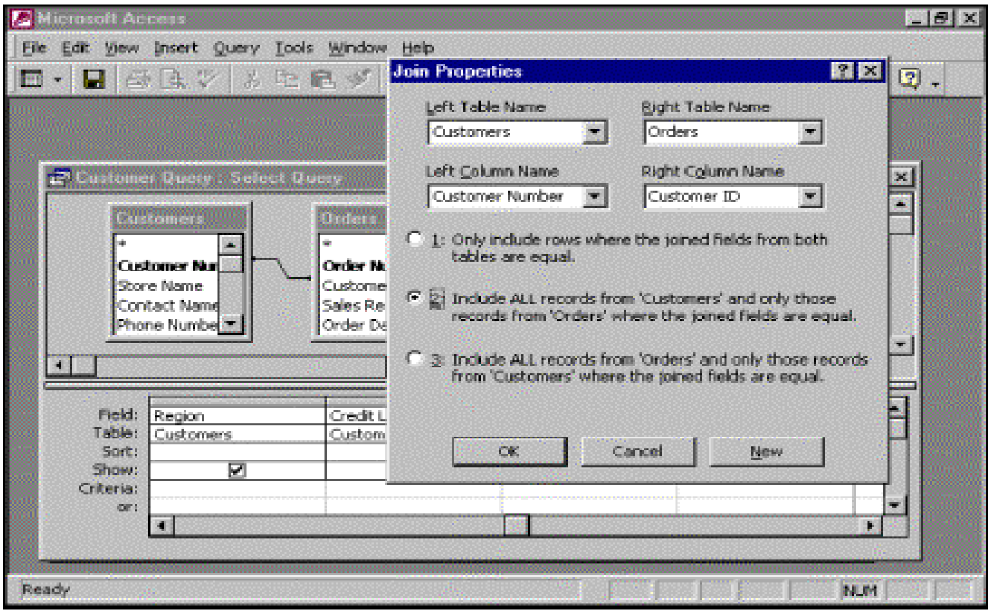
When Design view is opened to design a new query, the Show Table dialog box opens automatically so that multiple tables can be added. However, when Design view is opened to modify an existing query design (i.e., to add a table), the Show Table dialog box is to be opened manually.

1. Select the name of the query to which a base window toolbar.
2. Click the **Show Table** button on the **Query Design** toolbar.
3. Select the name of the table to be added to the query.
4. Select **Close** to close the Show Table dialog box.

In case of multiple tables query, the tables must be joined in order for the query to give accurate and meaningful results. If Access does not know how to relate the data between tables, it displays every combination of data between the two tables. For example, if one table has 20 records and the other has 5, then the RecordSet will contain 100 records and the results are virtually meaningless.

If the relationships have already been defined in the Relationships window, join lines between the field lists display automatically in Design view. Therefore, if a related table is added to a query, a join line appears automatically. Access also automatically creates a join if there is a field with the same name in both tables. If there is no predefined relationship or fields with the same name, such a relationship must be created.

The join type defined in the relationship is particularly important to queries. The default type is an inner join in which records are only included in the RecordSet if there is matching data in the join fields for both tables. An outer join can also be created in which all the records from the "one" table appear, even if there is no matching data in the "many" table.



Join Properties

Left Table Name: Customers

Right Table Name: Orders

Left Column Name: Customer Number

Right Column Name: Customer ID

○ 1: Only include rows where the joined fields from both tables are equal.

● 2: Include ALL records from 'Customers' and only those records from 'Orders' where the joined fields are equal.

○ 3: Include ALL records from 'Orders' and only those records from 'Customers' where the joined fields are equal.

Field: Region | Credit L
Table: Customers | Custom
Show: ☑

Figure 37: Query on joined tables

1. Select the join field in the field list of the first table.
2. Drag the field on top of the matching field in the field list of the second table.
3. Double-click the join line.
4. Select the desired join type.
5. Select **OK**.

Running a Query

A query can be run directly from Design view to display the RecordSet. This option is useful to test the design of the query to see if the resulting RecordSet contains the required information.

Running a query does not save the design of the query. If the RecordSet is closed after running a query, a message box opens, asking to save the changes.

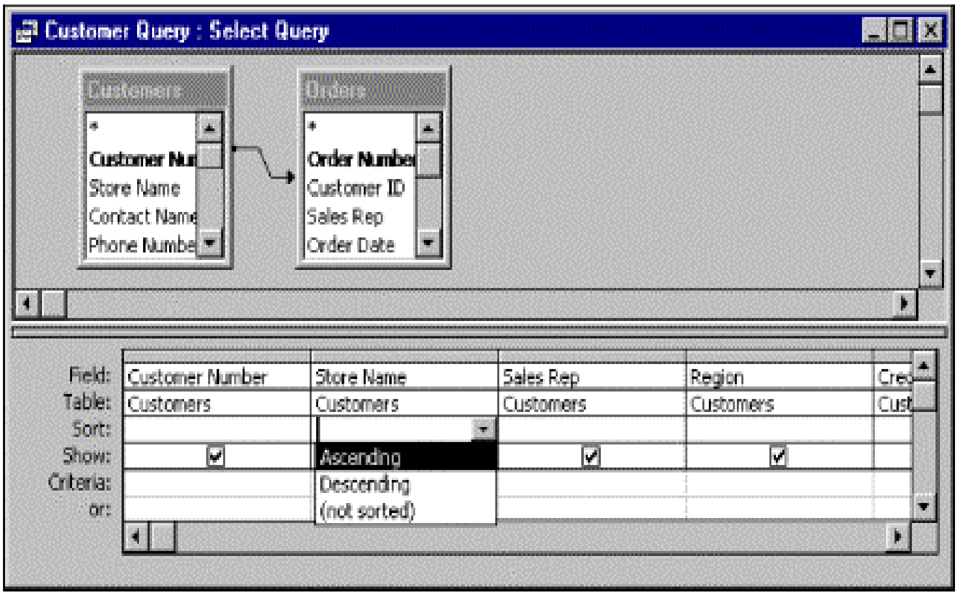
Run Query button

1. Click the **Run** button on the **Query Design** toolbar.

6.4 Adding Features in Query

Sorting a Query

When a query is run, the records in the RecordSet appear in the same order in which they appear in the table. The records can be ordered by either sorting the RecordSet or assigning a sort order in the query design. The RecordSet can be sorted just as a table. However, this must be done every time the query is executed. If the sort order is assigned in the query design, Access performs the sort automatically each time the query is executed.



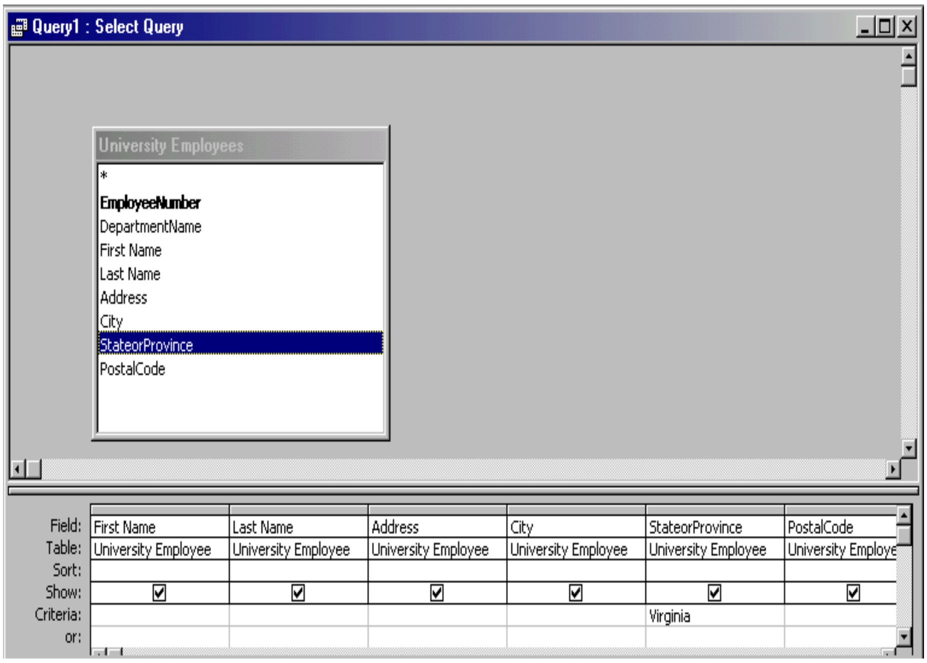
Field:	Customer Number	Store Name	Sales Rep	Region	Cre
Table:	Customers	Customers	Customers	Customers	Cust
Sort:					
Show:	☑		☑	☑	
Criteria:					
or:					

Figure 38: Sorting in a query

1. Select the **Sort** row under the field by which you want to sort.
2. Select the **Sort** list.
3. Select **Ascending** or **Descending**.

Data can be entered in the **Criteria** row of the query design grid to restrict the number of records that are returned in a query.

*To select records that meet a single value, enter the value in the **Criteria** row under the appropriate field. Access automatically inserts quotation marks (" ") around alphanumeric entries and number symbols (#) around date entries. If the entry is numeric, the number appears without quotation marks. When the query is run, only those records with values that match the criteria appear in the RecordSet.*



Field:	First Name	Last Name	Address	City	StateorProvince	PostalCode
Table:	University Employee	University Employee	University Employee	University Employee	University Employee	University Employe
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:					Virginia	
or:						

Figure 39: A sample query

For Example, Create a Query that filters all University Employees from the state of Virginia.

Write a new Query that includes the following fields: First Name, Last Name, Address, City, StateorProvince, PostalCode.

1. Select the **Criteria** row under the **StateorProvince** field.
2. Type the value for the criteria.
3. Press **[Enter]**.

Hiding a Field in a Query

A field can be used to select records that meet a certain criteria without displaying the field in the RecordSet. This option is useful when all the records meet the same specified criteria.

For example, you may want to create a query to display customers in the **Southeast** region. Therefore, you must add the **Region** field to the query in order to enter the criteria to select only the **Southeast** region. However, you might not want the field to appear in the RecordSet because you know that all the data in the **Region** field is the same (**Southeast**). In this case, you may want to hide the field you used for the criteria.

The design grid includes a **Show** row with a check box for each field. If the check box is selected, the field will appear in the RecordSet. If the check box is deselected, the field will be hidden in the RecordSet. All the **Show** field check boxes are selected by default.

Adding a Record Using a Query

A query can be used to update records in related tables. When information is entered into the join field for a primary table, Access automatically locates the corresponding information in the related table and enters it into the record.

For example, the **Orders** and **Customers** tables are related. When you enter the customer ID number in the appropriate field in the **Orders** table, Access completes the customer name and address information automatically through the relationship with the **Customers** table.

You can always edit information in queries based on a single table. In queries based on related tables, Access must be able to determine the relationship type (i.e., one-to-one or one-to-many), or you cannot edit information. In queries with tables that have a one-to-one relationship, you can always edit data. In queries with tables that have a one-to-many relationship, you may not be able to edit data at times. For example, this can happen if the join field from the "many" table is not in the query, since Access cannot create new records in the "many" table. For example, Access cannot create new orders for a customer if the query is based on the **Customers** table unless the **Customer ID** field from the **Orders** table is included in the query.

New Record button

Click the **New Record** button on the **Query Datasheet** toolbar.

1. Type the necessary data in the first field.
2. Move to the next field.
3. Continue to add data as necessary.
4. Save the current record.
5. Printing a Query

The RecordSet resulting from a query can be printed. It can be printed after running a query, while it appears on the screen, or by printing it directly from the Database window. If the RecordSet is printed from the Database window, Access runs the query and sends the results to the printer, rather than to the screen.

6.5 Conditional Queries

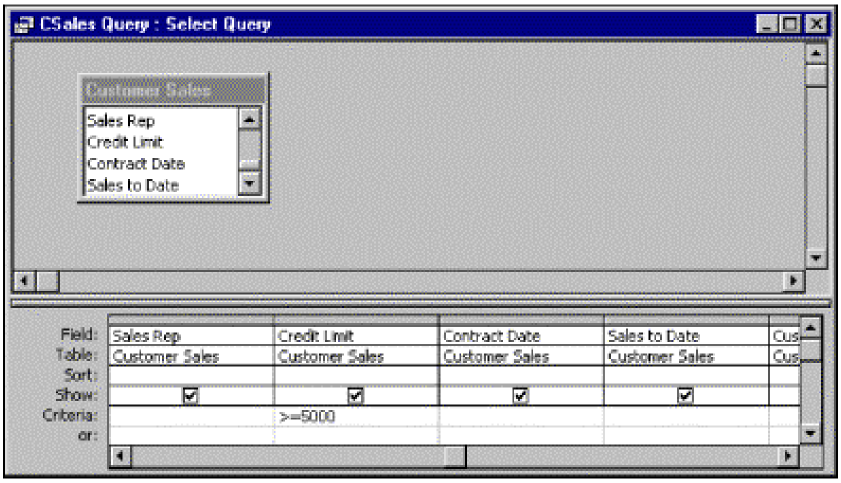
Using Comparison Operators

In order to select specific records the criteria is entered in the **Criteria** row of the query design grid. The simplest criterion requires that records match a single value to be included in the RecordSet.

Comparison operators can also be used to select a specific group of records in a table. For example, if you want to find all customers with credit limits less than 1000, or all customers with a contract date on or before January 2004, you can write an expression that defines the criteria using a combination of comparison operators and field values, such as <1000 or <=1/1/04. Comparison operators are symbols that represent conditions recognised by Access. The available comparison operators are

Operator	***Description***
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
=	equal to
<>	not equal to
Not	reverse logic

A comparison operator can be used to compare a specified value with all the values in a field. When the query is executed, only the records with values meeting the specified criteria appear in the RecordSet.



Field:	Sales Rep	Credit Limit	Contract Date	Sales to Date	Cus
Table:	Customer Sales	Customer Sales	Customer Sales	Customer Sales	Cus
Sort:					
Show:	☑	☑	☑	☑	
Criteria:		>=5000			
or:					

Figure 40: A sample conditional query

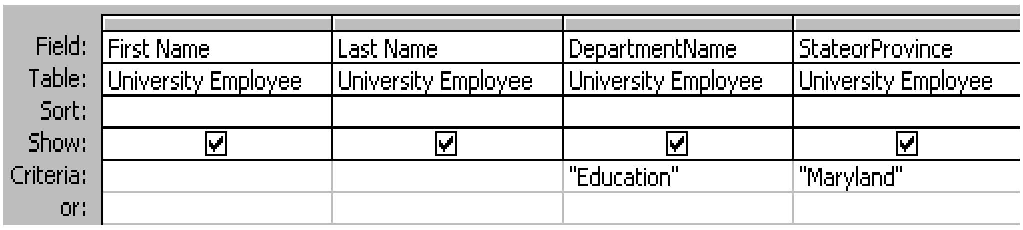
1. Select the **Criteria** row under the desired field.
2. Type a comparison operator and the value for the criteria.
3. Press **[Enter]**.

Using an AND Condition

Many times, a query requires more than one condition to obtain the desired result. For example, if you want to find all customers in PA with sales to date over 10,000, you would need two conditions: State=PA and Sales to Date>10000. The records must meet both conditions in order to be included in the RecordSet. To combine two criteria in this way, you use the **And** logical operator.

You can use the **And** operator in a single field or in different fields. In a single field, you can use the **And** operator to find records that fall into a range. For example, to find customers whose contract dates fall between 9/1/04 and 9/30/04, you type both criteria on a single line in the **Criteria** row under the appropriate field (i.e., >=**9/1/04 And** <=**9/30/04** in the **Contract Date** field).

The **And** operator also allows you to impose conditions in two different fields. For example, to find customers in PA with sales to date over 10,000, you type each criterion on a single line in the **Criteria** row under the appropriate fields (i.e., =**PA** in the **State/Province** field and >**10000** in the **Sales to Date** field).



Field:	First Name	Last Name	DepartmentName	StateorProvince
Table:	University Employee	University Employee	University Employee	University Employee
Sort:				
Show:	☑	☑	☑	☑
Criteria:			"Education"	"Maryland"
or:				

Create a Query that lists University Employees from the Education Department **AND** from the state of Maryland.

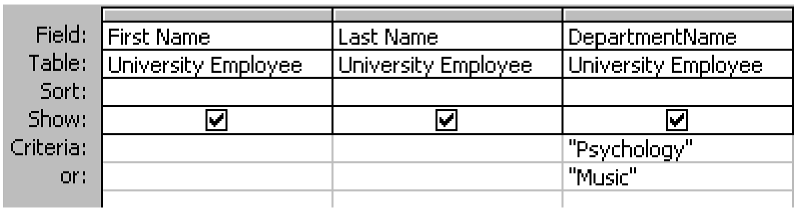
Using the University Employees database, create a new Query in Design view using the University Employees Table. Add the following fields: ***First Name, Last Name, DepartmentName, and StateorProvince.***

1. Select the **Criteria** row under the **DepartmentName** field.
2. Select the **Criteria** row under the **StateorProvince** field.
3. Switch to **Datasheet** view to view the Query results.

Using an OR Condition

Many times, a query requires more than one condition to obtain the desired result. For example, if you want to find all customers in PA or all customers with sales to date over $10,000, you would need two conditions: State=PA as well as Sales to Date>10000. The records only need to meet one of the conditions in order to be included in the RecordSet. To combine two criteria in this way, you use the **Or** logical operator.

*You can use the **Or** operator in a single field or in different fields. In a single field, you type the criteria on two separate lines under the same field. In different fields, you type the criteria on two separate lines under the appropriate fields. For example, to find all customers with contract dates on or before 1/1/04 or credit limits above 3,000, you type <=**1/1/04** in the **Criteria** row under the **Contract Date** field and >**3000** in the or row under the **Credit Limit** field.*



Field:	First Name	Last Name	DepartmentName
Table:	University Employee	University Employee	University Employee
Sort:			
Show:	☑	☑	☑
Criteria:			"Psychology"
or:			"Music"

*Create a Query that lists University Employees from the Psychology **OR** Music department*

Using the University Employees database, create a new Query in Design view using the University Employees Table. Add the following fields: **First Name, Last Name, and DepartmentName.**

1. Select the Criteria row under the DepartmentName **field.**
2. Select the **or** row under the **DepartmentName** field.
3. Switch to **Datasheet** view to view the Query results.

Using the BETWEEN AND Operator

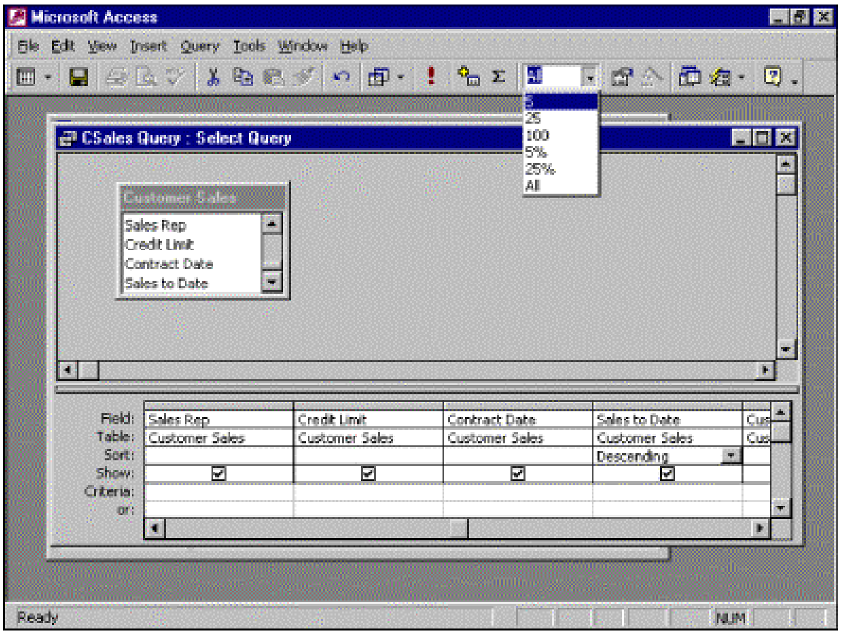
To find data that is between two values, the **Between** And operator can be used in a query. It can be used with a text, numeric, or date field. For example, to find all records of customers with credit limits between 1000 and 2000, enter **Between 1000 And 2000** in the **Criteria** row under the **Credit Limit** field.

1. Select the **Criteria** row under the desired field.
2. Type the **Between And** operator and the criteria.
3. Press **[Enter]**.

6.6 Advanced Query Features

Setting Top Values in a Query

The results of a query can be limited so that only the highest or lowest values for a field appear in a RecordSet. For example, you can set the top values of a **Quantity Sold** field to 10 to find the top ten best selling products. You can limit the number of records to a specific number or a percentage of all records being queried (i.e., top 25%). The field for which you are setting the top or bottom values must be sorted. If the field is sorted in descending order (Z to A, 9 to 0), the top values will be found. If the field is sorted in ascending order (A to Z, 0 to 9), the bottom values will be found.



Field:	Sales Rep	Credit Limit	Contract Date	Sales to Date	Cus
Table:	Customer Sales	Customer Sales	Customer Sales	Customer Sales	Cus
Sort:				Descending	
Show:	☑	☑	☑	☑	
Criteria:					
or:					

Figure 41: A query with advanced features

1. Select the **Sort** row under the desired field.
2. Select the **Sort** list.
3. Select the desired sort order.
4. Select the **Top Values** list on the **Query Design** toolbar.
5 Select the desired top value.

Creating a Function Query

Access allows creating a query that groups records by a selected field and applies a function that calculates values on other fields in the query as needed. ==For example, you can group records in a table by state and then select the **Count** function to find out how many customers (records) are in each state (field). You can also group by customer name (field) and calculate the **Sum** of each customer's orders (record values).

There are several types of functions, the most common of which are listed in the following table:

Function	***Description***
Sum	Sums the values in the calculated field
Average	Finds the average value of the calculated field
Count	Counts the number of records in the calculated field
Max	Finds the highest value in the calculated field
Min	Finds the lowest value in the calculated field

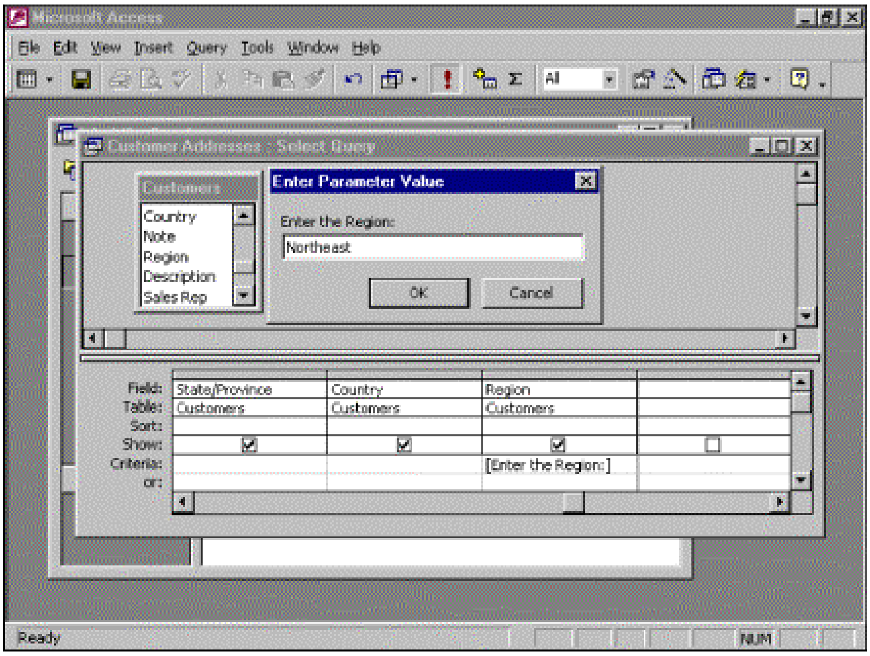
1. Select the **New** button on the Database window toolbar.
2. Select **Design View**.
3. Select **OK**.
4. Select the name of the table you want to add to the query.
5. Select **Add**.
6. Select **Close** to close the Show Table dialog box.

7. Select the field by which you want to group.
8. Select the field you want to calculate.
9. Select the **View** menu.
10. Select the **Totals** command.
11. Select the **Total** row under the field you want to calculate.
12. Select the **Total** list.
13. Select the desired function.

Creating a Parameter

A query can be executed with different criteria each time. To do this a parameter query can be created. A parameter query is a query that prompts the user for information when the query is run. Access then uses the information as the criteria and runs the query. The resulting RecordSet only includes those records that meet the criteria. This option allows one to avoid displaying the query in **Design** view each time the criteria is changed.

*You enter the text that will display in the prompt in the **Criteria** row under the appropriate field in the design grid, followed by a colon (**:**) and enclosed in square brackets (**[ ]**). You can set up a parameter query to prompt the user for more than one piece of information as well.*



Enter Parameter Value

Enter the Region: Northeast

Field:	State/Province	Country	Region	
Table:	Customers	Customers	Customers	
Sort:				
Show:	☑	☑	☑	☐
Criteria:			[Enter the Region:]	
or:				

Figure 42: A query with parameter

*1. Select the **Criteria** row under the desired field.*

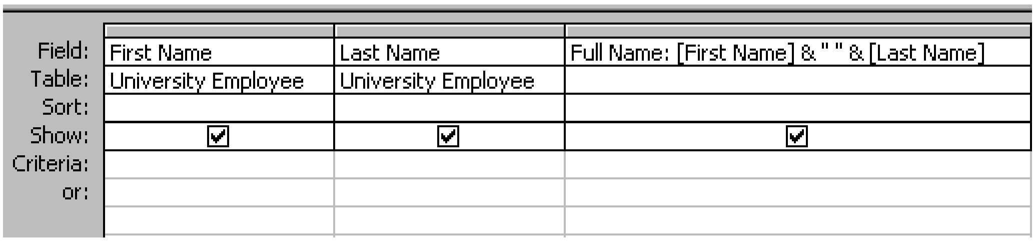
The insertion point appears in the **Criteria** row under the appropriate field.

1. Type the desired text for the prompt, followed by a colon (**:**) and enclosed in square brackets (**[ ]**).
2. Press **[Enter]**.

Creating a Concatenation in a Query

Access allows combining two or more fields into one field. This process, known as concatenation, enables us to add field name text strings to one another. The text strings can follow each other with or without spaces. You can add other characters between the text strings if needed. For example, you can combine the individual **City**, **State**, and **Postal Code** fields into one field called **Address**. You can have the comma and space characters appear between the field text strings. This concatenation process can be performed by creating a query to combine two or more fields.

When typing expressions for concatenation, the first part of the expression defines the name of the new field. The second part of the expression defines the fields which are to be concatenated. These field names must be surrounded by brackets. The ampersand (&) appears between the field name brackets. Any additional characters that are to appear between the fields are surrounded by double quotes. For example, the expression *Names: [First Name] &" - "&[Last Name]* ***creates a new field called*** *Names*. ***The new field contains the first name and last name separated by a space, a hyphen, and a space.***



Field:	First Name	Last Name	Full Name: [First Name] & " " & [Last Name]
Table:	University Employee	University Employee	
Sort:			
Show:	☑	☑	☑
Criteria:			
or:			

Using the **University Employees** database, create a **Concatenation** Query to combine **First Name** and **Last Name** into one field called **Full Name**.

1. Open the **Combine Names** Query from the **University Employees** database in Design View.
2. Select the next available field in Query design view and type the concatenation.

Using Multiple Tables in a Query

There may be instances when more than one table or query are added to a query. In order to do this, ensure that the field lists are joined to each other with a join line so that Access knows how to connect the information.

If tables in a query are not joined to one another, Access does not know which records are associated with which, so every combination of records between the two tables would appear in the query. Therefore, if each table had 50 records, the query's results would contain 2500 records (50×50), thereby rendering useless results.

If a relationship has been previously created between tables in the Relationship window, Access automatically displays join lines when related tables are added in query **Design** view. If the relationships have not been previously created, Access automatically creates joins if tables were added to a query as long as the tables each have a field with the same or compatible data type and one of the join fields is a primary key. If tables added to the query do not include any fields that can be joined, one or more extra tables or queries have to be added to serve solely as a bridge between the tables containing the required data.

Exercise 5

After doing so much about queries, you now create at least five queries relating to HMS. Some sample queries are listed below:

- Find the doctors who have not seen any patient.
- Create more relationship tables and track the history of the patients.
- Create more relationship tastes and track the payment to employees over last year. Also calculate the income tax return for each employee using a simple formula.

7.0 SUMMARY

This block is an attempt to initiate you into the practical world of Database Management System. We started the block with a brief but not sufficient description of a Hospital Management System and have identified a few tables. You can undertake a thorough study on this system such that you are able to design better tables for this system. We have tried to explain the process of creating various tables, their relationships, forms, reports and queries. However, you must use this information to develop the required exercises for hospital management. Thus, the block provides a flavor of various tools that are essential to build up a database application. Some of the basic tools which have been introduced in this block are:

- MS-Access Reports
- MS-Access Query Generator
- Visual Basic – a very brief introduction

You must keep working on these tools to obtain expertise over a period of time. However, it is strongly recommended that you also use certain open source tools like MySQL and PHP.

SOCIAS-IGNOU/P.O.10.5T/MAY, 2004