



Section - D

MS Access-2010



INTRODUCTION TO MICROSOFT ACCESS, 2010

LEARNING OBJECTIVES

- What is MS-Access?
- What are databases used for?
- Component of databases.
- Opening file in Access.

1.1 INTRODUCTION TO MICROSOFT ACCESS 2010

Microsoft Access 2010 for Windows is a Database Management System or DBMS, which helps us manage data stored in a computer database. In this chapter, the readers will be given exposure to the basics of Access and the various data types used in Access. A database is a tool for collecting and organizing information. Databases can store information about people, products, orders or anything else. A computerized database is a container of objects. One database can contain more than one table. For example, an inventory tracking system that uses three tables is not three databases, but one database that contains three tables.

An **Access** database stores its tables in a single file, along with other objects, such as forms, reports, macros and modules. Databases created in the **Access 2010** format have the file extension **.accdb** and databases created in earlier Access formats have the file extension **.mdb**.

Using **Access**, we can do the following:

- Add new data to a database, such as a new item in an inventory,
- Edit existing data in the database, such as changing the current location of an item,
- Delete information, perhaps if an item is sold or discarded,
- Organize and view the data in different ways,
- Share the data with others via reports, e-mail messages, an intranet or the Internet.

Microsoft Access 2010 can be started in Windows 95 or later versions of windows operating system in the following way:

- (i) Click on the **Start** button on the **Windows Taskbar** and then choose **Program >> Microsoft Access 2010**.
- (ii) After a short delay, the **Microsoft Access 2010** main window displays a dialogue box offering the user a choice of creating a new database or opening one of the existing databases as shown

in Fig. 1.1.1. The lower panel contains the names of the databases that have been opened recently.

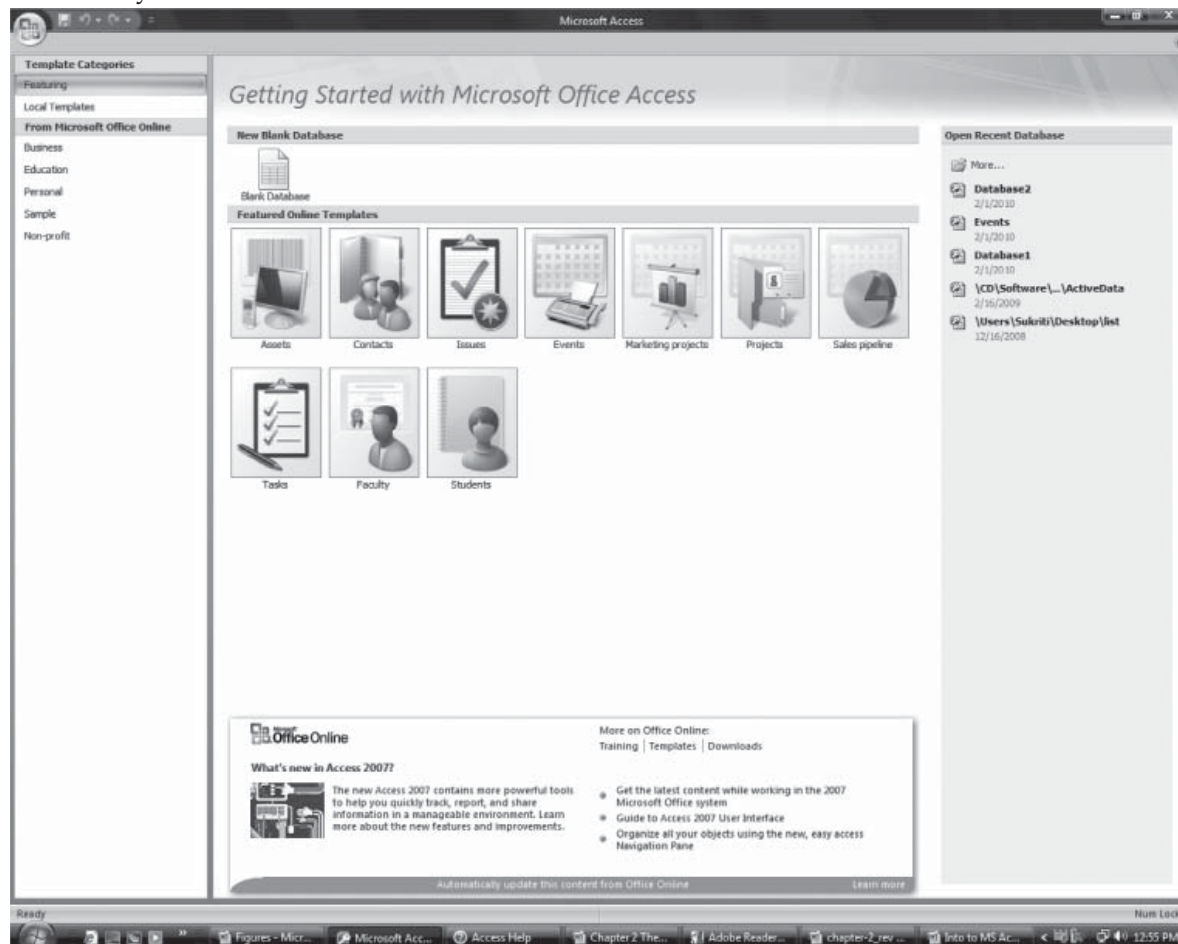


Fig. 1.1.1: Start-up MS Access 2010 Screen

1.1.1 The parts of an Access database

The following sections are short descriptions of the parts of a typical **Access** database.

1. Tables
2. Forms
3. Reports
4. Queries
5. Macros
6. Modules

1.1.2 Tables

A **database table** is similar in appearance to a spreadsheet, in that data is stored in rows and columns.



As a result, it is usually quite easy to import a spreadsheet into a database table. The main difference between storing data in a spreadsheet and storing it in a database is in how the data is organized.

To get the most flexibility out of a database, the data needs to be organized into tables so that redundancies don't occur. For example, if we're storing information about employees, each employee should only need to be entered once in a table that is set up just to hold employee data. Data about products will be stored in its own table and data about branch offices will be stored in another table. This process is called **Normalization**.

Each row in a table is referred to as a **Record**. Records are where the individual pieces of information are stored. Each record consists of one or more fields. Fields correspond to the columns in the table.

1.1.3 Forms

Forms are sometimes referred to as "Data Entry Screens", which are the interfaces we use to work with our data and they often contain command buttons that perform various commands. We can create a database without using forms by simply editing our data in the table datasheets. However, most database users prefer to use forms for viewing, entering and editing data in the tables.

Forms provide an easy-to-use format for working with the data and we can also add functional elements, such as command buttons, to them. We can program the buttons to determine which data appears on the form, open other forms or reports or perform a variety of other tasks. For example, we might have a form named "Customer Form" in which we work with customer data. The customer form might have a button which opens an order form where we can enter a new order for that customer.

Forms also allow us to control how other users interact with the data in the database. For example, we can create a form that shows only certain fields and allows only certain operations to be performed. This helps protect data and to ensure that the data is entered properly.

1.1.4 Reports

Reports are what we use to summarize and present data in the tables. A report usually answers a specific question, such as "How much money did we receive from each customer this year?" or "What cities are our customers located in?" Each report can be formatted to present the information in the most readable way possible.

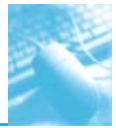
A report can be run at any time and will always reflect the current data in the database. Reports are generally formatted to be printed out, but they can also be viewed on the screen, exported to another program or sent as e-mail message.

1.1.5 Queries

Queries are the real workhorses in a database and can perform many different functions. Their most common function is to retrieve specific data from the tables. The data we want to see is usually spread across several tables and queries allow us to view it in a single datasheet. Also, since we usually don't want to see all the records at once, queries let us add criteria to "filter" the data down to just the records we want. Queries often serve as the record source for forms and reports.

1.1.6 Macros

Macros in Access can be thought of as a simplified programming language which we can use to add functionality to our database. For example, we can attach a macro to a command button on a form so that the macro runs whenever the button is clicked. Macros contain actions that perform



tasks, such as opening a report, running a query or closing the database. Most database operations that we do manually can be automated by using macros, so they can be great time-saving devices.

1.1.7 Modules

A **module** is a collection of declarations, statements and procedures that are stored together as a unit. A module can be either a class module or a standard module. Class modules are attached to forms or reports and usually contain procedures that are specific to the form or report they're attached to. Standard modules contain general procedures that aren't associated with any other object. Standard modules are listed under Modules in the Navigation Pane, whereas class modules are not.



1.2 CREATING DATABASES

The primary step involved in creating any application is to set up a database file. There are various methods of creating a database. The user could either start with creating a database from the scratch or create a database using the database wizard having predefined objects.

1.2.1 Create a Database using a Template

Access provides us wide variety of templates that can be used to speed up the database creation process. A template is a ready-to-use database containing all the tables, queries, forms and reports needed to perform a specific task. For example, there are templates that we can use to track issues, manage contacts or keep a record of expenses. Some templates contain a few sample records to help demonstrate their use. Template databases can be used as they are or we can customize them to better fit our needs.

If one of these templates fits our needs, using it is usually the fastest way to get a database started. However, if we have data in another program that we want to import into Access, it is better to create a database without using a template. Templates have a data structure already defined and it might require a lot of work to adapt our existing data to the template's structure. The steps to be followed are:

- (i) If we have a database open, click File and then  click **Close Database**  to display the **Getting Started with MS Office Access** page.
- (ii) Several featured templates are displayed in the middle of the **Getting Started with MS-Office Access** page and more become available when we click the links in the **Sample Templates** pane as shown in Fig. 1.2.1.

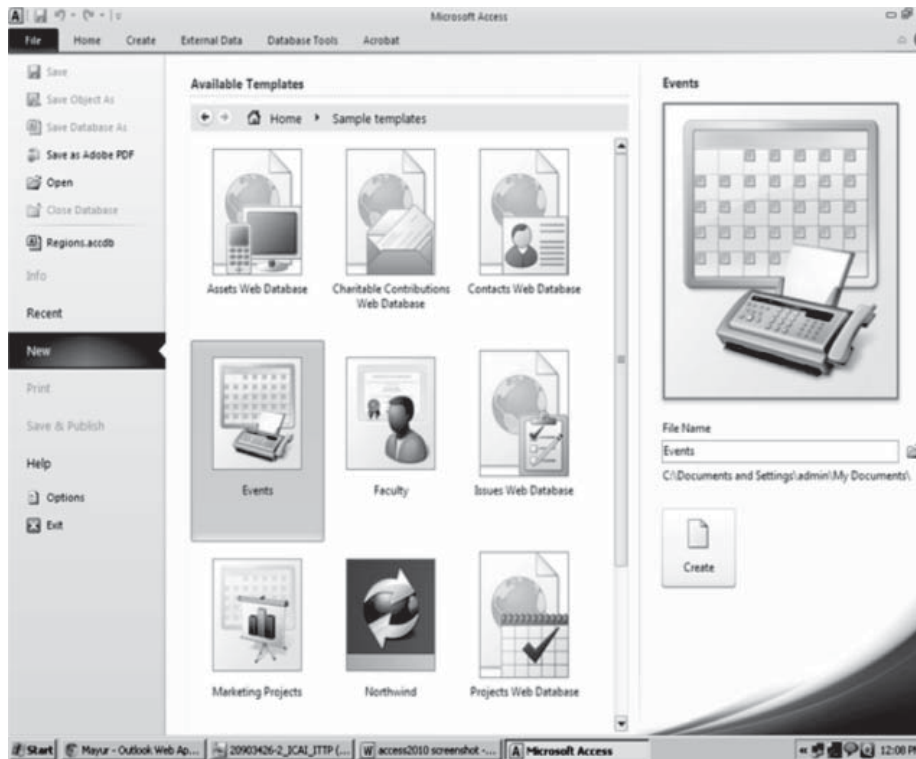


Fig. 1.2.1: MS Access showing Local Template Categories

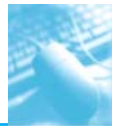
- (iii) Click the template we want to use. For example - Let us choose **Event** Template.
- (iv) Access suggests a file name for our database in the File Name box - we can change the file name, if we want. To save the database in a different folder from the one displayed below the file name box, click **Browse** to the folder in which we want to save it and then click **OK**.
- (v) Click **Create** (or Download for an Office Online template). **Access** creates or downloads the database and then opens it. A form is displayed in which we can begin entering data. If our template contains sample data, we can delete each record by clicking the record selector (the shaded box or bar just to the left of the record) and then click **Home >> Records >> Delete**.
- (vi) To begin entering data, click in the first empty cell on the form and begin typing. Use the **Navigation Pane** to browse for other forms we might want to use.

1.2.2 Create a Database without using a Template

If we are not interested in using a template, we can create a database by building our own tables, forms, reports and other database objects. In most cases, this usually involves one or both of the following :

Entering, pasting or importing data into the table that is created when we create a new database and then repeating the process with new tables that we create by using the Table command on the **Create** tab.

Importing data from other sources and creating new tables in the process.



1) Create a blank database

- (i) On the **Getting Started with MS Office Access** page, under **New Blank Database**, click **Blank Database**, as shown in Fig. 1.2.2.

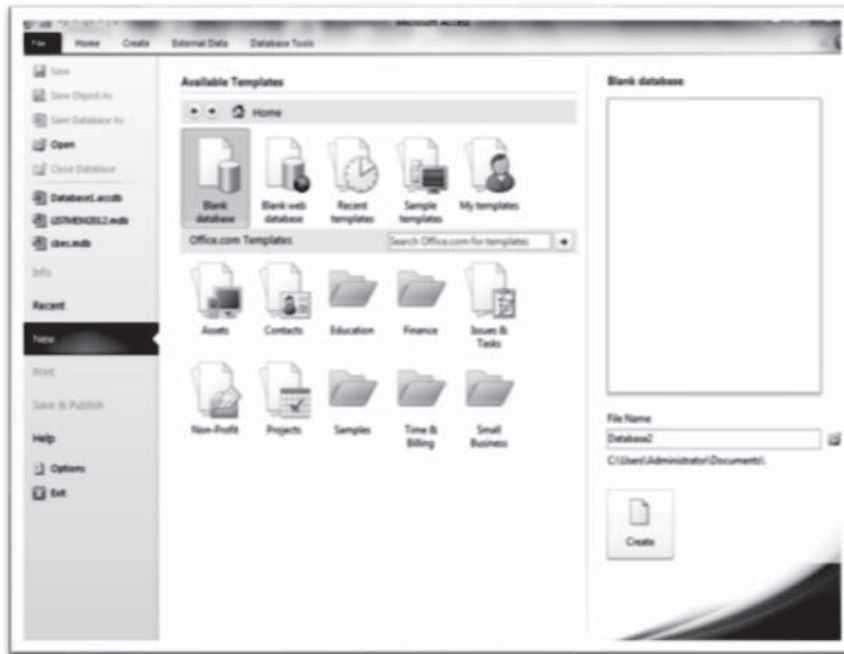


Fig. 1.2.2: Creating and Naming the Blank Database

- (ii) In the **Blank Database** pane, type a file name in the **File Name** box. If we do not supply a file name extension, **Access** adds it for us. To change the location of the file from the default, click **Browse** for a location to put for our database, (next to the File name box), browse to the new location and then click **OK**.
- (iii) Click **Create**, Access creates the database with an empty table named Table1 and then opens Table1 in **Datasheet** view. The cursor is paces in the first empty cell in the Click to Add column.
- (iv) Begin typing to add data or data can be pasted from another source.

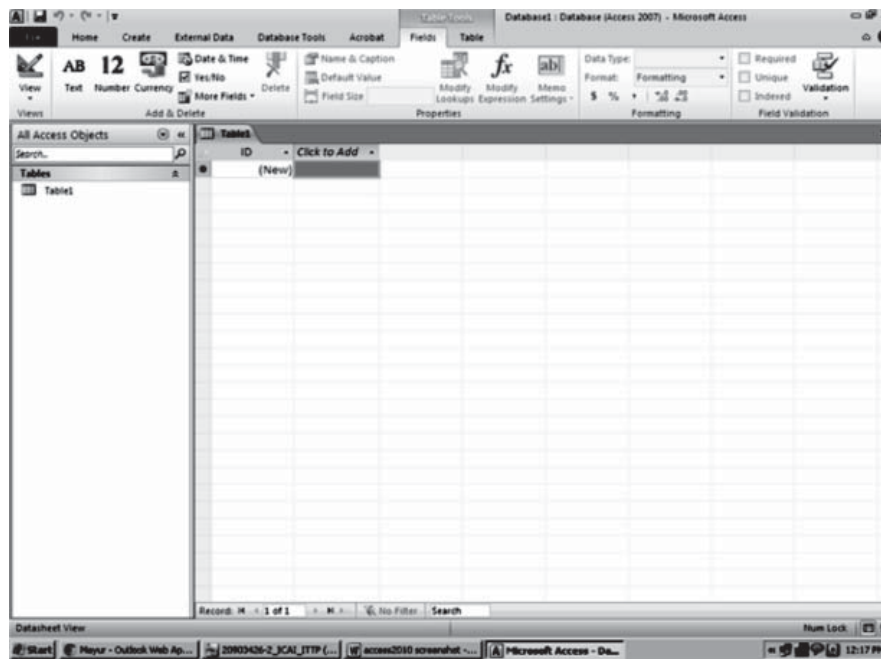



Fig. 1.2.3: Datasheet View

Entering information in Datasheet view (as shown in Fig. 1.2.3) is designed to be very similar to working in a **Microsoft Office Excel 2010** worksheet. The table structure is created while we enter data - any time we add a new column to the table, a new field is defined. Access automatically sets each field's data type, based on the data we enter.

If we do not want to enter information in Table1 at this time, click **Close** . If any changes are made in the table, **Access** prompts us to save changes to the table. Click **Yes** to save the changes, click **No** to discard them or click **Cancel** to leave the table open.

LEARNING OBJECTIVES

- To understand the concept of table.
- To introduce table and field properties.
- To know all data types.
- To work with database design and creation of a table.
- To save, delete, rename and open a table.
- To introduce the concept of SharePoint List.
- Form and its importance.
- General types of forms.
- General steps in creating a form.
- Creating a form using form view.
- Creating a form using Form Wizard.
- Form controls, its types and properties.
- Inserting controls on a form.

2.1 INTRODUCTION TO TABLES

When we create a database, we store our data in tables - subject-based lists of rows and columns. For instance, we might create a **Contacts** table to store a list of names, addresses and telephone numbers or a **Products** table to store information about products. We should always start the design of a database by first creating its tables - even before we create any other database objects.

2.1.1 WHAT IS A TABLE?

A **Table** is a database object used to contain data about a particular subject, such as employees or products. Each record in a table contains information about one item, such as a particular employee. A record is made up of fields, such as name, address and telephone number. A record is also commonly called a **Row** and a field is referred as a **Column**.



Customer Id	Company Name	First Name	Last Name
1	A	John	Bedecs
2	B	Joe	Axen
3	A	Anny	Smith
4	C	Harry	Potter

Fig. 2.1.1 : An example of a 'Customers' Table

In Fig. 2.1.1, point 1 indicates Record or Row whereas point 2 indicates Field or Column. A database can contain many tables, each storing information about a different subject. Each table can contain many fields of different types, including text, numbers, dates and pictures.

The following list shows some common examples of tables we might create.

- A Customers table that lists our company's customers and their addresses,
- A Catalogue of products sold, including prices and pictures for each item,
- A Tasks table that tracks tasks and due dates,
- An Inventory of equipment or stock on hand.

We should plan and design our database carefully to ensure its correctness and to avoid having to make too many changes later.

(1) Table and Field Properties

Tables and fields have properties that one can set to control their characteristics or behaviour.

- Table property :** In an **Access database**, table properties are attributes of a table that affect the appearance or behaviour of the table as a whole. A table opens in **Design view** and its properties are set in the table's property sheet. For example, one can set a table's **Default View** property to specify how the table is displayed by default.
- Field property :** A field property defines one of the field's characteristics or an aspect of the field's behaviour and applies to a particular field in a table through Datasheet view. One can also set any field property in **Design view** by using the **Field Properties** pane.

Datasheet View : In **Datasheet** view, we can enter data immediately and let Access build the table structure behind the scenes. Field names are assigned numerically (Field1, Field2 and so on...) and Access automatically sets each field's data type, based on the data we enter. In this, we can edit fields, add and delete data and search for data.



(2) Data Types

Every field has to be assigned a data type which indicates the kind of data that the field stores, such as large amounts of text or attached files. A data type is a field property, but it differs from other field properties as follows :

- We set a field's data type in the table design grid, not in the **Field Properties** pane.
- A field's data type determines what other properties the field has.

The **Data Type** property can be set only in the upper portion of table **Design view**. The data types and their explanation is highlighted in Table 2.1.1.

Setting	Type of Data	Size
Text	(Default) Text or combinations of text and numbers, as well as numbers that don't require calculations, such as phone numbers.	Up to 255 characters or the length set by the FieldSize property, whichever is less.
Memo	Lengthy text or combinations of text and numbers.	Upto 63,999 characters.
Number	Numeric data used in mathematical calculations.	1, 2, 4 or 8 bytes.
Date/Time	Date and time values for the years 100 - 9999.	8 bytes.
Currency	Currency values and numeric data used in mathematical calculations involving data with one to four decimal places. Accurate to 15 digits on the left side of the decimal separator and to 4 digits on the right side.	8 bytes.
AutoNumber	A unique sequential (incremented by 1) number or random number assigned whenever a new record is added to a table. AutoNumber fields can't be updated.	4 bytes (16 bytes if the FieldSize property is set to Replication ID).
Yes/No	Yes and No values and fields that contain only one of two values (Yes/No, True/False or On/Off).	1 bit.
OLE Object	An object (such as a Microsoft Excel spreadsheet, a Microsoft Word document, graphics, sounds or other binary data) linked to an embedded in a Microsoft Access table.	Up to 1 gigabyte (limited by available disk space)



Setting	Type of Data	Size
Hyperlink	Text or combinations of text and numbers stored as text and used as a hyperlink address A hyperlink address can have up to three parts: text to display – the text that appears in a field or control. address – the path to a file (UNC path) or page (URL) subaddress – a location within the file/page.	Each part of the three parts of a Hyperlink data type can contain up to 2048 characters.
Attachment	screentip – the text displayed as a tooltip. Any supported type of file.	To attach images, spreadsheet files, documents, charts and other types of supported files to the records in the database and to view and edit attached files.
Lookup Wizard	Creates a field that allows the user to choose	The same size as the primary key field use to perform the lookup, typically 4 bytes.

Table 2.1.1 : Data types and their explanation

Calculated Field: Result of calculation. The calculation must refer to other fields in the same table. You would use the expression builder to create the calculation.

(3) Table Relationships

Although each table stores data about a different subject, tables in a database usually store data about subjects that are related to each other. For example, a database might contain the following:

- A Customers table that lists our company's customers and their addresses.
- A Products table that lists the products that we sell, including prices and pictures for each item.
- An Orders table that tracks customer orders.

Because we store data about different subjects in separate tables, we need some way to tie the data together so that we can easily combine related data from those separate tables. To connect the data stored in different tables, relationships are created. A relationship is a logical connection between two tables that specifies fields that the tables have in common.

Keys: Fields that are part of a table relationship are called keys. A key usually consists of one field, but may consist of more than one field. There are two kinds of keys :

- (i) **Primary key :** A table can have only one primary key which consists of one or more fields that



uniquely identify each record that is stored in the table. Often, there is a unique identification number, such as an ID number or a code, that serves as a primary key.

- (ii) **Foreign key** : A foreign key contains values that correspond to values in the primary key of another table. A table can also have one or more foreign keys. For example, we might have an Orders table in which each order has a customer ID number that corresponds to a record in a Customers table. The customer ID field is a foreign key of the Orders table.

The correspondence of values between key fields forms the basis of a table relationship. We use a table relationship to combine data from related tables. For example, suppose that **Employee** and **Salary** are two database tables. In each **Employee** and **Salary** table, each record is identified by the primary key field, Employee_Id.

Benefits of using relationships

Keeping data separated in related tables produces the following benefits:

- **Consistency** : Because each item of data is recorded only once, in one table, there is less opportunity for ambiguity or inconsistency. For example, we store a customer's name only once in a table about customers, rather than storing it repeatedly (and potentially inconsistently) in a table that contains order data.
- **Efficiency** : Recording data in only one place implies less disk space. Moreover, smaller tables tend to provide data more quickly than larger tables. If separate tables for separate subjects are not used, null values (means the absence of data) and redundancy can waste space and impede performance.
- **Comprehensibility** : The design of a database is easier to understand if the subjects are properly separated into tables.

(4) Introduction to Navigation Pane

When a new or existing database is opened in **Office Access 2010**, the objects in our database - the tables, forms, reports, queries, macros and so on - appear in the **Navigation Pane**. Our tables reside in a group called **Tables**, forms in a group called **Forms** and so on. The Fig. 2.1.2 shows the groups in the **Object Type** category.

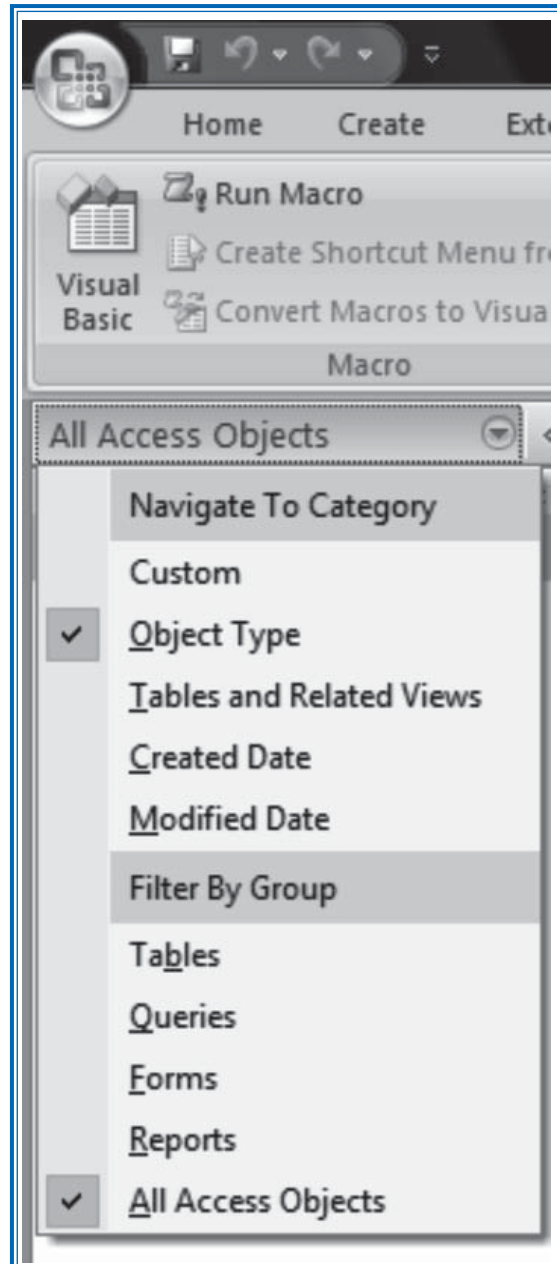



Fig. 2.1.2 : Navigation Pane



The **Navigation Pane** appears by default when a database in **Office Access 2010**. For example, if a data is to be entered into a new table row, the task can be performed from the **Navigation Pane**.

Prevent the Navigation Pane from appearing by default

- (i) Click the **File**  and then click **Access Options**. The **Access Options** dialog box appears.
- (ii) Click **Current Database** and, under **Navigation**, select or clear the **Display Navigation Pane** check box. By default, this option is selected. Fig. 2.1.3 depicts the same.

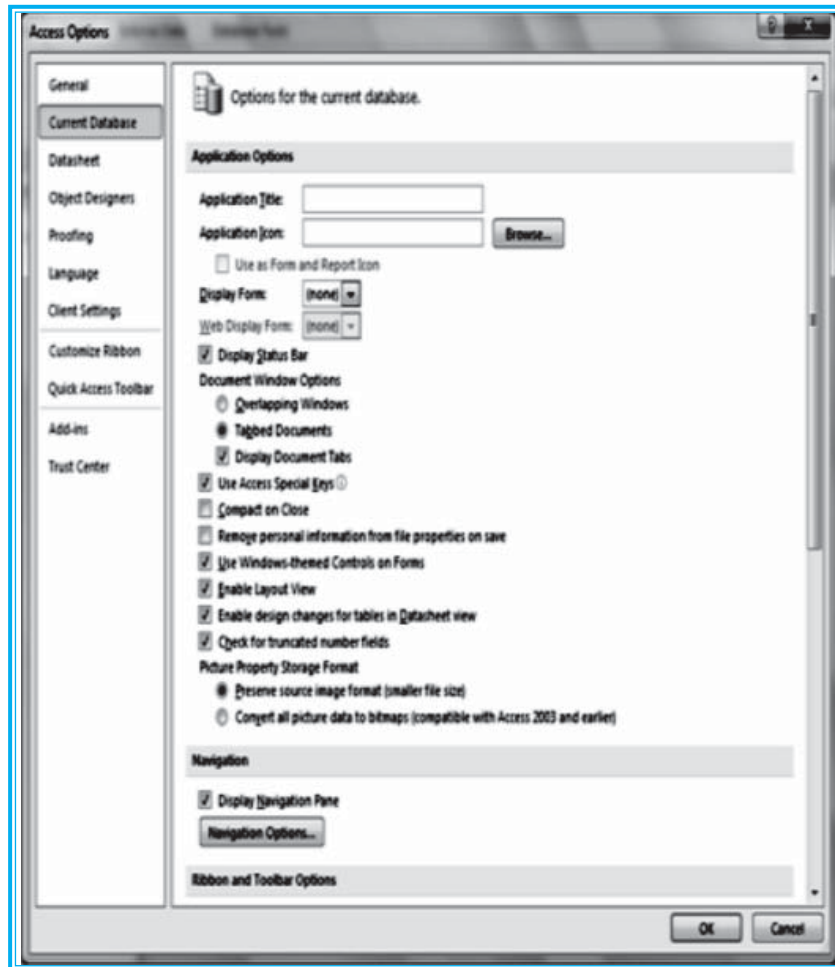


Fig. 2.1.3 : Access Option Dialog Box under Option 'Current Database'

Set global options for the Navigation Pane

Several global options in the **Navigation Pane** can be set or changed according to the user's requirements. The following Table 2.1.2 lists the options and describes how to use them.



Option	Purpose
Show Hidden Objects	Display hidden objects as semi-transparent disabled icons. Clear this option when we want to hide objects completely.
Show System Objects	Display system objects, such as system tables.
Single Click Open	Open objects with a single mouse click.
Double Click Open	Open objects with a double mouse click.

Table 2.1.2 : Options in Navigation Pane

To do so, left click the menu at the top of the **Navigation Pane** and then click **Navigation Options** as shown in Fig. 2.1.4.

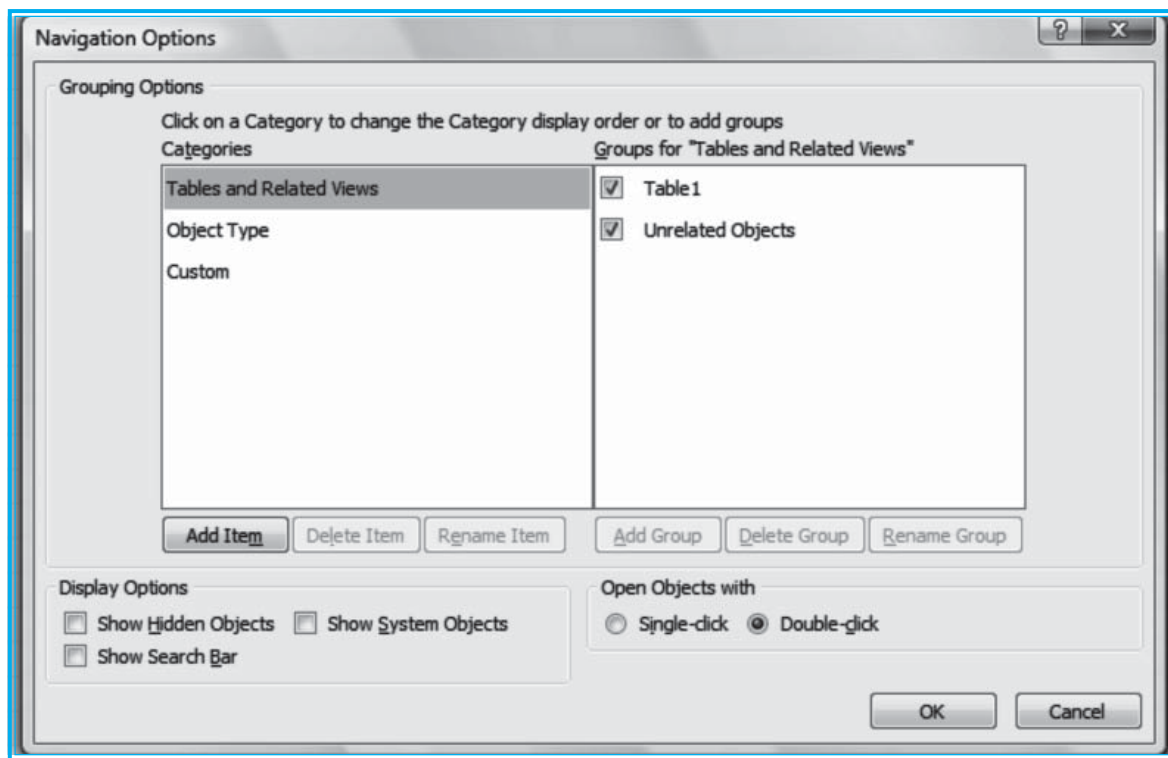


Fig. 2.1.4 : Navigation Options Dialog Box

(5) Database Design

A properly designed database provides us with the access to up-to-date, accurate information. Certain principles that guide the database design are as follows :

- (i) The first principle is that duplicate information (redundant information) is bad, as it wastes space and increases the likelihood of error and inconsistencies.
- (ii) The second principle is that the correctness and completeness of information is important. If our database contains incorrect information, any reports that pull information from the database will also contain incorrect information.



A good database design is, therefore, one that

- divides the information into subject-based tables to reduce redundant data,
- provides **Access** with the information it requires to join the information in the tables together as needed,
- helps support and ensure the accuracy and integrity of the information,
- accommodates the data processing and reporting needs.

The design process consists of the following steps :



- Determine the purpose of the database.
- Find and organize the information required.
- Divide the information into tables.
- Turn information items into columns.
- Specify primary keys.
- Set up the table relationships.
- Refine the design.
- Apply the normalization rules.

2.2 CREATE A NEW TABLE

A simple database, such as a Contact list, might use only a single table, whereas many databases, however, may use several tables. When a new database is to be created, a new file on the computer is to be created that acts as a container for all of the objects in our database, including our tables.


We can create a table by creating a new database, by inserting a table into an existing database or by importing or linking to a table from another data source - such as a Microsoft Office Excel 2007 Workbook, a Microsoft Office Word 2007 document, a text file or another database. When we create a new blank database, a new empty table is automatically inserted for us. We can then enter data to start defining our fields.

2.2.1 Create a new Table in a new Database

- Click the **File**  and then click **New**.
- In the **File**  **Name** box, type a file name. To change the location, click the folder icon to **Browse**.
- Click **Create**.

The new database is opened and a new table named **Table1** is created and opened in **Datasheet view**.

2.2.2 Create a new table in an existing Database

- Click the **File**  and then click **Open**.
- In the **Open** dialog box, select and open the database.

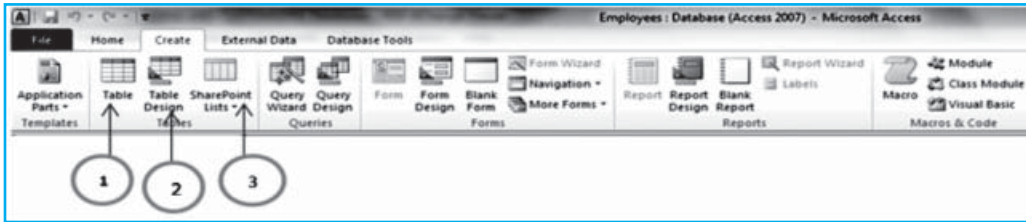


Fig. 2.2.1 : Tables Group

- (iii) On the **Create** tab, in the **Tables** group, click **Table**. (as depicted in point 1, Fig. 2.2.1). A new table gets inserted in the database and the table is opened in **Table tools** as shown in Fig. 2.2.2.

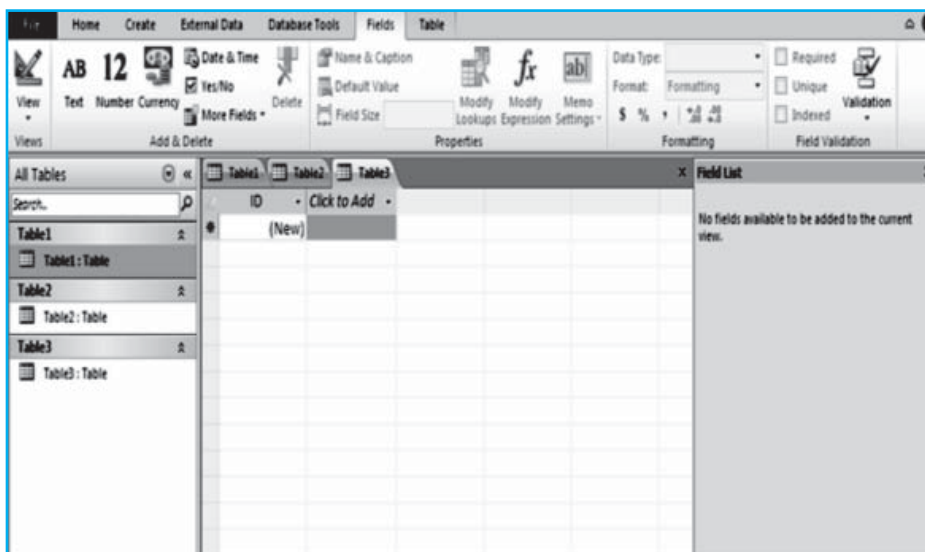


Fig. 2.2.2 : Datasheet view of the Table

2.2.3 Create a table based on a Table Template

To create a **Contacts**, **Tasks**, **Issues**, **Events** or **Assets** table, start with the **Table Templates** that come with **Office Access 2010**. The steps to be followed are as follows :


- (i) Click the **File**  and then click **Open**.
- (ii) In the **Open** dialog box, select and open the database.
- (iii) On the **Create** tab, in the **Templates** group, click **Application parts** and then from **Quick start** select one of the available templates from the list (Point no. 2, Fig. 2.2.1). A new table is inserted, based on the table template (shown in Fig. 2.2.3) we choose.



Fig. 2.2.3 : Table Templates

2.3 IMPORT OR LINK TO CREATE A TABLE




We can create a table by importing or linking to information stored elsewhere. For instance, we can import or link to information in an Excel worksheet, SharePoint list, an XML file, another Access database, a Microsoft Office Outlook 2010 folder and a number of other sources.

- When we import any information, we create a copy of the information in a new table in the current database.
- Conversely, when we link to the information, we create a linked table in the current database that represents a live link to the existing information that is stored elsewhere. Thus, when we change data in the linked table, we change it in the original source too. When information is changed in the original source by using another program, that change is visible in the linked table.

In some cases, we cannot make changes to our data source through a linked table, most notably when the data source is an Excel worksheet.

Note : SharePoint is a web site that can be displayed in a browser such as Internet Explorer and includes an interface for displaying lists and libraries in a secure database. **SharePoint** also contains various web-parts that can display sorted or filtered information in various ways or can display some kind of custom functionality. **SharePoint** can be used for something as simple as a list of contacts or for something as complex as a project management dashboard for a new store opening.

2.3.1 Create a new table by importing or linking to External Data

- (i) To use an existing database, on the **File**  menu, click **Open**.
- (ii) In the **Open** dialog box, select and open the database.
- (iii) To create a new database, on the **File**  menu, click **New**.
 - In the **File**  **Name** text box, type a file name. To change the location, click the folder icon.
 - Click **Create**.
 - The new database is opened and a new table named **Table1** is created and opened in **Datasheet view**.
- (iv) On the **External Data** tab, in the **Import** group, click one of the available data sources as shown in Fig. 2.3.1.

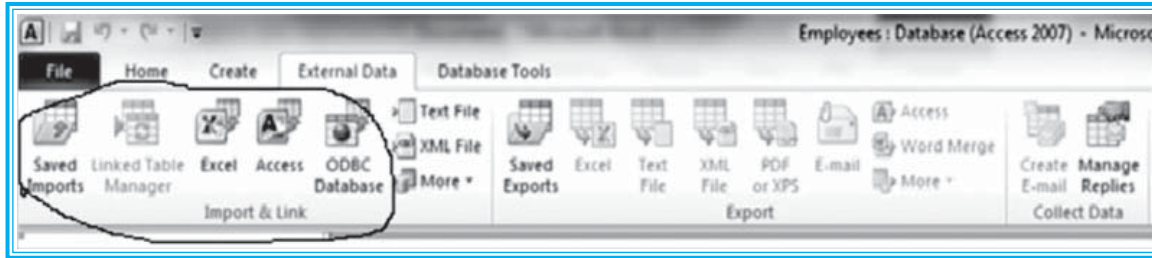


Fig. 2.3.1 : Import Group in External Data Tab

- (v) Follow the instructions in the dialog boxes. **Access** creates the new table and displays it in the **Navigation Pane**.

One can also import or link to a **SharePoint** list by using a command on the **Create** tab.

2.3.2 Create a Table based on a Share Point list

We can create a table in our database that imports from or links to a **SharePoint** list. The list can pre-exist which we can import or link to or we can create a new custom list or a new list based on a predefined template. The predefined templates in **Office Access 2010** include **Contacts**, **Tasks**, **Issues** and **Events** as shown in Fig. 2.3.2.

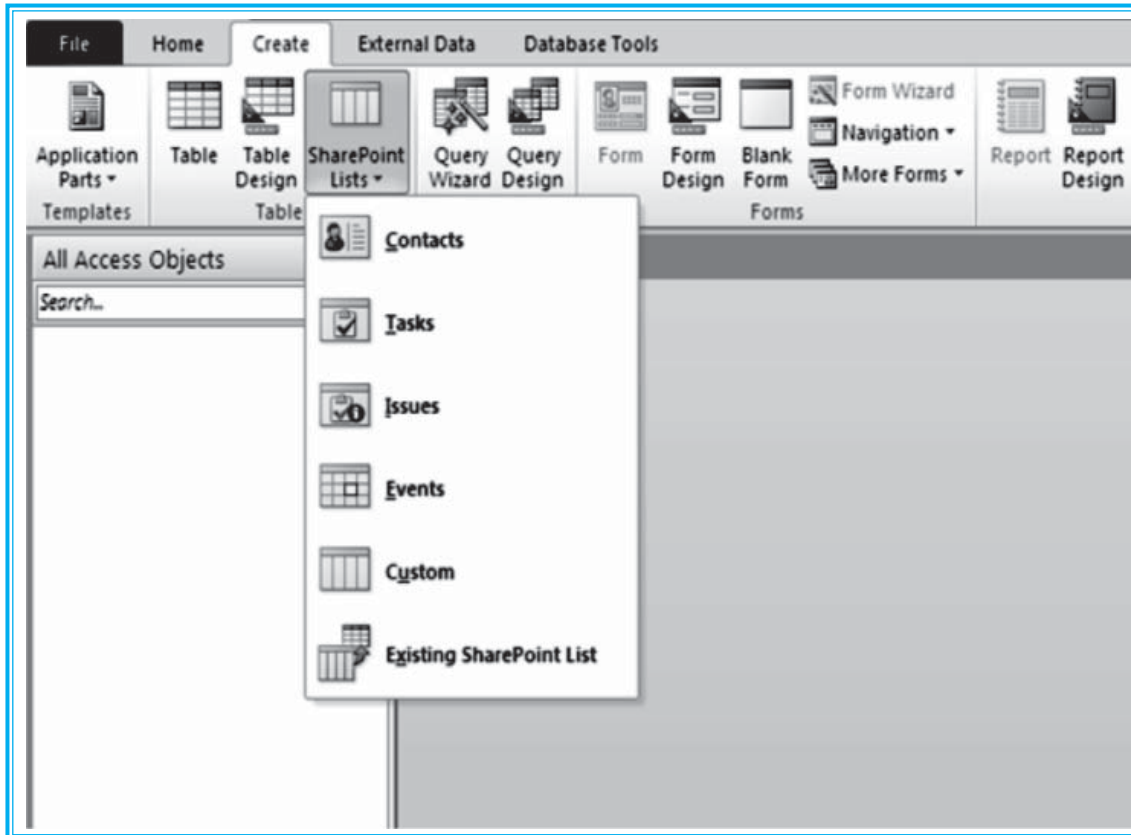



Fig. 2.3.2 : Options in SharePoint Lists



First, open the database in which we want to create the table.

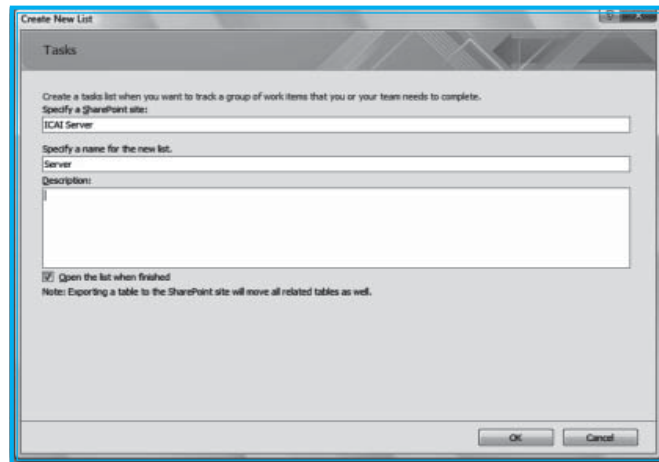
- (i) Click the **File**  and then click **Open**.
- (ii) In the **Open** dialog box, select and open the database.
- (iii) On the **Create** tab, in the **Tables** group, click **SharePoint Lists**. (point 3, Fig 3.2.1)

The Fig. 2.3.2 gets opened showing predefined templates in **Office Access 2010 - Contacts, Tasks, Issues and Events**.

(iv) Then, do one of the following:

1) Create the list based on a template

- (i) Click either **Contacts, Tasks, Issues** or **Events**. Fig. 2.3.3 shows the selection of option **Tasks** from the group of **SharePoint List**.
- (ii) In the **Create New List** dialog box, type the **URL** for the **SharePoint** site where we want the list to reside. Also, enter the name of the new **SharePoint** list and its description. If we want the linked table to open after it is created, select the **Open the list when finished** check box (it is selected by default). Otherwise, clear the check box.



(iii) Click **OK**.

2) Create a new custom list

- (i) Click **Custom** in Fig. 2.3.2. The **Custom** Dialog Box shown in Fig. 2.3.4 gets opened.
- (ii) In the **Create New List** dialog box, type the URL for the **SharePoint** (point 1, Fig. 2.3.4) site where we want the list to reside. Also, type the name of the new **SharePoint list** (point 2, Fig. 2.3.4) and its description.
- (iii) If we want the linked table to open after it is created, select the **Open the list when finished** check box (it is selected by default). Otherwise, clear the check box.
- (iv) Click **OK**.

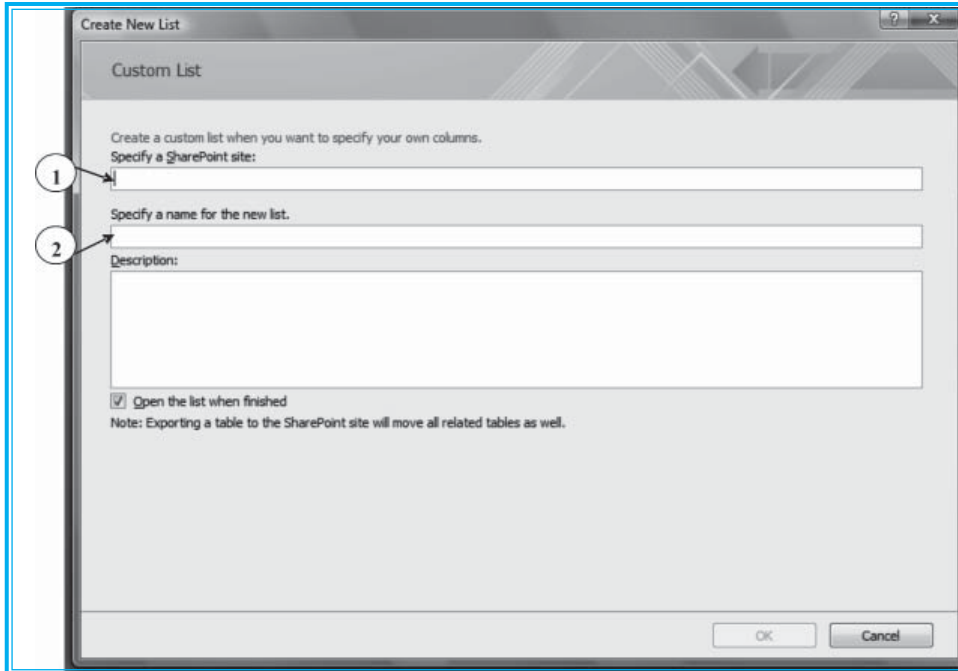


Fig. 2.3.4 : 'Custom' option Dialog Box from SharePoint List

3) Import the data from an existing list

- (i) Click **Existing SharePoint List** in Fig. 2.3.2.
- (ii) In the **Get External Data** dialog box, type the **URL** for the **SharePoint** site that contains the list as depicted in Fig. 2.3.5.
- (iii) Click **Import the source data into a new table in the current database** (point 1, Fig. 2.3.5).
- (iv) Click **Next** and select a check box next to any **SharePoint lists** we want to import.
- (v) Click **OK**.

4) Link to an existing list

- (i) Click **Existing SharePoint List**. The dialog Box shown in Fig. 2.3.5 gets opened.
- (ii) In the **Get External Data** dialog box, type the **URL** for the **SharePoint** site that contains the list.
- (iii) Click **Link to the data source by creating a linked table** (point 2, Fig. 2.3.5).
- (iv) Click **Next**.
- (v) Select a check box next to any **SharePoint lists** to which we want to link.
- (vi) Click **OK**.

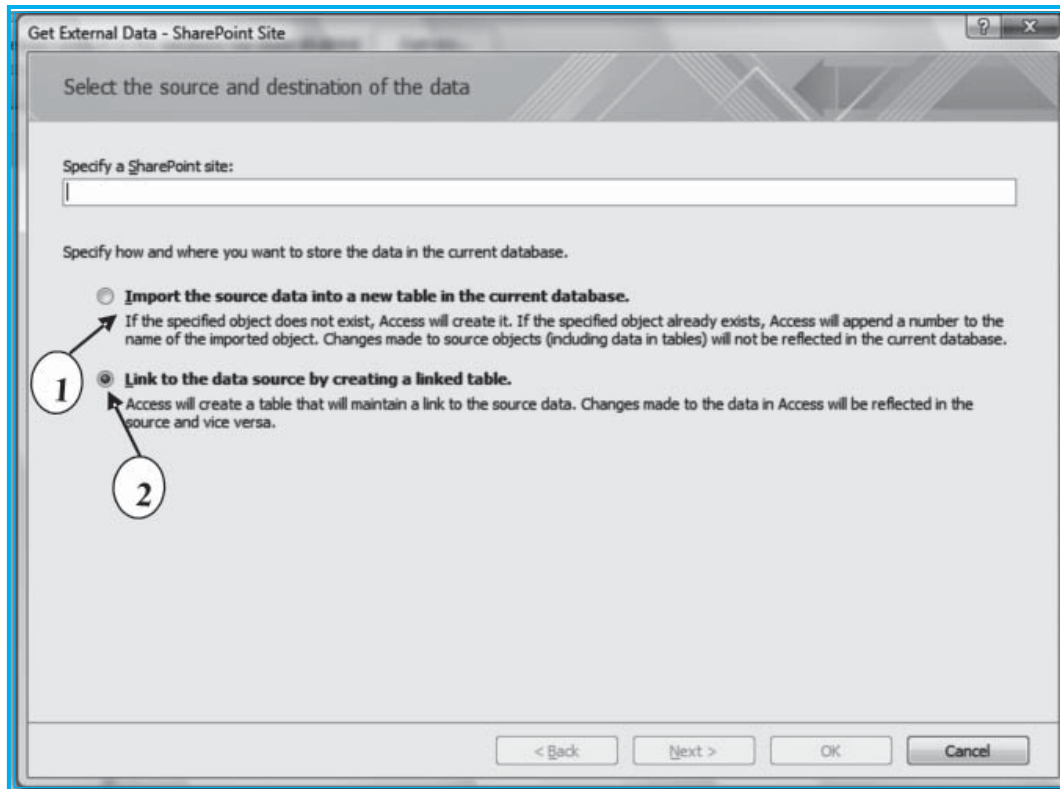


Fig. 2.3.5 : Existing SharePoint List Dialog Box

2.4 ADD FIELDS TO A TABLE IN DATASHEET VIEW

We store the information items we want to track in fields (also called columns). For example, in a **Employee_Details** table, we might create fields for EmpCode, LastName, FirstName, Telephone_Number and Address, among others.

A field defines certain characteristics and are to be selected carefully. For example, every field has a Name that uniquely identifies the field within a table. A field also has a data type that's chosen to match the information to be stored. The data type determines the values that can be stored and the operations that can be performed, as well as how much storage space to set aside for each value. Every field also has an associated group of settings called **properties** that define the appearance or behaviour characteristics of the field. For example, the **Format property** defines a field's display layout - that is, how it should appear when displayed.

When we create a new table, the table opens in **Datasheet view**. We can immediately add a field by typing some information in the **Add New Field** column as shown in Fig. 2.4.1, point 1.

Emp_ID	Emp_FirstName	Emp_LastName	Emp_Gender	Click to Add
001	Joe	Wiley	M	
002	Smith	Sin	M	
003	Parley	John	F	
004	Park	Ray	M	
*				

Fig. 2.4.1 : 'Add New Field' Column Header

2.4.1 Add a new field to an existing table

- (i) Click the **File** button and then click **Open**.
- (ii) In **Open** dialog box, select and open the database.
- (iii) In the **Navigation Pane**, double-click one of the available tables to open it.
- (iv) Type the data in the cell below the **Click to Add** column header.

2.4.2 Add a new field to a new table

- (i) Click the **File** button and then click **Open**.
- (ii) In **Open** dialog box, select and open the database.
- (iii) On the **Create** tab, in the **Tables** group, click **Table** (shown in point 1, Fig. 2.2.1).

Access inserts a new table in the database and opens it in **Datasheet** view. Type the data in the cell below the **Click to Add** column header. When we enter data in the new column, **Office Access 2010** uses the information we type to recognize the appropriate data type for the field. For example, if a date is inserted in a column, such as 1/1/2010, **Office Access 2010** recognizes it as a date and sets the data type for the field to Date/Time. If **Access** doesn't have enough information from what we enter to guess the data type, the data type is set to **Text**. The Table 2.4.1 shows how automatic data type detection works in **Datasheet** view.

If we type	Office Access 2010 creates a data type with a field of
Mary	Text
45,000.98	Number, Double
45000	Number, Long Integer
1/12/2009 or 1st December 2009	Date/Time

Table 2.4.1: Default data type by MS Access 2010

2.4.3 Explicitly set the Data Type/Format

If we want to explicitly set the data type and format for a field, overriding the choice that **Office Access 2010** makes, we can do so by using the commands in the **Data Type & Formatting** group on the **Datasheet** tab as shown in Fig. 2.4.2.

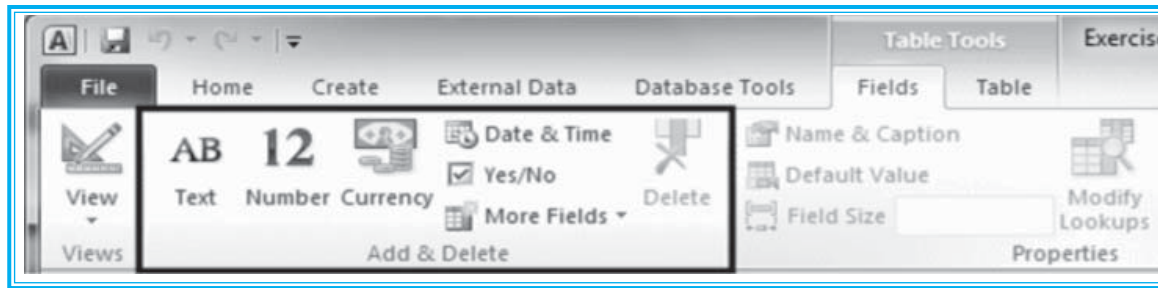


Fig. 2.4.2 : Data Type & Formatting Option in Datasheet View

Do the following steps:

- (i) On the **Field** tab, in the **Add & Delete** group, click **Data Type**.
- (ii) Choose the **data type/format** of our choice.

When we add a field by typing information into the cell below the **Click to Add** header, **Office Access 2010** automatically assigns a name to the field. These names start with **Field1** for the first field, **Field2** for the second field and so on. It is a good practice to use more descriptive field names. One can rename the field by right-clicking its header and then clicking **Rename Column** on the shortcut menu.

Field names can consist of up to 64 characters (letters or numbers), including spaces. We should try to give fields descriptive names so that we can easily identify them when we view or edit records.

2.4.4 Add fields by using field templates

A **Field Template** is a predefined set of characteristics and properties that describes a field. The field template definition includes a field name, a data type, a setting for the field's format property and a number of other field properties that, when taken together, form a model that serves as the basis for creating a new field.

One can use the **Field Templates** task pane to choose from a list of predefined fields than to manually create a field. **Office Access 2010** comes with a set of built-in field templates that can save our considerable time when we create fields. To create a new field using a field template, we display the **Field Templates** pane and then drag and drop one or more templates to the table that is opened in **Datasheet view**.

- (i) Open up the table in a **Datasheet view**.
- (ii) On the **Field** tab, in the **Add & Delete** group, click **More Fields** as shown in point 1, Fig. 2.4.3.

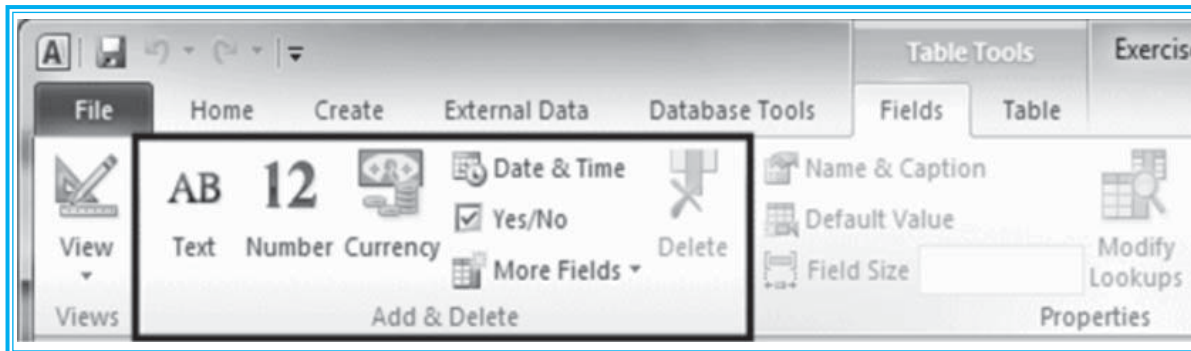
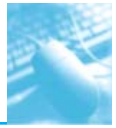


Fig. 2.4.3 : Fields & Columns Option in Datasheet View



- (iii) Select one or more fields in the **Field Templates** pane and drag them to the table. The fields appear in the datasheet as shown in Fig. 2.4.4.

Field templates are models, which provide a basis from which we can create a new field. In addition to creating a new field from a field template, we can also create a new field from an existing field in another table.

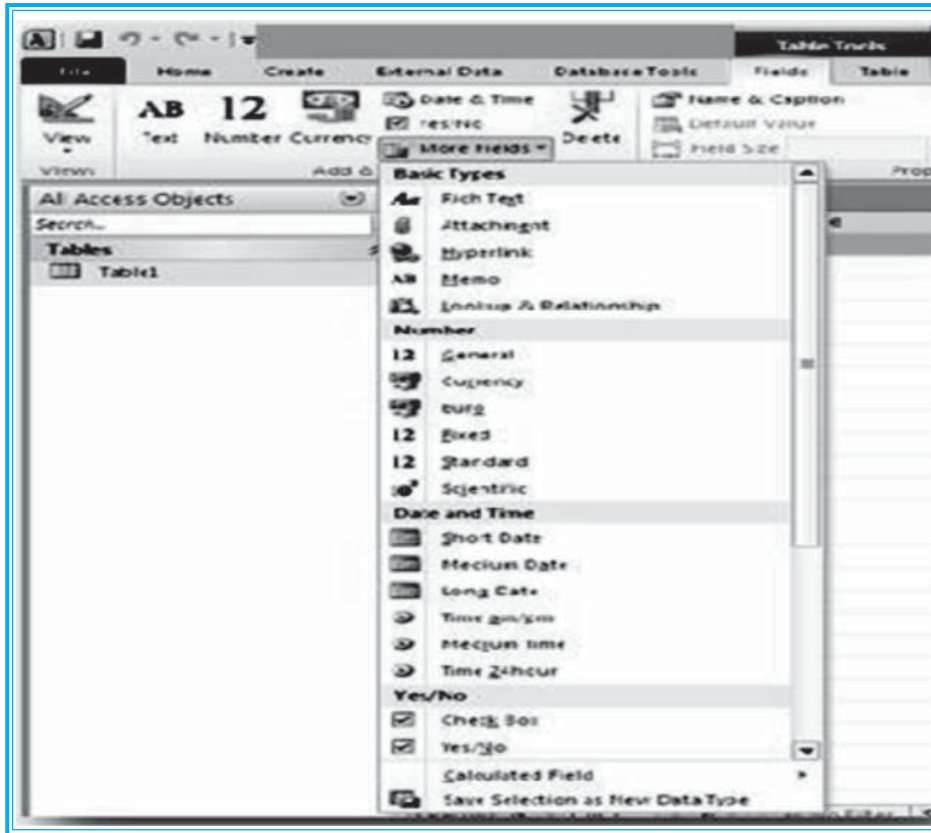


Fig. 2.4.4 : Field Template in Datasheet View

2.4.5 Add a field from an existing table

If a database contains multiple tables, we can add a field from another table. In a relational database, we store information in separate subject-based tables and then define relationships to bring the information together as needed. **Office Access 2010** allows user to create a new field by adding it from another related table or by adding it from another table for which no relationship is created. We add the field by using the **FieldList**, which shows us fields available in other tables in our database.

1) Open and add a field from the Field List pane

- (i) Click the **File** button and then click **Open**.
- (ii) In the **Open** dialog box, select and open the database.
- (iii) In the **Navigation Pane**, double-click the table in which we want to add an existing field. The table opens in **Datasheet view**.



- (iv) On the **Datasheet** tab, in the **Fields & Columns** group, click **Add Existing Fields**, as shown in point 2, Fig. 2.4.3. The **Field List** pane appears as depicted in Fig. 2.4.5.

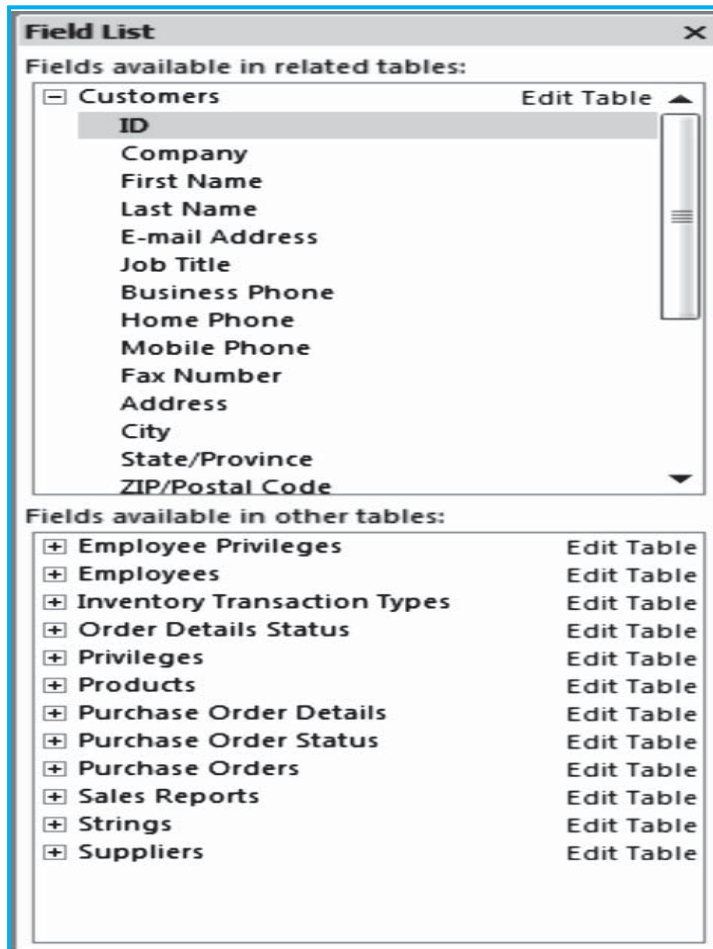


Fig. 2.4.5 : FieldList Pane

The **Field List** pane lists all of the other tables in our database, grouped into categories.

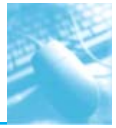
- (v) Click the plus sign (+) next to a table to display the list of fields in that table and drag the field we want from the Field List pane to the table in Datasheet view. The Lookup Wizard starts. Follow the instruction to complete the wizard.

2.5 SAVE A TABLE

Once a table's fields are added and its design is set, one should save it. When a new table is to be saved for the first time, a name is to be given to it that describes the information it contains. We can use up to 64 characters (letters or numbers), including spaces, for example, a table name may be Customers, Parts Inventory, Products and so on.

To save a table, do any of the following :

- (i) Click the **File**  and then click **Save**.



- (ii) Right-click the table's document tab and then click **Save** on the shortcut menu.
- (iii) Click **Save** on the **Quick Access Toolbar**.

If we are saving the table for the first time, type a name for the table and then click **OK**.

2.6 SET THE PRIMARY KEY

The primary key of a table consists of one or more fields that uniquely identify each row we store in the table. The primary key is never empty or null - there is always a value and it rarely (ideally never) changes. **Access** uses primary key fields to quickly bring together data from multiple tables. We should always specify a primary key for a table. **Access** automatically creates an index for the primary key, which helps speed up queries and other operations and ensures that every record has a value in the primary key field and that it is always unique. For example, we might have an **Employee** table where each employee has a unique employee_ID, which can be made as a primary key of the table.

When we create a new table in **Datasheet view**, **Access** automatically creates a primary key for us and assigns it a field name of ID and the **AutoNumber** data type. The field is hidden in **Datasheet view**, but we can see it if we switch to **Design view**.

2.6.1 Switch to Design View

To switch to **Design View**, do any of the following:

- (i) Right click the document tab and then click **Design View**.
- (ii) Right-click the table name in the **Navigation Pane** and then click **Design View**.
- (iii) Click **Design View** in the **Access** status bar.

To change or remove the primary key or to set the primary key for a table that doesn't already have one, we must use **Design view**.

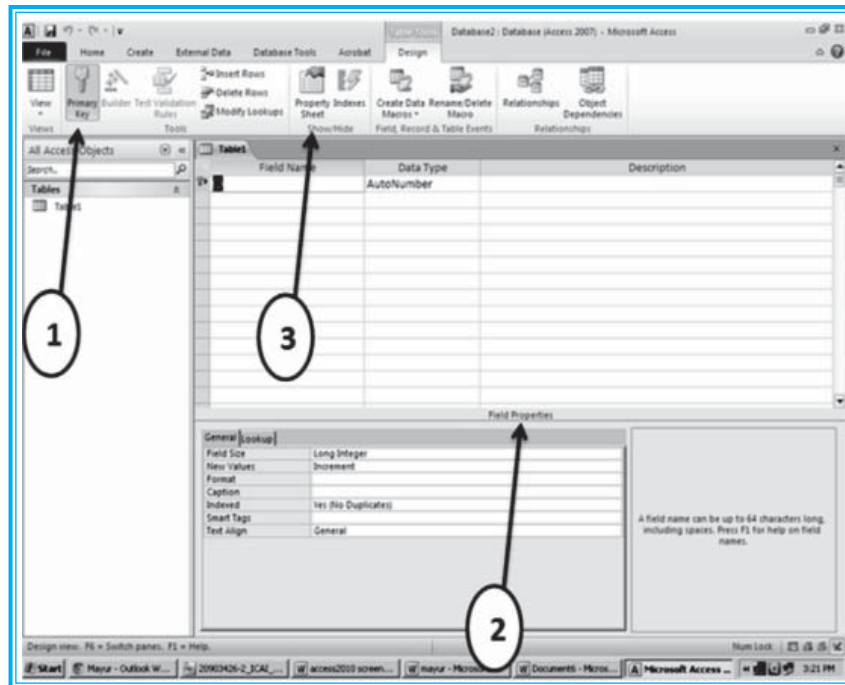


Fig. 2.6.1 : Design View

1) Set or change the primary key

When a new table is created without setting a primary key, Access prompts the user to create one. If we choose **Yes**, **Access** creates an **ID** field that uses the **AutoNumber** data type to provide a unique value for each record. If the table already includes a **AutoNumber** field, **Access** uses it as the primary key. Do the following:

- Open the table in **Design view**.
- Select the field or fields we want to use as the primary key. To select one field, click the row selector for the field we want. To select more than one field, hold down the **CTRL** key and then click the row selector for each field.
- On **Design** tab in **Tools** group, click **Primary Key** as shown in Fig. 2.6.1, point 1. A key indicator is added to the left of the field or fields that we specify as the primary key.

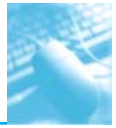
2) Remove the primary key

- Open the table in **Design view**.
- Click the row selector for the current **primary key**. If the primary key consists of a single field, click the **row selector** for that field. If the primary key consists of multiple fields, click the **row selector** for any field in the primary key.
- On the **Design** tab, in the **Tools** group, click **Primary Key**. (point 1, Fig. 2.6.1).

The key indicator is removed from the field or fields we previously specified as the primary key.

2.7 SET FIELD PROPERTIES

One can control the appearance of information, prevent incorrect entries, specify default values, speed up searching and sorting and control other appearance or behavior characteristics by setting



field properties. For example, we can format numbers to make them easier to read or we can define a validation rule that must be satisfied for information to be entered in a field.

The data type of the field determines the properties we can set. For example, the **Format**, **InputMask** and **Caption** properties affect how our information appears in table and query datasheets. In addition, any controls on new forms and reports that are based on the fields in the table inherit these same property settings by default.

1) Set a field property for a table in Datasheet view

- (i) Open the table in **Datasheet view**.
- (ii) Click in the field for which we want to set the property.
- (iii) On the **Field** tab, in the **Field Validation** group, select one of the following commands in Fig. 2.7.1:
 - **Unique** - Sets the **Indexed** property.
 - **Is Required** - Toggles the setting of the Required property. If it was set to **Yes**, it is set to **No**. If it was set to **No**, it is set to **Yes**.

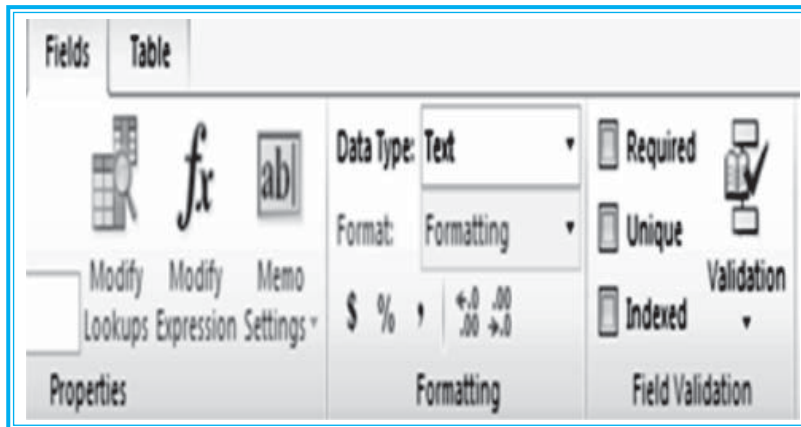


Fig. 2.7.1 : Datasheet View

We can set only a few of the available field properties in **Datasheet view**. To set additional field properties, we must open the table in **Design view**. To have access to and set the complete list of field properties, we must use **Design view**.

2) Set a field property for a Table in Design view

- (i) Open the table in **Design view**.
- (ii) In the upper portion of the table **Design grid**, click the field for which we want to set properties (or move to the field by using the arrow keys).
- (iii) **Access** displays the properties for this field in the lower portion of the table **Design grid**. The data type of the field determines the properties we can set.
- (iv) In the lower portion of the window, under **Field Properties**, (point 2, Fig. 2.6.1) click the box for the field property we want to set. Alternatively, we can press **F6** and then move to the property by using the arrow keys.




- (v) Type a setting for the property or, if an arrow appears at the right side of the property box, click the arrow to choose from a list of settings for the property. The following Table 2.7.1 lists the available field properties.

Use this Field Property	To
Field Size	Set the maximum size for data stored as a Text , Number or AutoNumber data type.
New Values	Set whether an AutoNumber field is incremented or assigned a random value.
Format	Customize the way the field appears when displayed or printed.
Caption	Set the text displayed by default in labels for forms, reports and queries.
Indexed	Speed up access to data in this field by creating and using an index.
Smart Tags	Attach a smart tag to this field.
Text Align	Specify the default alignment of text within a control.

Table 2.7.1: Field Property

To save the changes made, do any of the following:

- (i) Click **Save** on the **Quick Access Toolbar**.
- (ii) Right-click the table's document tab and then click **Save** on the shortcut menu.
- (ii) Click the **File**  and then click **Save**.

The following table provides additional information about the **FieldSize**, **Format** and **New Values** field properties.

2.8 SET TABLE PROPERTIES

In addition to field properties, we can set properties that apply to an entire table and to entire records. We set these properties in the table's property sheet.

1) Set a Table property

- (i) Open the table in **Design view**.
- (ii) On the **Design** tab, in the **Tools** group, click **Property Sheet** (point 3, Fig. 2.6.1). The table's property sheet is displayed.
- (iii) Click the box for the property we want to set and type a setting for it as shown in Fig. 2.8.1.



Property Sheet	
Selection type: Table Properties	
General	
Display Views on SharePoint	Follow Database Setting
Subdatasheet Expanded	No
Subdatasheet Height	0"
Orientation	Left-to-Right
Description	
Default View	Datasheet
Validation Rule	
Validation Text	
Filter	
Order By	
Subdatasheet Name	[Auto]
Link Child Fields	
Link Master Fields	
Filter On Load	No
Order By On Load	Yes

Fig. 2.8.1: Property Sheet

The following Table 2.8.2 lists the available table properties.


Use this Table Property	To
Display Views On Share Point	Specify whether forms and reports associated with this table should be available on the View menu in Windows Share Point Services if the database is published to a SharePoint site.
Subdatasheet Expanded	Set whether to expand all subdatasheets when we open the table.
Subdatasheet Height	Specify whether to expand to show all available subdatasheet rows (the default) when opened or set the height of the subdatasheet window to show when opened.
Orientation	Set the view orientation, according to whether our language is read left-to-right or right-to-left.
Description	Provide a description of the table.
Default View	Set Datasheet , Pivot Table or PivotChart as the default view when we open the table.
Validation Rule	Supply an expression that must be true whenever we add a record or change a record.
Validation Text	Enter text that appears when a record violates the Validation Rule expression.



Filter	Define criteria to display only matching rows in Datasheet view .
OrderBy	Select one or more fields to specify the default sort order of rows in Datasheet view .
SubdatasheetName	Specify whether a subdatasheet should appear in Datasheet view and if so, which table or query should supply the rows in the subdatasheet.
LinkChildFields	List the fields in the table or query used for the subdatasheet that match this table's primary key field(s).
LinkMasterFields	List the primary key field(s) in this table that match the child fields for the subdatasheet.
FilterOnLoad	Automatically apply the filter criteria in the Filter property (by setting to Yes) when the table is opened in Datasheet view .
OrderByOnLoad	Automatically apply the sort criteria in the OrderBy property (by setting to Yes) when the table is opened in Datasheet view .

Table 2.8.2 : Table Properties in /Access

To save the changes made, do any of the following:

- (i) Click **Save** on the **Quick Access Toolbar**.
- (ii) Right-click the table's document tab and then click **Save** on the shortcut menu.
- (iii) Click the **File**  and then click **Save**.

2.9 DELETING A TABLE

If we have a table we don't need anymore, we can remove it from our database. Because we cannot delete a table if it is opened, we must first close it. To remove a table from our database, do any of the following:

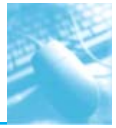
- (i) In the **Navigation Pane**, right click the table and click **Delete**.
- (ii) In the **Navigation Pane**, click the table to select it. Then, on the **Ribbon**, click **Home**. In the **Records** group, click **Delete**.
- (iii) In the **Navigation Pane**, click the table to select it and press **Delete**.

2.10 RENAMING A TABLE

If for some reason, the name of the table is to be changed, it can be done by right-clicking the table in the **Navigation Pane** and clicking **Rename**.

Sources:

- [1] <http://cisnet.baruch.cuny.edu/holowczak/classes/2200/access/accessall.html>
- [2] Microsoft Office Access Help



2.11 CREATING FORMS

This topic deals with forms, their importance, general types of forms available under **MS-Access** and how to create simple form using form wizard and form design view. It also put focus on the various controls, their types and properties which could be used while designing a form using form design view. The chapter also explores the topic on how to insert controls in a form.

2.11.1 Introduction

Although **Access** provides a convenient spreadsheet-style datasheet view for entering data, it is not an ideal tool for safe and secured data entry. In such situation Microsoft Access 2010 offers a facility called forms through which the user can enter data to their respective databases. A form provides the user with a better interface to work with. Microsoft Access 2010 form offers many new options for creating a form which make form creation much easier. Generally forms can be created using the form wizard or by opting for form design view where the user needs to create the form manually. The form should be created in such a way that it reflects its user friendliness and the user should feel at ease while using it.

2.11.2 Definition

A form is a database object that the user can use to enter, edit, display data from a table or a query. The user can use the forms to control access of data in a database.

2.11.3 Form and Its Types

As mentioned above, the user can create a form in **MS-Access** in either of the following two ways:-

- Choose the **Form Wizard** provided by **Microsoft Access**. Opting for form wizard make the form creation an easy task. But the form thus created is very simple with little options.
- Create form from scratch using **Form Design** view. Opting for form design view provides the user with more controls to customize and design the form in a complex yet effective way.

The general types of forms that MS-Access offers are as follows:-

2.11.3.1 Simple Form

This is the simplest and most common type of form available in **MS-Access**. It lets the user create a form that lets him / her to enter information for one record at a time.

2.11.3.2 Split Form

This form lets the user see the datasheet in the upper section and a form in the lower section for entering information about the record selected in the datasheet.

2.11.3.3 Multiple Items

This form lets the user to see multiple records in a datasheet, with one record per row.

2.11.3.4 Blank Form

This form lets the user start with a blank form and then add fields and other controls manually.

2.11.3.5 PivotChart

PivotChart lets the user create a PivotChart form.

2.11.3.6 Form Wizard

Form wizard lets the user create a form instantly using step by step instructions.



2.11.3.7 Datasheet

Datasheet lets the user see a form in a datasheet view. The user can see the data in a row and column format in which the user can provide limit on which fields are displayed and editable.

2.11.3.8 Modal Dialog

Modal Dialog lets the user create a dialog box which may be used by a user to host or carry other controls. A dialog box is characterized as modal if the user must close it before continuing with another task on the same application.

2.11.3.9 PivotTable

In **MS-Access 2010** the user instead of creating PivotTable forms, can open an existing form in PivotTable view and build the layout of the view or use the PivotTable wizard to help you design the PivotTable view for a new form.

2.11.3.10 Form Design

Form Design lets the user create a blank form and displays it in design view.

2.11.4 General Steps to create a form

The general steps to create a form in **MS-Access** are as follows:-

- Choose the table or query for creating the form.
- Specify the type of form by selecting from **Forms** option under **Create** menu tab.
- The user may need to provide necessary details for successful creation of form.
- For activities adding records, editing records, etc., the user may switch to form view. And for customization of form, the user may go to design view.
- After the creation, the newly created form should be saved under a meaningful and valid name.

2.11.5 Create a Form using Form Wizard

Form wizard may be used by a user to automatically create a simple form. Although, the form wizard provides less control in creating a form but the user has the option to customize using form design view.

Source:

Database : Northwind 2010

Table : Customers

Steps:-

1. On the **Create** tab, in the **Forms** group, click **More Forms** and choose **Form Wizard** as shown in Fig. 2.11.1.



Fig. 2.11.1: Form Wizard (i)

2. Now, specify the table or query for which the user wants to create the form as shown in Fig. 2.11.2.

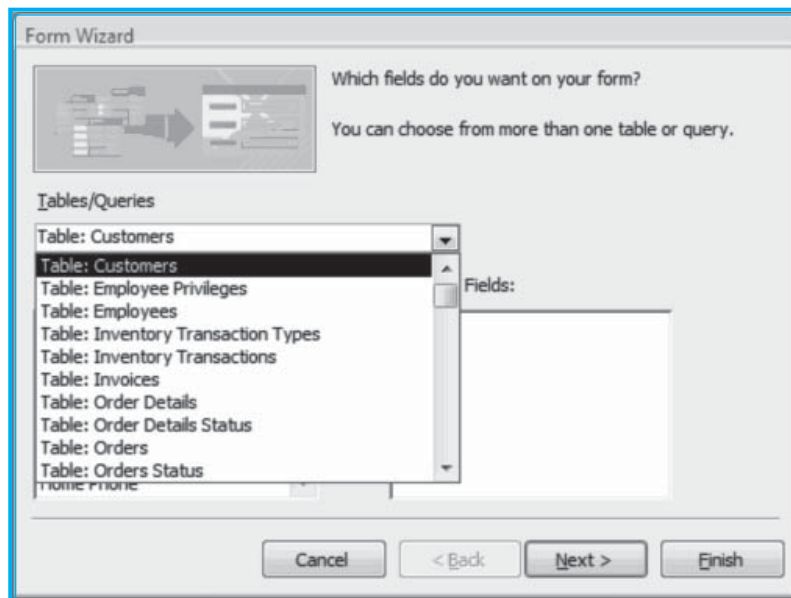
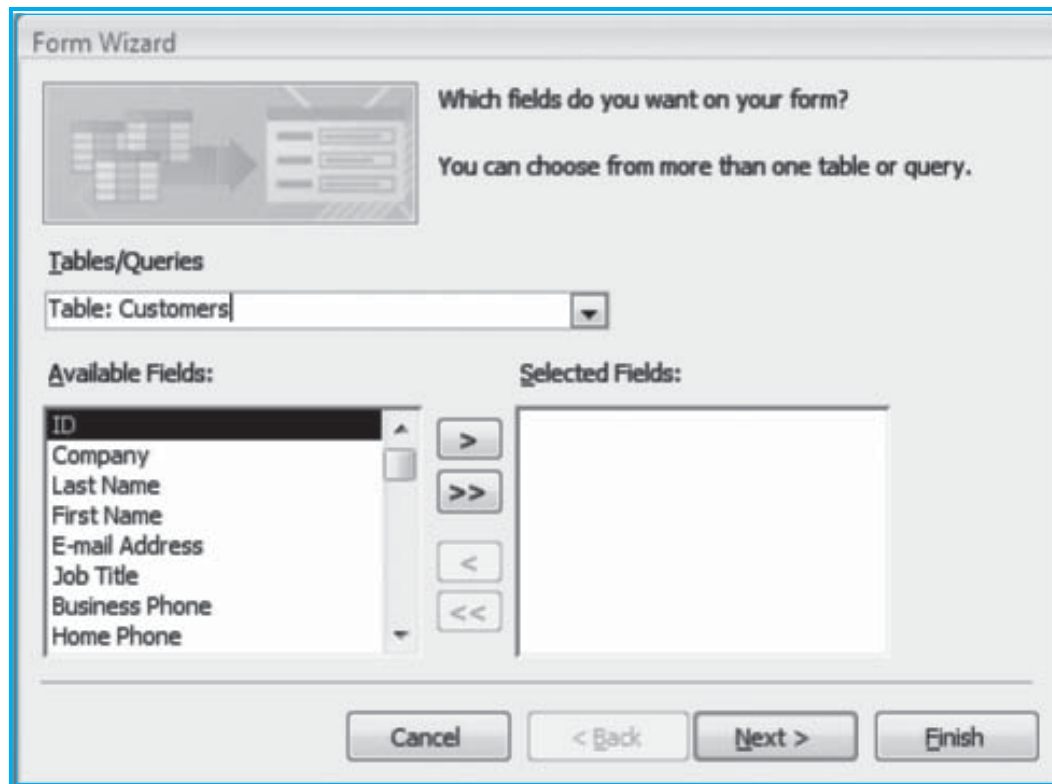


Fig. 2.11.2: Form Wizard (ii)

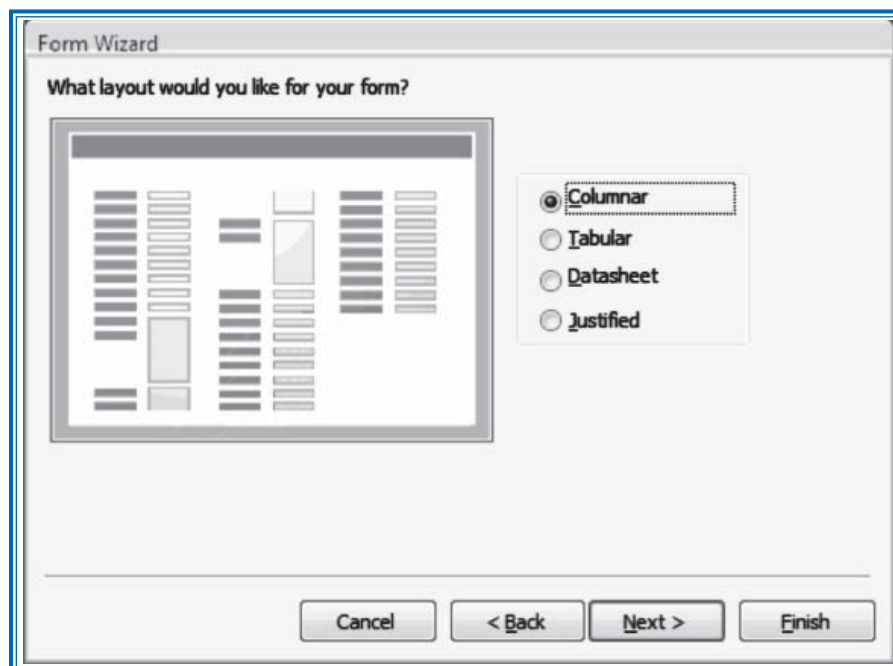
After choosing the table or query, the user will see a screen similar to Fig. 2.11.3 as shown below:-



The 'Form Wizard' dialog box is shown at step (iii). It asks 'Which fields do you want on your form?' and 'You can choose from more than one table or query.' Below this, there is a 'Tables/Queries' list with 'Table: Customers' selected. To the right, there are two lists: 'Available Fields' and 'Selected Fields'. The 'Available Fields' list contains: ID, Company, Last Name, First Name, E-mail Address, Job Title, Business Phone, and Home Phone. The 'Selected Fields' list is currently empty. Between the two lists are four buttons: '>', '>>', '<', and '<<'. At the bottom of the dialog are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

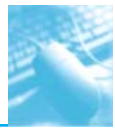
Fig. 2.11.3: Form Wizard (iii)

After selecting the desired field(s) from the table or query, the following screen will appear as shown in Fig. 2.11.4.



The 'Form Wizard' dialog box is shown at step (iv). It asks 'What layout would you like for your form?'. On the left, there is a preview window showing a form layout with various fields. On the right, there are four radio button options: 'Columnar' (selected), 'Tabular', 'Datasheet', and 'Justified'. At the bottom of the dialog are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'.

Fig. 2.11.4: Form Wizard (iv)



Here, the user needs to specify the form layouts which are as follows:-

Columnar: By default, the form layout is Columnar as shown in Fig. 2.11.4.

Tabular: As shown in Fig. 2.11.5.

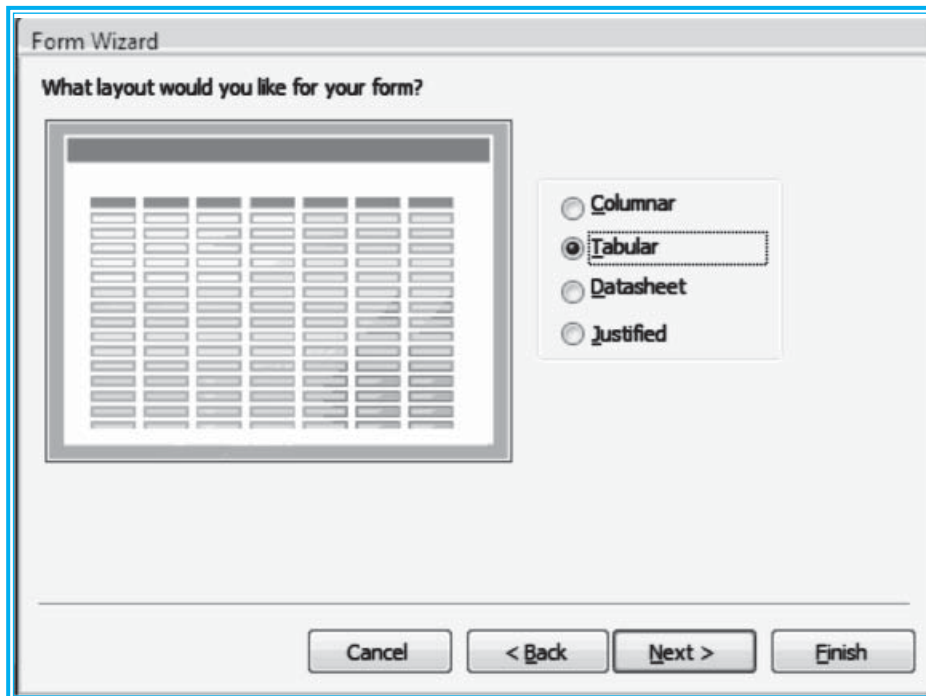


Fig. 2.11.5: Form Wizard (v)

Datasheet: As shown in Fig. 2.11.6.

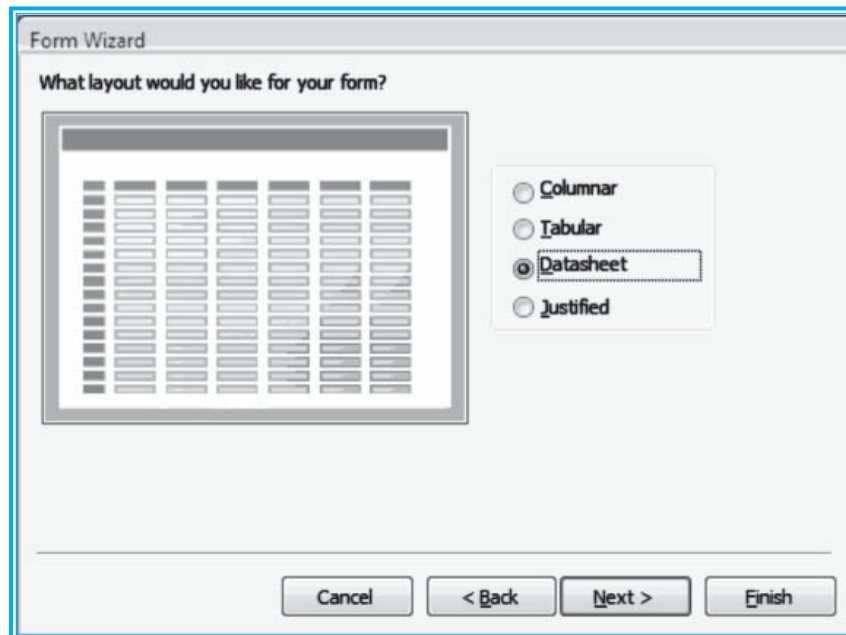


Fig. 2.11.6: Form Wizard (vi)



Justified: As shown in Fig. 2.11.7.

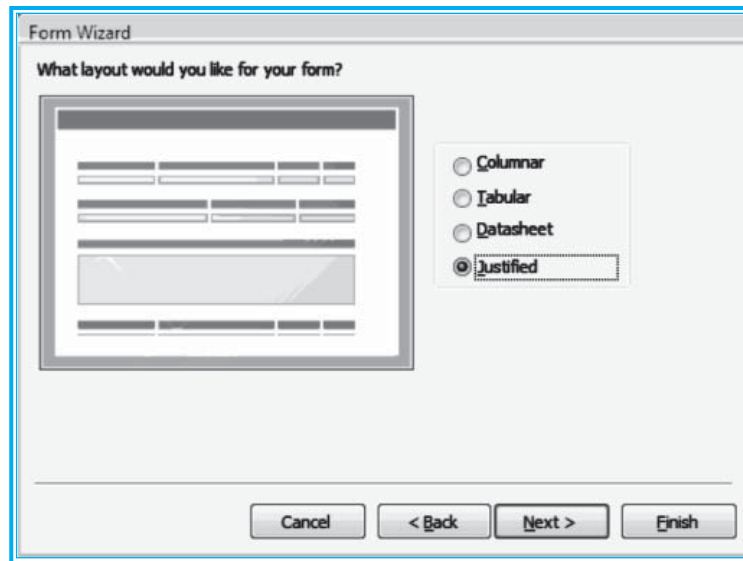


Fig. 2.11.7: Form Wizard (vii)

4. The user may then specify the title of the form and may further specify whether they want to open the newly created form in form view to carry out form related activities like addition of records, editing existing records, etc or to modify the form in form design view for further customization as shown in Fig. 2.11.8.

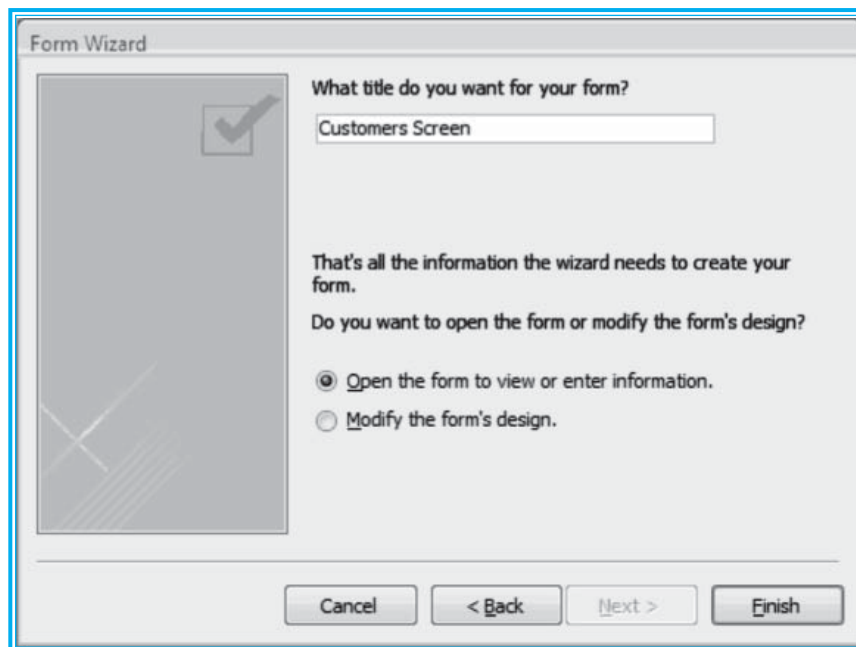


Fig. 2.11.8 : Form Wizard (viii)



After making the choice, the user can click **Finish** to see a screen similar to the Fig. 2.11.9.

Fig. 2.11.9 : Resultant Screen

2.12 CREATE FORM USING FORM VIEW

The user may use the form view available under **Forms** Group of **Create** tab as shown in Fig. 2.12.1.



Fig. 2.12.1: 'Form' option

Source:

Database : Northwind 2010

Table : Customers



Steps:-

1. Select the desired table or query for the form to be created.
2. Click the **Form** view option available under **Forms** group of **Create** tab as shown in Fig. 2.12.1.
3. The user will be shown a screen similar to Fig. 2.12.2.

Fig. 2.12.2: Form Design Screen

4. The user may further go to design view to customize the form as per the requirements.
Now, before proceeding with the creation of form through design view, the user needs to understand the various form controls that are available to them for placing in the form as per the field types.

2.13 CONTROLS

Control is an object on a form such as labels, text boxes, buttons etc.

The various form controls available under **Control** group, as shown in Table 2.13.1, are as follows:-

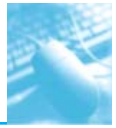
Logo		Lets the user to insert a picture into the form.
Title		Lets the user specify a title of the form.
Insert Page Number		Lets the user insert page number.
Date & Time		Insert current date and time.



Text Box		Displays and allows users to edit data.
Label		Displays static text that typically doesn't change.
Button		Also called a command button. Calls macros or runs VBA code when clicked.
Combo Box		A drop-down list of values.
List Box		A list of values that is always displayed on the form or report.
Subform / Subreport		Displays another form or report within the main form or report.
Line		A graphical line of variable thickness and color, which is used for separation.
Rectangle		A rectangle can be any color or size or can be filled in or blank; the rectangle is used for emphasis.
Bound Object Frame		This frame holds an OLE object or embedded picture that is tied to a table field.
Option Group		Holds multiple option buttons, check boxes or toggle buttons.
Check Box		A two-state control, shown as a square that contains a check mark if it's on and an empty square if it's off.
Option Button		Also called a radio button, this button is displayed as a circle with a dot when the option is on.
Toggle Button		This is a two-state button-up or down-which usually uses pictures or icons instead of text to display different states.
Tab Control		Displays multiple pages in a file folder type interface.
Insert Page		Adds a "page" on the form or report. Additional controls are added to the page and multiple pages may exist on the same form.
Insert Chart		This chart displays data in a graphical format.



Unbound Object Frame		This frame holds an OLE object or embedded picture that is not tied to a table field and can include graphs, pictures, sound files and video.
Image		Displays a bitmap picture with very little overhead.
Insert or Remove Page Break		This is usually used for reports and indicates a physical page break.
Insert Hyperlink		This control creates a link to Web page, a picture, an e-mail address or a program.
Attachment		This control manages attachments for the Attachment data type.
Line Thickness		This control is used to specify the line thickness.
Line Type		This control is used to specify the line type.
Line Color		This control is used to specify the line color.
Special Effect		This control is used to specify special effect for a control on a form like flat, raised etc.
Set Control Defaults		<p>This control doesn't add a control to a form. Instead, it determines whether a wizard is automatically activated when you add certain controls.</p> <p>Note: The Option Group, Combo Box, List Box, Subform / Subreport, Bound and Unbound Object Frame and Command Button controls all have wizards that Access starts when the user add a new control.</p>
Select All		This control is used to select all the control present on a form.
Select		This control is used to select a particular control present on a form.
Control Wizard		
Insert ActiveX Control		This control is used to insert ActiveX controls like RealPlayer G2 Controls, Windows Media Player, Yahoo! Toolbar etc.



2.13.1 Types of Controls

These are the three basic categories of controls available under MS-Access, which are as follows:-

Bound Controls: Bound controls are those controls which are bound to a table field. When the user enters a value into a bound control, Access automatically updates the table field in the current record. Most of the controls that let the user enter information can be bound. These include **OLE (Object Linking and Embedding)** fields. Controls can be bound to most data types, including text, dates, numbers, Yes / No, pictures and memo fields.

Unbound Controls: Unbound controls are those controls which retain the entered value, but they don't update any table fields. The user can use these controls for text label display, for controls such as lines and rectangles or for holding unbound OLE objects (such as bitmap pictures or logo) that aren't stored in a table but on the form itself. Unbound controls are also known as variables or memory variables.

Calculated Controls: Calculated controls are those controls which are based on expressions, such as functions or calculations. Calculated controls are also unbound because they don't update table fields. An example of a calculated control is $\text{=[Quantity]} \times \text{[Price]}$. This control calculates the total of two table fields for display on a form but is not bound to any table field.

There are two ways to add a control in a form which is as follows:

- User may click a button in the Design ribbon's Controls group and draw a new unbound control on the form.
- User may drag a field from the Field List to add a bound control to the form.

A bound control is one that is linked to a table field, while an unbound control is one that is not bound to a table field. A control bound to a table places the data directly into the table by using the form.

2.13.2 Properties

Properties are named attributes of controls, fields or database objects that are used to modify the characteristics of a control, field or object. Examples of these attributes are the size, color, appearance or name of an object. Properties are used extensively in forms and reports to change the characteristics of controls. Each control on the form has properties. The form itself also has properties. A property can also modify the behavior of a control, determining, for example, whether the control is read-only or editable and visible or not visible.

Properties are displayed in a property sheet also called property window. Whenever a user clicks any control present on the form, the property sheet shows the related properties of that control. The users can then change or modify the associated properties of that control from the property sheet. The property sheet groups the types of property of an associated control and presents them to the user under specific tabs which are as follows:-

Format: These properties determine how a label or value looks: font, size, color, special effects, borders and scroll bars.

Data: These properties affect how a value is displayed and the data source it is bound to: control source, input masks, validation, default value and other data type properties.

Event: Event properties are named events, such as clicking a mouse button, adding a record, pressing a key for which you can define a response (in the form of a call to a macro or a VBA procedure) and so on.

Other: Other properties show additional characteristics of the control, such as the name of the control or the description that displays in the status bar.

All: It lets the user see all the properties for a control.



2.14 INSERTING CONTROLS ON A FORM

2.14.1 Insert Logo

1. Click **Logo** option under **Header/Footer** group. The user will be prompted to specify image or picture as shown in Fig. 2.14.1.

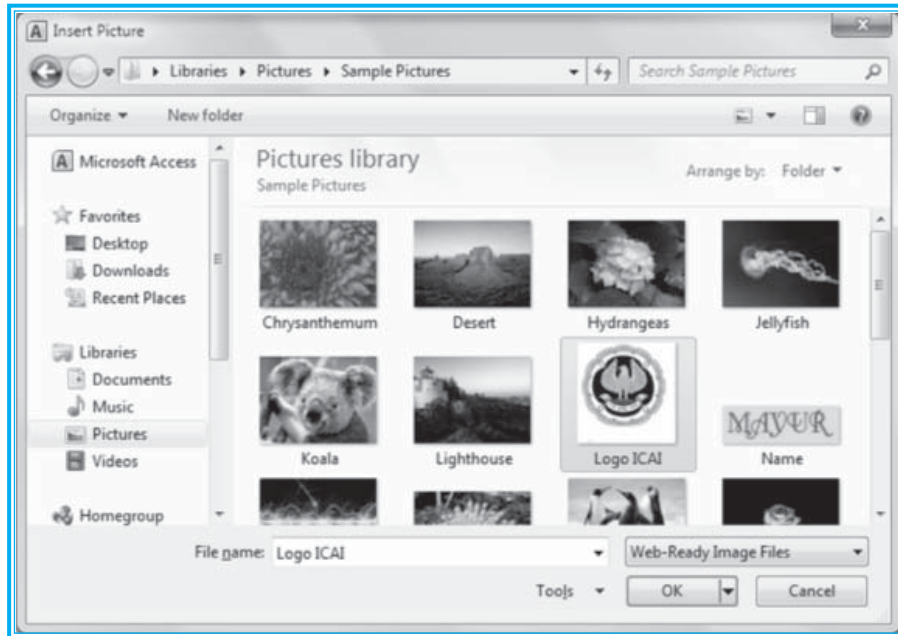


Fig. 2.14.1: Insert Logo (i)

2. After choosing an image or picture, the user will be required to click **Open**.
3. When the user will click **Open**, picture will be inserted in the **Form Header** section as shown in Fig. 2.14.2.

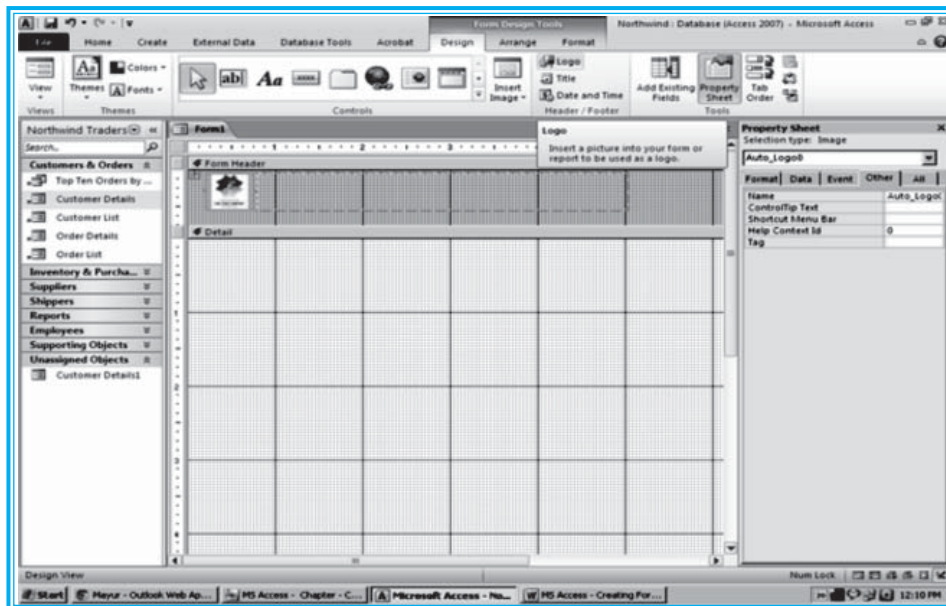


Fig. 2.14.2: Insert Logo (ii)



Note: The user may move the image or picture used as a logo from Form Header section to Detail section, as per requirement.

2.14.2 Insert Title

1. Click **Title** option under **Header / Footer** group. The user will be prompted to specify a title on the **Form Header** section as shown in Fig. 2.14.3.

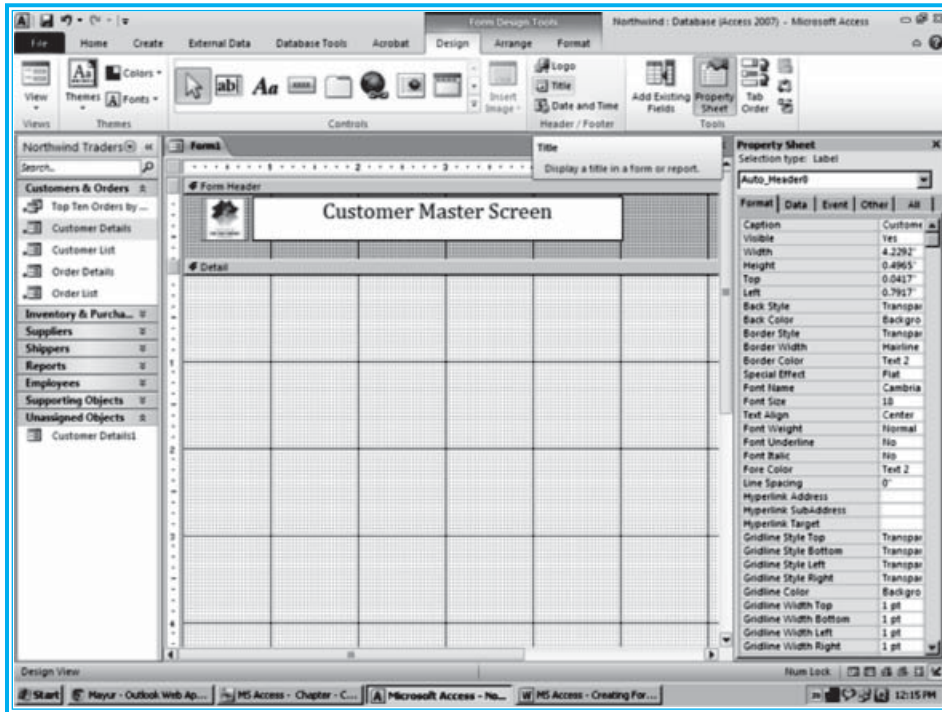


Fig. 2.14.3: Insert Title

2.14.3 Insert Date & Time

1. Click **Insert Date & Time** option under **Header / Footer** group. The user will be prompted to specify some additional information as shown in Fig. 2.14.4.



Fig. 2.14.4: Insert 'Date & Time'

2. After providing additional information as per requirement, the user will be shown a screen similar to Fig. 2.14.5 (A) in form design view. The output will be shown in form view as shown in Fig. 2.14.5 (B).

Fig. 2.14.5 (A): Form Header (i)

Fig. 2.14.5 (B): Form Header (ii)



2.14.4 Insert Label

1. Click **Label** option under **Controls** group.
2. Draw a **Label** in **Detail** section as shown in Fig. 2.14.6 (A) (**Form Design View**) and Fig. 2.14.6 (B) (**Form View**):

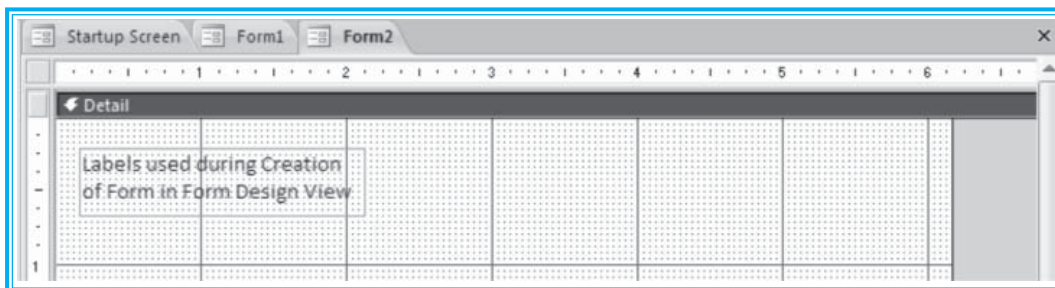


Fig. 2.14.6 (A): Insert Label (i)

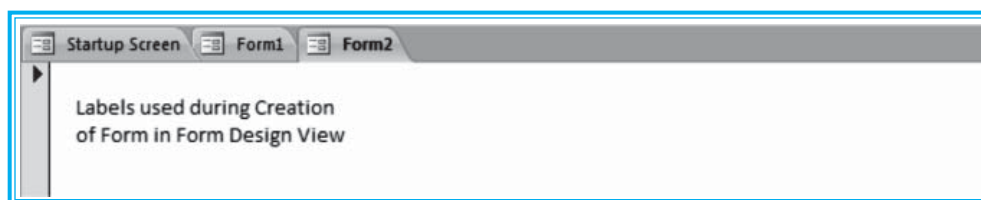


Fig. 2.14.6 (B): Insert Label (ii)

2.14.5 Insert Text Box

To add unbound Text Box

1. Click **Text Box** option under **Controls** group.
2. Draw a text box in **Detail** section as shown in Fig. 2.14.7.

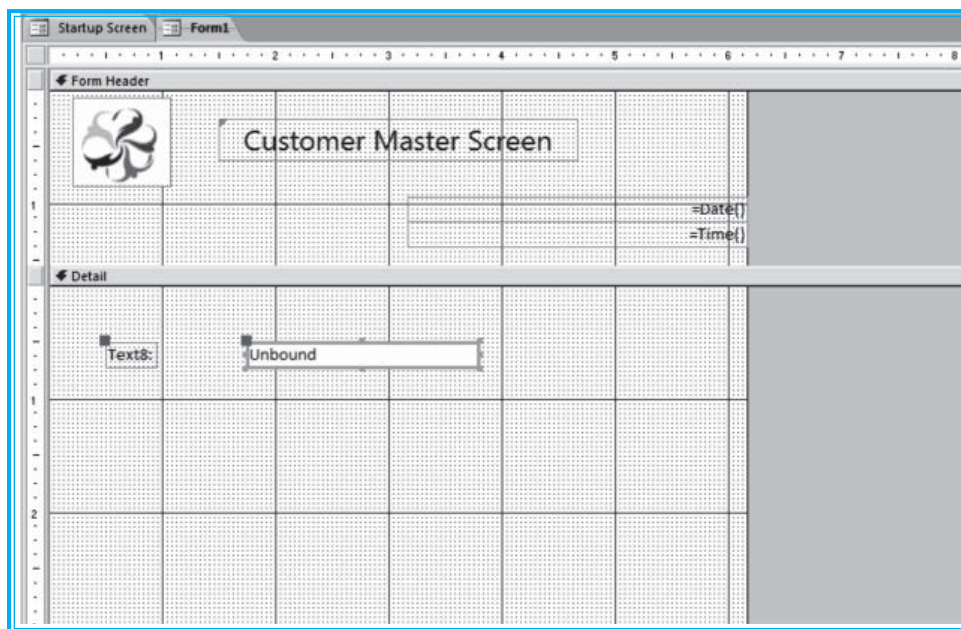


Fig. 2.14.7: Insert Text Box (i)



To add bound Text Box

1. Click **Add Existing Fields** option under **Tools** group as shown in Fig. 2.14.8.



Fig. 2.14.8: Add Existing Fields

2. After clicking **Add Existing Fields** option, a screen similar to Fig. 2.14.9 will be shown to the user:-

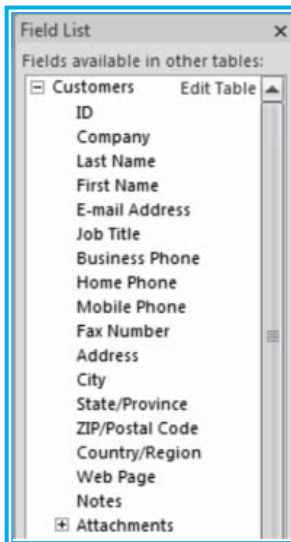


Fig. 2.14.9: Field List Dialog Box

3. The user may drag and drop the required fields in the detail section from the Field List dialog box showing Fig. 2.14.9. The user will be shown in Fig. 2.14.10 (A) (Form Design View) and Fig. 2.14.10 (B) (Form View):

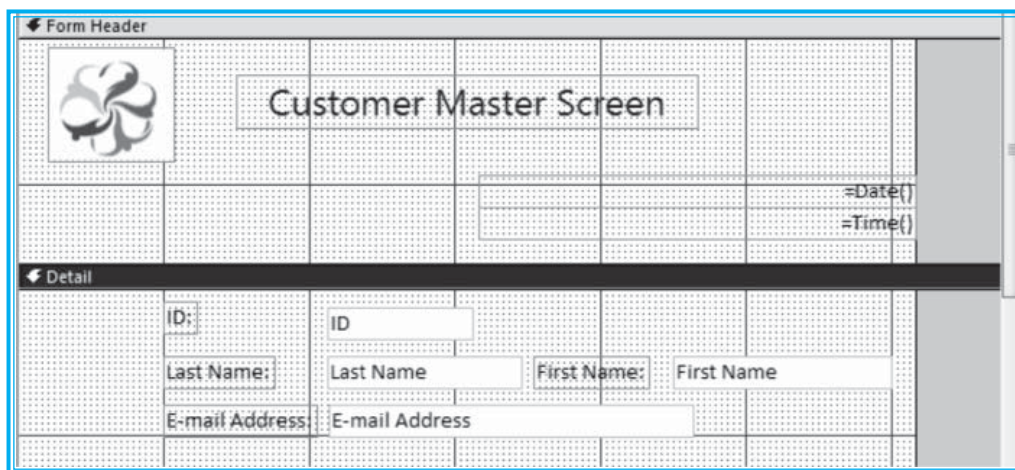


Fig. 2.14.10 (A): Form Design View of Text Box

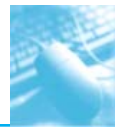


Fig. 2.14.10 (B): Form view of Text Box

2.14.6 Insert Button

Insert Unbound Button

1. Click **Button** option under **Controls** group.
2. Draw a button in the **Details** section as shown in Fig. 2.14.11.

Fig. 2.14.11: Insert Button

3. Provide a meaningful name to the newly created Button as shown in Fig. 2.14.12 (A) (Form Design View) and Fig. 2.14.12 (B) (Form View):

Fig. 2.14.12 (A): Naming a Button

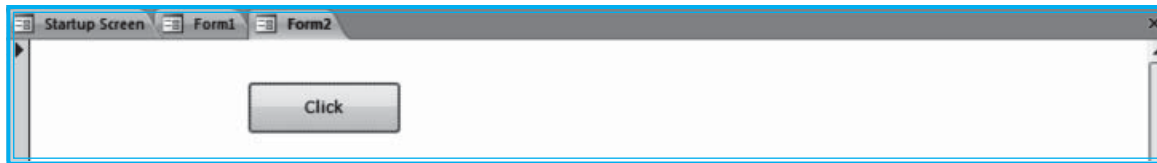


Fig. 2.14.12 (B): Form view of Button

Events Associated with a Button

The user may attach an event procedure on an activity associated with the button as follow:-

- On Click
- On Got Focus
- On Lost Focus
- On Dbl (Double)Click
- On Mouse Down
- On Mouse Up
- On Mouse Move
- On Key Down
- On Key Up
- On Key Press
- On Enter
- On Exit

As stated in the above sections, events are procedures which get triggered on an associated activity of a control.

Insert bound Button

1. Before clicking the **Button** option under **Controls** groups, the user must ensure that the **Use Control Wizard** option under **Controls** group is activated. Use Control Wizard is a toggle button, once clicking it will activate it and another time clicking will deactivate it.
2. Click the **Button** option under **Controls** groups.
3. Draw the **Button** in the **Details** section of the form. The user will see a screen similar to Fig. 2.14.13.

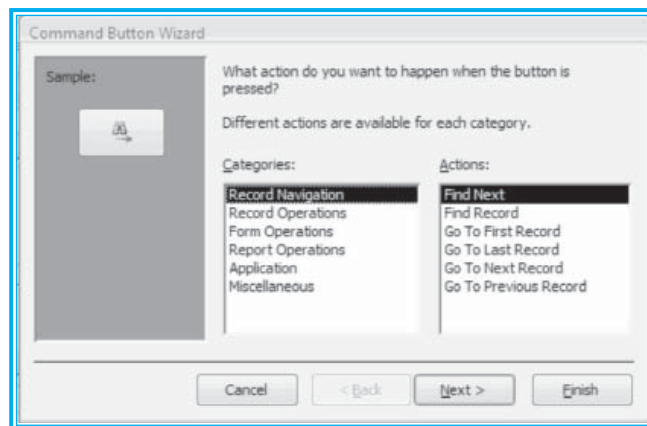


Fig. 2.14.13: Command Button Wizard (i)



4. The user needs to specify the Category and associated Action related with the newly created Button. Following table lists all the categories and associated actions to be found under **Command Button Wizard**:-

Categories	Actions
Record Navigation	Find Next Find Record Go to First Record Go to Last Record Go to Next Record Go to Previous Record
Record Operations	Add New Record Delete Record Duplicate Record Print Record Save Record Undo Record
Form Operations	Apply Form Filter Close Form Open Form Print a Form Print Current Form Refresh Form Data
Report Operations	Mail Report Open Report Preview Report Print Report Send Report to File
Application	Quit Application
Miscellaneous	Auto Dialer Print Table Run Macro Run Query

Table 2.14.1: Command Button Categories & Actions

5. After specifying the category and associated action related with the newly created Button, the user will be shown the following Fig. 2.14.14.

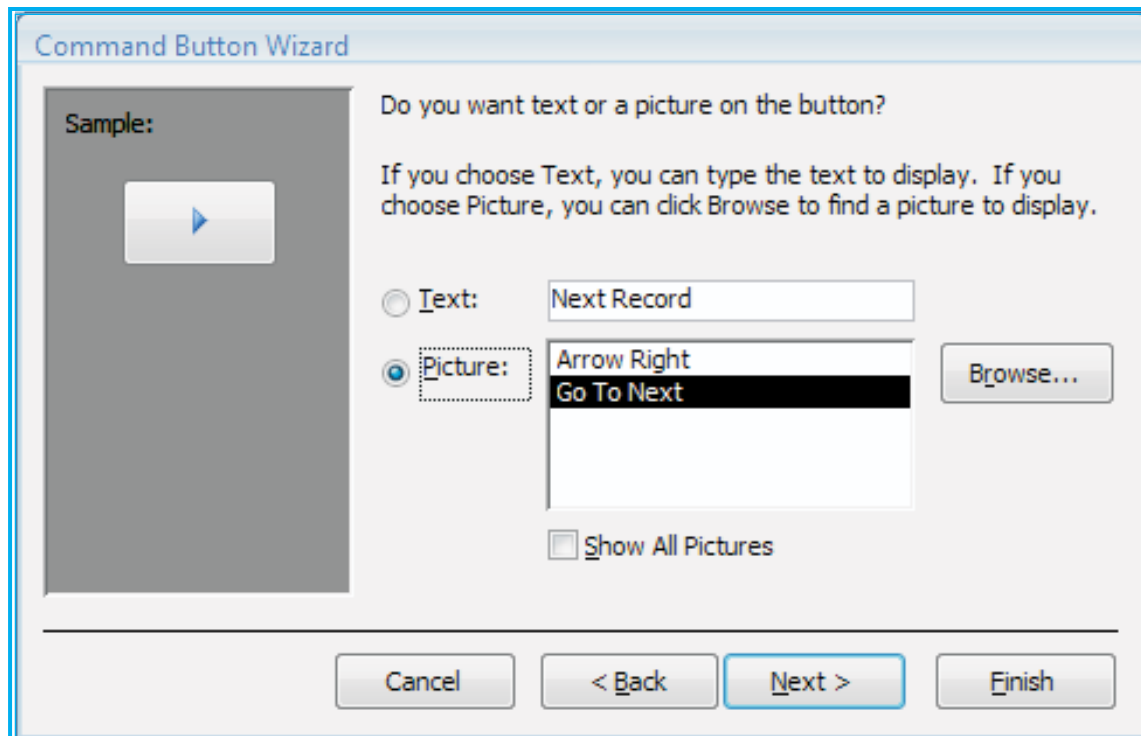


Fig. 2.14.14: Command Button Wizard (ii)

6. The user may specify to use text or a picture on the button as per the requirements. Further, the user will be shown the following Fig. 2.14.14.

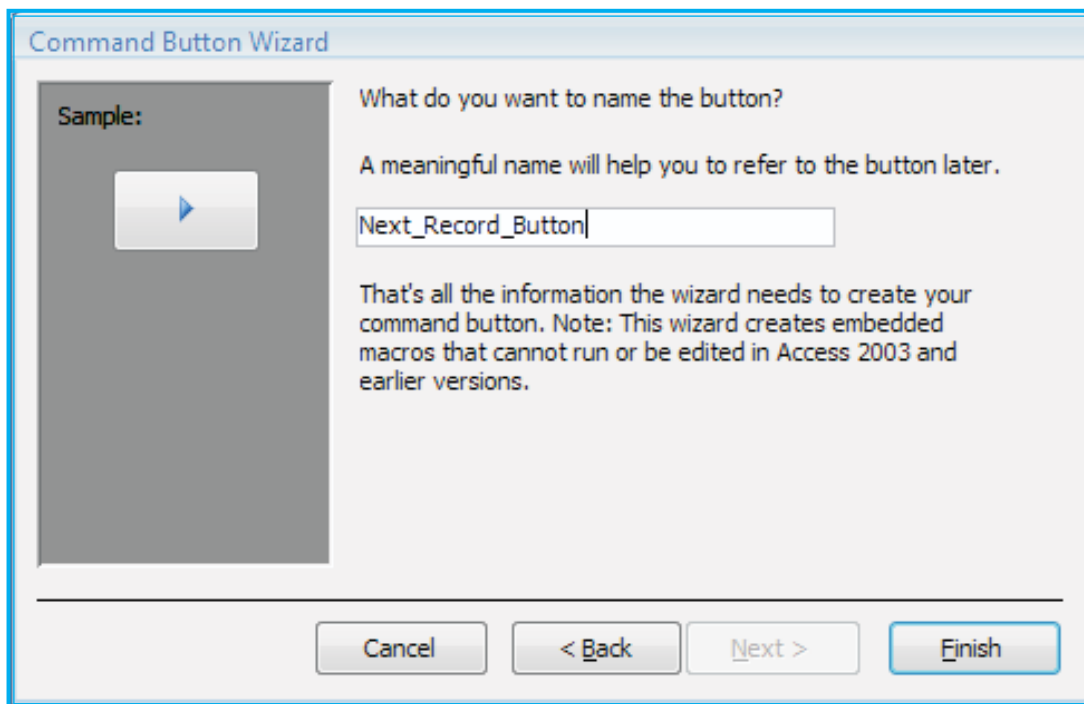
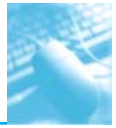


Fig. 2.14.15: Command Button Wizard (iii)



7. The user needs to give a meaningful name to the Button which could be user in future to refer to the button control as shown in Fig. 2.14.15.
8. Click **Finish** to close the wizard.

2.14.7 Insert Combo Box

1. Click the **Combo Box** option under **Controls** group.
2. Draw the **Combo Box** in the **Details** section of the form. The user will see a screen similar to Fig. 2.14.16.

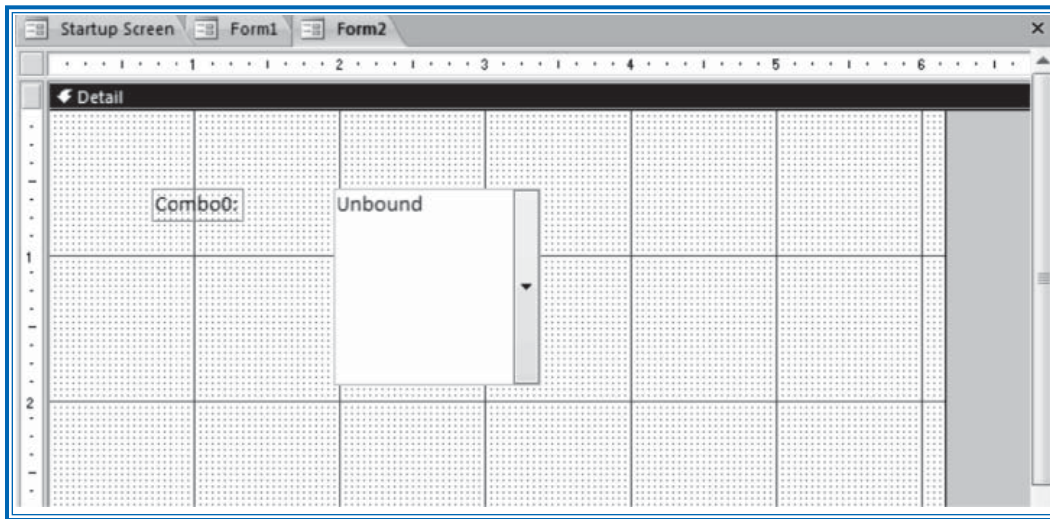


Fig. 2.14.16: Insert Combo Box (i)

3. Select the concerned **Combo Box** control and click the **Property Sheet** option under **Tools** group.
4. On the **Property Sheet**, Click the **Data Tab**.
5. As shown in Fig. 2.14.17, under Row Source Type of Property Sheet, select Value List to supply user defined values to the Combo Box control. Alternatively, the user may also select Table / Query or Field List to associate the Combo Box control with desired table and field.

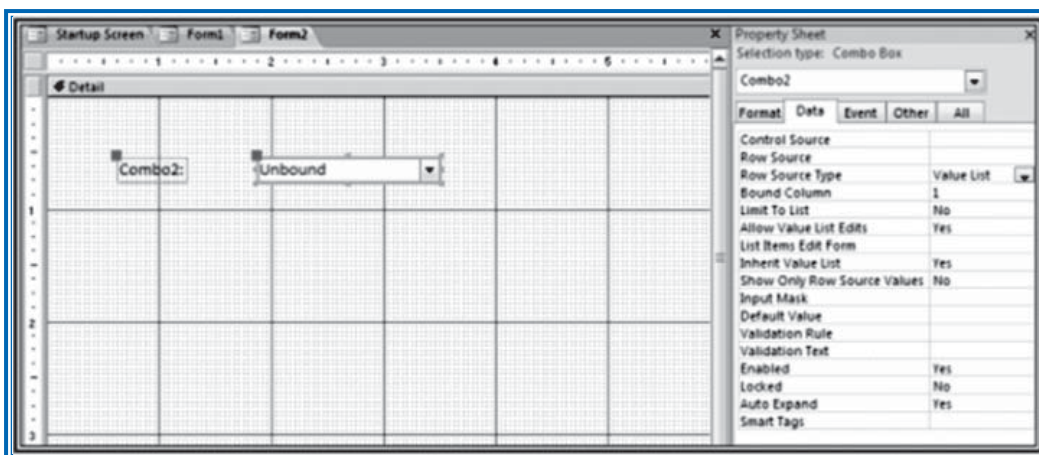


Fig. 2.14.17: Insert Combo Box (ii)



6. Selecting Row Source will show the user a screen similar to Fig. 2.14.18.

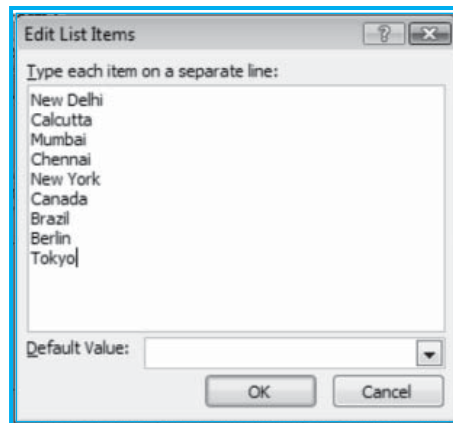


Fig. 2.14.18: Insert Combo Box (iii)

7. The user needs to type in the items that the user wishes to include in the **Combo Box** control as shown in Fig 2.14.18.
8. Click **OK** and see the output in **Form View** as shown in Fig. 2.14.19:

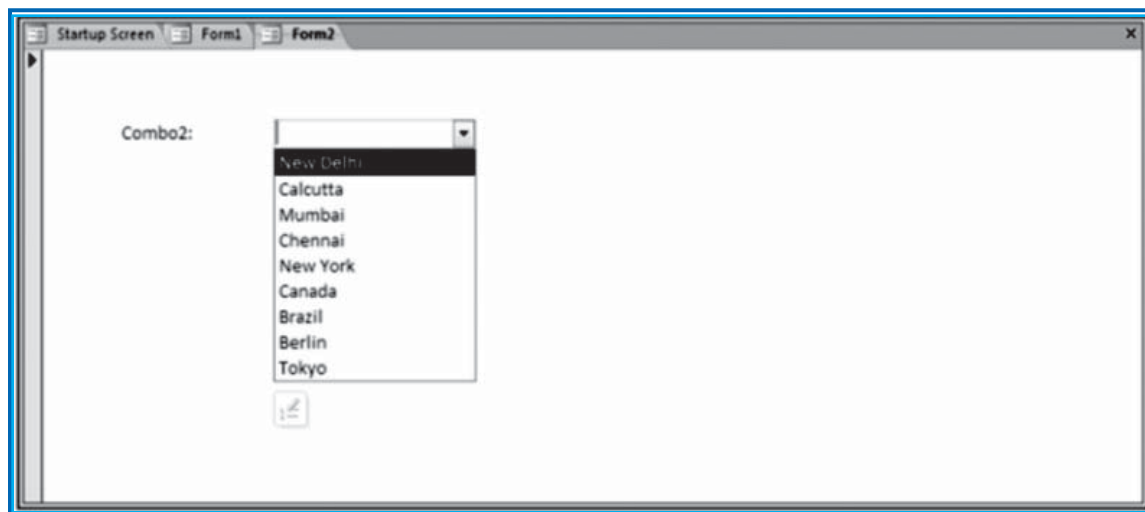


Fig. 2.14.19: Insert Combo Box (iv)

2.14.8 Insert List Box

1. Click the **List Box** option under **Controls** group.
2. Draw the **List Box** in the **Details** section of the form. The user will see a screen similar to Fig. 2.14.20.

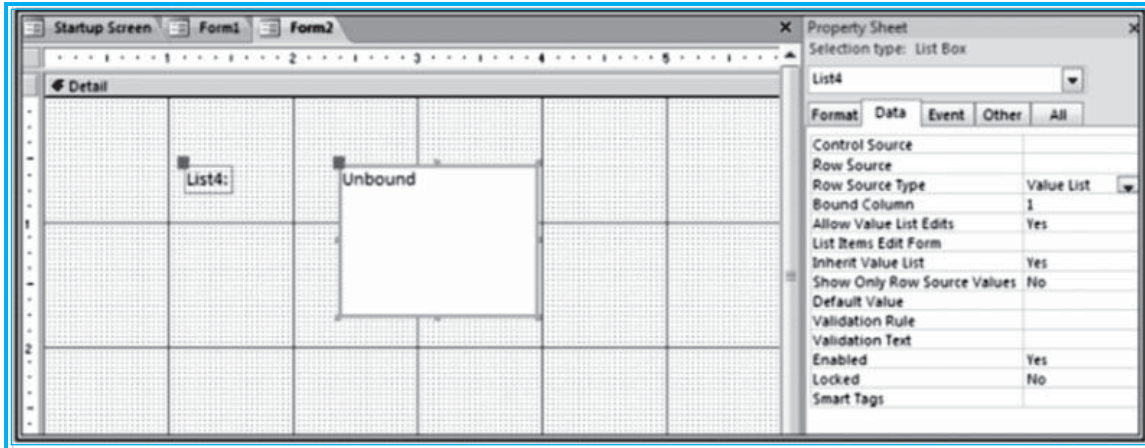


Fig. 2.14.20: Insert List Box (i)

3. Select the concerned **List Box** control and click the **Property Sheet** option under **Tools** group.
4. On the **Property Sheet**, Click the **Data** Tab.
5. As shown in Fig. 2.14.20, under Row Source Type of Property Sheet, select Value List to supply user defined values to the List Box control. Alternatively, the user may also select Table / Query or Field List to associate the List Box control with desired table and field.
6. Selecting Row Source will show the user a screen similar to Fig. 2.14.21:

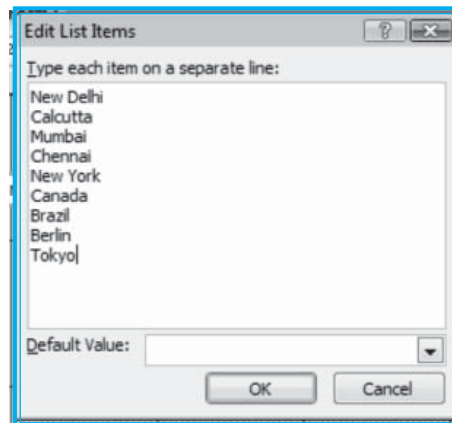


Fig. 2.14.21: Insert List Box (ii)

7. The user needs to type in the items that the user wishes to include in the List Box control as shown in Fig. 2.14.21.
8. Click **OK** and see the output in **Form** View as shown in Fig. 2.14.22:

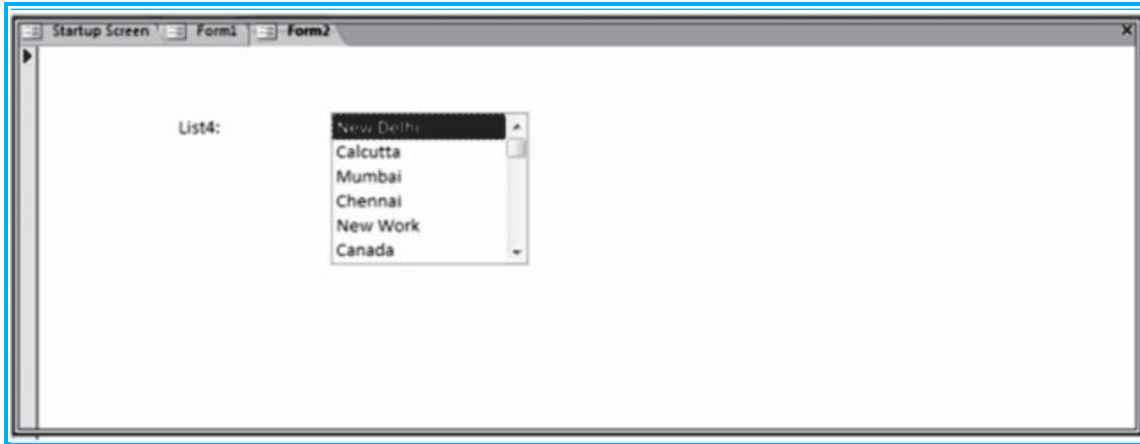


Fig. 2.14.22: Insert List Box (iii)

2.14.9 Insert Check Box, Option Button and Toggle Button

Bound Controls:

1. Create a table name Membership_Statistics with the structure as shown in Fig. 2.14.23:

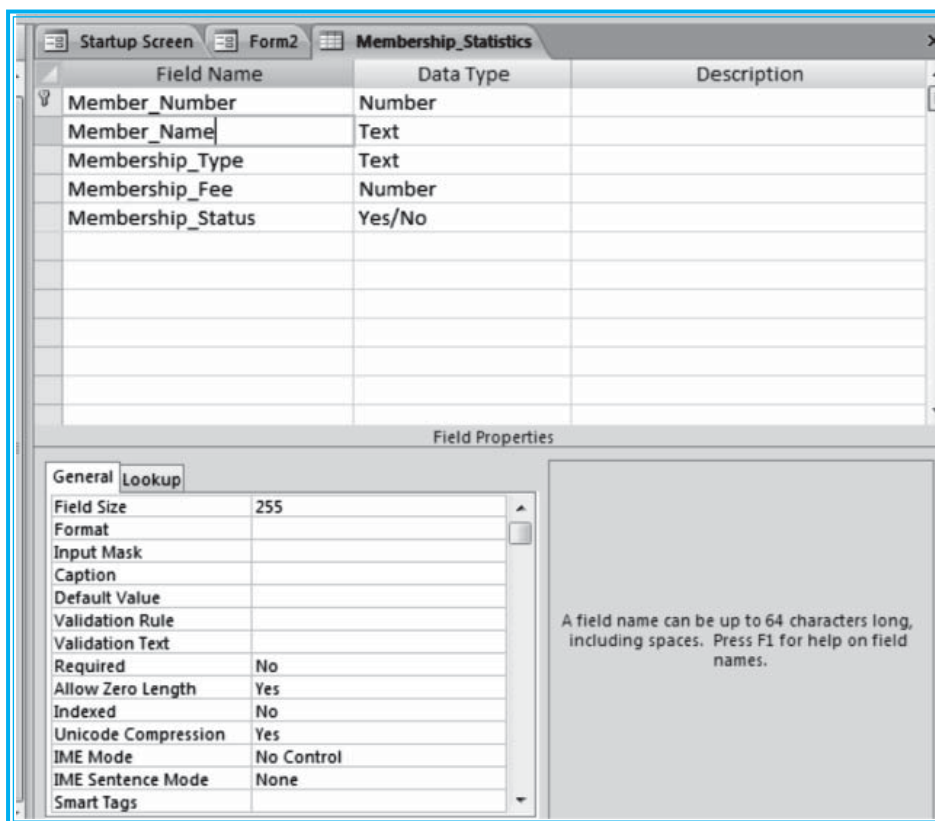
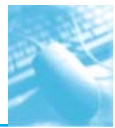


Fig. 2.14.23: Insert Bound Controls (i)



2. Add the following data as shown in Fig. 2.14.24:

Member_Number	Member_Name	Membership_Type	Membership_Fee	Membership_Status	Click to Add
101	Mayur Kumar	Life Member	7000	<input checked="" type="checkbox"/>	
102	Vikas Shinde	Annual	2500	<input checked="" type="checkbox"/>	
103	Anoj Das	Life Member	7000	<input type="checkbox"/>	
104	Ankur Khairwal	Semi Annual	1600	<input checked="" type="checkbox"/>	

Fig. 2.14.24: Insert Bound Controls (ii)

3. Save and close the table.
4. Open a form in **design view** and click the field list pane as shown in Fig. 2.14.25:

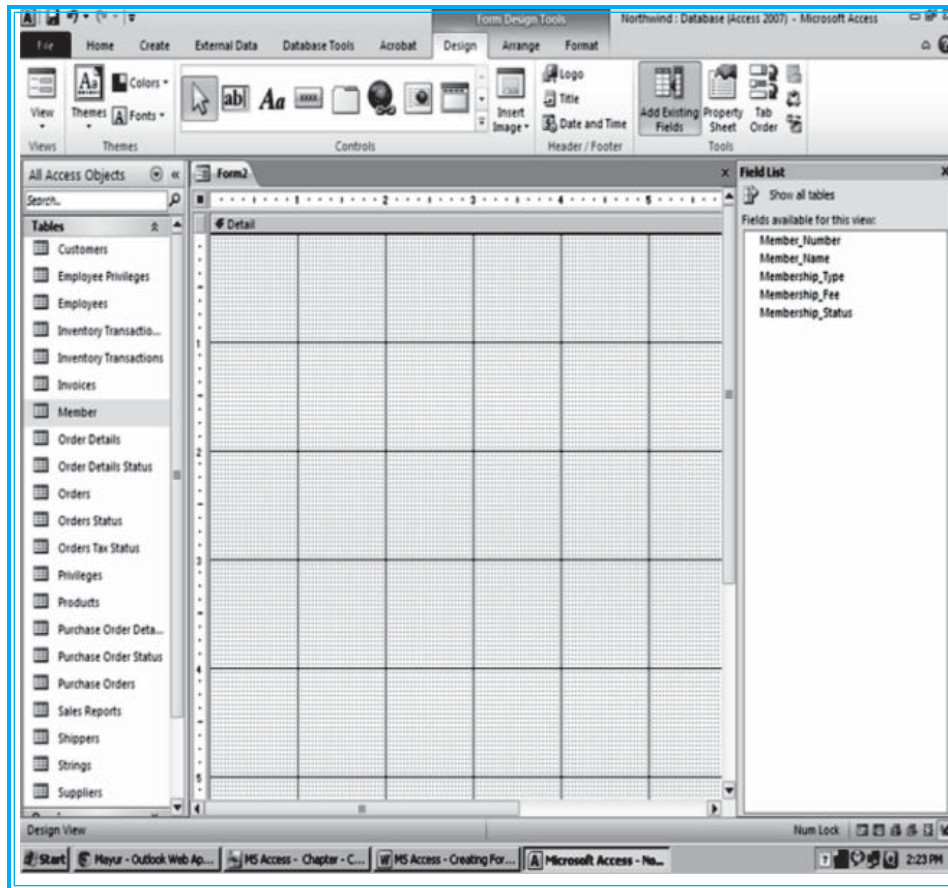


Fig. 2.14.25: Insert Bound Controls (iii)

5. Add the fields from field list pane one by one. The user will see a screen similar to Fig. 2.14.26:

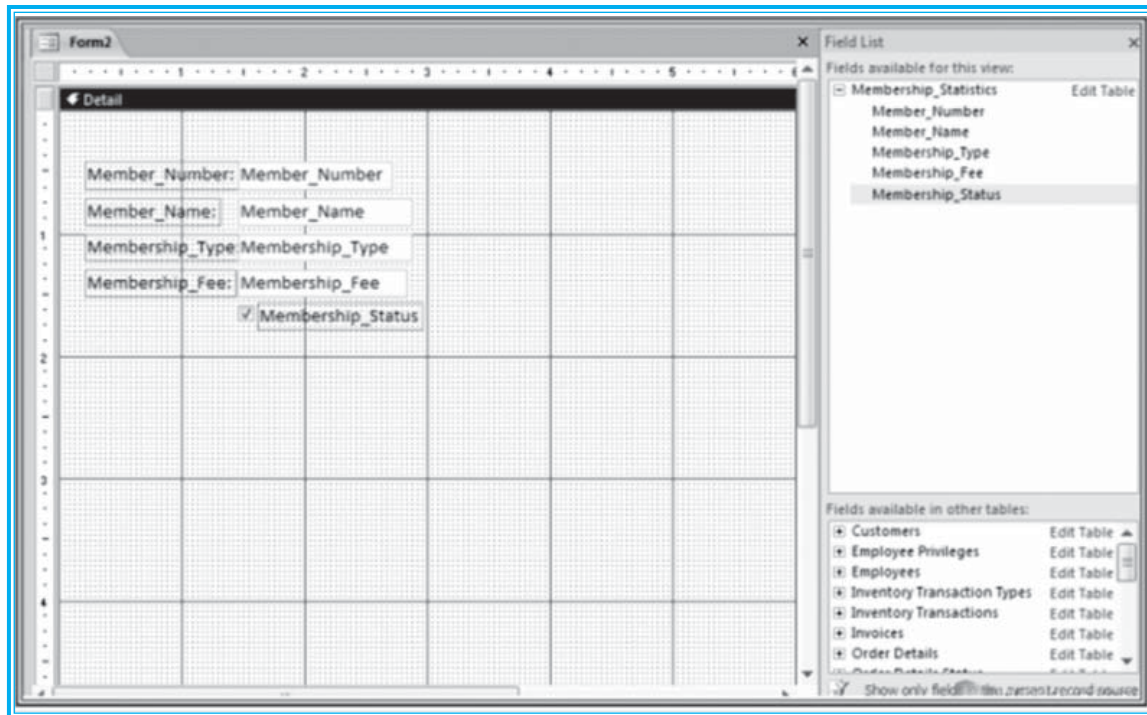


Fig. 2.14.26: Insert Bound Controls (iv)

Note: Depending on the setting of the field's Display Control property, a check box, a text box or a combo box is created and bound to the field. When the user creates a new Yes/No field in a table, the default setting of the field's Display Control property is Check Box.

The form when shown in form view will look similar to Fig. 2.14.27:

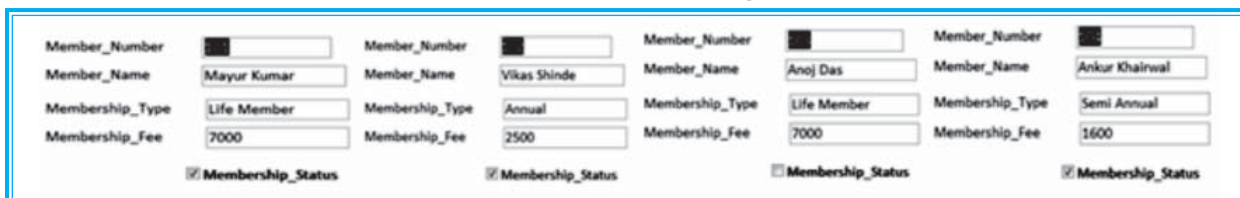
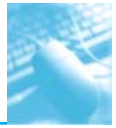


Fig. 2.14.27: Form in Form View

2.14.10 Insert Option Group

An option group displays a limited set of alternatives and only one option in an option group can be selected at a time. An option group consists of a group frame and a set of check boxes, toggle buttons or option buttons. If an option group is bound to a field, only the group frame itself is bound to the field, not the controls contained inside the frame. Instead of setting the Control Source property for each control in the option group, the user sets the Option Value property of each control to a number that is meaningful for the field to which the group frame is bound. When the user selects an option in an option group, Access sets the value of the field to which the option group is bound to the value of the selected option's Option Value property. The value of an option group can only be a number, not text.



1. Create a table Customer_Status with the structure as shown in Fig. 2.14.28:

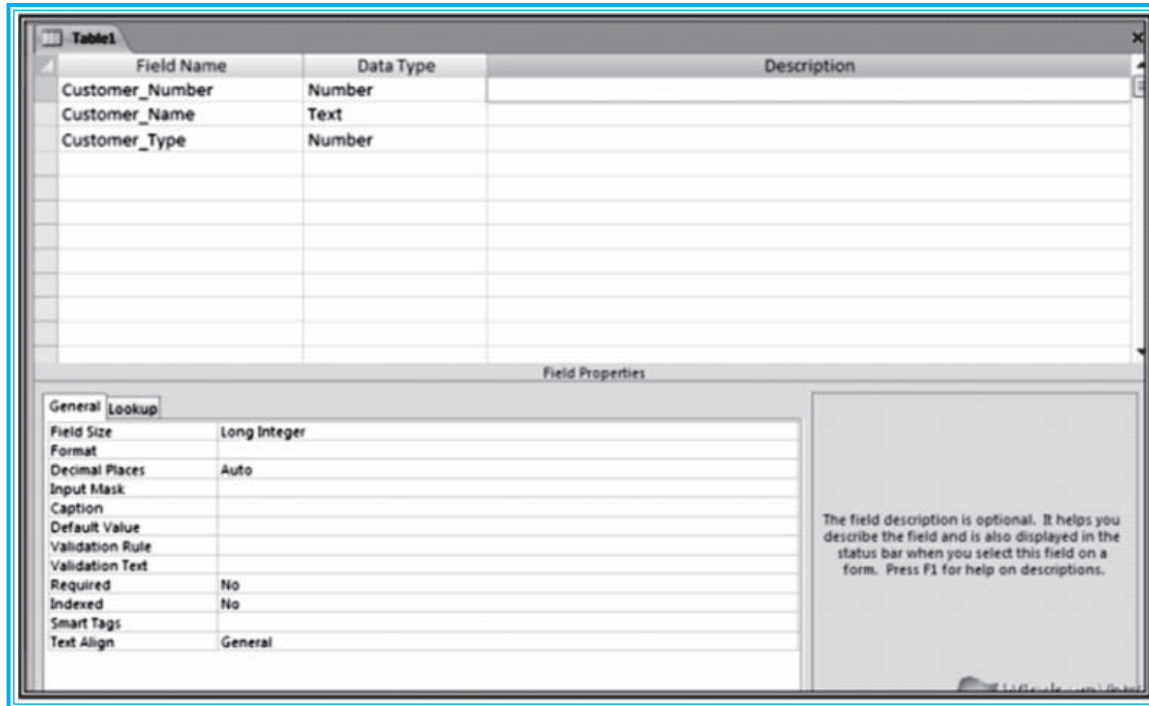


Fig. 2.14.28: Insert Option Group (i)

2. Add the following data as shown in Fig. 2.14.29.

Customer_Number	Customer_Name	Customer_Type	Click to Add
101	Mayur Kumar	1	
102	Vikas Shinde	2	
103	Anoj Das	1	
104	Ankur Khairwal	3	
*			

Fig. 2.14.29: Insert Option Group (ii)

3. Save and close the table.
4. Open a form in **design** view and click the field list pane and select Customer_Status Table.
5. Add Customer_Number and Customer_Name field from the field list pane.
6. Now, select option group from **Design** tab. Add three **Option Button** and provide name to them as shown in Fig. 2.14.30:



Fig. 2.14.30: Insert Option Group (iii)

7. Select the Option Group control and in its property sheet click Data tab. Specify the control source as Customer_Type as shown in Fig. 2.14.31:

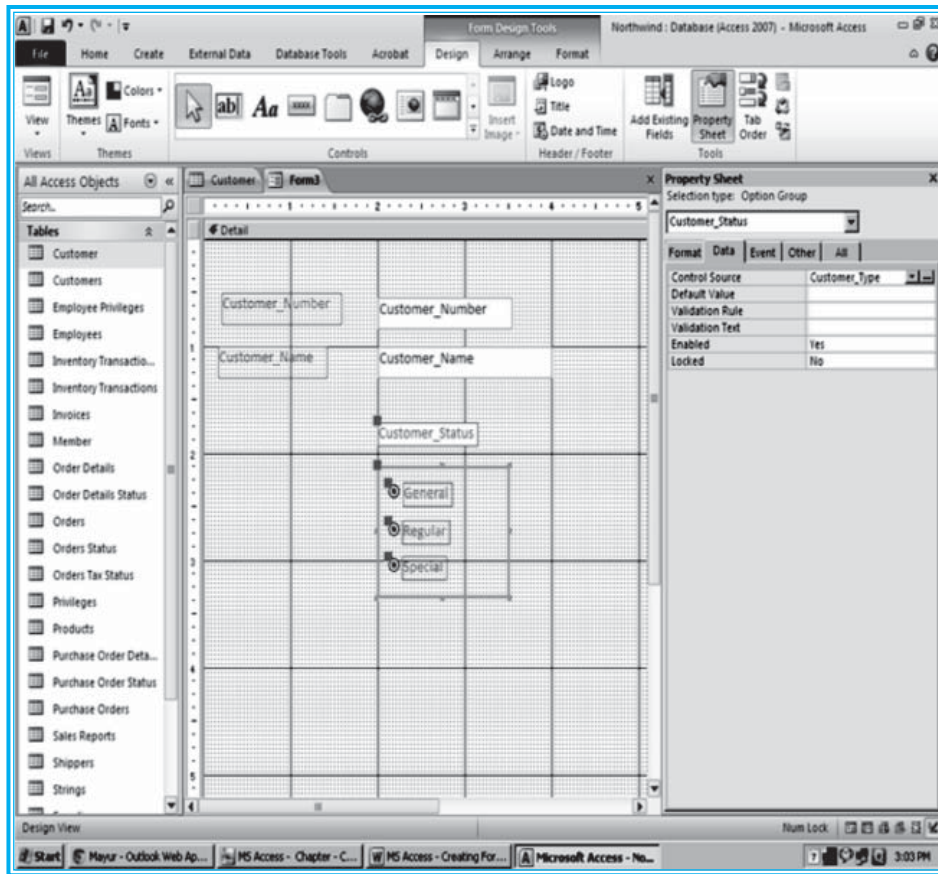


Fig. 2.14.31: Insert Option Group (iv)

8. Now, click individual **Option** Buttons and specify their option value as follows:-

Option Button	Option Value
General	1
Regular	2
Special	3

9. Select the **Form View** to see a form similar to Fig. 2.14.32.



Fig. 2.14.32: Insert Option Group (v)

Note: As the user will go to the next record, the value of Customer Status will change as per the data entered in the table and the same will be reflected on the form, too.

2.14.11 Insert Image

1. Select the image from Controls under **Design** tab.
2. Click on insert image, then browse it.
3. Specify the image that user wants to use in the form.
4. Click the form view to see the output as shown in Fig. 2.14.33.

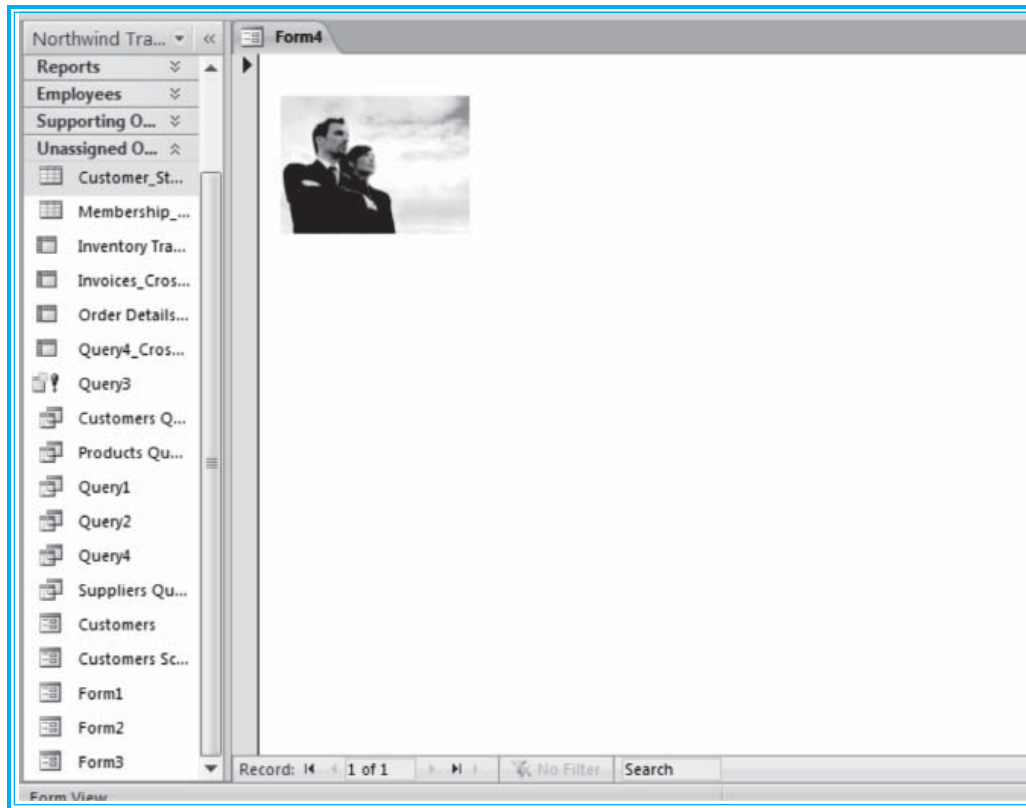


Fig. 2.14.33: Insert Image

2.14.12 Insert Hyperlink

1. Open a form in **design view**.
2. Select the hyperlink from controls under **Design** tab.
3. Specify the text the user wants to have for the hyperlink message along with the file name to open as shown in Fig. 2.14.34:

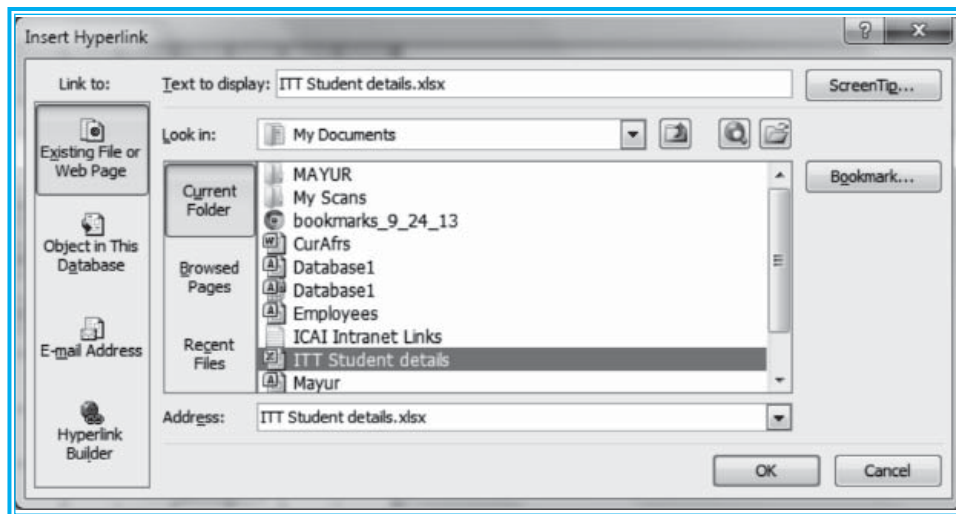


Fig. 2.14.34: Insert Hyperlink (i)



4. Click the **Form View** to see the output as shown in Fig. 2.14.35.

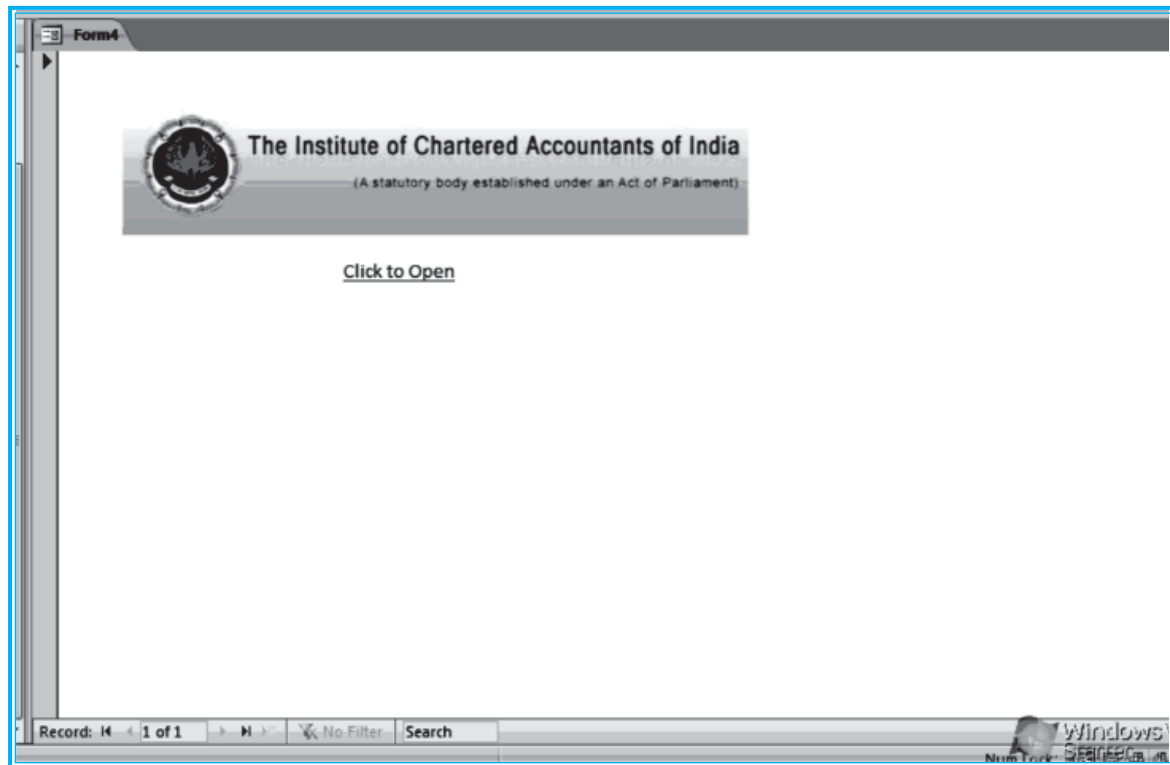


Fig. 2.14.35: Insert Hyperlink (ii)

LEARNING OBJECTIVES

In this chapter, we will learn:

- Types of queries.
- General steps in creating a query.
- Query using query wizard & design view.
- Sorting a query.
- Running & saving query.
- Create parameter query.
- Creating query using multiple tables.
- Reports & its importance.
- Types of reports.
- Steps in creating a report.
- Creating a report with the report wizard & design view.
- Saving & printing report.

3.1 INTRODUCTION

At times the data stored in a single table is so large in volume that it becomes difficult for a user to find specific information. Further, it becomes more complex when two or more tables get involved in searching specific information. In such situation, it is a good idea to use the concept of queries. Query is a mechanism for retrieving information from a database and consists of questions presented to the database in a predefined format. A user may use a query to answer a simple question, to perform calculations, to combine data from different tables or even to add, change or delete data from a table(s). A user may use queries to filter the desired data, to perform calculations with the data and to summarize the data present in table(s). Queries can also be used to supply data for a form or report. Additionally, queries can also be used to automate many data management tasks and to review changes in the data before actually committing the changes the user wants to make.



3.2 DEFINITION

Query can be defined as an operation that extracts record(s) from a database based on a given condition. A query consists of search criteria expressed in a database language called SQL. For example, the query can specify that only certain columns or only certain records be included in a result.

3.3 QUERY AND ITS TYPES

The user can create a query in MS-Access in either of the following two ways:-

- Choose the **Query Wizard** provided by **Microsoft Access** for the ease of user to build a query.
- Create own queries from scratch using query design.

Now, before a user start to build query, he / she should know the general types of queries that MS-Access offers as follows:-

3.3.1 Select Query

The **select query** is the simplest and most common type of query available in MS-Access. Such queries may be used to select and display data from either one or more table(s) depending upon the requirement. The user-determined criteria convey to the database about the selection of records from table(s). After the select query is called, it creates a "virtual" table where the data can be changed one record at a time.

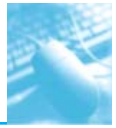
3.3.2 Crosstab Query

Crosstab queries are useful for summarizing information, calculating statistics, spotting bad data and looking for trends. The results of a crosstab query are read-only - data cannot be added, edited or deleted in a crosstab result. An aggregate function, such as sum or count, is used to help summarize the data. At times, the user wants to create a query for some specific purpose such as to group data into categories and display values in a spreadsheet like format with summary totals. Such queries can also be used to compare values and observe the trends in the data and to obtain summary data such as monthly, quarterly, semiannually or annually sales figures.

3.3.3 Action Query

Action query is used to create new table(s), delete rows from existing table(s) and update records or creating entirely new ones in a table(s). When the action query is called, the database undergoes a specific action depending on what was specified in the query itself. Action queries are very popular in data management because they allow for many records to be changed at one time unlike only single record in a select query. There may be four sub-types of action queries which are as follows:-

- **Append Query:** The append query takes the set results of a query and "appends" them to an existing table. In other words, we can say that, an append query copies specified or all the records from one table to the bottom of another table. This query is useful when two different tables with similar structures are used. If the source table has more fields than the target tables, extra fields are ignored. Moreover, if the source table has fewer fields than the target table, fields with matching names are updated and any additional fields are left blank.
- **Delete Query:** The delete query deletes all records in an underlying table from the set results of a query. In simpler terms, we can say that, the delete query is used to delete a group of records that meets specific search criteria. The process for setting up a delete query depends on whether the user wants to delete records from single table or from multiple tables that have an established and valid relationship among them.



- **Make Table Query:** The make table query creates a table based on the set results of a query. In simpler terms, we can say that, the make table query is used to create a new table and populate it with data from one or more existing tables. By creating a make table query, the user actually created a backup copy of a table. Moreover, it may be useful, when the user wants to save delete records in a separate history table for future use. It may also be useful in situations, where a subset of data needs to be generated as a report and may be created as a table for exporting to other applications.
- **Update Query:** The update query allows for one or more field in table(s) to be updated. In other words, we can say that, the update query allows the user to make global changes to data in one or more tables.

3.3.4 Parameter Query

A **parameter query** is used to pass a parameter to a different query such as an action or a select query. The parameter can either be a value or a condition. This parameter conveys to the other query what is supposed to be done. It allows for a dialog box where the end user can enter whatever parameter value they wish each time the query is run. The parameter query can be seen as a modified select query.

3.3.5 Aggregate Query

Aggregate query is a special type of query which works on other queries such as selection, action or parameter but instead of passing a parameter to another query it totals up the items by selected groups. It essentially creates a summation of any selected attribute in a table. The SQL aggregate functions available to Microsoft Access are:

- Sum
- Avg
- Min
- Max
- First
- Last
- Group By
- Count
- StDev
- Var
- Expression
- Where

3.4 GENERAL STEPS TO CREATE A QUERY

The general steps to create a query in MS-Access are as follows:-

- Choose the tables or queries as sources of data.
- Specify the fields that the user wishes to include from the data sources.
- Optionally, specify criteria to limit the records that the query returns.
- Run the query to see the results.



Query wizard may be used by a user to automatically create a simple query. Although, the query wizard provides less control in handling a query but surely, query once created using query wizard can always be modified using design view.

Source:

Table : Customers

1. On the **Create** tab, in the **Queries** group, click **Query Wizard**, as shown in Fig. 3.5.1



- Fig. 3.5.2: New Query Dialog Box*



We will see a screen similar to Fig. 3.5.3 as shown below:-

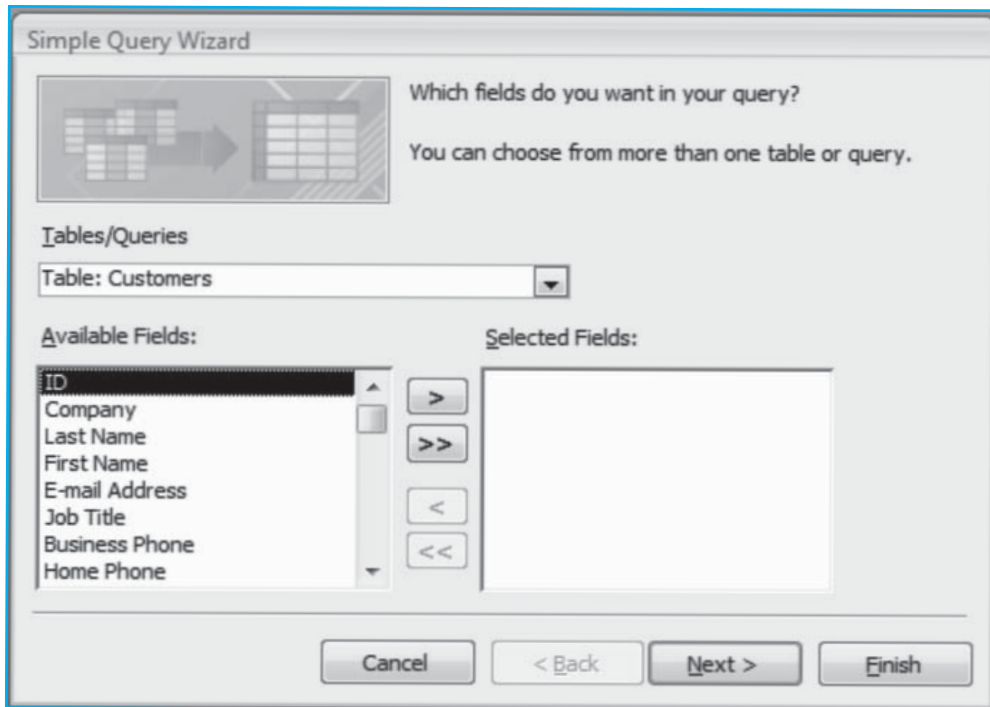


Fig. 3.5.3: Simple Query Wizard (i)

3. After selecting the desired field(s) from the table, the result is as shown in Fig. 3.5.4.

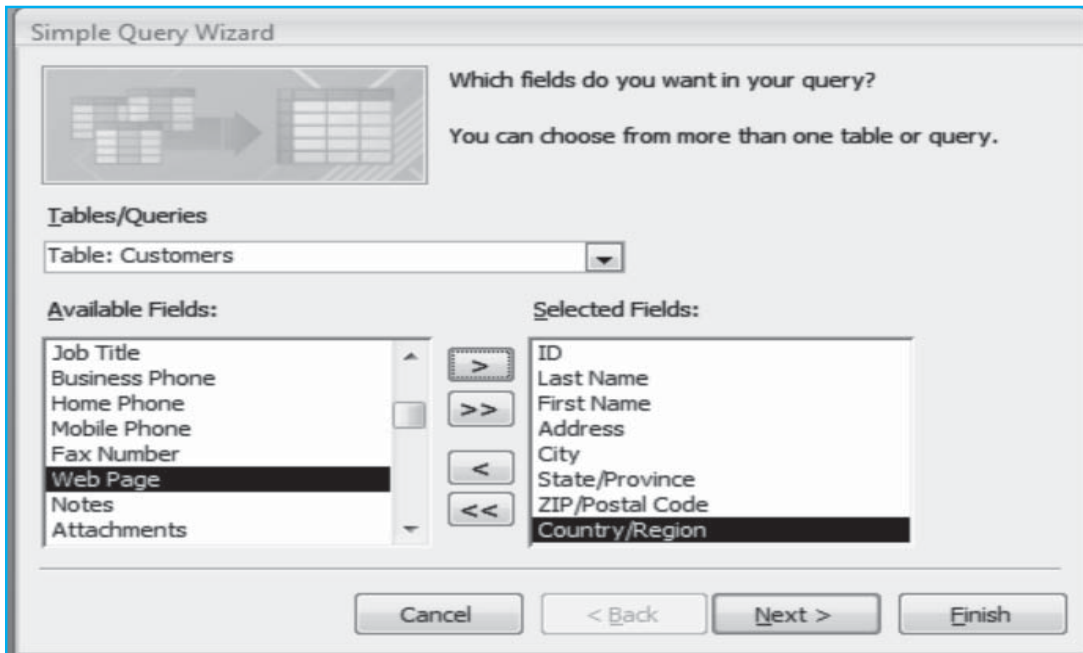


Fig. 3.5.4: Simple Query Wizard (ii)

The user can add up to 255 fields from as many as 32 tables or queries.



For each field, the user depending upon the requirement may perform the following two steps:-

- Under **Tables/Queries**, click the table or query that contains the field.
 - Under **Available Fields**, double-click the field to add it to the **Selected Fields** list.
4. When the user has added all the desired fields, he / she may click **Next** and may see Fig. 3.5.5 as follows:

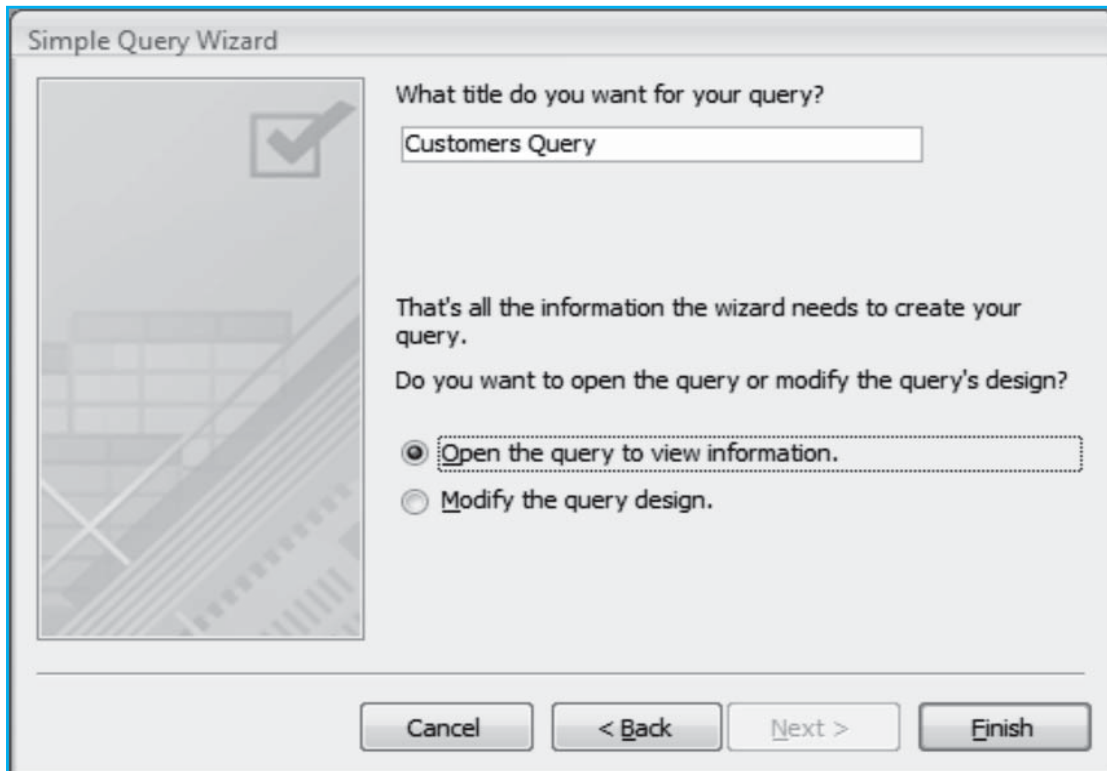


Fig. 3.5.5: Simple Query Wizard (iii)

Although, the query wizard suggests a query title for the query being prepared still the user may specify his / her own query title. For this, the user has to type the desired query title inside the text box provided for the same as shown in Fig. 3.5.5. Moreover, the query wizard provides the user with a choice to either open the query to view information or to modify the query in query design view.



5. The user may click **Finish** to complete the query wizard and may see the results of the query as shown in Fig. 3.5.6 as follows:-

ID	Last Name	First Name	Address	City	State/Provir	ZIP/Postal C	Country/Reg
1	Bedecs	Anna	123 1st Street	Seattle	WA	99999	USA
2	Gratacos Solso	Antonio	123 2nd Street	Boston	MA	99999	USA
3	Axen	Thomas	123 3rd Street	Los Angeles	CA	99999	USA
4	Lee	Christina	123 4th Street	New York	NY	99999	USA
5	O'Donnell	Martin	123 5th Street	Minneapolis	MN	99999	USA
6	Pérez-Olaeta	Francisco	123 6th Street	Milwaukee	WI	99999	USA
7	Xie	Ming-Yang	123 7th Street	Boise	ID	99999	USA
8	Andersen	Elizabeth	123 8th Street	Portland	OR	99999	USA
9	Mortensen	Sven	123 9th Street	Salt Lake City	UT	99999	USA
10	Wacker	Roland	123 10th Street	Chicago	IL	99999	USA
11	Krschne	Peter	123 11th Street	Miami	FL	99999	USA
12	Edwards	John	123 12th Street	Las Vegas	NV	99999	USA
13	Ludick	Andre	456 13th Street	Memphis	TN	99999	USA
14	Grilo	Carlos	456 14th Street	Denver	CO	99999	USA
15	Kupkova	Helena	456 15th Street	Honolulu	HI	99999	USA
16	Goldschmidt	Daniel	456 16th Street	San Francisco	CA	99999	USA
17	Bagel	Jean Philippe	456 17th Street	Seattle	WA	99999	USA
18	Autier Miconi	Catherine	456 18th Street	Boston	MA	99999	USA
19	Eggerer	Alexander	789 19th Street	Los Angeles	CA	99999	USA
20	Li	George	789 20th Street	New York	NY	99999	USA
21	Tham	Bernard	789 21th Street	Minneapolis	MN	99999	USA
22	Ramos	Luciana	789 22th Street	Milwaukee	WI	99999	USA
23	Entin	Michael	789 23th Street	Portland	OR	99999	USA
24	Hasselberg	Jonas	789 24th Street	Salt Lake City	UT	99999	USA
25	Rodman	John	789 25th Street	Chicago	IL	99999	USA

Fig. 3.5.6: Resultant Sheet

Now, if the user has included any numeric field (number field) from table(s), the wizard asks the user, whether, he / she may want the query to return details or summary data. The following step by step discussion will explain few more options which the user may use for a specific need:-

Source:

Database : Northwind 2010

Table : Products

- Step 1 and Step 2 given above for Customers table step by step instruction to be repeated. After selecting the desired field(s) from the table, the result will be shown as Fig 3.5.7.

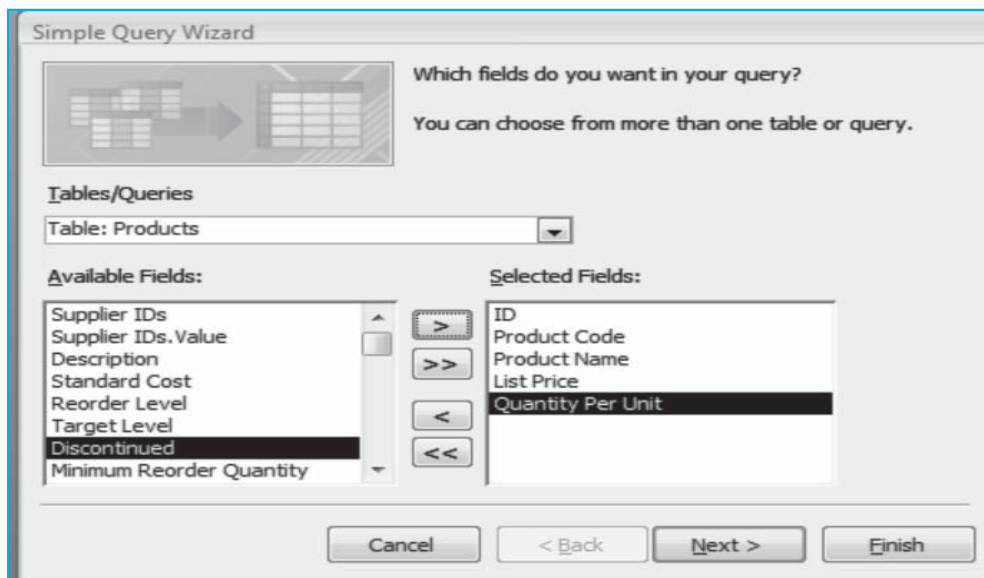


Fig. 3.5.7: Simple Query Wizard (iv)



2. When the user has added all the desired fields, he / she may click **Next** and may see Fig. 3.5.8 as follows:-

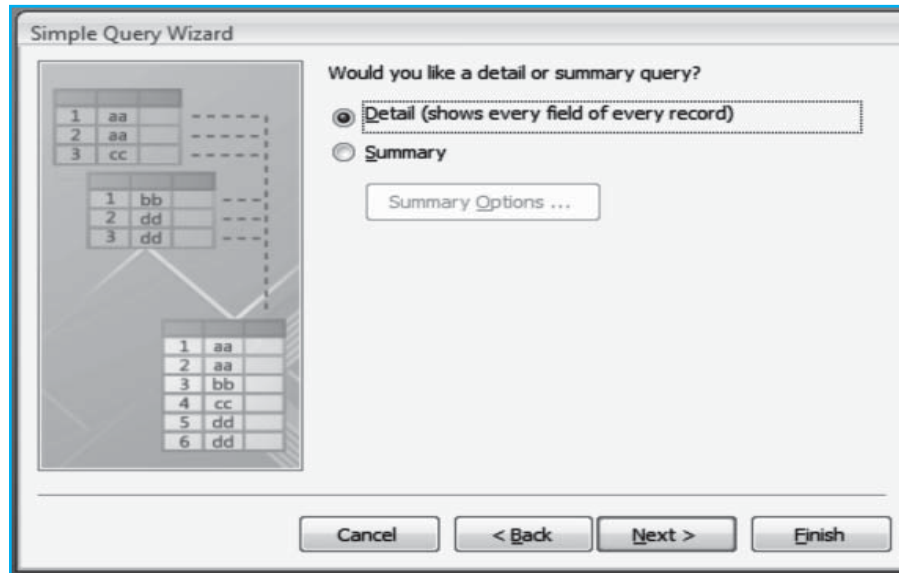


Fig. 3.5.8: Simple Query Wizard (v)

The user will be provided with two choices and are supposed to do one of the following activities:-

- If the user wishes to see individual records, they will be required to click **Details** and further, the user will be required to go directly to step 5.
- If the user wishes to see summarized numeric data, such as averages, they will be required to click **Summary**, followed by **Summary Options** as shown in Fig. 3.5.9.

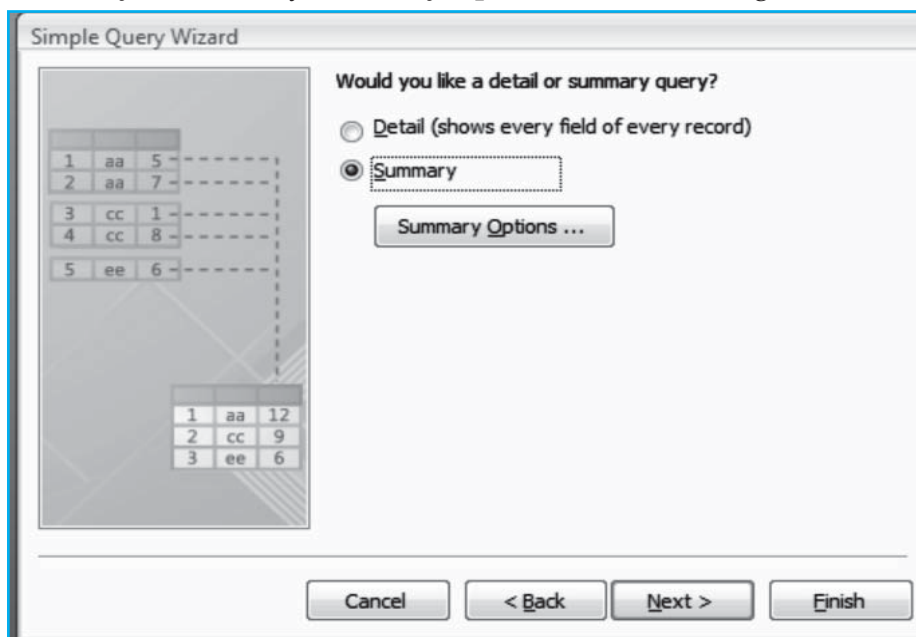


Fig. 3.5.9: Simple Query Wizard (vi)



3. The following Fig. 3.5.10 will be provided to the user for further inputs, when the user will click **Summary Options**:-

Field	Sum	Avg	Min	Max
List Price	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☐ Count records in Products

Fig. 3.5.10: Summary Options (i)

4. In the **Summary Options** dialog box, the user will be required to specify which fields he or she wants to summarize and how the user wants to summarize the data. Only number fields will be listed in the dialog box. For each number field, the user may choose one of the following functions:
- **Sum:** If the user wants that the query should returns the sum of all the values of the field.
 - **Avg:** If the user wants that the query should returns the average of the values of the field.
 - **Min:** If the user wants that the query should returns the smallest value of the field.
 - **Max:** If the user wants that the query should returns the largest value of the field.

After providing all the necessary details, the user will be required to click **ok** as shown in Fig 3.5.11.



Field	Sum	Avg	Min	Max
List Price	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☐ Count records in Products

Fig. 3.5.11: Summary Options (ii)

5. The user will be taken to figure 4.9, where the user will be required to click Next.
6. The user will be required to perform Step 4 and Step 5 given above for Customers table step by step instructions. The resultant screen will look as shown in Fig. 3.5.12.

ID	Product Code	Product Name	Quantity Per Unit	Sum Of List Price
1	NWTB-1	Northwind Traders Chai	10 boxes x 20 bags	\$18.00
3	NWTCO-3	Northwind Traders Syrup	12 - 550 ml bottles	\$10.00
4	NWTCO-4	Northwind Traders Cajun Seasoning	48 - 6 oz jars	\$22.00
5	NWTO-5	Northwind Traders Olive Oil	36 boxes	\$21.35
6	NWTJP-6	Northwind Traders Boysenberry Spread	12 - 8 oz jars	\$25.00
7	NWTDNF-7	Northwind Traders Dried Pears	12 - 1 lb pkgs.	\$30.00
8	NWTS-8	Northwind Traders Curry Sauce	12 - 12 oz jars	\$40.00
14	NWTDNF-14	Northwind Traders Walnuts	40 - 100 g pkgs.	\$23.25
17	NWTCFV-17	Northwind Traders Fruit Cocktail	15.25 OZ	\$39.00
19	NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	10 boxes x 12 pieces	\$9.20
20	NWTJP-6	Northwind Traders Marmalade	30 gift boxes	\$81.00
21	NWTBGM-21	Northwind Traders Scones	24 pkgs. x 4 pieces	\$10.00
34	NWTB-34	Northwind Traders Beer	24 - 12 oz bottles	\$14.00
40	NWTCM-40	Northwind Traders Crab Meat	24 - 4 oz tins	\$18.40
41	NWTSO-41	Northwind Traders Clam Chowder	12 - 12 oz cans	\$9.65
43	NWTB-43	Northwind Traders Coffee	16 - 500 g tins	\$46.00
48	NWTCA-48	Northwind Traders Chocolate	10 pkgs	\$12.75
51	NWTDNF-51	Northwind Traders Dried Apples	50 - 300 g pkgs.	\$53.00
52	NWTG-52	Northwind Traders Long Grain Rice	16 - 2 kg boxes	\$7.00
56	NWTP-56	Northwind Traders Gnocchi	24 - 250 g pkgs.	\$38.00
57	NWTP-57	Northwind Traders Ravioli	24 - 250 g pkgs.	\$19.50
65	NWTS-65	Northwind Traders Hot Pepper Sauce	32 - 8 oz bottles	\$21.05
66	NWTS-66	Northwind Traders Tomato Sauce	24 - 8 oz jars	\$17.00
72	NWTD-72	Northwind Traders Mozzarella	24 - 200 g pkgs.	\$34.80
74	NWTDNF-74	Northwind Traders Almonds	5 kg pkg.	\$10.00
77	NWTCO-77	Northwind Traders Mustard	12 boxes	\$12.00

Fig. 3.5.12: Resultant Screen



3.6 CREATING SELECT QUERY USING DESIGN VIEW

The user may use the query design view to create a select query manually. The query design view provided by MS-Access offers more control during the design of query. The user may commit some design mistakes while creating a query using query design view. Moreover, designing a query may take longer time than creating the same query through a query wizard.

Source:

Database : Northwind 2010

Table : Customers

Steps:-

1. On the **Create** tab, in the **Queries** group, click **Query Design**, as shown in Fig. 3.6.1.

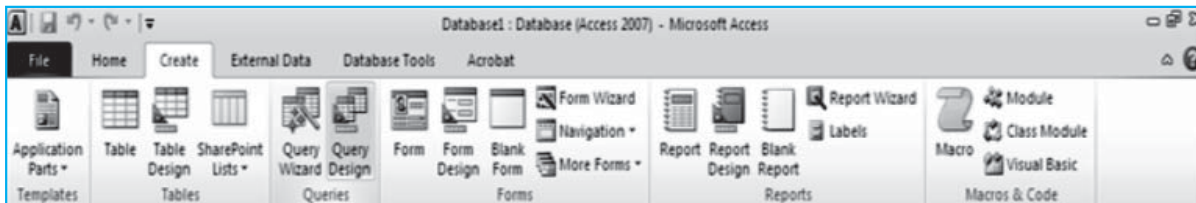


Fig. 3.6.1: 'Query Design' option

2. Following Fig. 3.6.2 containing the details of tables and queries available under **Show Table** dialog box will be shown to the user:-

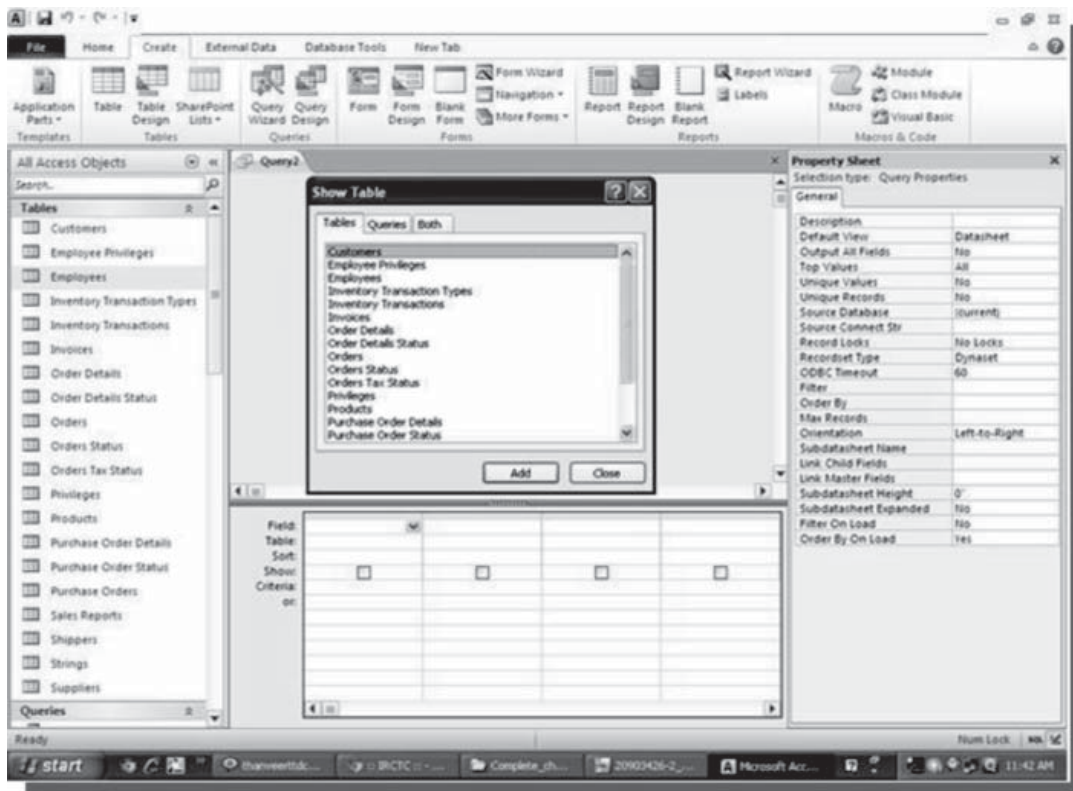


Fig. 3.6.2: 'Show Table' Dialog box (i)



3. Choose any table / query from the list provided in the **Show Table** dialog box and click **Add**. The following Fig. 3.6.3 will be shown to the user:-

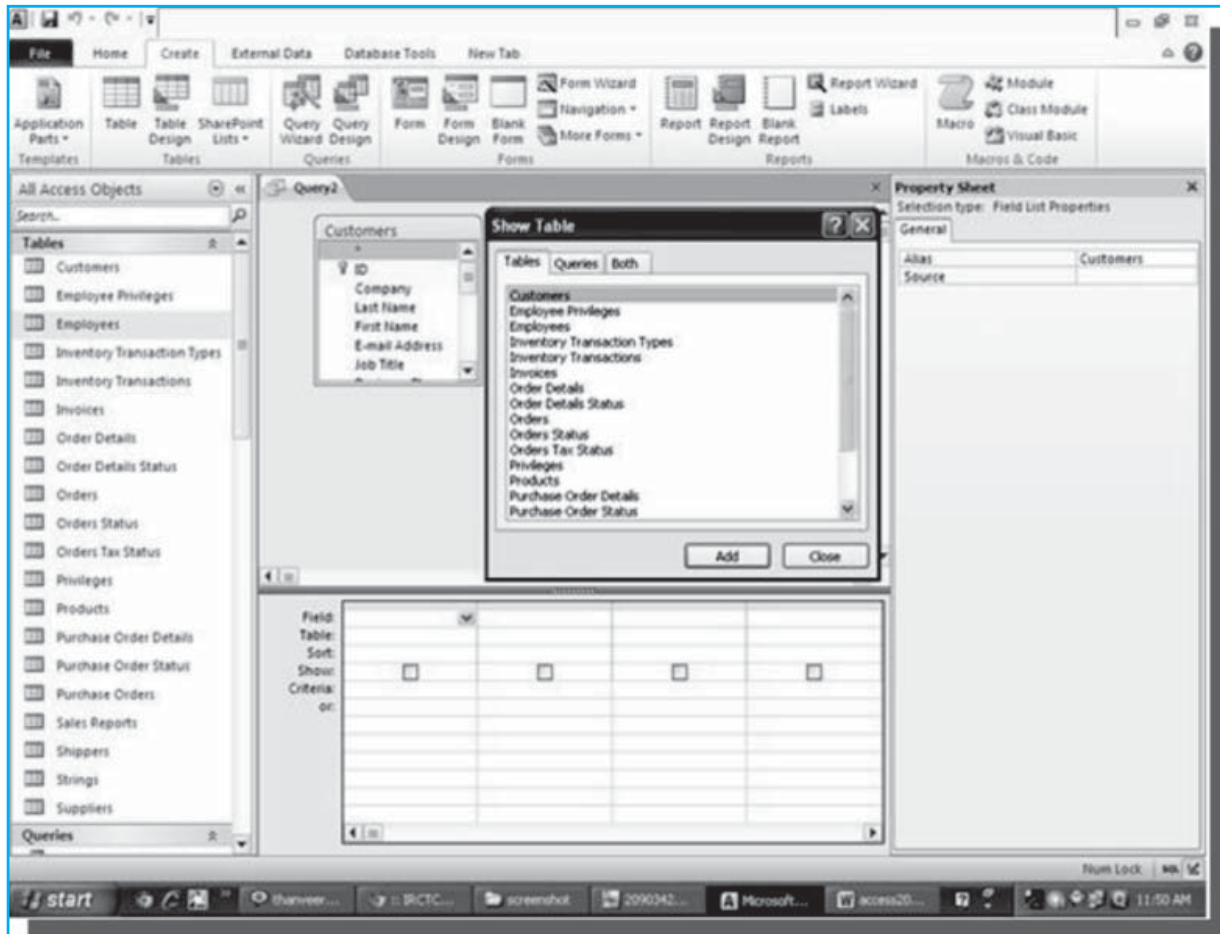


Fig. 3.6.3: 'Show Table' Dialog Box (ii)

4. If the user wants to add more table(s) or query(s), he / she may do so as per requirement. Further, the user has to click **Close** to see Fig. 3.6.4.

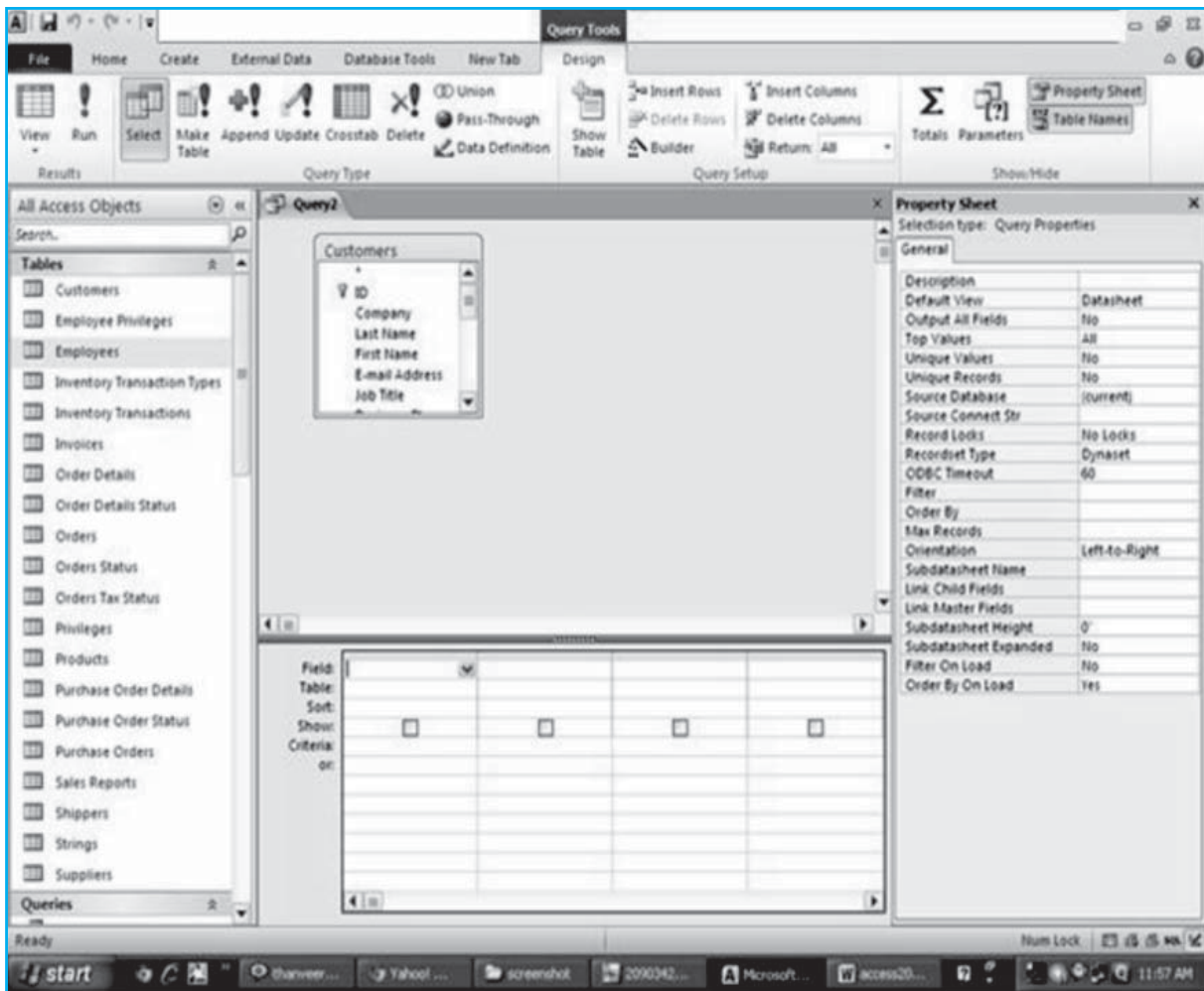


Fig. 3.6.4: On Close of 'Show Table'

4. To choose field(s), the user will be required to click the down-arrow available with the Field option as shown in Fig. 3.6.5. The table name appears on the table option. The user may choose as many fields as required for the query by repeating this step.

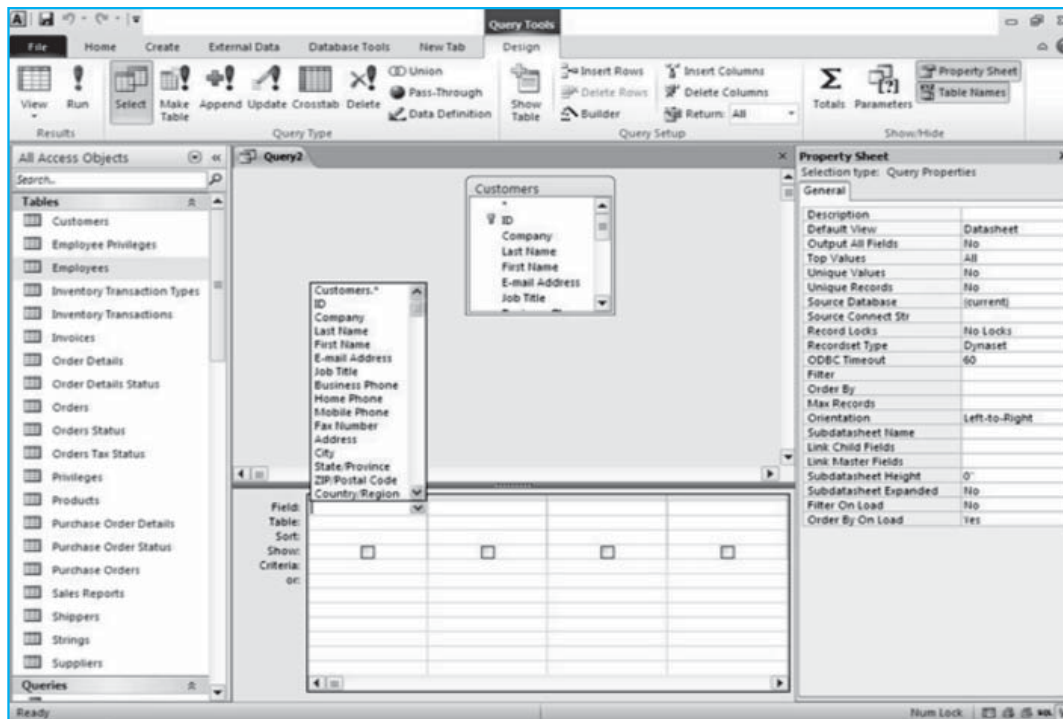


Fig. 3.6.5: Query Tools (i)

After selecting the desired number of fields, the screen will look as Fig. 3.6.6 which is as follows:-

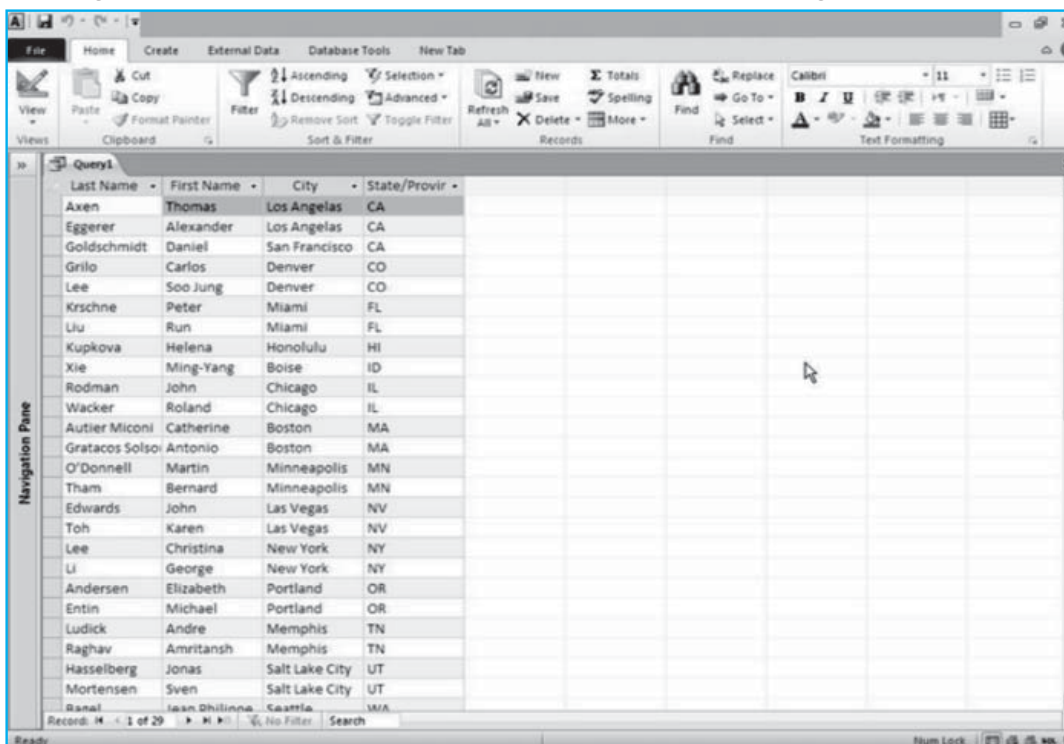


Fig. 3.6.6: Query Tools (ii)



5. On the **Design** tab, in the **Results** group, click **Run**, as shown in Fig. 3.6.7.

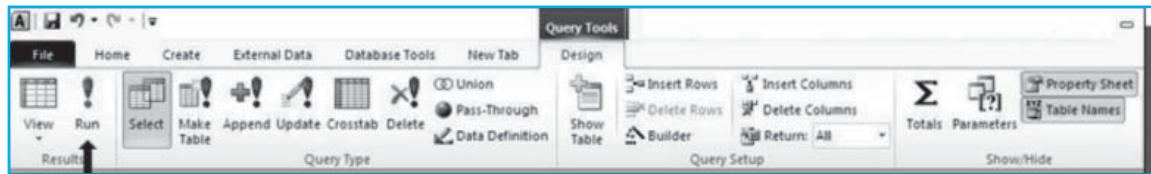


Fig. 3.6.7: 'Run' option of Query

6. The user will be shown a screen similar to Fig. 3.6.8 as follows:-

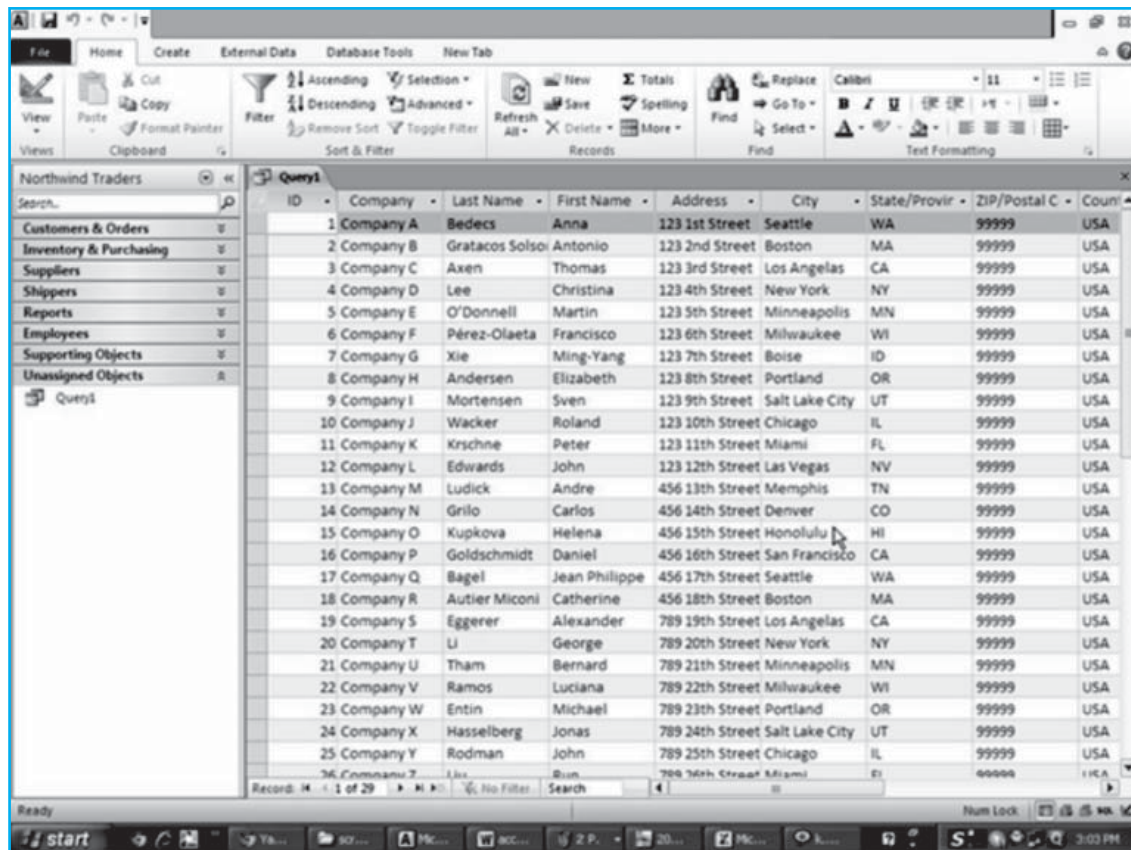


Fig. 3.6.8: On executing a Query

Note: For saving a query the user may follow the normal steps for saving a file in MS-Office environment. Alternatively, the user may follow the steps mentioned as follows to save a query:-

1. Right click **Query** tab as shown in Fig. 3.6.9 as follows:-

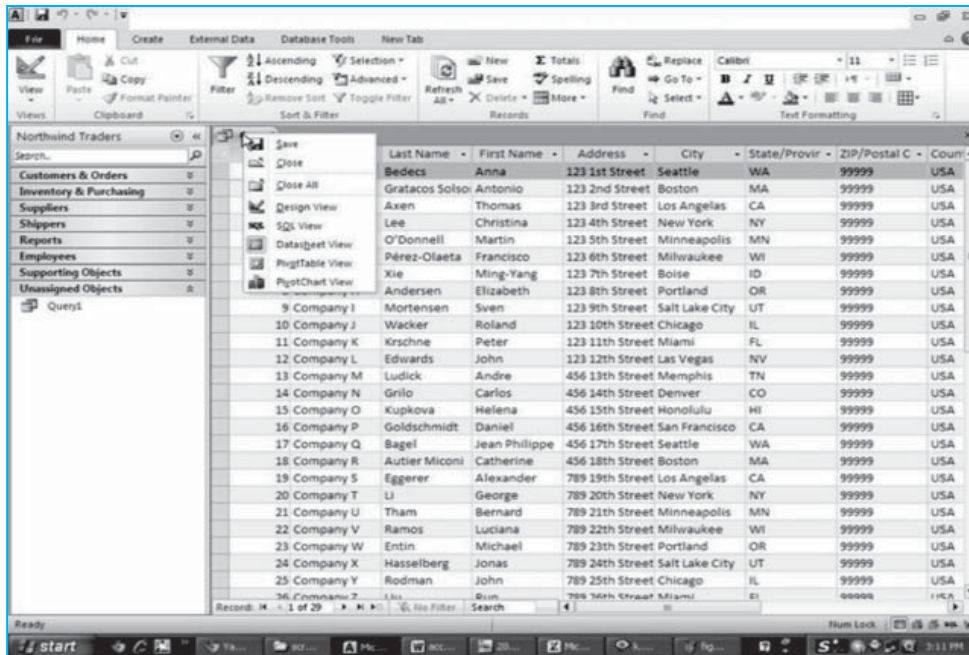


Fig. 3.6.9: Saving a Query

2. Choose **Save** option from the drop down box available. The user will be shown the following Fig. 3.6.10.

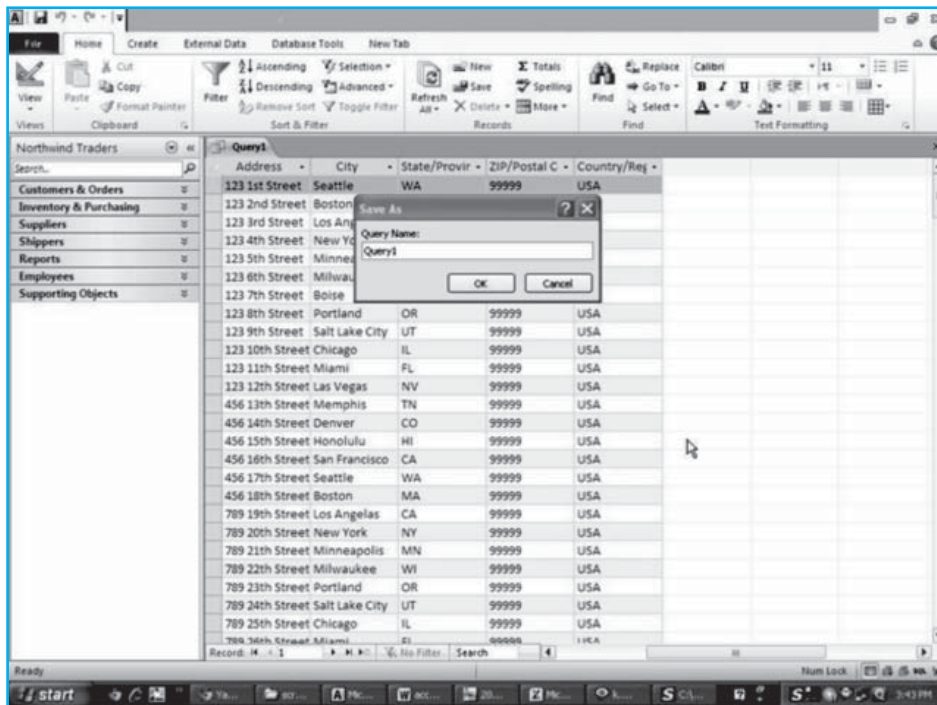


Fig. 3.6.10: Naming a Query

3. Provide a meaningful name to the query and click **OK** to complete the save process.



3.7 DISPLAY ALL RECORDS AND ALL FIELDS

Sometime the user wants to see all complete information contained in the table. The query design view offers an option that lets the user display all of the fields and all the records from the table as shown in Fig. 3.7.1, which is as follows:-

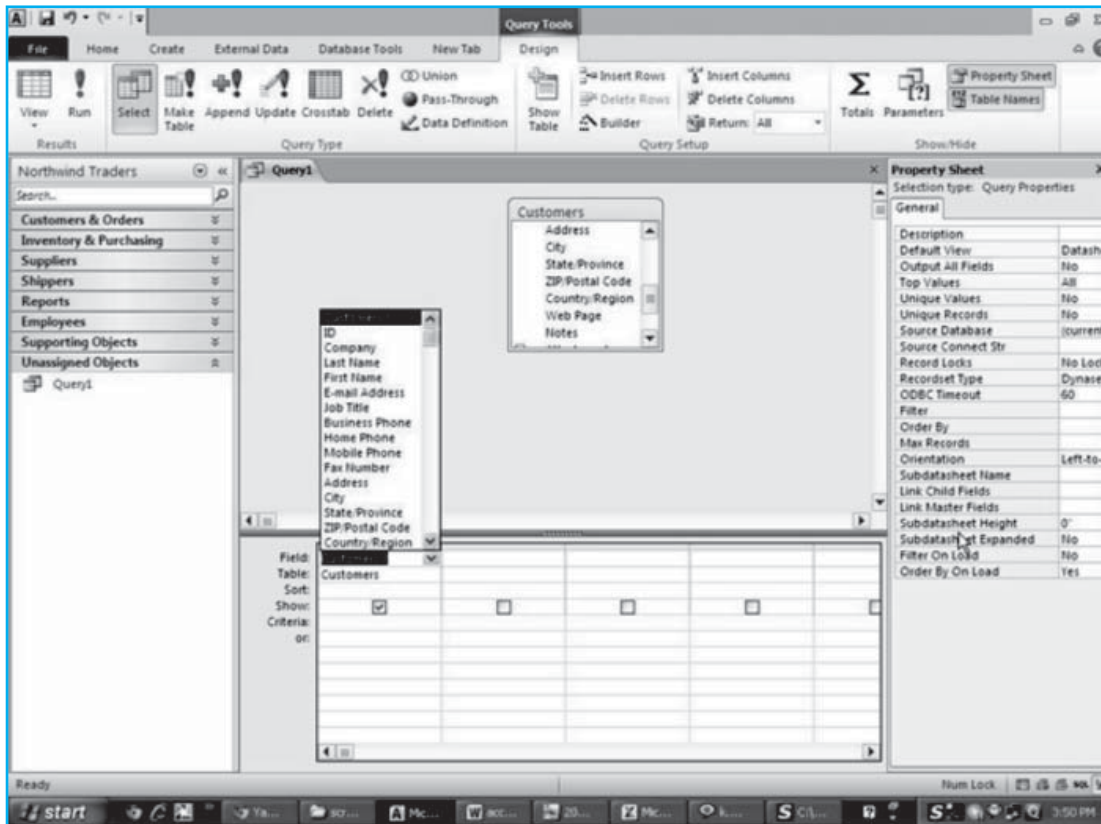


Fig. 3.7.1: Query Design View (i)

3.8 CHANGE FROM DATASHEET VIEW TO QUERY DESIGN VIEW

Once the user has seen the result of the query created, he or she may wish to do some modification in the query as per a particular requirement, in such case, the user has to switch from datasheet view to query design view by clicking **View**, under results group in **Design** tab option and choosing **Design** view as shown in Fig. 3.8.1.

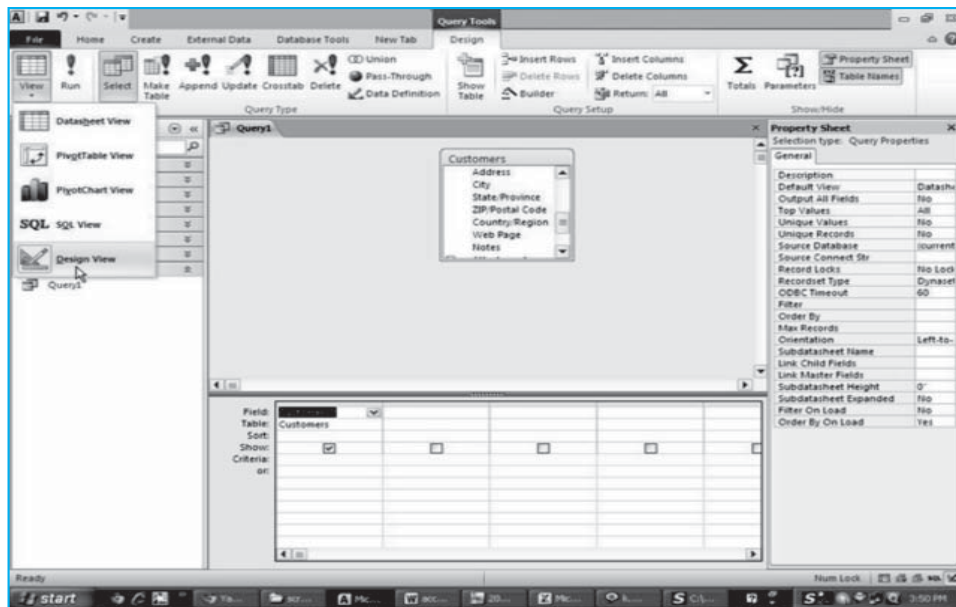


Fig. 3.8.1: Query Design View (ii)

3.9 SORTING A QUERY

The query created by the user may be sorted in ascending and descending mode, as per requirement. For sorting, the user may specify the order of sorting available under **Sort** option as shown in Fig. 3.9.1, which is as follows:-

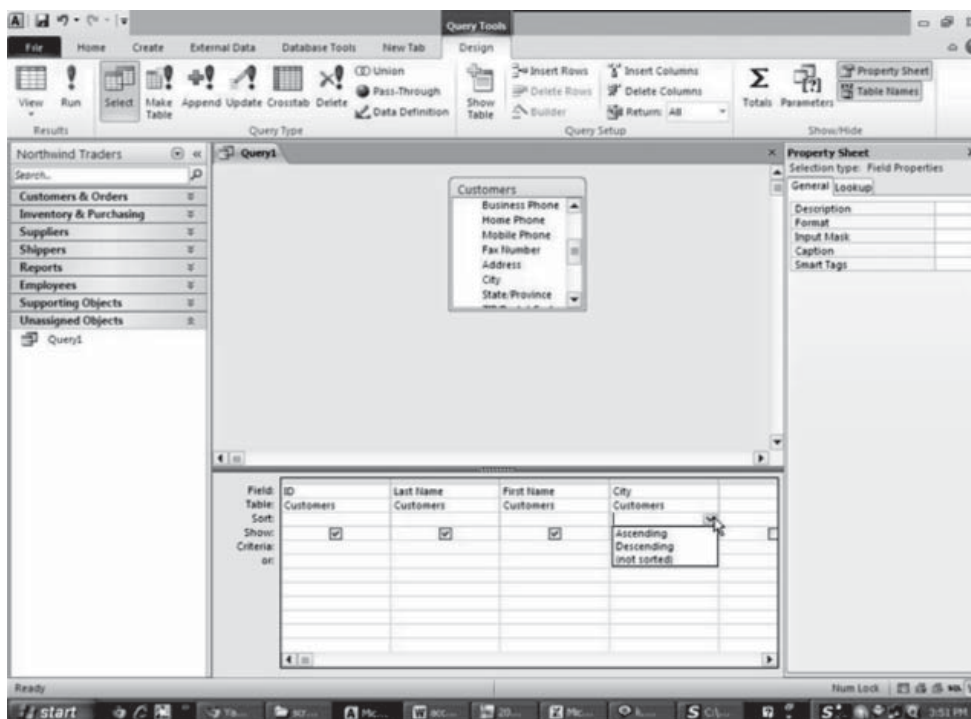


Fig. 3.9.1: Query Sorting



3.9.1 Sorting Multiple Columns in a Query

For sorting the result of a query based on multiple fields, the user may specify sort within a sort. For example, taking the customer table as source, the user may sort the query result by state and then within a state, the user may sort by city and within a city, the user may sort as per last name and then by first name as shown in Fig. 3.9.2, which is as follows:-

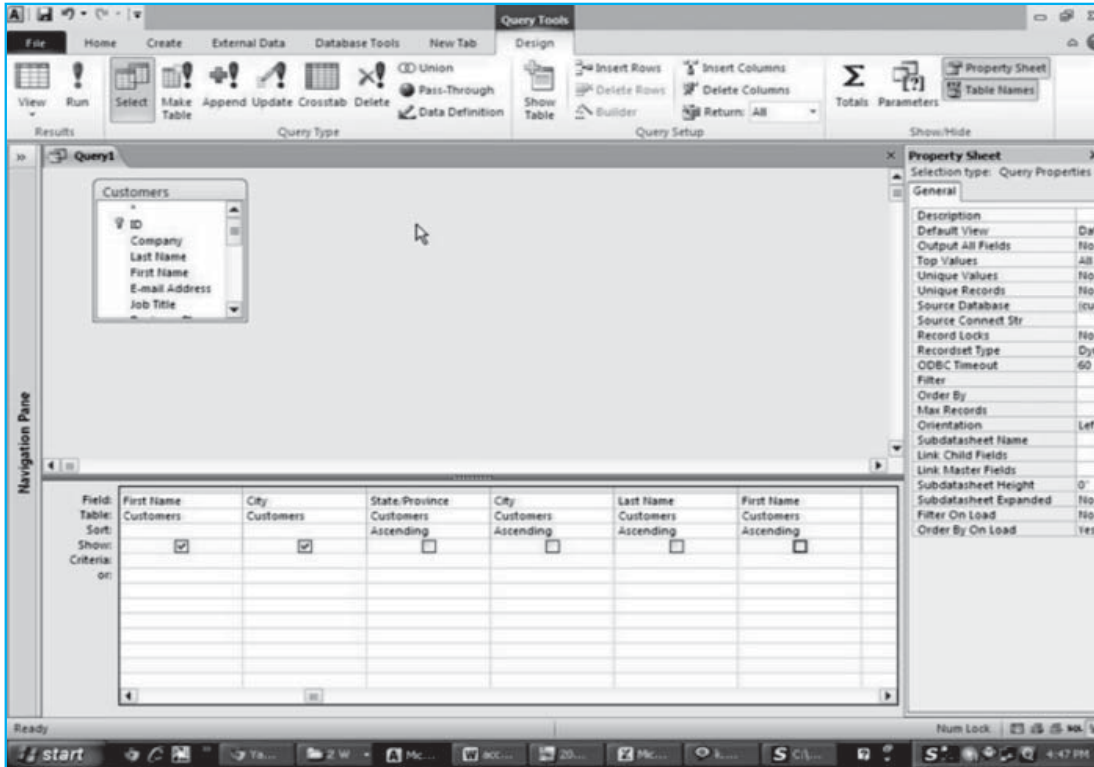


Fig. 3.9.2: Sorting multiple columns

Note: The user may want to sort by state, then by city within a state and then by last name within a state and then by first name within last name. The user is required to enter the sort in the following order: state, city, last name, first name. The sort order may not agree with the order in which the user wants to display fields in the result. In such a case, the user may first choose the required fields for display in the query result and along with that, the user then mention the fields as per sort order. To prevent a field from displaying twice, the user will have to deselect the Show box on the Show row option.

The result of the query thus generated is shown in Fig. 3.9.3, which is as follows:-



Last Name	First Name	City	State/Province
Axen	Thomas	Los Angeles	CA
Eggerer	Alexander	Los Angeles	CA
Goldschmidt	Daniel	San Francisco	CA
Grilo	Carlos	Denver	CO
Lee	Soo Jung	Denver	CO
Krschne	Peter	Miami	FL
Liu	Run	Miami	FL
Kupkova	Hefena	Honolulu	HI
Xie	Ming-Yang	Boise	ID
Rodman	John	Chicago	IL
Wacker	Roland	Chicago	IL
Autier Miconi	Catherine	Boston	MA
Gratacos Solso	Antonio	Boston	MA
O'Donnell	Martin	Minneapolis	MN
Tham	Bernard	Minneapolis	MN
Edwards	John	Las Vegas	NV
Toh	Karen	Las Vegas	NV
Lee	Christina	New York	NY
Li	George	New York	NY
Andersen	Elizabeth	Portland	OR
Entin	Michael	Portland	OR
Ludick	Andre	Memphis	TN
Raghav	Amritansh	Memphis	TN
Hasselberg	Jonas	Salt Lake City	UT
Mortensen	Sven	Salt Lake City	UT
Ranal	Isan Dhillon	Seattle	WA

Fig. 3.9.3: Query Result

3.10 RETRIEVING SPECIFIC RECORDS

As per requirements, the user may wish to retrieve specific records, instead of retrieving all the records. For example, the user may wish to retrieve only those records where the product list price is greater than \$ 18.00. In such cases, the user may specify some criteria (condition) to retrieve specific records. For this purpose, logical operators are used. The valid logical operators which could be used by the user to specify criteria are as follows:-

Logical Operators			
Operator	Meaning	Field Type	Entry Format
=	Equal to	Character Number Date	= "DE" = 5 = #2/16/88#
<>	Not equal to	Character Number Date	<> "DE" <> 5 <> #2/16/88#
>	Greater than	Character Number Date	> "DE" > 5 > #2/16/88#
>=	Greater than or equal to	Character Number Date	>= "DE" >= 5 >= #2/16/88#



<	Less than	Character Number Date	< "DE" < 5 < #2/16/88#
<=	Less than or equal to	Character Number Date	<= "DE" <= 5 <= #2/16/88#
In	Equal to any item in a list	Character Number Date	In ("DE", "NJ") In (5, 9, 17) In (#2/16/88#, #2/3/ 90#, #12/15/88#)
Not In	Not equal to any item in a list	Character Number Date	Not In ("DE", "NJ") Not In (5, 9, 17) Not In (#2/16/88#, #2/3/ 90#, #12/15/88#)
Between	Between two values, greater than or equal to one and less than or equal to the other	Character Number Date	Between "C" And "F" Between 5 And 10 Between #1/1/88# And #12/31/88#
Not Between	Not between two values	Character Number Date	Not Between "C" And "F" Not Between 5 And 10 Not Between #1/1/88# And #12/31/88#
Is Null	The value is missing from the field	Character Number Date	Is Null Is Null Is Null
Is Not Null	The value is not missing from the field	Character Number Date	Is Not Null Is Not Null Is Not Null
Like	Like a specified pattern. * means any series of characters. ? means any single character.	Character Number Date	Like "S*" Like "1*" Not Applicable
Not Like	Not like a specified pattern. * means any series of characters. ? means many single character.	Character Number Date	Like "S*" Like "1*" Not Applicable

Table 3.10.1: Logical operators

Taking the above mentioned example of product table, the screen will look like Fig. 3.10.1 which is as follows:-

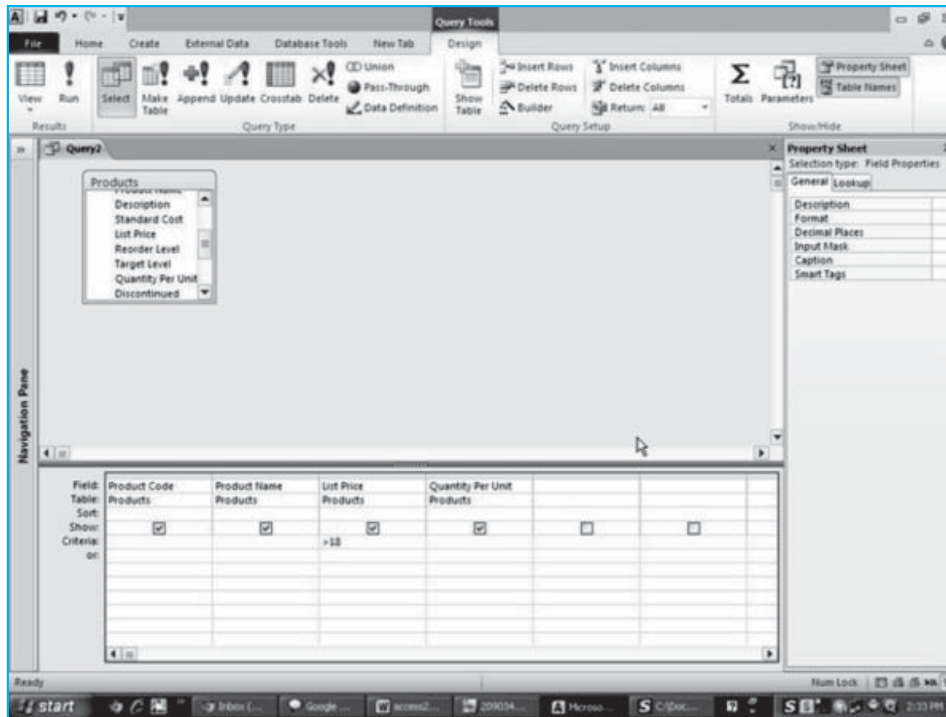


Fig. 3.10.1: Resultant Screen (i)

The resultant screen would be like Fig. 3.10.2 which is as follows:-

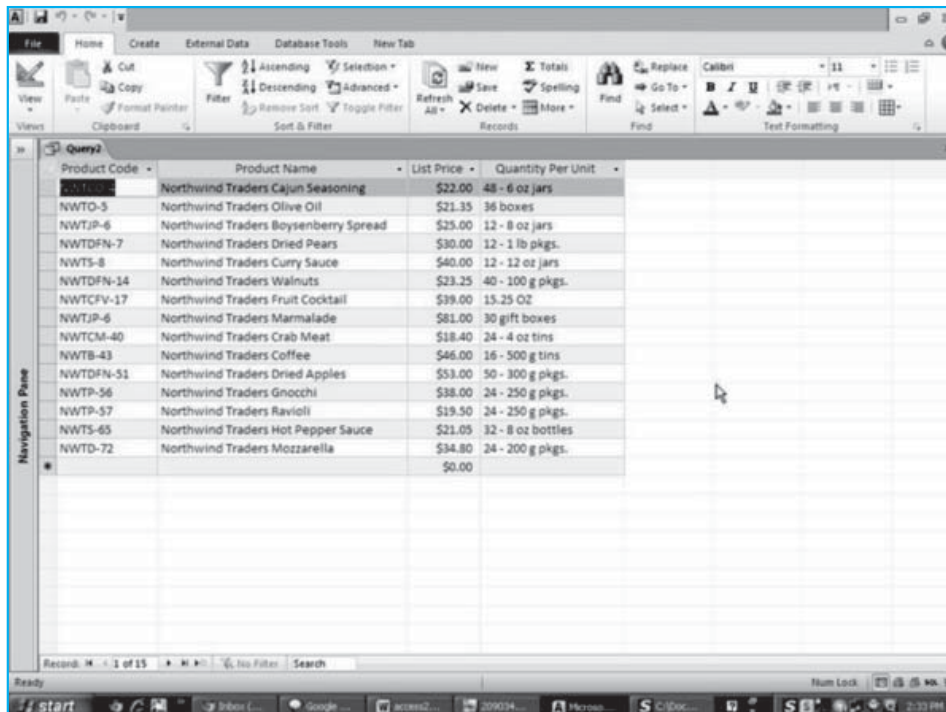
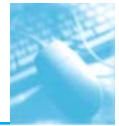


Fig. 3.10.2: Resultant Screen (ii)



3.10.1 Applying Multiple Criteria

Multiple criteria could be applied to the same table as shown in Fig. 3.10.3, which is as follows:-

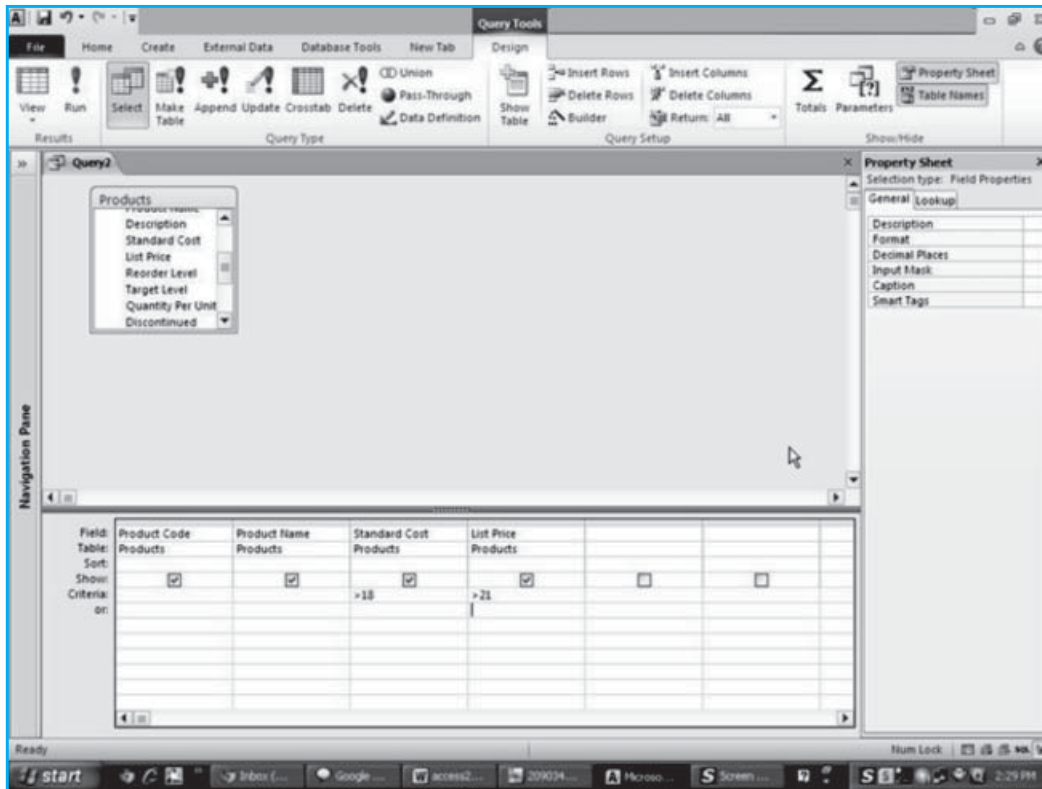


Fig. 3.10.3: To apply multiple criteria

Here, the user wants to view only those records from the product table where the Standard Cost of a product is above 18 and the List Price is above 21. The resultant screen is shown as Fig. 3.10.4, which is as follows:-



Product Code	Product Name	Standard Cost	List Price
NWTFN-7	Northwind Traders Boysenberry Spread	\$18.75	\$25.00
NWTFN-7	Northwind Traders Dried Pears	\$22.50	\$30.00
NWTS-8	Northwind Traders Curry Sauce	\$30.00	\$40.00
NWTCFV-17	Northwind Traders Fruit Cocktail	\$29.25	\$39.00
NWTJP-6	Northwind Traders Marmalade	\$60.75	\$81.00
NWTB-43	Northwind Traders Coffee	\$34.50	\$46.00
NWTFN-51	Northwind Traders Dried Apples	\$39.75	\$53.00
NWTP-56	Northwind Traders Gnocchi	\$28.50	\$38.00
NWTD-72	Northwind Traders Mozzarella	\$26.10	\$34.80
		\$0.00	\$0.00

Fig. 3.10.4: Resultant Screen (iii)

3.11 CREATING QUERIES USING MULTIPLE TABLES

The user can create a query using two or more tables. In such cases, it is necessary to establish proper relationship between the concerned tables. Following are the steps which will show how to create query using two tables:-

Source:-

Database : Northwind 2010

Table 1 : Products

Table 2 : Purchase Order Details

1. Repeat the steps for opening **Query Design** View as explained earlier.
2. Repeat steps for inserting the table(s) as explained earlier.
3. Choose the fields as shown in Fig. 3.11.1, which is as follows:-

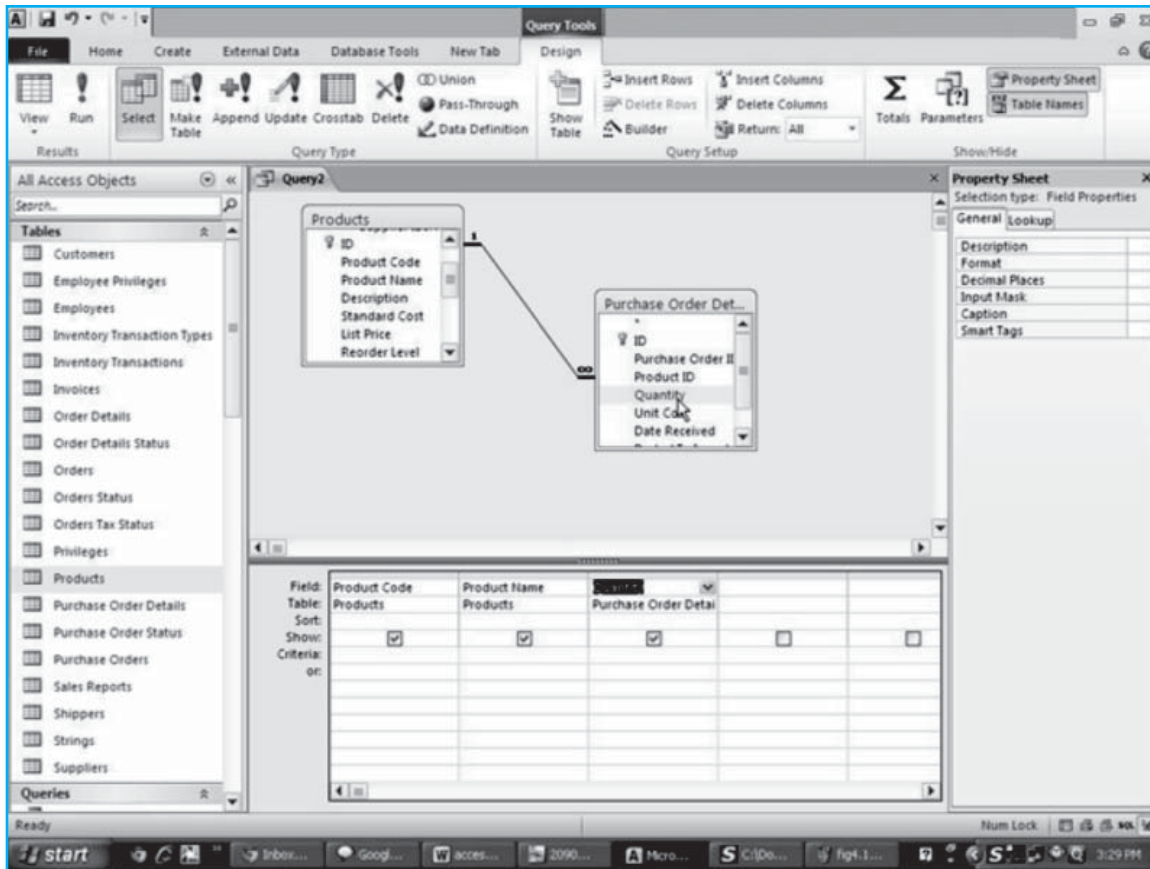


Fig. 3.11.1: Query Design using two tables

4. On the **Design** tab, in the **Results** group, click **Run** and see the result as shown in Fig. 3.11.2 as follows:-



Product Code	Product Name	Quantity
NWTB-1	Northwind Traders Chai	40
NWTB-1	Northwind Traders Chai	40
NWTB-1	Northwind Traders Chai	1
NWTCO-3	Northwind Traders Syrup	100
NWTCO-3	Northwind Traders Syrup	50
NWTCO-4	Northwind Traders Cajun Seasoning	40
NWTCO-4	Northwind Traders Cajun Seasoning	40
NWTO-5	Northwind Traders Olive Oil	40
NWTJP-6	Northwind Traders Boysenberry Spread	100
NWTJP-6	Northwind Traders Boysenberry Spread	10
NWTFN-7	Northwind Traders Dried Pears	40
NWTS-8	Northwind Traders Curry Sauce	40
NWTS-8	Northwind Traders Curry Sauce	25
NWTFN-14	Northwind Traders Walnuts	40
NWTCFV-17	Northwind Traders Fruit Cocktail	40
NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	20
NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	30
NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	10
NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	25
NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	20
NWTJP-6	Northwind Traders Marmalade	40
NWTJP-6	Northwind Traders Marmalade	40
NWTBGM-21	Northwind Traders Scones	20
NWTB-34	Northwind Traders Beer	60
NWTB-34	Northwind Traders Beer	100
NWMTL-34	Northwind Traders Beer	300

Fig. 3.11.2: Resultant Screen

Note: The same example can be performed by the user by opting for **Query Wizard**.

3.12 CREATE MAKE TABLE QUERY

There are instances, when a user wants to save the result of a query as a table, which containing multiple and selected fields from more than one table, for future use. This can be achieved by using **Make Table** available under **Query Type** option as shown in Fig. 3.12.1. A query saved using make table option retrieves data from one or more tables and then loads the result set into a new table. That new table can be saved in the database that is the working database or the user may also create it in another database.

Source:-

- Database : Northwind 2010
- Table 1 : Products
- Table 2 : Purchase Order Details



Fig. 3.12.1: 'Make Table' option

Once, query is created, as explained in the above sections, the user is required to do the following steps:-

1. Click **Make Table** available under **Query** Type option, the user will see a screen similar to the one shown in Fig. 3.12.2, which is as follows:-

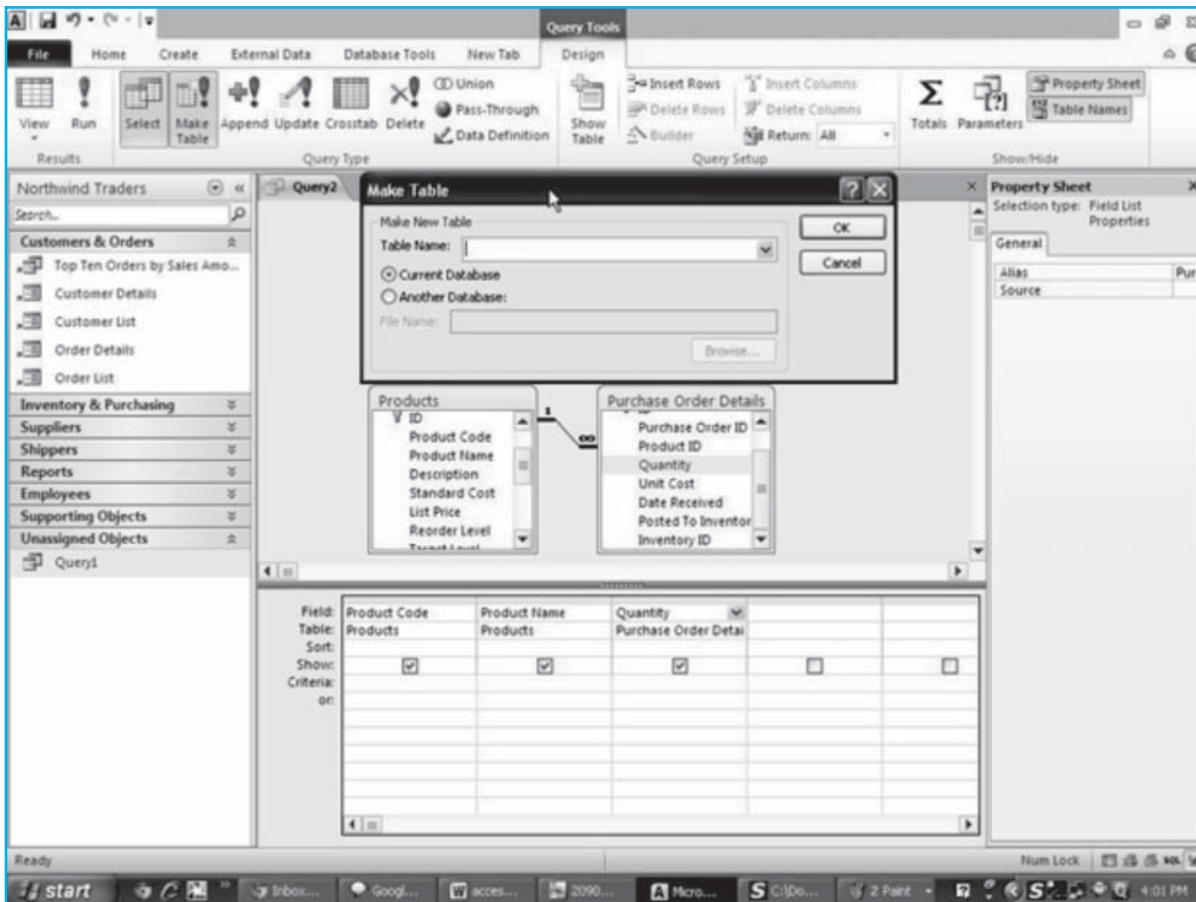


Fig. 3.12.2: 'Make Table' Dialog Box (i)

2. The user will be required to give some meaningful name to the new table as shown in Fig. 3.12.3, which is as follows:-

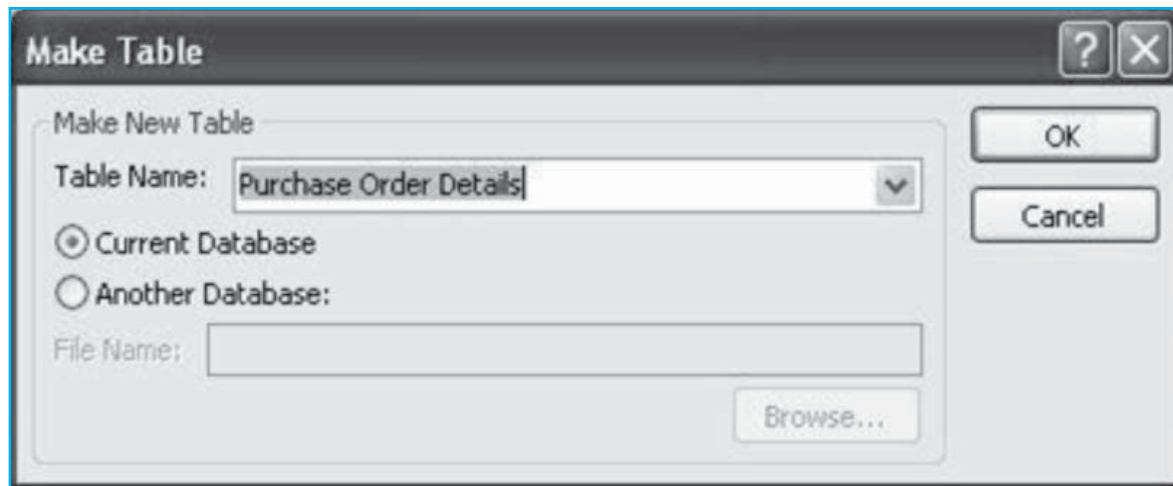


Fig. 3.12.3: 'Make Table' Dialog Box (ii)

3. Further, the user is required to specify the location of the table as either Current Database or as Another Database. In the case the user choose Current Database, he or she needs to simply click OK, else, the user will have to specify the location of the database in which the user wished to save the file and after specifying the location of the database, the user needs to click OK to finish the activity.

3.13 CREATE PARAMETER QUERY

There are times, when a user wants to supply value to a query as parameter. This can be achieved by using parameter query. A parameter query prompts a user for some input before running a query. To achieve this, the user needs to enclose a question in square brackets ([]). Taking the previous example in to consideration, if the user wishes to see the purchase order quantity of a selected product, the user needs to do the following steps:-

Source:-

Database : Northwind 2010

Table 1 : Products

Table 2 : Purchase Order Details

1. The user needs to mention [Which Product Name?] in the criteria row as shown in Fig. 3.13.1, as follows:-

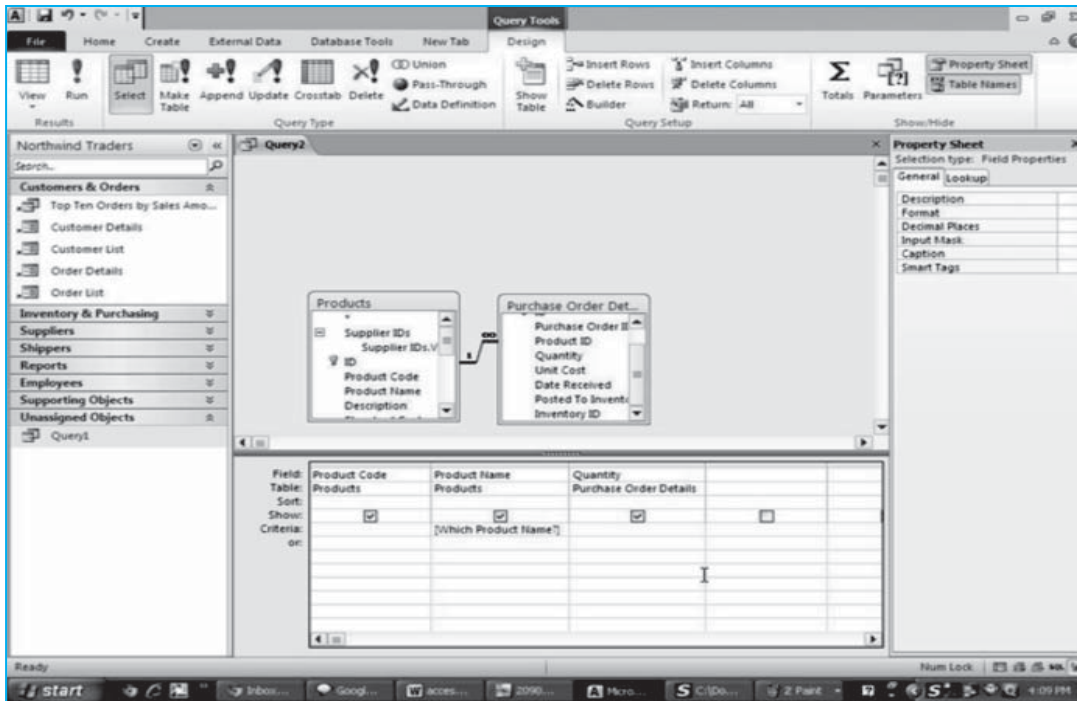


Fig. 3.13.1: To Supply value to a query

- The user then needs to click **Run** and the a screen similar to the one shown in Fig. 3.13.2 will be displayed to the user:-

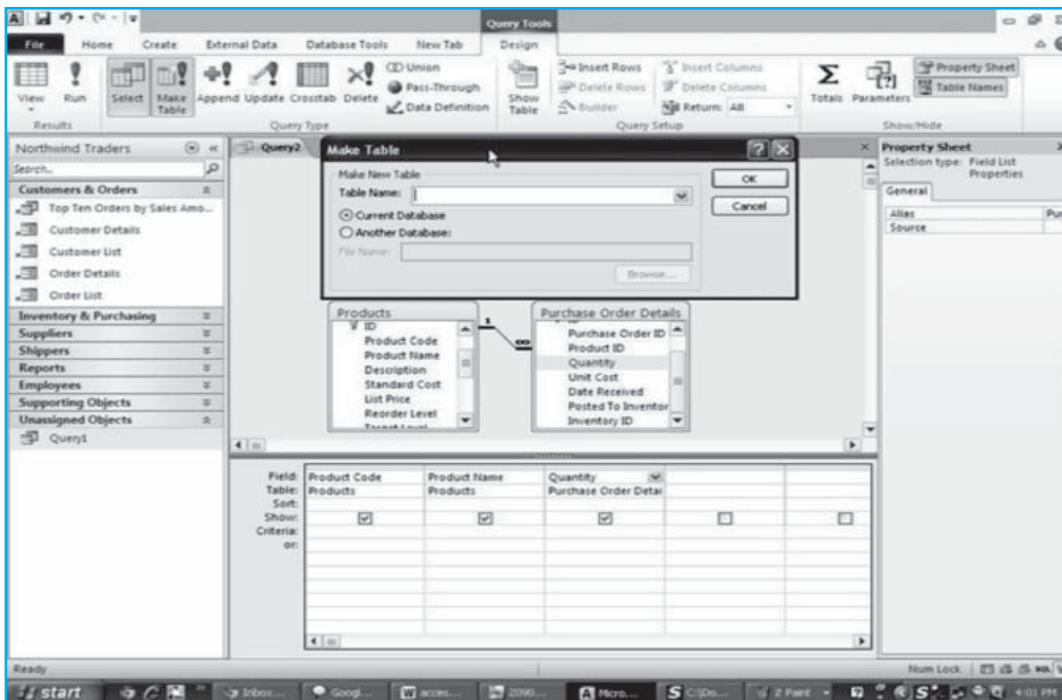


Fig. 3.13.2: On 'Run' of Query



3. Click **OK** and the user will be shown the result as shown in the following Fig. 3.13.3.

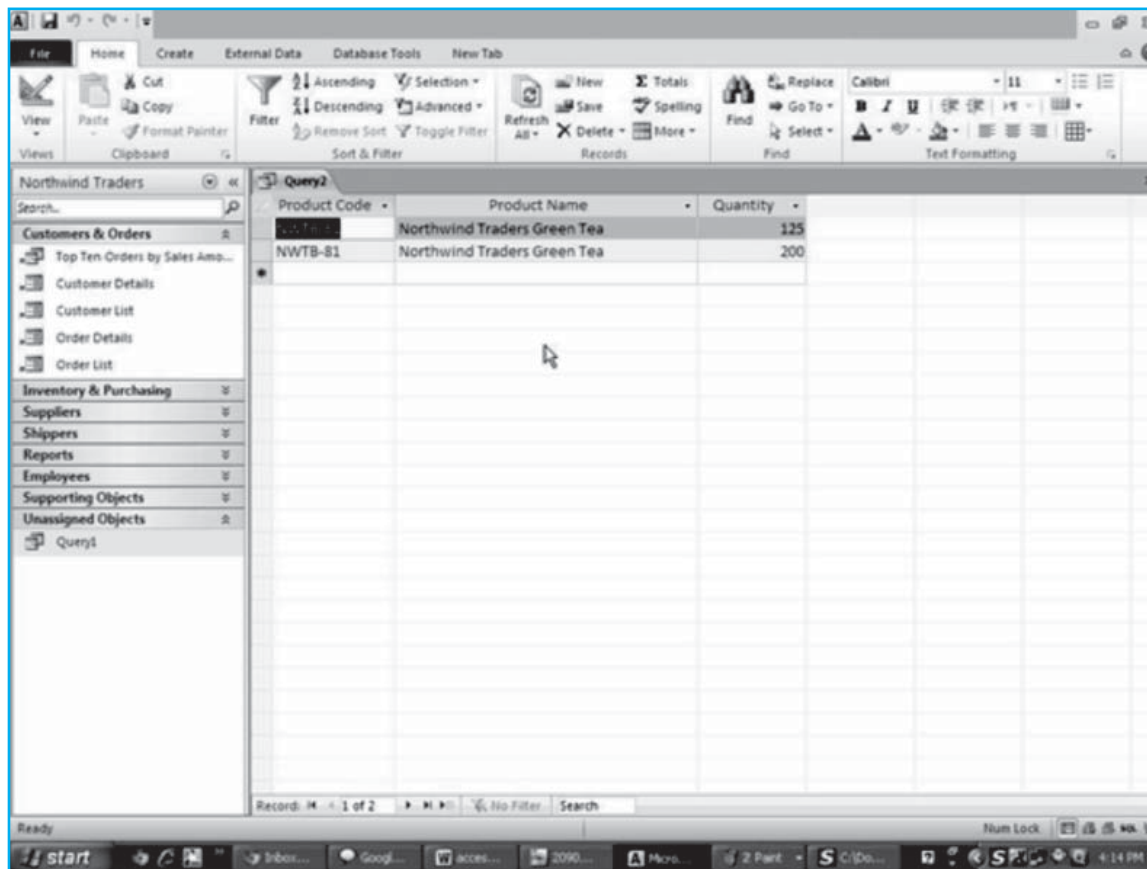
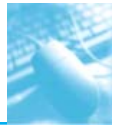


Fig. 3.13.3: Resultant Screen

Following Table 3.13.1 provides some options which the user can use in parameter query:-

Like "*" & [Prompt] & "*"	Returns all records that contain the value you enter. Example: If you enter ad , Access returns all records that include the sequence ad anywhere in the field.
Like "*" & [Prompt]	Returns all records that end with the value you enter. Example: If you enter S , Access returns all records that end with S .
Like [Prompt] & "*"	Returns all records that begin with the value you enter. Example: If you enter S , Access returns all records that begin with S .
> [Prompt] Note: You can also use < (less than) ,<= (less than or equal to) >=, >= (greater than or equal to), or <> (not equal)	Find all records with a value greater than the value you enter. Example: If you enter 5 , Access returns all records that are greater than 5 .

Table 3.13.1: Parameter Query Option



This topic deals with reports, their importance, general types of reports available under MS-Access and how to create simple reports using form wizard and report design view. It also put focus on the various report sections which could be used while designing a report using report design view.

3.14 INTRODUCTION

Reports provide the most flexible way of viewing and printing summarized information. Reports display information with the desired level of detail, while enabling the user to view or print the information in almost any format. The user can add multilevel totals, statistical comparisons and pictures and graphics to a report.

3.15 DEFINITION

A database report presents information retrieved from a table or query in a preformatted, attractive manner.

3.16 REPORTS AND ITS TYPES

There are basically following types of reports provided by MS-Access:-

3.16.1 Tabular Report

Tabular reports (also known as groups/totals reports) are similar to a table that displays data in neat rows and columns. Tabular reports, unlike forms or datasheets, usually group data by one or more fields. Often, tabular reports calculate and display subtotals or statistical information for numeric fields in each group. Some reports include page totals and grand totals. You can even have multiple snaked columns so that you can create directories (such as telephone books). These types of reports often use page numbers, report dates or lines and boxes to separate information. Reports may have color and shading and display pictures, business graphs and memo fields. A special type of summary tabular report can have all the features of a detail tabular report but omit record details.

3.16.2 Columnar Report

Columnar reports generally display one or more records per page, but do so vertically. Columnar reports display data very much as a data-entry form does but are used strictly for viewing data and not for entering data. Another type of columnar report displays one main record per page (like a business form) but can show many records within embedded subforms. An invoice is a typical example. This type of report can have sections that display only one record and at the same time have sections that display multiple records from the many side of a one-to-many relationship and even include totals.

3.16.3 Labels

Another type of columnar report displays one main record per page (like a business form) but can show many records within embedded subforms. An invoice is a typical example. This type of report can have sections that display only one record and at the same time have sections that display multiple records from the many side of a one-to-many relationship and even include totals.

3.17 GENERAL STEPS IN CREATING A REPORT

The report process begins with your desire to view your data in a table, but in a way that differs from datasheet display. You begin with a design for this view; Access begins with raw data. The



purpose of the report is to transform the raw data into a meaningful set of information. The process of creating a report involves several steps:

- Defining the report layout
- Assembling the data
- Creating the report design using the Access Report Design window
- Printing or viewing the report

3.18 CREATING A REPORT WITH REPORT WIZARDS

Report wizard helps the user to create the report in a easy and efficient manner. Report Wizards give the user a basic layout for the report. The report can further be customized as per requirement. Report Wizards simplify the layout process of the controls that are meant to be used in the report. It helps the user to create a report by asking a series of questions about the type of report that the user wants to create and then automatically creates the report for the user.

1. Select the **Report Wizard** button in the **Reports** group of the **Create** Ribbon tab as shown in Fig 3.18.1.

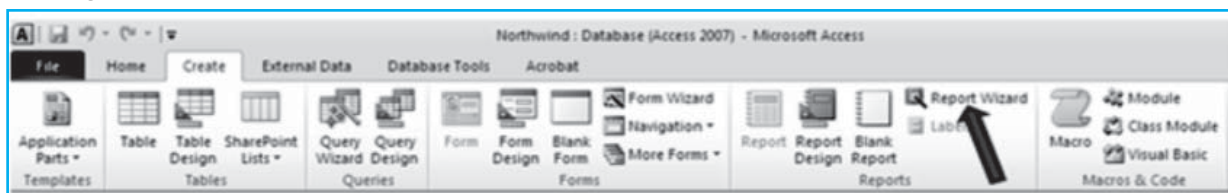


Fig. 3.18.1: Report Wizard (i)

2. Select a query or a table for the report as shown in Fig. 3.18.2:

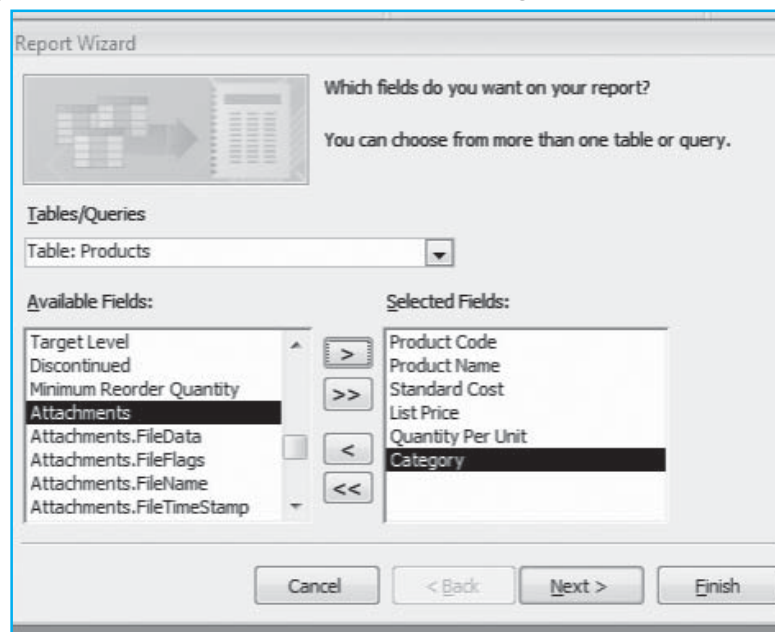


Fig. 3.18.2: Report Wizard (ii)

Note: The user can also choose fields from more than one table or query.



3. Next, specify the fields that should be included in the table as shown in Fig. 3.18.2.
4. Now, the user is prompted through a dialog box which enables the user to specify the field(s) for grouping data in the report as shown in Fig. 3.18.3 (A) and Fig. 3.18.3 (B):

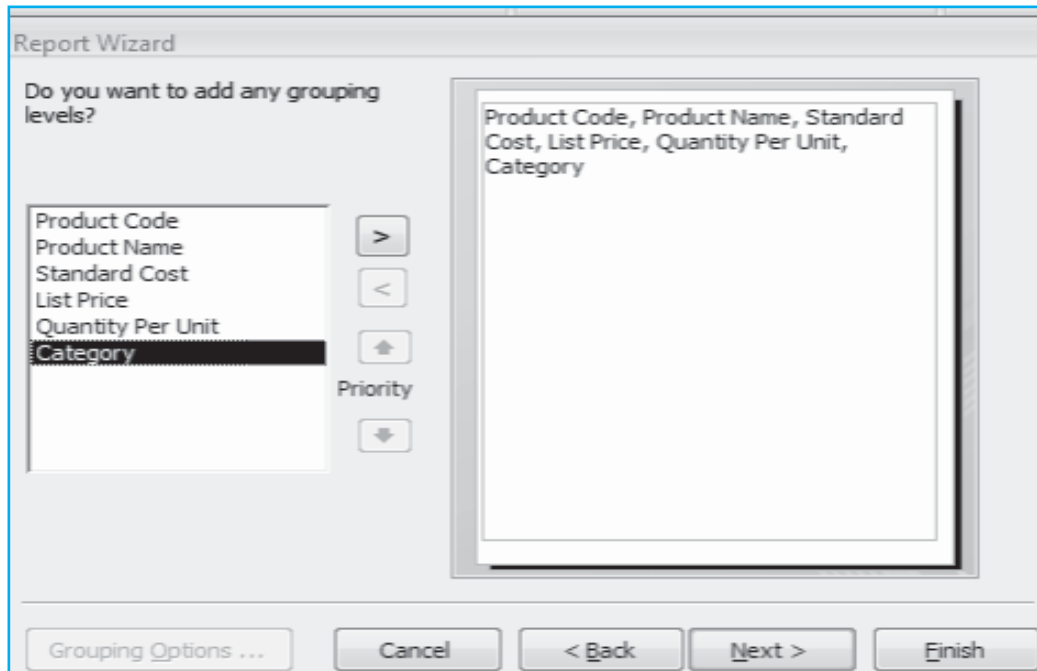


Fig. 3.18.3 (A): Before Grouping

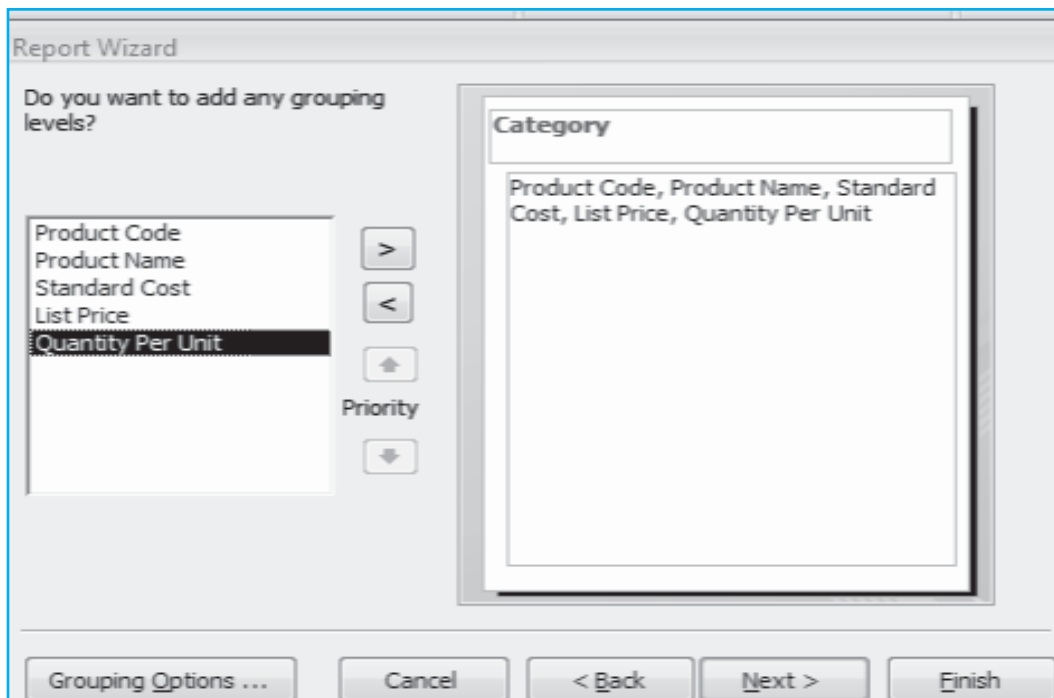


Fig. 3.18.3 (B): After Grouping



The user may click the Grouping Options button provided at the bottom left corner of the Group dialog box to display another dialog box, which enables the user to further define how the report uses the group field. For instance, the user can choose to group by only the first character of a field chosen for grouping. This means that all records with the same first character in the grouping field are included as a single group. If the user group a customers table by the CustomerName and specify to group on the first character of the CustomerName field, a group header and footer appears for the set of all customers whose name begins with the same character. There would be a group for all records with a CustomerName beginning with the letter A, another group for all records with CustomerName beginning with the letter B and so on.

The Grouping Options dialog box, which is displayed when the user click the Grouping Options button enables the user to further define the grouping. The selection is dependent on the type of data being grouped. The Grouping intervals list box displays different values for the various data types as also shown in Fig. 3.18.4:

Text: Normal, 1st Letter, 2 Initial Letters, 3 Initial Letters, 4 Initial Letters, 5 Initial letters

Numeric: Normal, 10s, 50s, 100s, 500s, 1000s, 5000s, 10000s, 50000s, 100000s.

Date: Normal, Year, Quarter, Month, Week, Day, Hour, Minute.

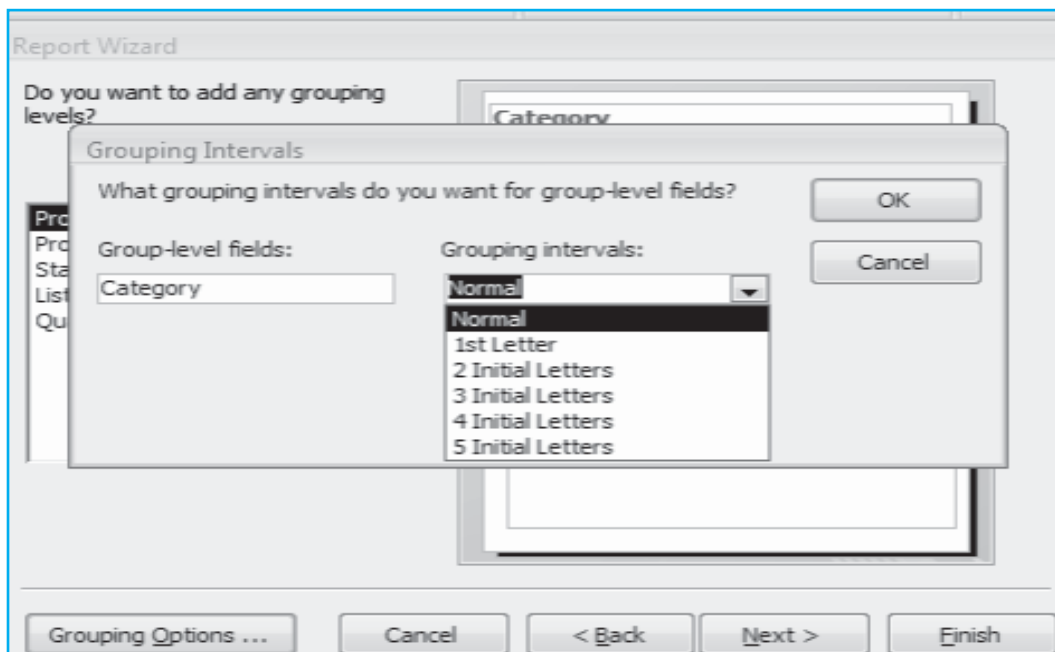


Fig. 3.18.4: Report Wizard (iii)

- Further, the user is prompted to specify the sorting order as ascending or descending as shown in Fig. 3.18.5. The user can select fields that they have not already chosen to group and can use them as sorting fields. The fields selected in the sorting dialog box does not affect grouping.

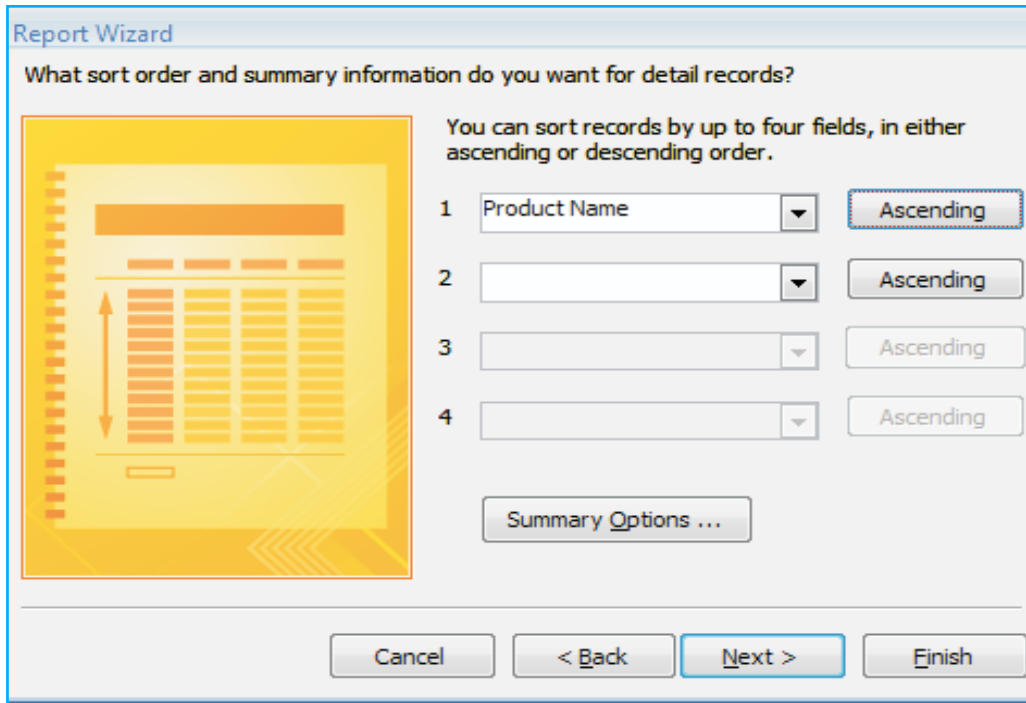


Fig. 3.18.5: Report Wizard (iv)

6. The user may use the **Summary Option** button which opens up a dialog box which provides additional display options for numeric fields as shown in Fig. 3.18.6.

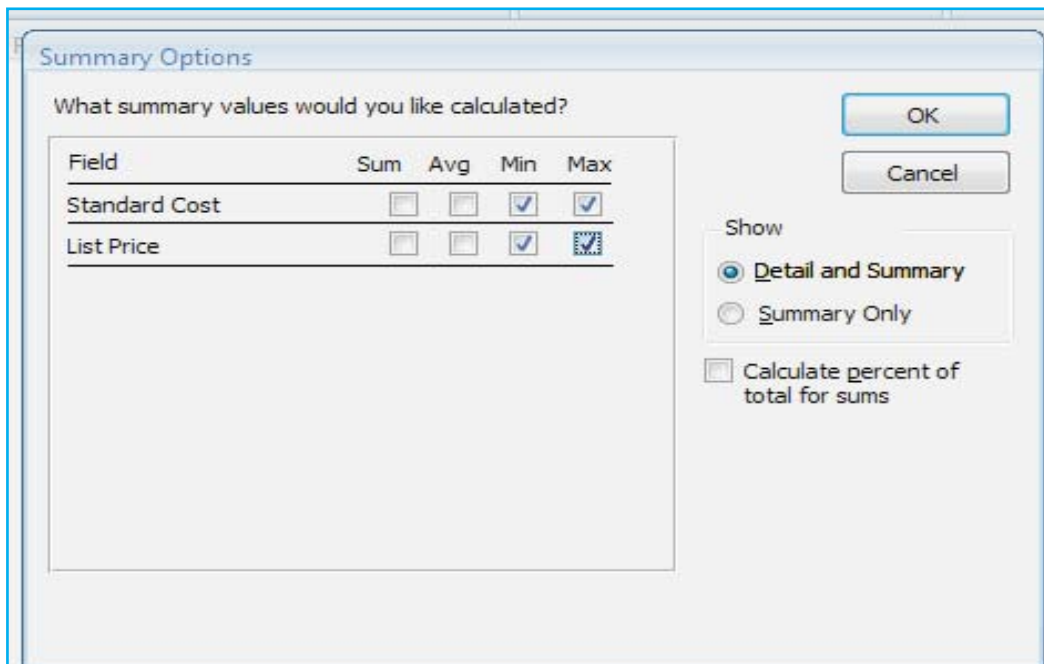


Fig. 3.18.6: Summary Options Dialog Box



- Further, the user is prompted to specify the layout of the report. The dialog box shown in Fig. 3.18.7, enables the user to determine the basic layout of the data. The user can also specify the orientation of the report as portrait or landscape.

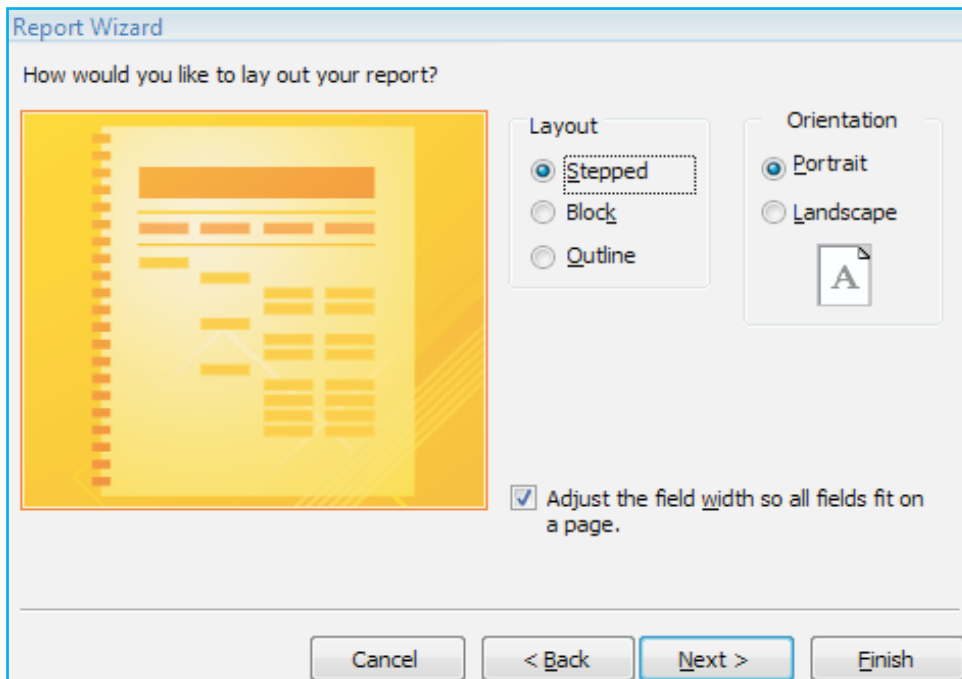


Fig. 3.18.7: Report Wizard (v)

- The user is further prompted to specify the title of the report as shown in Fig. 3.18.8:

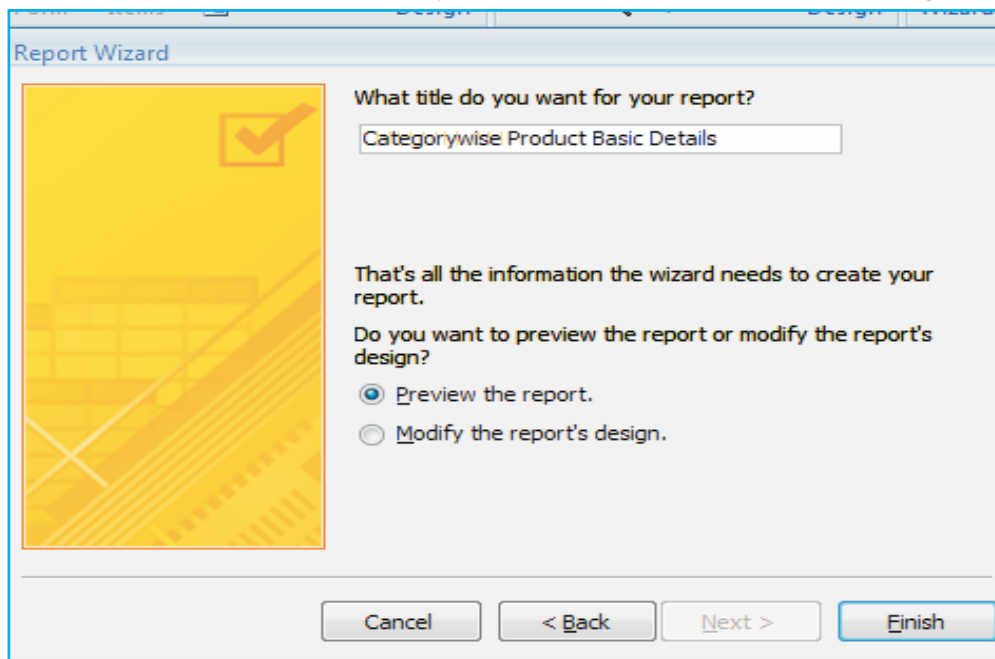


Fig. 3.18.8: Report Wizard (vi)

10. The user may click finish to see a report as shown in Fig. 3.18.9 (A).

Categorywise Product Basic Details

Category	Product Name	Product Code	Standard Cost	List Price	Quantity Per Unit
Baked Goods & Mixes					
	Northwind Traders Brownie Mix	NWTBGM-85	\$9.00	\$12.49	3 boxes
	Northwind Traders Cake Mix	NWTBGM-86	\$10.50	\$15.99	4 boxes
	Northwind Traders Chocolate Bis	NWTBGM-19	\$6.90	\$9.20	10 boxes x 12 pieces
	Northwind Traders Scones	NWTBGM-21	\$7.50	\$10.00	24 pkgs. x 4 pieces
Summary for 'Category' = Baked Goods & Mixes (4 detail records)					
Min				\$6.90	\$9.20
Max				\$10.50	\$15.99
Beverages					
	Northwind Traders Beer	NWTB-34	\$10.50	\$14.00	24 - 12 oz bottles
	Northwind Traders Chai	NWTB-1	\$13.50	\$18.00	10 boxes x 20 bags
	Northwind Traders Coffee	NWTB-43	\$34.50	\$46.00	16 - 500 g tins
	Northwind Traders Green Tea	NWTB-81	\$2.00	\$2.99	20 bags per box
	Northwind Traders Tea	NWTB-87	\$2.00	\$4.00	100 count per box
Summary for 'Category' = Beverages (5 detail records)					
Min				\$2.00	\$2.99
Max				\$34.50	\$46.00
Candy					

Fig. 3.18.9 (A): Report Resultant

Fig 3.18.9 (B) shows the report in report design view, from where the user can further customize the report.

The screenshot displays the Microsoft Access interface with the 'Categorywise Product Basic Details' report in design view. The report is structured as follows:

- Report Header:** Contains the title 'Categorywise Product Basic Details'.
- Page Header:** Contains a table with columns: Category, Product Name, Product Code, Standard Cost, List Price, and Quantity Per Unit.
- Category Header:** Contains a table with columns: Category, Product Name, Product Code, Standard Cost, List Price, and Quantity Per Unit.
- Detail:** Contains a table with columns: Category, Product Name, Product Code, Standard Cost, List Price, and Quantity Per Unit.
- Category Footer:** Contains a table with columns: Category, Product Name, Product Code, Standard Cost, List Price, and Quantity Per Unit.
- Page Footer:** Contains a table with columns: Category, Product Name, Product Code, Standard Cost, List Price, and Quantity Per Unit.
- Report Footer:** Contains a table with columns: Category, Product Name, Product Code, Standard Cost, List Price, and Quantity Per Unit.

The report is displayed in a window titled 'Categorywise Product Basic Details' with a standard Windows XP interface. The Windows logo and 'Windows Vista' text are visible in the bottom right corner.

Fig. 3.18.9 (B): Report Design View



3.19 CREATING REPORT IN REPORT DESIGN VIEW

The report design view gives the user a high level of control over the controls on a report, as well as the report itself. The report design view is divided into a number of sections. The following Access sections are available in report design view:

Report Header Section

Controls in the Report Header section are printed only once at the beginning of the report. A common use of a Report Header section is as a cover page or a cover letter or for information that needs to be communicated only once to the user of the report. The user can also have controls in the Report Header section print on a separate page, which enables them to create a title page and include a graphic or picture in the section. The Force New Page property in the Report Header section needs to be set to After Section that will place the information in the report header into a separate page.

Page Header section

Text or text box controls in the Page Header section normally print at the top of every page. If a report header on the first page is not on a page of its own, the information in the Page Header section prints just below the report header information on the first page. Page headers serve as column headers in group/total reports; they can also contain a title for the report. The user can also change special effects such as color, shading, borders, line thickness, font type and font size for each control. Both the Page Header and Page Footer sections can be set to one of four settings through the Report's properties as follows:

- **All Pages:** The page header and page footer print on every page.
- **Not with Report Header:** Neither the page header nor footer prints on a page with the report header.
- **Not with Report Footer:** The page header does not print with the report footer. The report footer prints on a new page.
- **Not with Report Header/Footer:** Neither the page header nor the footer prints on a page with the report header or footer.

Group Header Section

A **Group Header** section normally displays the name of the group. Access knows when all the records in a group have been displayed in a Detail section when the group name changes. It is possible to have multiple levels of group headers and footers. However, in some reports the user ou might have groups of information with date values. The user could group these sections by year or month and year and within those sections by another group such as category.

Detail Section

The **Detail section** processes every record in the data and is where each value is printed. The Detail section frequently contains a calculated field such as profit that is the result of a mathematical expression.

Group Footer Section

The **Group Footer** section is used to calculate summaries for all the detail records in a group. The value of this text box control is automatically reset to 0 every time the group changes.



Page Footer Section

The **Page Footer** section usually contains page numbers or control totals. In very large reports, such as when there are multiple pages of detail records with no summaries, the user may want page totals as well as group totals. The user can also print the date and the time printed. A text box control with the following expression in the Control Source property can be used to display page number information that keeps track of the page number in the report:

= "Page: " & [Page] & " of " & [Pages]

Report Footer Section

The **Report Footer** section is printed once at the end of the report after all the detail records and group footer sections are printed. Report footers typically display grand totals or other statistics such as averages, percentages, etc for the entire report. The Report Writer in Access is a two-pass report writer, capable of preprocessing all records to calculate the totals such as percentages needed for statistical reporting. This capability enables the user to create expressions that calculate percentages as Access processes those records that require foreknowledge of the grand total.

1. Click the **Report Design** button of **Reports Section** under **Create** tab as shown in Fig. 3.19.1.



Fig. 3.19.1: 'Report Design' Option

2. Click **Add Existing Fields** button on **Tools** section and select the desired table(s) / query(s) as shown in Fig. 3.19.2.

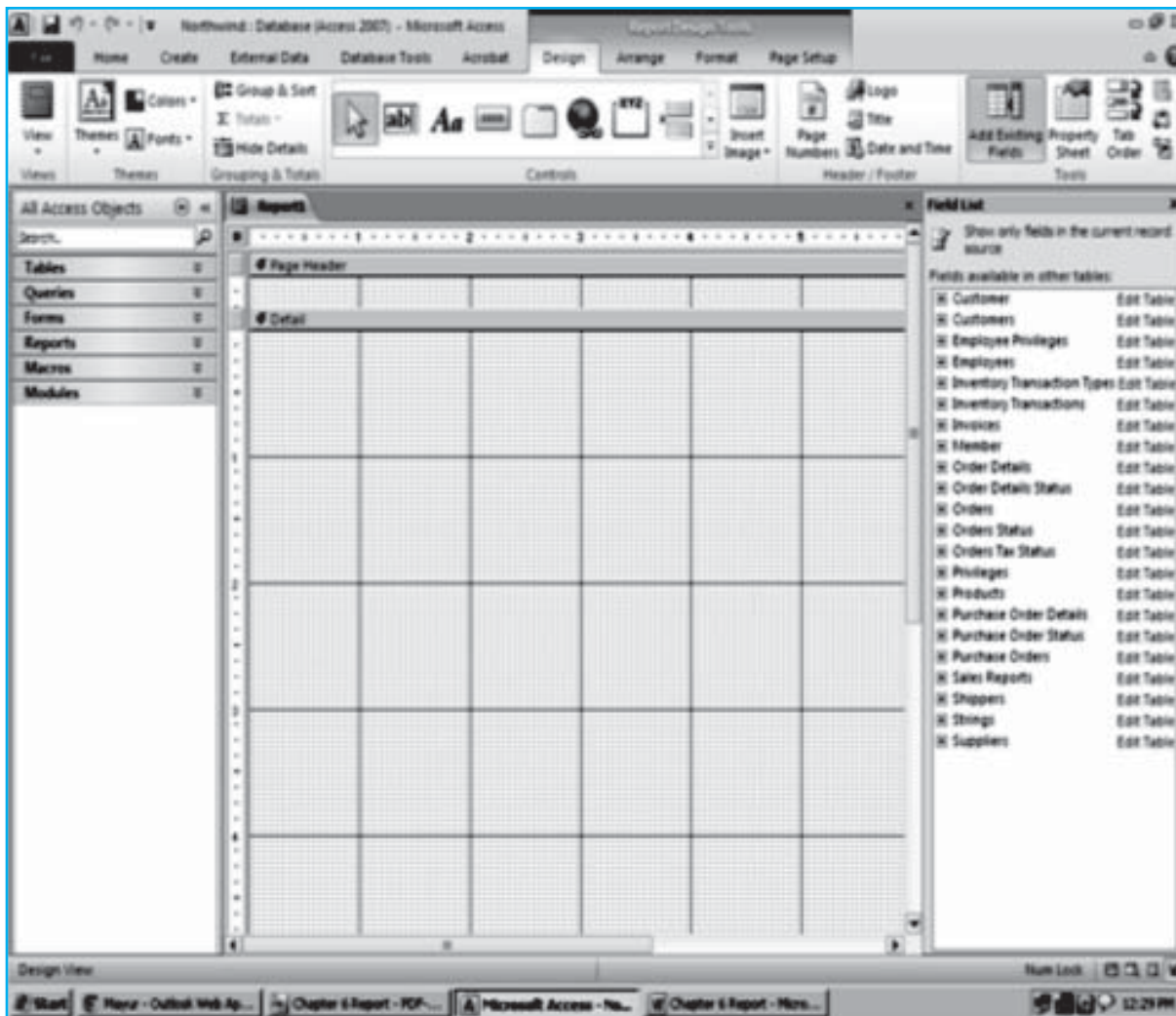


Fig. 3.19.2: Report Design Tools (i)

Note: By default the report contains only Page Header, Details and Page Footer section in the report design view. IF the user wishes to have Report Header and Report Footer section then he or she may right click inside the report design window and include Report Header / Report Footer.

3. The user may specify the Report Title as shown in Fig. 3.19.3.

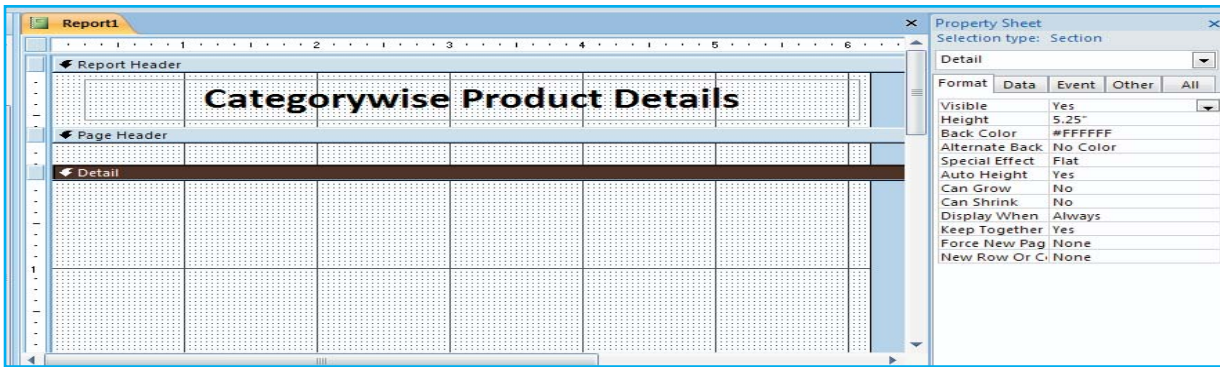


Fig. 3.19.3: Report Design Tools (ii)

4. Right Click on the report **design view** to select **Sorting and Grouping** as shown in Fig. 3.19.4.

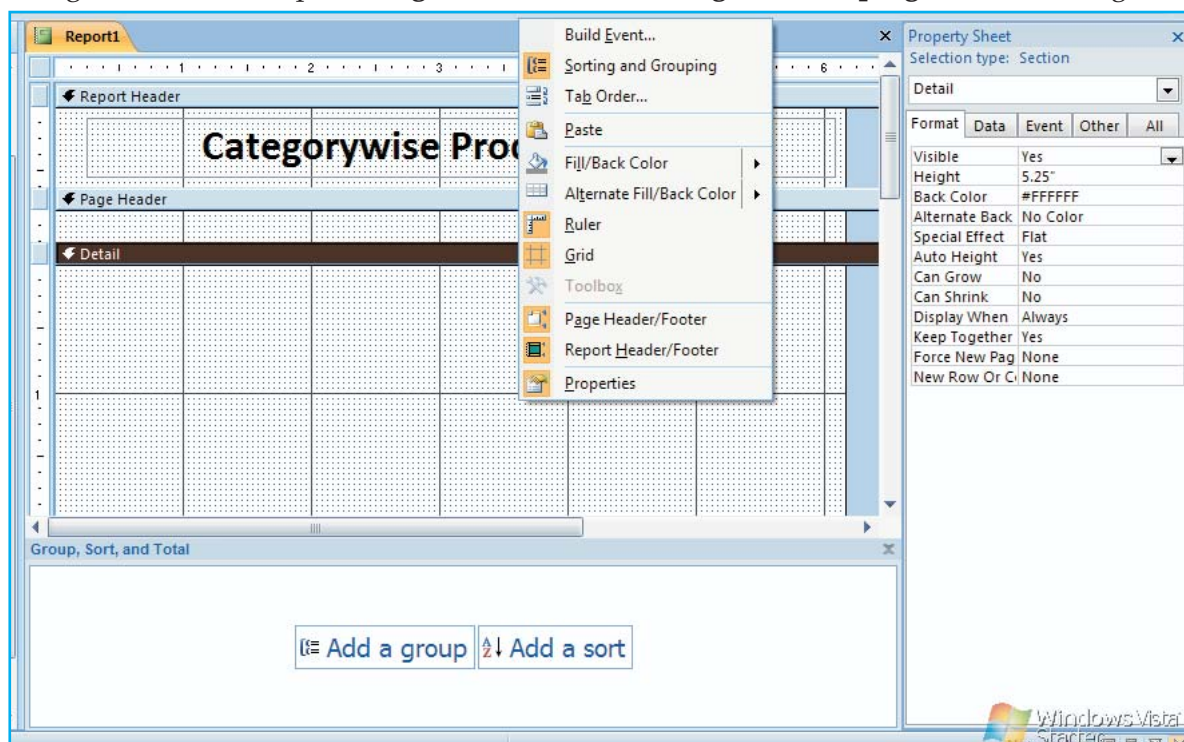


Fig. 3.19.4: Report Design Tools (iii)

5. Click the **Add a group** button shown in Fig. 3.19.4.

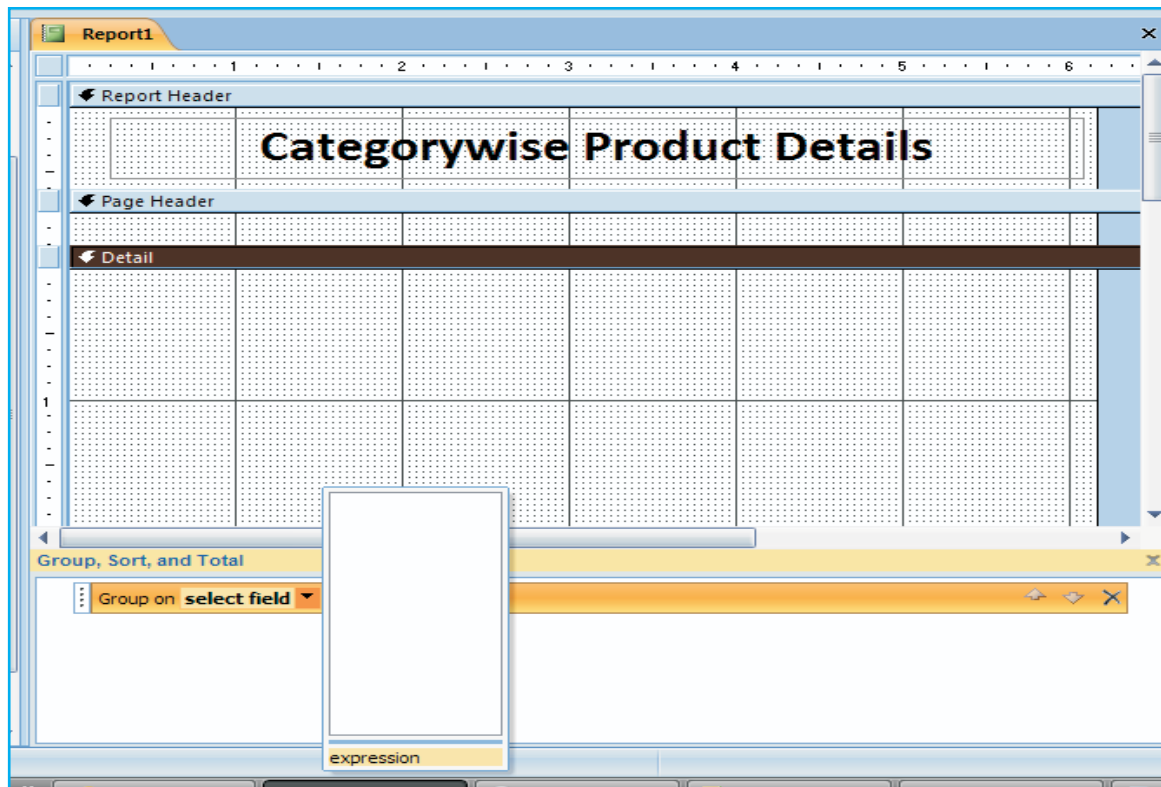


Fig. 3.19.5: Report Design Tools (iv)

6. Either select from the list of available fields in the select field dropdown box of **Group** on as shown in Fig. 3.19.5 or Click the expression option as shown in Fig. 3.19.5.
7. For grouping, specify the table to be used and select the field from the list of available fields as shown Fig. 3.19.6.

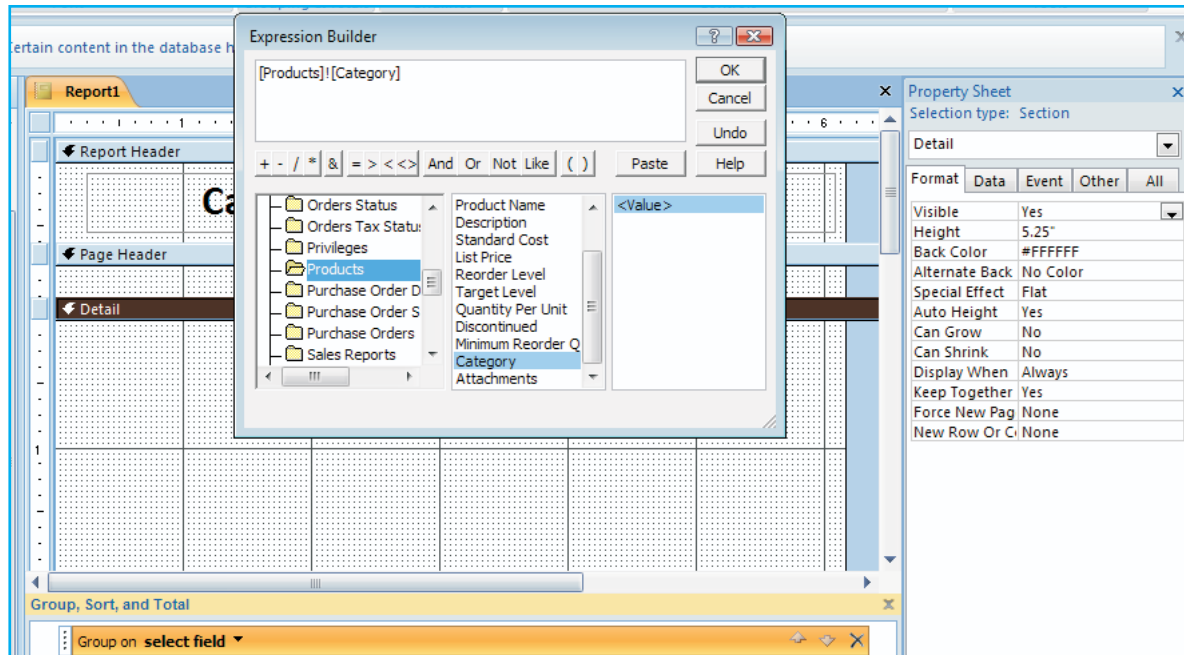


Fig. 3.19.6: Report Design Tools (v)

8. Click **OK** to return to the report design. A separate section as **Group Header** Section will appear as shown in Fig. 3.19.8.

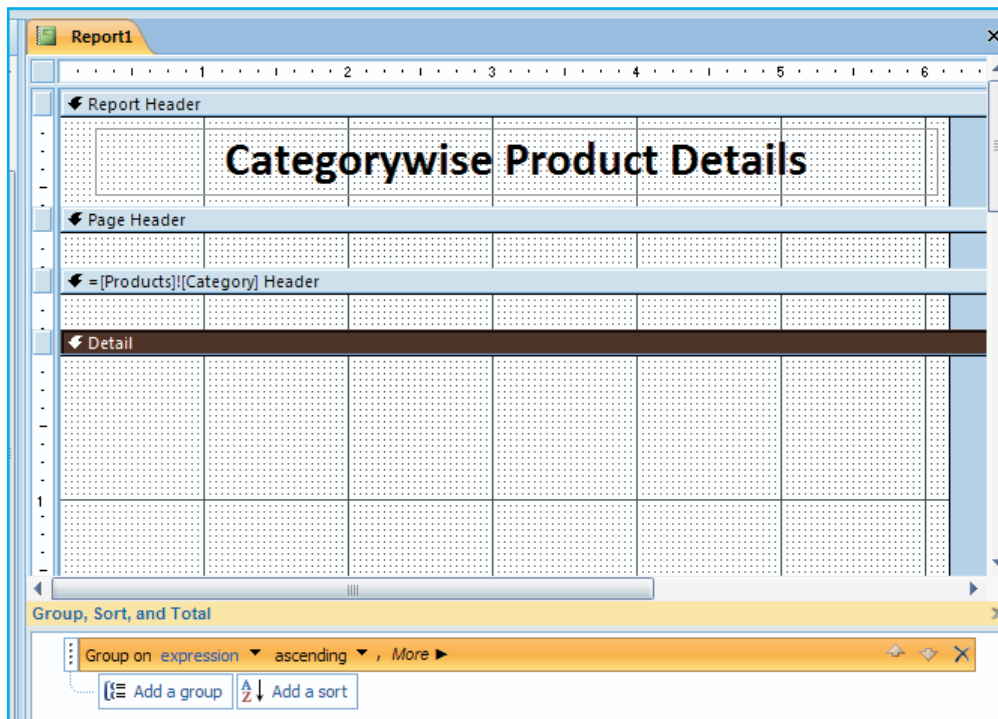


Fig. 3.19.7: Report Design Tools (vi)



9. Now, from the field list pane select the table and put the fields as shown in Fig. 3.19.8.

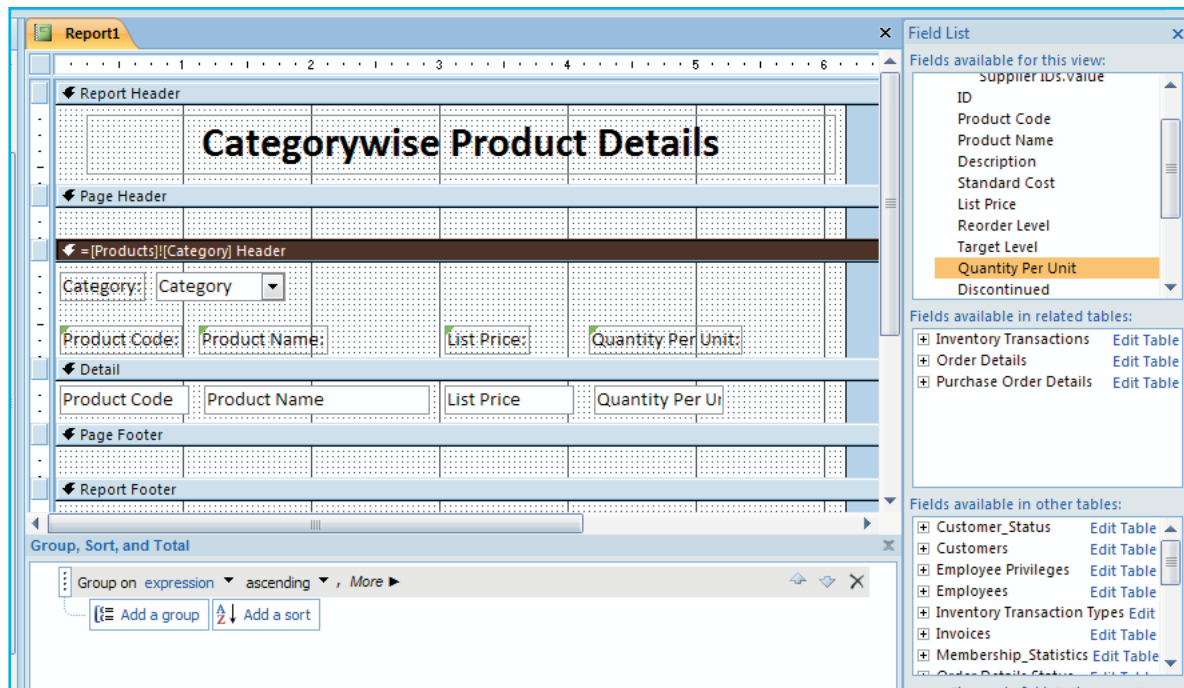


Fig. 3.19.8: Report Design Tools (vii)

10. Further, as per Fig. 3.19.9, the user can attach totals.

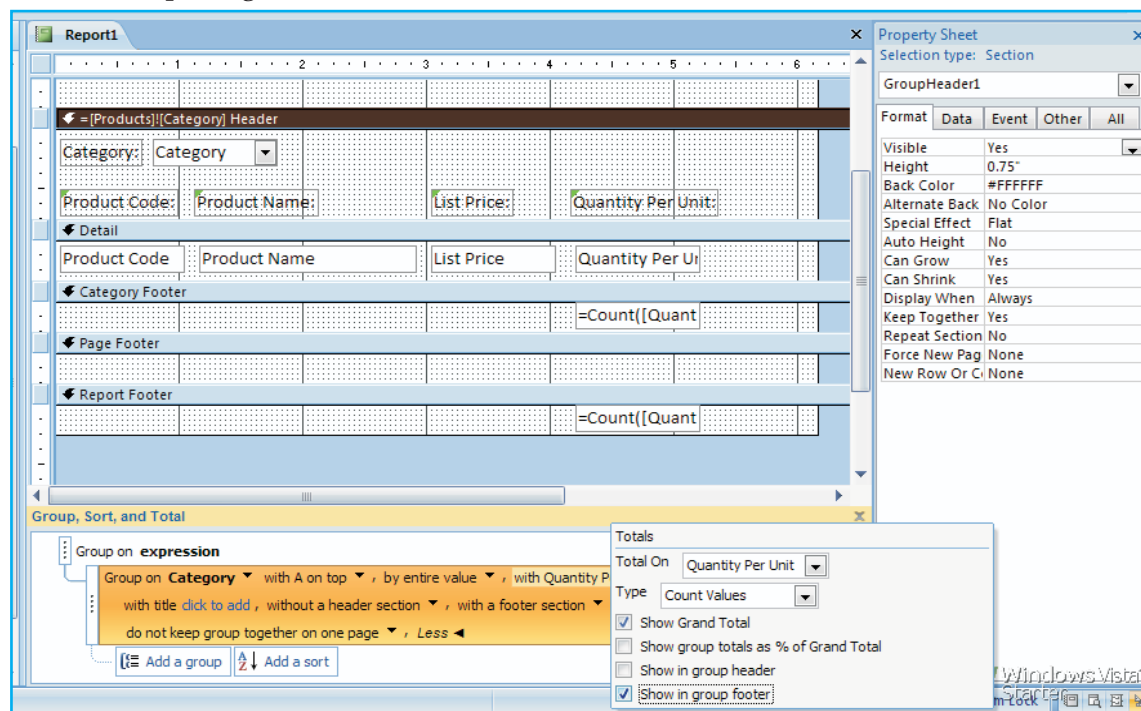


Fig. 3.19.9: Report Design Tools (viii)



11. The resultant output that will generated as a report will be similar to the one shown in Fig. 3.10.10.

Categorywise Product Details			
Category: Baked Goods &			
Product Code:	Product Name:	List Price:	Quantity Per Unit:
NWTBGM-86	Northwind Traders Cake M	\$15.99	4 boxes
NWTBGM-21	Northwind Traders Scones	\$10.00	24 pkgs. x 4 pie
NWTBGM-19	Northwind Traders Chocola	\$9.20	10 boxes x 12 p
NWTBGM-85	Northwind Traders Browni	\$12.49	3 boxes
4			
Category: Beverages			
Product Code:	Product Name:	List Price:	Quantity Per Unit:
NWTB-81	Northwind Traders Green T	\$2.99	20 bags per box
NWTB-43	Northwind Traders Coffee	\$46.00	16 - 500 g tins
NWTB-34	Northwind Traders Beer	\$14.00	24 - 12 oz bottle
NWTB-1	Northwind Traders Chai	\$18.00	10 boxes x 20 b

Fig. 3.19.10: Resultant Report

3.20 SAVING AND PRINTING A REPORT

The process of saving and printing a report is similar to that of a normal document under windows environment.