Onlinevarsity
your e-way to learning

Managing Data - The Microsoft Access Way

Are you □□□□□□□□□□ with Onlinevarsity.com?	R G E T R E E S I D
Did you □□□□□□□□ this ebook?	D O O W L N A D
Do you have a □□□□□ copy of this ebook?	L G A E L
Are you a victim of □□□□□□?	P C I A R Y

Answers: REGISTERED, DOWNLOAD, LEGAL, PIRACY

Download a **LEGAL** copy of this ebook which you can say is **mine**

because **PIRACY** is a **crime**

Managing Data - The Microsoft Access Way
Learner's Guide

APTECH LIMITED
Contact E-mail: ov-support@onlinevarsity.com

Edition 2 - 2013

Aptech®
Unleash your potential

Aptech™
COMPUTER EDUCATION
Unleash your potential

Dear Learner,

We congratulate you on your decision to pursue an Aptech course.

Aptech Ltd. designs its courses using a sound instructional design model – from conceptualization to execution, incorporating the following key aspects:

- Scanning the user system and needs assessment

Needs assessment is carried out to find the educational and training needs of the learner

Technology trends are regularly scanned and tracked by core teams at Aptech Ltd. TAG* analyzes these on a monthly basis to understand the emerging technology training needs for the Industry.

An annual Industry Recruitment Profile Survey# is conducted during August - October to understand the technologies that Industries would be adapting in the next 2 to 3 years. An analysis of these trends & recruitment needs is then carried out to understand the skill requirements for different roles & career opportunities.

The skill requirements are then mapped with the learner profile (user system) to derive the Learning objectives for the different roles.

- Needs analysis and design of curriculum

The Learning objectives are then analyzed and translated into learning tasks. Each learning task or activity is analyzed in terms of knowledge, skills and attitudes that are required to perform that task. Teachers and domain experts do this jointly. These are then grouped in clusters to form the subjects to be covered by the curriculum.

In addition, the society, the teachers, and the industry expect certain knowledge and skills that are related to abilities such as *learning-to-learn, thinking, adaptability, problem solving, positive attitude etc*. These competencies would cover both cognitive and affective domains.

A precedence diagram for the subjects is drawn where the prerequisites for each subject are graphically illustrated. The number of levels in this diagram is determined by the duration of the course in terms of number of semesters etc. Using the precedence diagram and the time duration for each subject, the curriculum is organized.

- Design & development of instructional materials

The content outlines are developed by including additional topics that are required for the completion of the domain and for the logical development of the competencies identified. Evaluation strategy and scheme is developed for the subject. The topics are arranged/organized in a meaningful sequence.

The detailed instructional material – Training aids, Learner material, reference material, project guidelines, etc.- are then developed. Rigorous quality checks are conducted at every stage.

- Strategies for delivery of instruction

  Careful consideration is given for the integral development of abilities like thinking, problem solving, learning-to-learn etc. by selecting appropriate instructional strategies (training methodology), instructional activities and instructional materials.

  The area of IT is fast changing and nebulous. Hence considerable flexibility is provided in the instructional process by specially including creative activities with group interaction between the students and the trainer. The positive aspects of Web based learning –acquiring information, organizing information and acting on the basis of insufficient information are some of the aspects, which are incorporated, in the instructional process.

- Assessment of learning

  The learning is assessed through different modes – tests, assignments & projects. The assessment system is designed to evaluate the level of knowledge & skills as defined by the learning objectives.

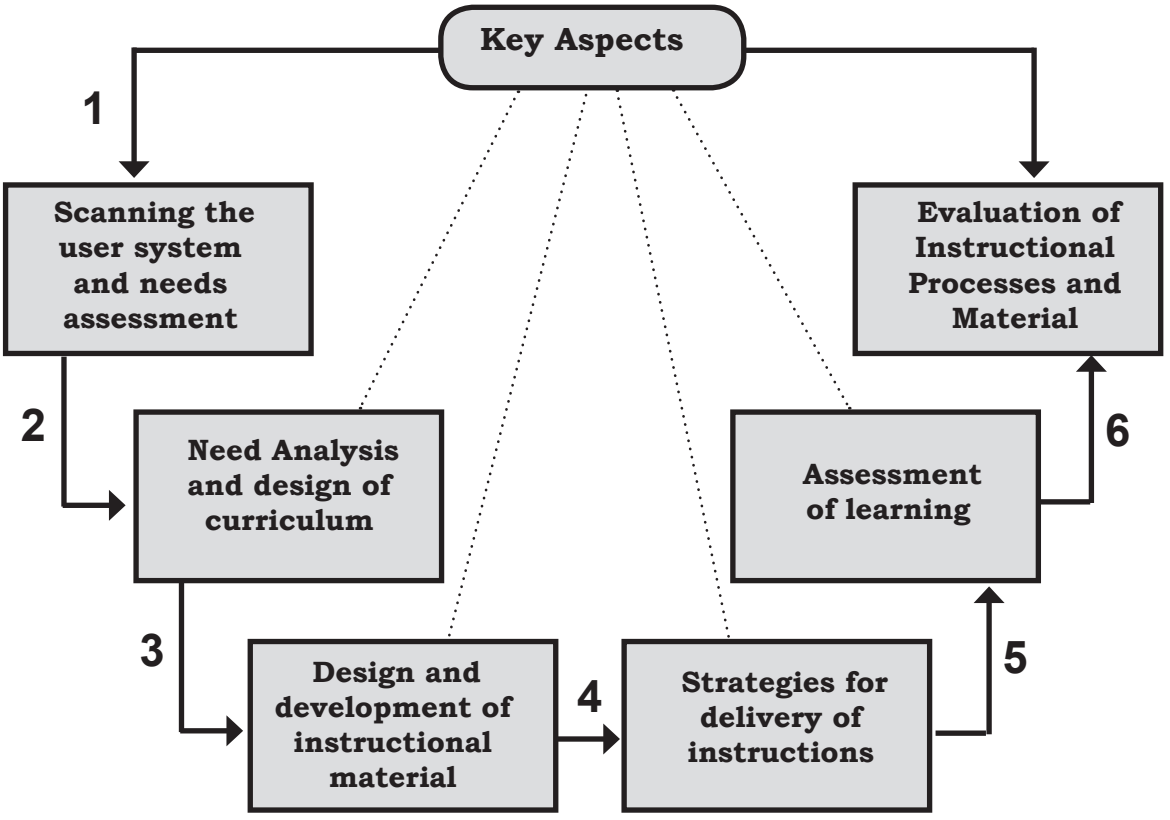
- Evaluation of instructional process and instructional materials

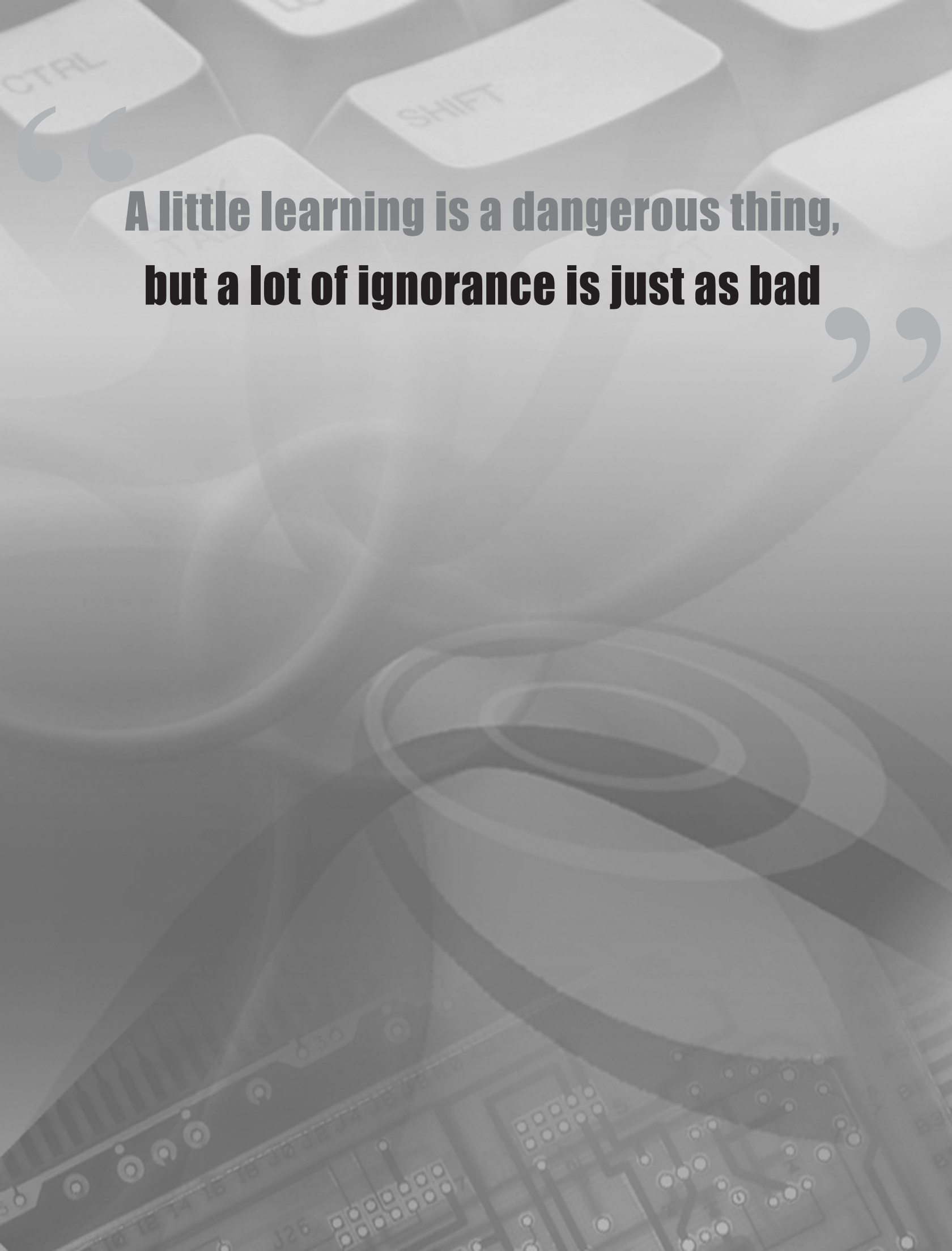
  The instructional process is backed by an elaborate monitoring system to evaluate - on-time delivery, understanding of a subject module, ability of the instructor to impart learning. As an integral part of this process, we request you to kindly send us your feedback in the reply pre-paid form appended at the end of each module.

*TAG – Technology & Academics Group comprises of members from Aptech Ltd., professors from reputed Academic Institutions, Senior Managers from Industry, Technical gurus from Software Majors & representatives from regulatory organizations/forums.

Technology heads of Aptech Ltd. meet on a monthly basis to share and evaluate the technology trends. The group interfaces with the representatives of the TAG thrice a year to review and validate the technology and academic directions and endeavors of Aptech Ltd.

Aptech New Products Design Model




A little learning is a dangerous thing,
but a lot of ignorance is just as bad

Preface

A database refers to an organized collection of useful information. Traditionally, data in a database was stored by using various forms of files and folders. This method of storing and managing data was very time-consuming, tedious, and prone to human errors. Thus, experts soon realized the need for database management and automating the processes of data storage and data manipulation. Thus, database management plays an important role in the modern computing era.

A Database Management System (DBMS) is software that allows creating and managing databases for storing, managing, and retrieving data. Since the tables in DBMS are not related, there is no consistency between data contained in tables. To overcome this limitation, E.F.Codd introduced the concept of Relational Database Management System (RDBMS). An RDBMS is a DBMS that allows presenting data in the form of tables and relating various tables within a database with one another to ensure data consistency. Today, there are varieties of RDBMS software products released by different manufacturers in the market. One among them is Microsoft Access. Access 2010 is the latest version of this software. An Access 2010 database can contain multiple objects such as tables, queries, forms, and reports.

The book begins with an introduction to Access 2010 and covers features such as database creation and management as well as creation of tables and queries. The book also covers creation of advanced forms and reports. Finally, the book concludes with creation of macros, importing and exporting of data, implementing database security, and publishing a database on the Web.

The knowledge and information in this book is the result of the concentrated effort of the Design Team, which is continuously striving to bring to you the latest, the best and the most relevant subject matter in Information Technology. As a part of Aptech's quality drive, this team does intensive research and curriculum enrichment to keep it in line with industry trends and learner requirements.

We will be glad to receive your suggestions.

Design Team

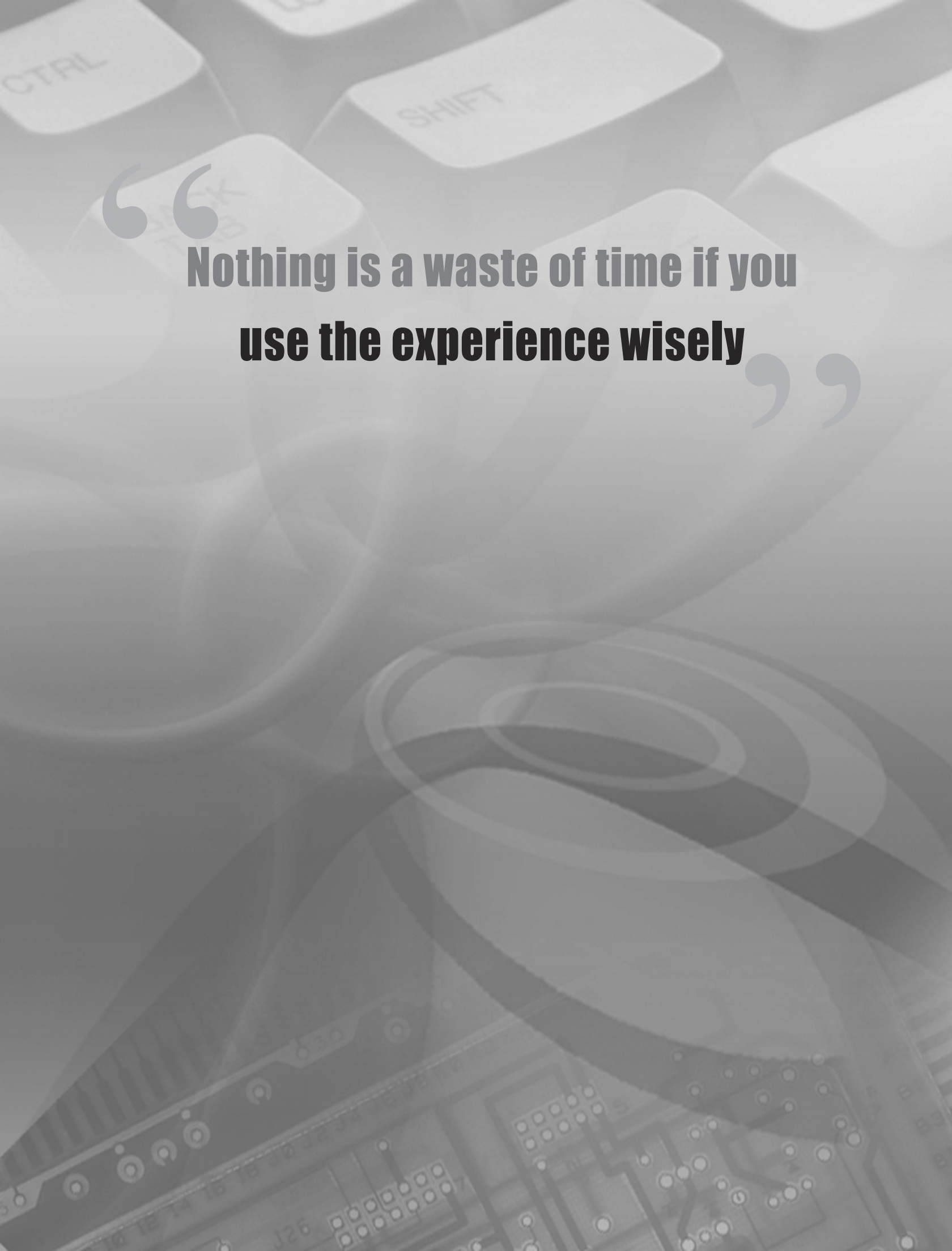

“Nothing is a waste of time if you
use the experience wisely”

Table of Contents

Sessions

“It is hard to fall, but it is worse never to have tried to succeed.”

Session - 1
Introduction to Microsoft Access 2010

Welcome to the Session, **Introduction to Microsoft Access 2010**.

This session explains the concepts of databases and their different types. It also describes a Relational Database Management System (RDBMS) and compares it with a Database Management System (DBMS). Then, the session describes the features of Access 2010 which is a powerful database tool for managing databases. Finally, the session explains creation of database in Access 2010 and introduces Application Parts which are templates used for performing common tasks.

In this Session, you will learn to:

- Define and describe a database
- Explain the need and purpose of a database management system
- Explain DBMS and RDBMS
- Explain the benefits of Access 2010
- Explain the User Interface (UI) of Access 2010
- List and describe the different approaches to create a database
- Explain Application Parts

1.1 Introduction

A database can be defined as an organized collection of data. Over the years, the concept of a database has evolved to manage large amount of data effectively in a database system.

Consider a scenario of a company that wants to maintain the personal and professional details of their employees in a payroll management system. The information should be stored such that it can be easily accessed, managed, and retrieved. In such a case, the company can store data by creating a database. A database represents a data structure used to store, access, and organize interrelated data. A database generally consists of one or more tables. A table is a collection of data arranged as a two-dimensional list.

A telephone directory is analogous to a database. A telephone directory consists of a list of records, with each record containing a name, address, and telephone number as shown in figure 1.1. A database file that is stored in a computer system is similar to this.

Serial Number	Name	Address	Telephone Number
1	John Smith	Hill Street	555-242-8283
2	Jill Martin	K Stadium	254-763-7876
3	Jane Rock	Lake Avenue	376-897-4533
4	David Aiden	Kate Street	908-765-5642

Figure 1.1: Telephone Directory

1.1.1 Types of Database

Data in a database can be stored in different ways depending upon the data model that is used. The data model is a structure that represents the way in which data is stored and accessed. There are different types of databases, each of which is differentiated by the data model used.

The different types of database models are as follows:

- Flat Model
- Hierarchical Model
- Network Model
- Relational Model

Figure 1.2 displays some of the data models.

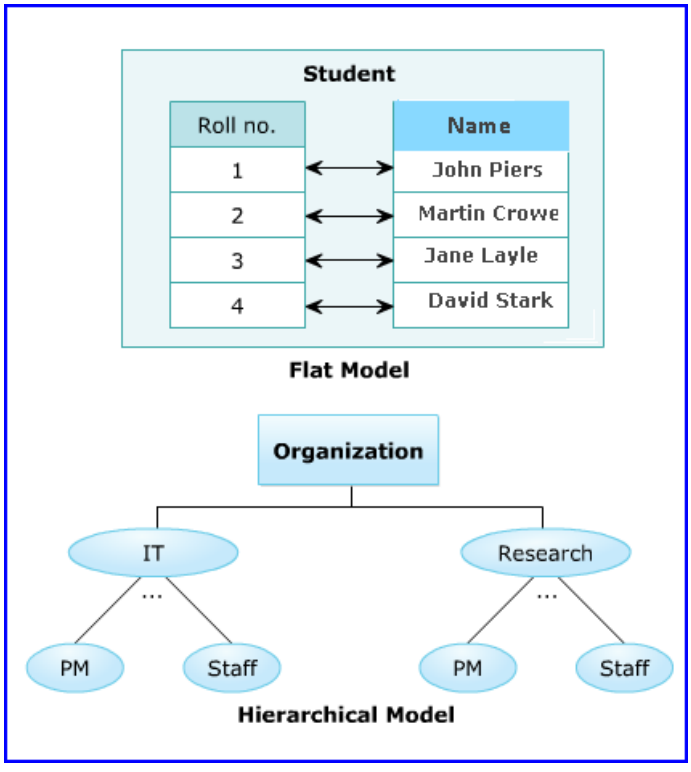


Figure 1.2: Examples of Data Models

The database models are explained in detail as follows:

➔ **Flat Model**

Represents data in two-dimensional format. In this model, data is stored in flat files, such as text files and is accessed sequentially.

➔ **Hierarchical Model**

Represents data in a tree-like structure. This is similar to the file structure in **Windows Explorer**. In this model, each parent node can have multiple child nodes, but a child node can have only one parent node. For example, each organization has multiple departments, such as IT, Research, and Accounts. Each department has a Project Manager and team members.

➔ **Network Model**

Expands the hierarchical model by providing relationships between multiple records. It includes multiple records, where each record is capable of including multiple parents and child nodes. For example, consider a medical database consisting of records for doctors, patients, and nurses. Each patient is associated with doctors as well as nurses. Similarly, one doctor can treat multiple patients.

➔ **Relational Model**

Represents data in the form of tables. You can associate relationships between the fields contained in the tables.

1.2 Need for a Database Management System (DBMS)

Consider a savings bank file processing system supported by a specific operating system. In such a file processing system, the customer records and savings account details are stored in permanent system files. Application programs are written to add an account, deposit to/withdraw from an account, find the balance, and so on. As the system develops, new application programs are written as and when required and new files are created. Over a period of time, the files may exist in different formats, as the programming language varies.

This can lead to several problems in file-based approach, which are as follows:

➔ **Data Redundancy and Inconsistency** – Same information is present at multiple locations which leads to duplication (redundancy). This data needs to be updated regularly. However, there is a chance that data may not be correctly updated at all locations.

➔ **Access Difficulty** – For any unusual request, new programs may need to be written for generating the required data.

➔ **Data Isolation** – Data exists in different files and formats due to which it becomes difficult to write programs that can access the data in different formats.

➔ **Concurrency** – In a DBMS, multiple users can access the data. Hence, there is a chance that the data may not always be accurate when accessed by different users.

➔ **Security Issues** - It is difficult to enforce access to data according to the various roles through the application programs.

➔ **Data Integrity** – It is difficult to change or enforce constraints with file-processing approach.

➔ **Data Not Shareable** – There is a chance that the data might not be shared between users.

All these issues led to the development of DBMS.

1.2.1 DBMS

Database management is the process of organizing and maintaining databases so that the data is easily available to users. The software application that allows you to manage databases is called a DBMS. It is also known as a Database Manager. A DBMS allows you to add, delete, and retrieve large volumes of data from a database.

Some of the popular DBMS products include DB2 and FileMaker.

Figure 1.3 depicts the concept of a DBMS.

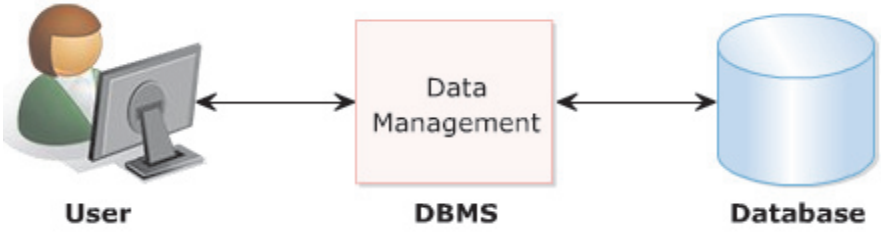


Figure 1.3: DBMS

A DBMS application performs various tasks for managing data to ensure reliability and security.

The different tasks are as follows:

➔ **Data Definition**

Allows creating, defining, and modifying the structure of a database.

➔ **Data Manipulation**

Allows performing various operations such as inserting, deleting, and modifying data.

➔ **Data Recovery and Concurrency**

Recovers the data that is lost due to system failure. In addition, it allows multiple users across the network to concurrently access data from the database.

➔ **Data Dictionary Maintenance**

Maintains a data dictionary which contains database information, such as available files, values contained within each file, and so on.

1.2.2 Advantages of a DBMS

A DBMS acts as an interface between a user and database. It provides several advantages in terms of integrity and administration of data. Some of the advantages of a DBMS are as follows:

➔ **Data Independence**

Allows accessing and using data independently without exposing the internal data structure. This means that the structure is not exposed to the application software that is accessing the database.

➔ **Efficient Data Access**

Allows accessing data that is stored on any external storage device with maximum efficiency.

➔ **Data Integrity and Security**

Ensures that the data in a database is valid, up-to-date, and consistent. This is achieved by accessing the data and making changes to the data at one location. In addition, it protects the data by providing access to data based on the credentials assigned to different users.

➔ **Data Administration**

Allows managing the data storage and retrieval for a centralized database which is accessed by multiple users to store and retrieve data.

➔ **Concurrent Access and Crash Recovery**

Allows multiple users to simultaneously use the database without impacting the performance. In addition, it protects users in case of any system failure.

A DBMS provides an efficient platform to manage and store data in a database. It allows multiple users to access and modify the database at the same time. This can lead to the loss of data integrity, as multiple users might be updating data at the same time. This also leads to security issues as unauthorized users might gain access and modify the database, as they will have access to all the files.

1.3 Relational Database Management System (RDBMS)

An RDBMS is a database management system that allows users to store data in the form of related tables. For example, users can relate a table containing student data (name and address) with a table containing academic data (subject and grades).

The concept of RDBMS was first introduced by Dr. E. F. Codd. It is one of the most widely used types of DBMS. The different RDBMS products that are available today are Microsoft SQL Server, Oracle, Microsoft Access, and so on. Some examples of RDBMS applications are airline reservations, banking applications, hospital management systems, and so on.

1.3.1 DBMS versus RDBMS

An RDBMS is an extension of DBMS. However, there are many differences between DBMS and RDBMS.

Table 1.1 lists the differences between DBMS and RDBMS.

DBMS	RDBMS
Does not support relationships between tables	Allows setting up relationships between tables within a database

Concepts

DBMS	RDBMS
Uses any of the database models including network, hierarchical, and so on	Uses the relational database model
Stores and retrieves limited amount of data	Stores and retrieves large amount of data
Allows working with databases containing relatively less records	Allows working with large databases

Table 1.1: Differences Between DBMS and RDBMS

1.4 Features and Benefits of Access 2010

Microsoft Access 2010 is an RDBMS. It enables the developers to collect large volumes of data and organize them into categories of related information. Thus, it allows to not only store, organize, and manage data, but also retrieve and present data in various formats and reports.

Access 2010 provides many new features and powerful tools that remain applicable as the data grows.

Some of the features of Access 2010 are as follows:

- **Build Database Faster and Easier** – Built-in templates and reusable components helps to create database easily and very fast. In other words, databases can be created using the templates or **Application Parts** with modular components and built-in components. For example, to create a database of events, a user can simply make use of the built-in Events template, with ready-to-use column names and structures.

- **Attractive Forms and Reports** – Access 2010 helps the developer to create informative and attractive forms and reports. Conditional formatting rules can be applied to display information that meets certain criteria. Conditional formatting supports the use of data bars.

- **Easy Access to Tools** – The **Ribbon** helps to access the most commonly used commands. Users can create custom tabs or customize the built-in tabs in Access 2010. **Backstage** view helps to manage the database from a centralized location.

- **Addition of Complex Expressions** – Expression builder has been enhanced to help the developer in writing formulas and expressions using IntelliSense. It also reduces errors.

- **Centralized Data** – Access 2010 can bring data from various sources together and thus, improves the quality of work. It allows users to import and link data from different external sources, such as MS Excel, MS Outlook, and so on.

- **Web Access** – Databases are available on the Web using the Access 2010 services on Microsoft SharePoint Server 2010. Thus, databases can be posted online and can be accessed, viewed, and edited online.

Web forms and reports can be accessed using a browser and the changes are synchronized with the SharePoint server.

➔ **Themes** – Access 2010 allows users to easily customize the appearance of a database by applying theme to it. It works in a similar fashion as in MS Word and MS PowerPoint. Themes help to maintain consistency in the look of the different objects.

➔ **Adding New Fields** – Access 2010 allows users to add common fields to the database by using the **Quick Start** fields. By default, the nine **Quick Start** fields that are provided by Access 2010 are namely, `Name`, `Category`, `Address`, `Payment Type`, `Phone`, `Priority`, `Start And End Dates`, `Status`, and `Tags`. Each of these **Quick Start** fields has preset field options that can be customized. For example, the `Payment Type` **Quick Start** field includes the other fields such as `Cash`, `Credit Card`, `Debit Card`, and so on. It can be customized by right-clicking the field and choosing **Edit List Items**.

1.4.1 New User Interface

The earlier version, MS Access 2007, had introduced a new User Interface (UI). The UI makes it easy for developers and users to find commands and features that were present in toolbars and complex menus. The three main components of the user interface are as follows:

➔ **Ribbon** – Present at the top of the program window as tabs that contains a group of commands.

➔ **Backstage View** – Displays a group of commands that appears on the **File** tab of the **Ribbon**.

➔ **Navigation Pane** – Appears on the left side of the program window and helps users to work with the database objects.

These three components provide an environment for the creation of databases. Access 2010 has enhanced this existing UI.

1.4.2 Ribbon

Access 2010 has the same user interface that is common to other Office 2007 and 2010 applications.

Menus and toolbars which existed in versions earlier to Access 2007, were been replaced by the **Ribbon** in Access 2007. There are five main tabs namely, **File**, **Home**, **Create**, **External Data**, and **Database Tools**. The tabs group commonly used commands based on their functionalities. The **Ribbon** can be hidden and restored by double-clicking the active command tab.

Figure 1.4 displays the **Home** tab present in Access 2010.

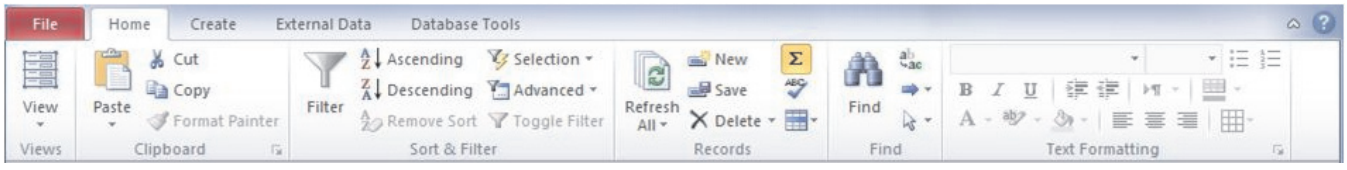


Figure 1.4: Home Tab

Access 2010 also provides contextual command tabs which contain commonly required commands for the object that the user is working with. For example, contextual tabs are displayed for the **Table Tools** when the user works with the data table. In other words, the contextual command tabs are displayed depending on the object the user is working with. It appears next to the standard command tab.

The **Quick Access** toolbar appears adjacent to the **Ribbon** and can be customized with common commands. The default commands displayed on the **Quick Access** toolbar are **Save**, **Undo**, and **Redo**. The placement of the toolbar can be modified and the size can also be adjusted.

Some of the commands in the **Ribbon** provide choices, whereas others launch a command.

The **Ribbon** also contains a type of control known as gallery. These controls show a preview of style rather than the commands, so that the user can view results before committing.

Figure 1.5 displays a gallery.

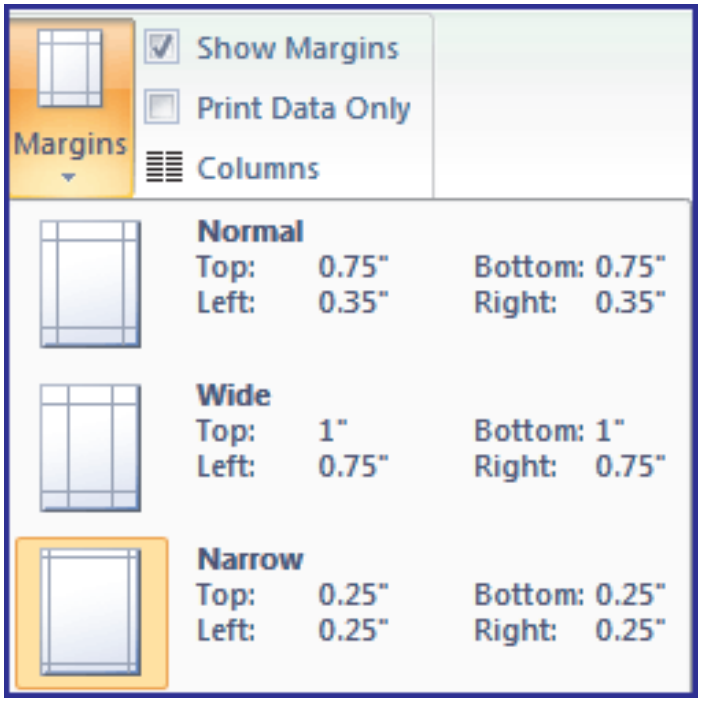


Figure 1.5: Gallery

Concepts

1.4.3 Backstage View

Access 2010, like other Office applications, contains a new feature, **Backstage** view. The **Backstage** view contains commands that can be applied to the entire database. In other words, it allows users to perform common file tasks from a central location. It contains commands, such as **Compact & Repair Database** and **Print**. These commands appear on the left side of the screen.

Figure 1.6 displays the **Backstage** view when the user clicks the **Info** command.

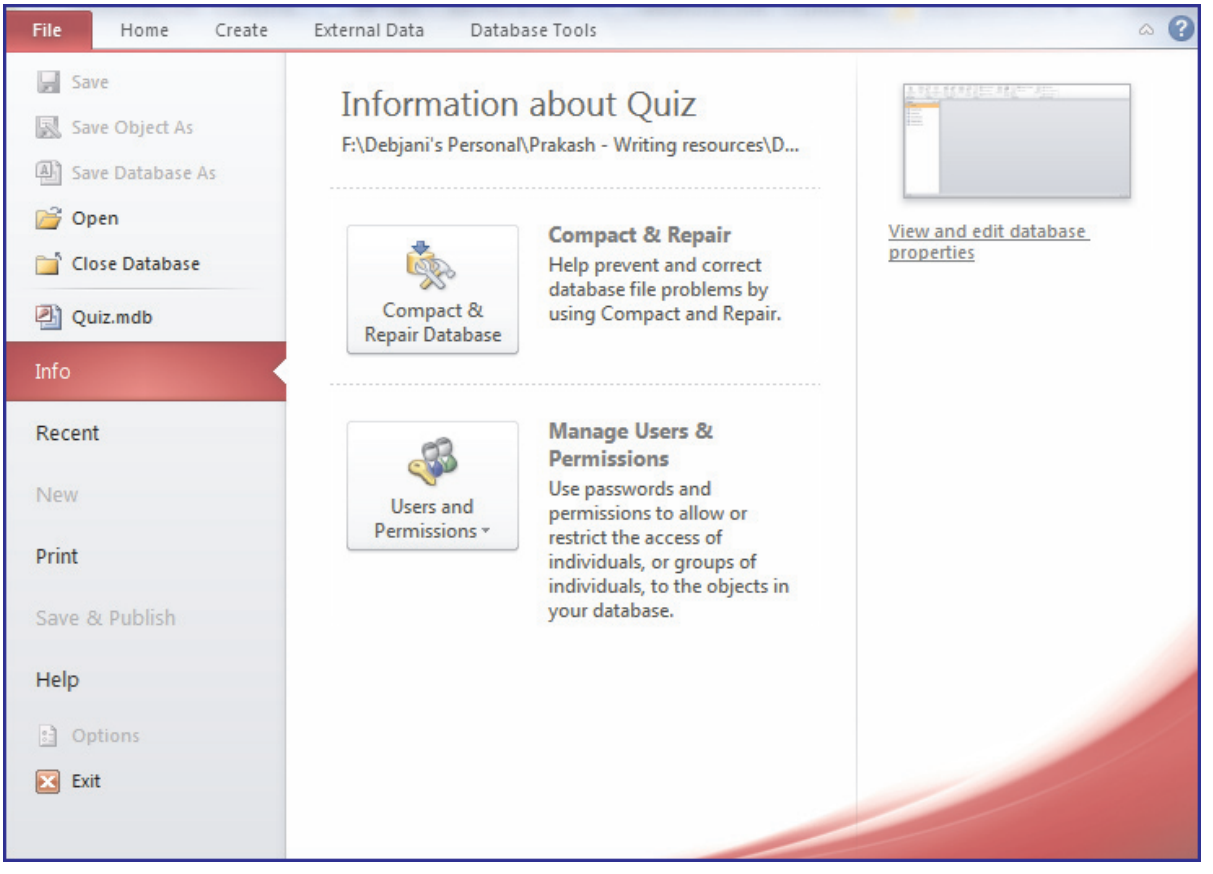


Figure 1.6: Backstage View

1.4.4 Navigation Pane

In earlier version of Access, a **Database** window existed, but from Access 2007 onwards, the Navigation Pane has replaced the **Database** window. The Navigation Pane lists and provides access to the objects of the current database. In other words, it helps to organize the database. It organizes by categories and groups, but users can create a custom categorization scheme. A newly created database uses the **Object Type** category which has various kinds of database objects.

The database objects include tables, forms, reports, pages, modules, and so on. An object can be opened by double-clicking the object or by selecting the object and pressing the **ENTER** key or by right-clicking the object and selecting **Open** from the context menu.

Concepts

1.5 Working with Databases

Access 2010 is a powerful database building tool that helps users to perform different operations with a database. It helps to build database for all kinds of business from small to medium-sized organization. The different operations that can be performed are discussed in detail.

1.5.1 Creating a Database

When Access 2010 is launched for the first time, the **Backstage** view is displayed. The **Backstage** view is used by a user to create, open, or close a database, as well as perform other operations such as customizing options.

There are different methods for creating a database that are as follows:

➔ **Create a new blank database**

To create a new blank database, perform the following steps:

1. Click **Start** → **All Programs** → **Microsoft Office** → **Microsoft Access 2010**. This will open Microsoft Access and the **Backstage** view will be displayed.
2. Click **File** and **New** from the left pane of the dialog box.
3. Select **Blank database** from the **Available Templates** pane.
4. Type a name for the database file in the **File Name** box present on the right under the **Blank Database**.
5. Click **Create**. The new database has been created and a new table named `Table1` is displayed in the **Datasheet View**.

Figure 1.7 displays the newly created database with a default table.

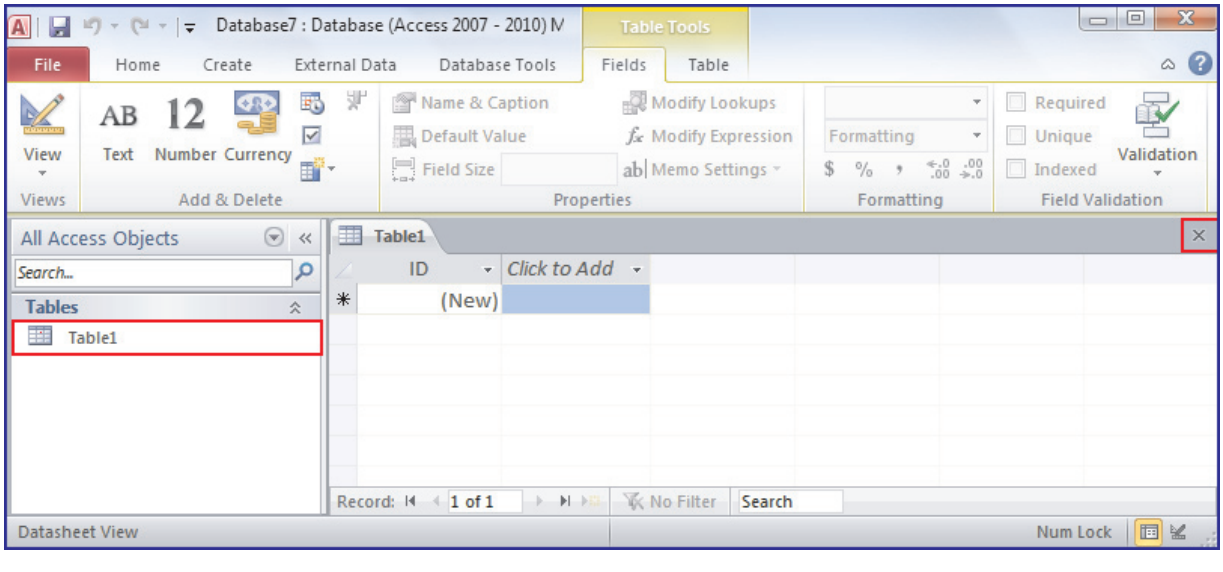


Figure 1.7: New Database with Default Table

Concepts

You can create or add data to the table. The table structure is created as the data is entered. Click the **Close** button, if you do not want to enter data in the table.

➔ Create a new database from templates

To create a new database from a template, perform the following steps:

1. Start **Access 2010** and display the **Backstage** view.
2. Click **New** from the left pane of the **Backstage** view.
3. Click **Sample templates** to display the **Available Templates**.

Figure 1.8 displays the **Available Templates** in the **Backstage** view.

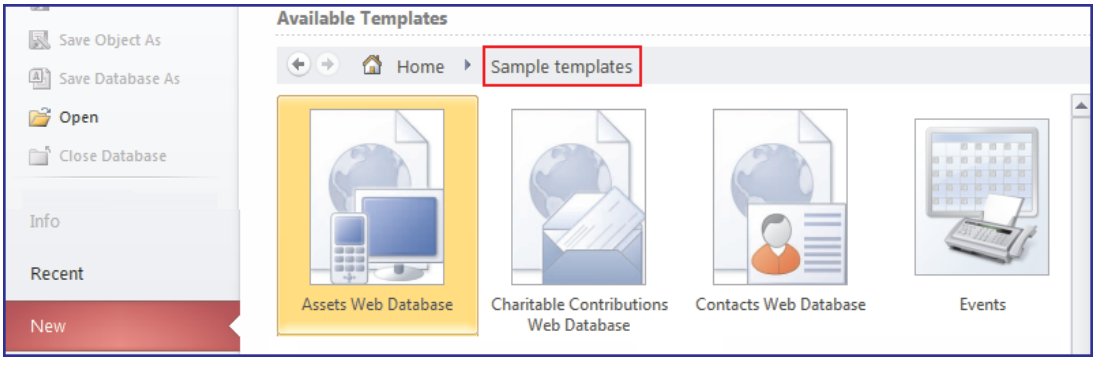


Figure 1.8: Available Templates

4. Select the desired templates.
5. Type the file name in the **File Name** box present on the right.
6. Click **Create**. The new database is created from the template and opened.

➔ Create a database from Office.com templates

To create a new database based on an **Office.com** template, perform the following steps:

1. Start **Access 2010** to display the **Backstage** view.
2. Click **New** from the left pane of the **Backstage** view.
3. Click a category under **Office.com Templates** to display the template.

Concepts

Figure 1.9 displays the categories under **Office.com Templates**.

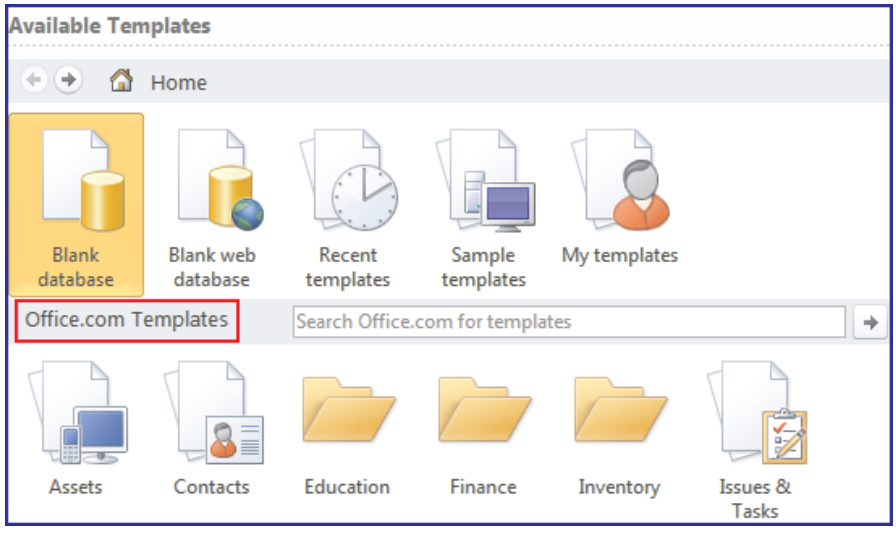


Figure 1.9: Office.com Templates

4. Select the desired template.
5. Type the file name in the **File Name** box present on the right.
6. Click **Download**. Access 2010 creates the database after downloading the required templates and stores it in the document folder.

1.5.2 Saving a Database

Each Access 2010 database can be considered as a folder storing the user's data as database objects. Each of these database objects are considered by Access 2010 as separate documents which are required to be saved and opened individually. You cannot save the whole database at one go, but can save objects stored within the database. There can be multiple objects, such as forms for entering data, queries for searching data, reports for analyzing data, and tables for storing data. Working with a database means working with many of these objects.

Trying to close the database will prompt the user to save any unsaved work. To save an object, you need to select the object and then, select **File** → **Save**. If the object is being saved for the first time, then the **Save As** dialog box is displayed to the user for typing the desired object name.

Note - To save an already saved object you can select **Save** command from the **Quick Access** toolbar or **Save** from the **Backstage** view.

Concepts

1.5.3 Opening and Closing a Database

Before data is entered or an object is modified, the database should be open.

To open a database, perform the following steps:

1. Click **File** tab to display the **Backstage** view.
2. Select **Open** to display the **Open** dialog box.
3. Locate and select the desired database.
4. Click **Open**. The database opens.

Note - Warning messages may appear when the database is opened. A yellow bar with a warning message is displayed below the **Ribbon**, if the database contains customized functions. Click **Enable Content** to display the database correctly. Click **Yes**, if a warning message stating 'Do you want to make this file a Trusted Document?' is displayed. This will enable all the objects.

To open a recently used database, perform the following steps:

1. Click **File** → **Recent** from the **Backstage** view.

   Figure 1.10 displays **Recent Databases** under the **Recent** tab.

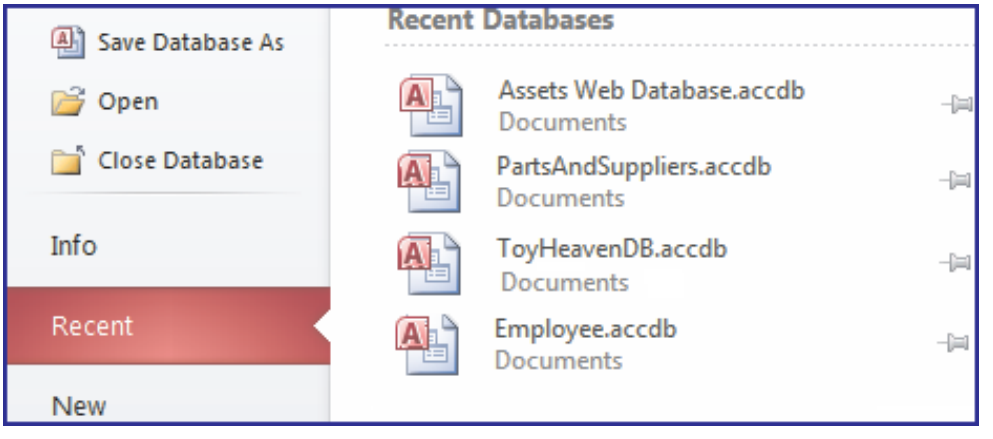


Figure 1.10: Recent Databases

2. Select the desired database to open the database.

To close a database, perform the following steps:

1. Click **File** → **Close Database** from the **Backstage** view.

   If any object is unsaved, a dialog box is displayed. Select **Yes** or **No** or **Cancel** to save the object, close without saving, or leave the database open respectively.

Concepts

1.6 Compact and Repair a Database

As database files grow in size, they can get corrupted or damaged. Access 2010 provides the **Compact & Repair Database** command to solve these issues. Compacting does not compress the data, but reduces the size of the database file by deleting the unwanted space.

The **Compact & Repair Database** command helps to prevent and correct the following problems:

➔ **Growing Database Files**

Database files increase in size as the design changes or data is added or updated. The increase in size of database files is mostly due to addition of data. There may be other reasons also. For example, to accomplish various tasks, Access 2010 creates many temporary objects which remain in the database, even after they are no longer required. When the database objects are deleted the disk space is not automatically reclaimed and the database file still uses that space.

This degrades the performance of the database and this is visible by files taking longer time to open and queries taking longer time to execute.

➔ **Corrupt Database Files**

A database file can get corrupted when it is shared over the Web and multiple users access it at the same time over the network. The risk increases especially when the `Memo` field is frequently edited. This type of corruption results in loss of forms or database design and can be mitigated using the **Compact & Repair Database**.

Splitting a database can prevent the corruption of database files and reduce the loss of data. This can be achieved by keeping the data in a separate file that are not directly accessed by the user.

Access 2010 will prompt the user to repair and open a corrupted database with the command, **Compact & Repair Database**. Depending on the success of repair of the corrupted file, an appropriate message is displayed. Users can determine what is required to be performed to recover from the backup.

1.7 Working with Objects

Each object in Access 2010 is treated as a separate document, which is required to be saved and opened individually in order to work with it.

The different activities that can be performed with an object are as follows:

➔ **Open an Object**

Before any modification or updation can be performed on an object, the object must be opened.

To open an object, perform the following steps:

1. Locate object in the **Navigation Pane**.

2. Double-click the desired object. It will appear as a new tab in the **Document Tabs** bar.

Figure 1.11 shows the selected objects in the **Document Tabs**.

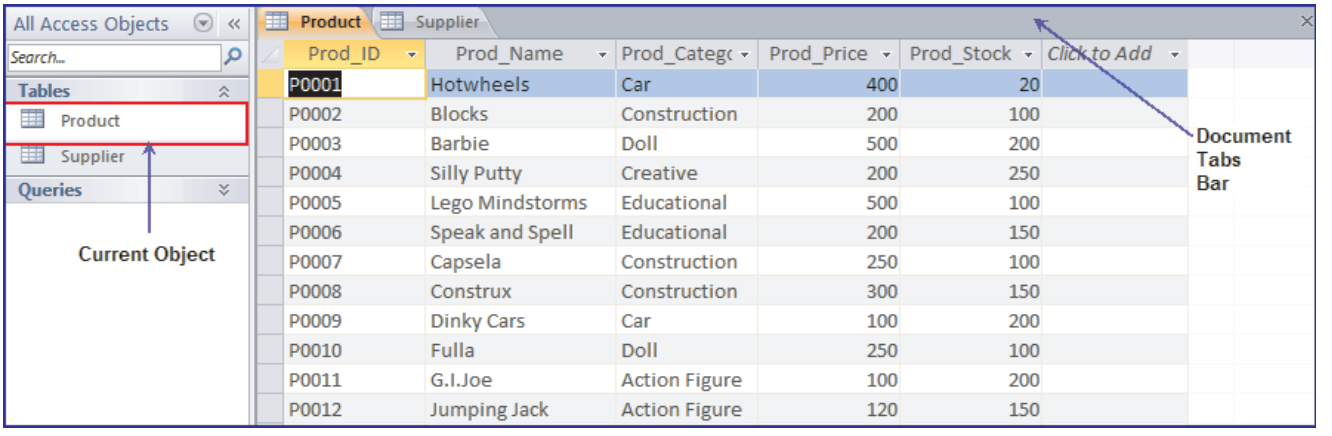


Figure 1.11: Document Tabs

Note - Most recently opened object will be displayed as the current object in the main window. Click the **Document Tabs** bar to view an open object.

➔ Save an Object

Access 2010 will prompt the user to save any unsaved objects before the database is closed. Saving work helps the user from losing any information.

To save an object, perform the following steps:

1. Select a object by clicking the **Document Tabs** bar.
2. Select **File** → **Save Object As** from the **Backstage** view to save the object. This displays the **Save As** dialog box.

Figure 1.12 displays the **Save As** dialog box.

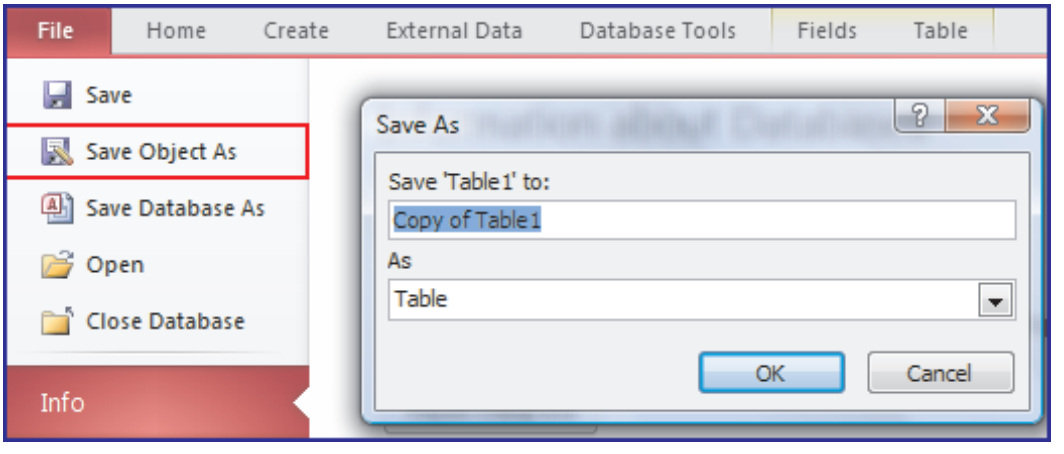


Figure 1.12: Save As Dialog Box

Concepts

3. Type the object name, if the object is being saved for the first time.
4. Click **OK**.

➔ **Close an Object**

1. Select the object from the **Document Tabs** bar.
2. Click the button present on the right of the **Document Tabs** bar to close the opened object.

Figure 1.13 displays the **Close** button on the **Document Tabs** bar.

Product | Supplier

Prod_ID	Prod_Name	Prod_Categ	Prod_Price	Prod_Stock	Click to Add
P0001	Hotwheels	Car	400	20	
P0002	Blocks	Construction	200	100	
P0003	Barbie	Doll	500	200	
P0004	Silly Putty	Creative	200	250	
P0005	Lego Mindstorms	Educational	500	100	
P0006	Speak and Spell	Educational	200	150	
P0007	Capsela	Construction	250	100	
P0008	Construx	Construction	300	150	
P0009	Dinky Cars	Car	100	200	
P0010	Fulla	Doll	250	100	
P0011	G.I.Joe	Action Figure	100	200	
P0012	Jumping Jack	Action Figure	120	150	

Figure 1.13: Close Button on Document Tabs

Access 2010 will prompt the user to save any unsaved document.

➔ **Rename an Object**

1. Right-click the object in the **Navigation Pane**.
2. Select **Rename**.

Concepts

Figure 1.14 displays the **Rename** option for an object.

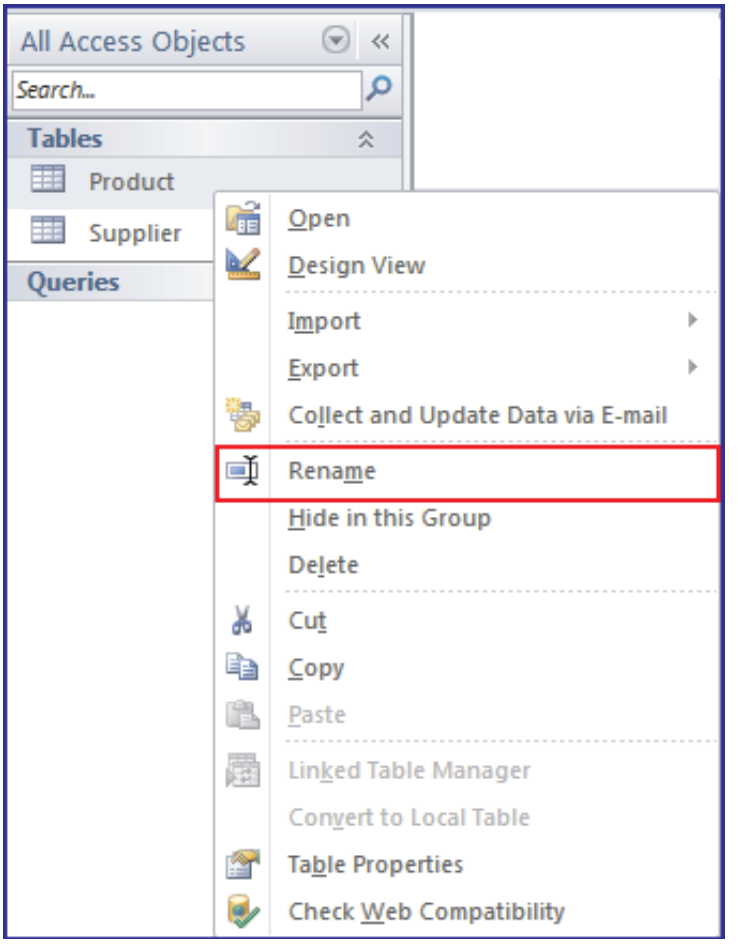


Figure 1.14: Rename Option

3. Type the new name and press the **ENTER** key.

1.8 Working with Application Parts

One of the main goals of Access 2010 is to enable users to reuse the database design rather than create from scratch. This has been achieved in Access 2010 with the help of **Application Parts**. **Application Parts** enables adding readymade tables and forms to the Access 2010 database. In other words, it allows adding pre-built components. It can include a single table or several database objects, such as forms, reports, relationship, and so on. **Application Parts** gallery displays the pre-built parts. Access 2010 includes a number of blank, pre-designed forms as well as **Quick Start** tables that can be added to the database.

To add an application part to the database, perform the following steps:

1. Open the **Access 2010** database.
2. Click **Create** tab.

Concepts

3. Click **Application Parts** to display the gallery containing the objects.
4. Click **Contacts**. This will create a table with built-in fields. It will include queries, forms, and reports.

   Figure 1.15 displays the selection of **Contacts** in the **Application Parts**.

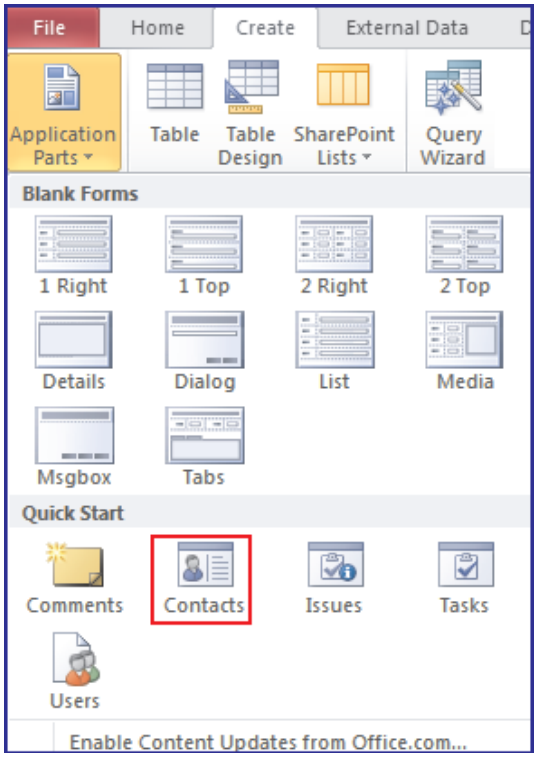


Figure 1.15: Contacts - Application Parts

5. Save it is as an object by deleting the other objects which are not required.
6. Click **File** tab in the **Backstage** view.
7. Click **Save and Publish**. This displays **Save Database As** pane on the right.
8. Select **Template (*.accdt)** to display the **Create New Template from This Database** dialog box.

Concepts

Figure 1.16 displays the **Create New template from This Database** dialog box.

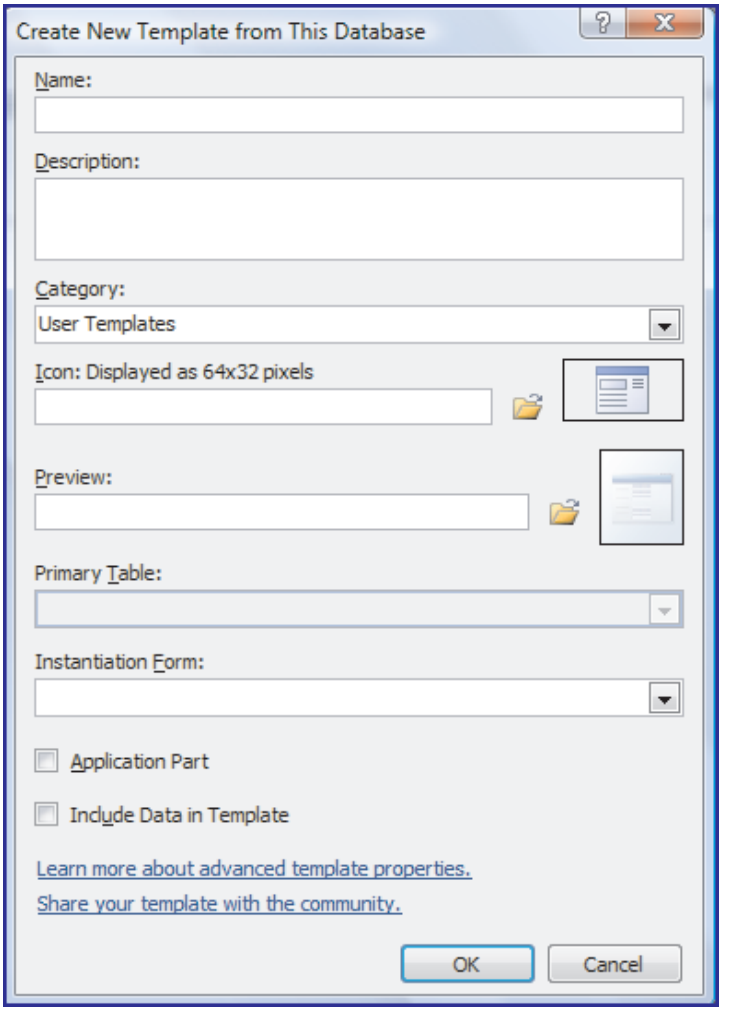


Figure 1.16: Create New Template from This Database Dialog Box

Table 1.2 displays the list of fields shown in the **Create New Template from This Database** dialog box.

Fields	Description
Name	Is required and helps to set the name of the Application Part
Description	Appears as tooltip
Category	Helps to specify the section in the Application Parts gallery where the part will appear
Icon	Helps to set an icon for the newly defined parts
Primary Table	Helps to display the part as the primary table
Instantiation Form	Helps to run the form after the part is instantiated and also gets deleted when closed. Useful as a splash screen
Application Part	Helps to place it in the Access Getting Started as well as in Application Parts Gallery when checked

Fields	Description
Include Data in Template	Saves the data along with the template when checked

Table 1.2: Fields Shown in the Create New Template from This Database Dialog Box

To create a new template, perform the following steps:

1. Type the details.
2. Click **OK** to save it.
3. Click **Application Parts** to display the newly created part in the **Application Parts** gallery.

Or

1. Click **File** tab to display the Backstage view.
2. Click **New** → **My Templates** to display the newly created template.

Thus, it can be said that **Application Parts** help to save time and simplify the task of adding commonly used components, such as tables, forms, and so on.

1.9 Check Your Progress

1. Match the Access 2010 UI components with their corresponding descriptions.

Component		Description	
(A)	Ribbon	1	Template that can be added to the user's database
(B)	Backstage View	2	Provides access to the objects of the current database
(C)	Navigation Pane	3	Allows user to perform common file tasks from a central location
(D)	Application Parts	4	Provides contextual command tabs

(A)	A - 3, B - 2, C - 4, D - 1	(C)	A - 4, B - 3, C - 2, D - 1
(B)	A -2, B - 3, C - 4, D - 1	(D)	A - 1, B - 2, C - 3, D - 4

2. Identify the different types of database models.

(A)	Flat	(C)	Multiple
(B)	Relational	(D)	Hierarchical

3. ____ allows you to store data in the form of related tables.

(A)	Database Management System	(C)	Relational Database Management System
(B)	Hierarchical Database Model	(D)	Excel table

4. Which of the following view is displayed when the database is closed without closing Access 2010?

(A)	Design View	(C)	Datasheet View
(B)	Navigation View	(D)	Backstage View

Concepts

5. Which of the following options shows the correct steps to save an object in the Access 2010 database?

(A)	Quick Access Toolbar → Save Table	(C)	Database Tools → Save Table
(B)	File → Save Object As	(D)	Table → Save Object As

1.9.1 Answers

1.	C
2.	A, B, D
3.	C
4.	D
5.	B

Summary

- A database can be defined as an organized collection of data.
- The different types of database models are namely, flat model, hierarchical model, network model, and relational model.
- Database management is a process of maintaining databases so that the data is easily available to users.
- An RDBMS is a database management system that allows you to store data in the form of related tables.
- Access 2010 is a powerful database building tool that helps users to perform different operations with a database.
- The three main UI components of Access 2010 are namely, Ribbon, Backstage view, and Navigation Pane.
- When Access 2010 is started for the first time, the Backstage view is displayed which is used to create, open, or close a database.
- Application Parts enable users to add readymade tables and forms to an Access 2010 database.

Try it Yourself

1. Mallace is a well-known IT education academy situated in New Jersey. The administration department of the academy manages the records of their faculties, students, and the different courses manually. The academy has decided to computerize the data using Access 2010 database.

   Assume that you are one of the developers of Mallace academy and you have to perform the following tasks:

   a. Create a database named `db_MallaceFaculty` using a suitable template available in Access 2010 database.

   b. Organize the objects created in the database depending on their object type, such as tables, forms, or reports.

Session - 1 (Workshop)

Introduction to Microsoft Access 2010

In this workshop, you will learn to:

- Use Backstage view in Access 2010
- Use Navigation Pane to manage objects in Access 2010
- Understand the various elements of the Ribbon in Access 2010

1.1 Introduction to the Access 2010 User Interface

You will view and practice how to use user interface of Access 2010.

- Working with Backstage view
- Working with Navigation Pane
- Working with different command tabs

Note - Please refer to the respective lab deliverable for demonstrations and guided simulations.

Session - 2

Working with Tables and Data

Welcome to the Session, **Working with Tables and Data**.

This session explains the creation of tables in Access 2010 and the data types that can be used for storing different types of data. Further, this session explains field properties and data entry validations. The session also describes sorting and filtering of data. Lastly, the session explains creation of relationships and importing of data to a new table.

In this Session, you will learn to:

- Explain creation of tables
- Explain data types
- Explain field and field properties
- Explain data entry validations
- Describe sorting and filtering of data
- Explain relationships
- Explain importing data

2.1 Introduction

Consider a situation wherein you want to create a list of customers visiting your shop. The list can be created based on the names of the customers, such as John Sanders, Larry Juvent, Mark Pollock, and so on. This will lead to a one-dimensional list because all the values in the list belong to the same category. Now, to include more information about the customers such as address, e-mail, phone, and so on, you need to create a detailed list. However, if all this data is included in a single category, the list would become very confusing. Therefore, to arrange the data properly, you will need to divide it into separate categories such as Name, Address, E-mail, Phone, and so on. This type of arrangement allows a user to look up data according to its category. Such a distribution of data is the basis of a table.

2.2 Creating Tables

A table is a collection of values arranged as a two-dimensional list according to the categories the values belong to. Figure 2.1 shows an example structure of a table created to store data about the customers of a shop.

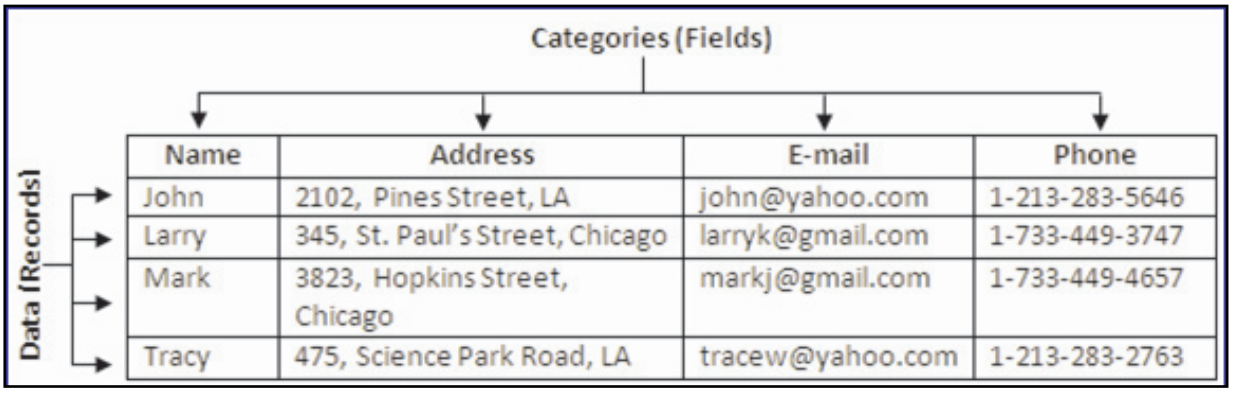


Figure 2.1: Structure of a Customer Table

Figure 2.1 shows the structure of a `Customer` table consisting of categories such as `Name`, `Address`, `E-mail`, and `Phone`. The categories are also known as the attributes or fields of a table. The data about the customers namely, `John`, `Larry`, `Mark`, and `Tracy` is added as individual records. Each row in the table corresponds to one record.

2.2.1 Overview of Tables

Access 2010 enables you to create your own tables within a database. A table in Access 2010 is a two-dimensional structure consisting of fields or columns and records. It is used to store data about a particular entity or object having certain characteristics or attributes. A field is a category or an attribute used to store data about a particular object. Each piece of data is stored under the respective category to which it belongs. A complete or incomplete set of values belonging to each category of a particular item in the table is called a record.

A table name should be meaningful and should reflect the type of the information that is to be stored into it.

Concepts

For example, the table shown in figure 2.1 can be named as `CustomerDetail` as it is storing data about the customers. Similarly, the column names should reflect the data stored within such as `Name`, `Address`, and so on. The column names should be short and can include an underscore '_' to combine two words.

2.2.2 Creating Tables Using Design View

Consider that you have created a new database named `St.XavierSchoolDB` in Access 2010. A new table named `Table1` is created by default and is presented in the **Datasheet View** as shown in figure 2.2.

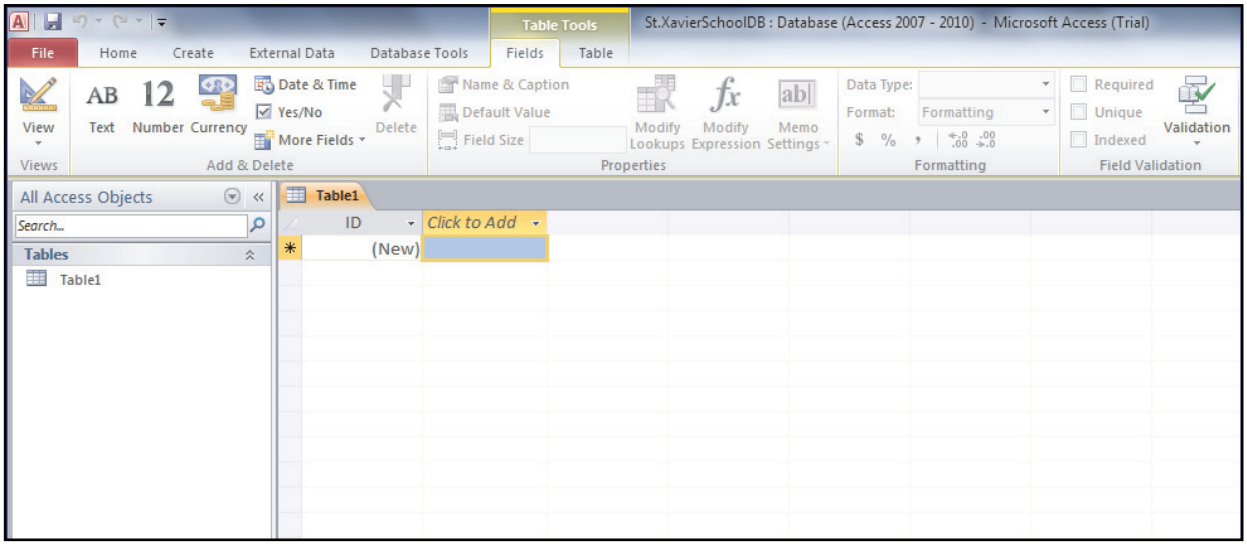


Figure 2.2: Table1 Created in the St.XavierSchoolDBDatabase

Additional tables can be created from the **Tables** group of the **Create** tab on the **Ribbon**. This tab consists of options such as **Table**, **Table Design**, and **SharePoint Lists** as shown in figure 2.3.

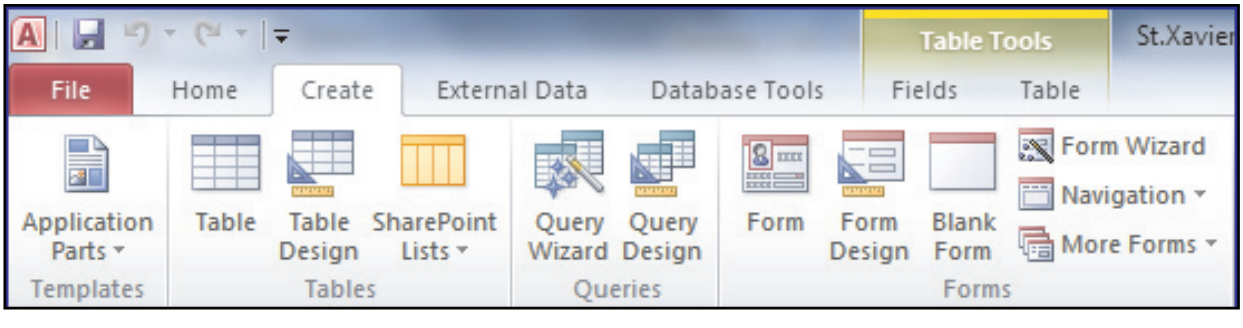


Figure 2.3: Create Tab

To create a new table in the `St.XavierSchoolDB` database, click **Table** option in the **Tables** group of the **Create** tab. A new table named `Table2` will be created as shown in figure 2.4 and will be opened in the **Datasheet View** in the right pane.

Concepts

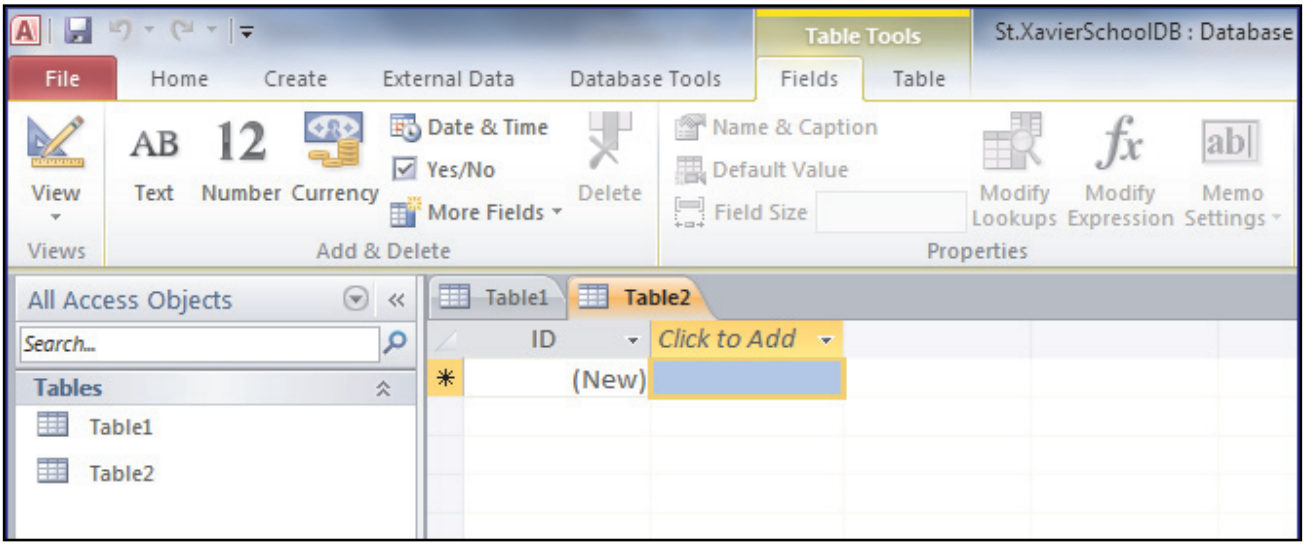


Figure 2.4: Table2 Created and Opened in Datasheet View

The **Fields** tab of **Table Tools** group is automatically activated to allow users to add fields to the table.

To delete the table, right-click `Table2` in the **Tables** navigation pane and select **Delete**.

Users can also create tables using the **Design View**. The steps for this are as follows:

1. Click **Create** → **Table Design** in the **Tables** group. A new table named `Table2` will be opened in the **Design View** as shown in figure 2.5.

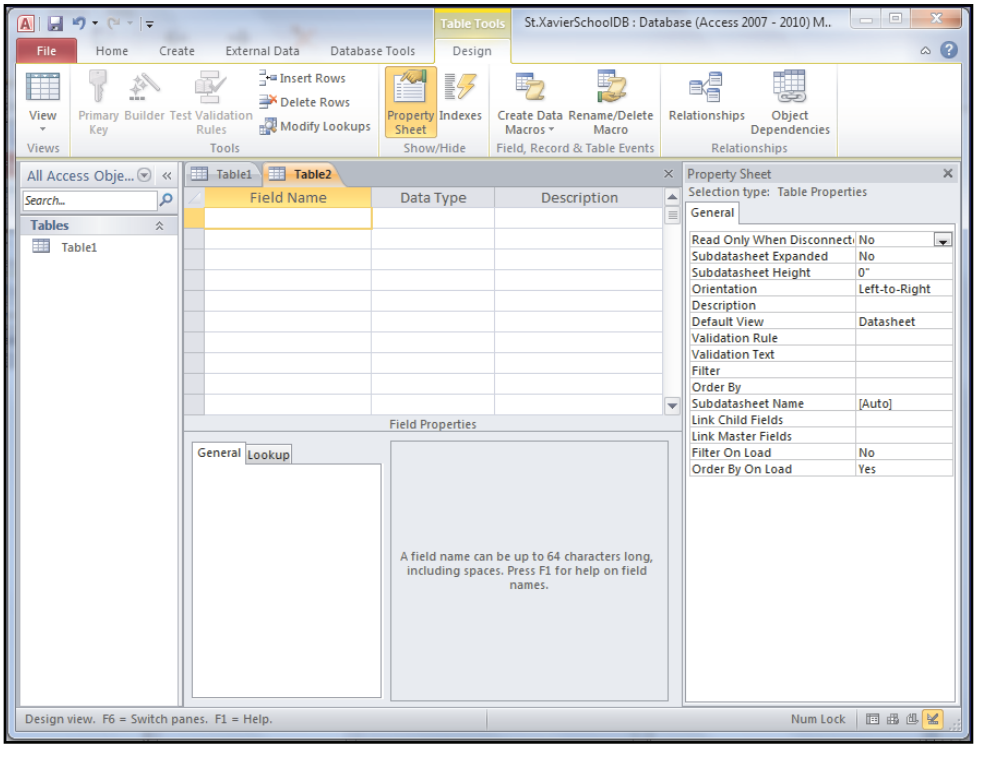


Figure 2.5: Creating a Table in Design View

Concepts

Clicking inside any cell of the **Field Name** column, opens the **Property Sheet** window on the right side showing the general properties of the table as shown in figure 2.5. The **Field Properties** pane is displayed at the bottom of the table with two tabs namely, **General** and **Lookup** that are used to set the properties of the fields of a table. The table is designed by adding information about fields such as **Field Name**, **Data Type**, and **Description** as shown in figure 2.5.

2. Right-click the **Table2** tab and select **Save** from the menu as shown in figure 2.6.

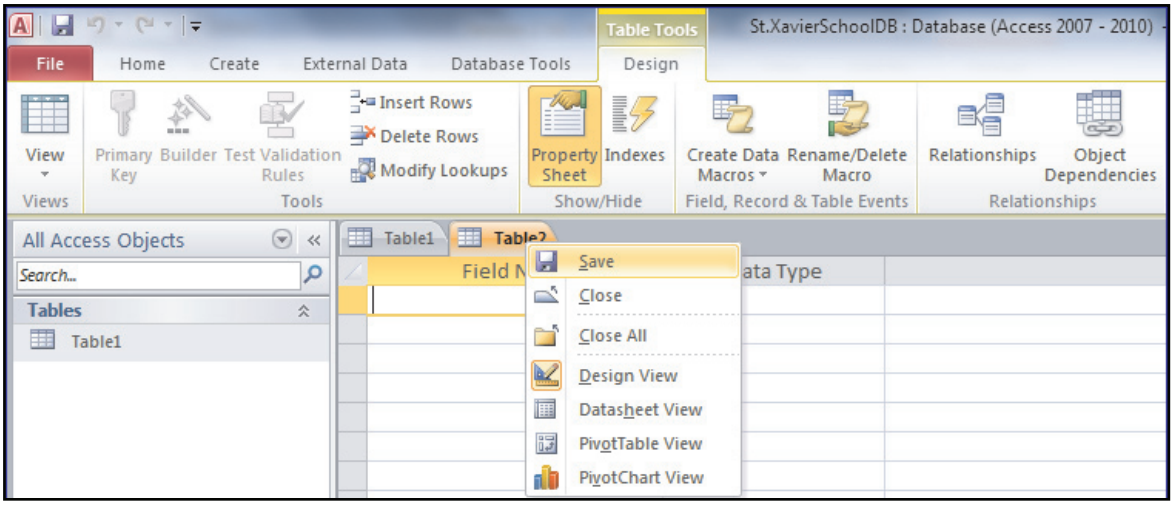


Figure 2.6: Saving a Table

3. Type the name '`Student _ Details`' in the **Save As** dialog box and click **OK**. A message box appears saying '**There are no fields in this table…..** '.

4. Click **OK**. The table will not be saved because there are no fields created. To create fields, first, the names and data types of the fields need to be defined.

By default, a table is displayed in the middle of the screen as a rectangular window with a tab. Access 2010 allows removing the tabs and to display tables without them. To do this, perform the following steps:

1. Click **File** → **Options**.
2. Click **Current Database** in the left pane.
3. In the right pane, in the **Document Window Option** group, click the **Tabbed Documents** option button and clear the **Display Document Tabs** check box.
4. Click **OK**. A message asking permission to close and reopen the database will be displayed.
5. Click **OK**.

The left side of the table name displays the table icon. Right-clicking the icon opens a menu that allows a user to save or change the view of the table.

When a table is displayed as a tab, the top right section of the tab shows a close button × that can be used to close the table.

Concepts

2.3 Data Types

The fields of a table have properties which define the attributes and behavior of the field. Data type is a property of a field that determines the kind of data that the field will store. For example, a field with **Text** as its data type can store textual as well as numeric data (which will be treated as text) whereas a field with **Number** as its data type can store only numeric data.

A data type also determines the following qualities of a field:

- ➔ How the field can be used in expression?
- ➔ What will be the maximum size of the field?
- ➔ Whether index can be applied to the field?
- ➔ Which data formats can be stored in the field?

2.3.1 Overview

While creating a new field, one needs to specify the data type and optionally, its description and other field properties.

When a table is created in the **Datasheet View** by using an existing field of another table or a field template, the data type is automatically applied according to the one defined in the other table or template. When a field is created by entering data in **Datasheet View**, a data type is assigned to the field depending on the value entered by the user as shown in figure 2.7.

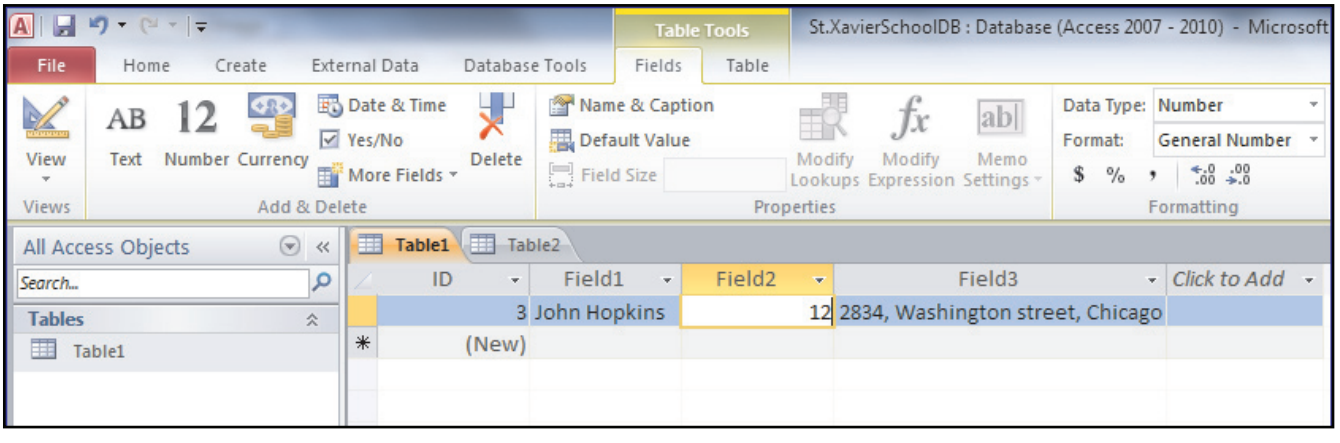


Figure 2.7: Data Type Assigned in the Datasheet View

Figure 2.7 shows **Table1** in the **Datasheet View** consisting of one record. When data is entered in the **Datasheet View**, the data type gets automatically assigned according to the type of data. This can be seen in the **Data Type** drop-down list of the **Formatting** group of the **Fields** tab. The selected field `Field2` consists of the value `12` and the **Data Type** drop-down list shows the data type **Number** assigned to `Field2`.

Concepts

When a user enters values with different data types in a field, Access 2010 may display a prompt asking the user to decide a data type for the field. The data type of a field can be changed using the **Data Type** drop-down list in the **Datasheet View**.

The **Fields** tab also consists of the **Add & Delete** group consisting of the data types such as **Text**, **Number**, and so on, that allows a user to directly add or delete fields of the selected data type in the **Datasheet View**.

2.3.2 Data Types in Access 2010

Access 2010 provides several data types to store different types of data. The data type can be selected through the **Data Type** drop-down in **Design View** as shown in figure 2.8.

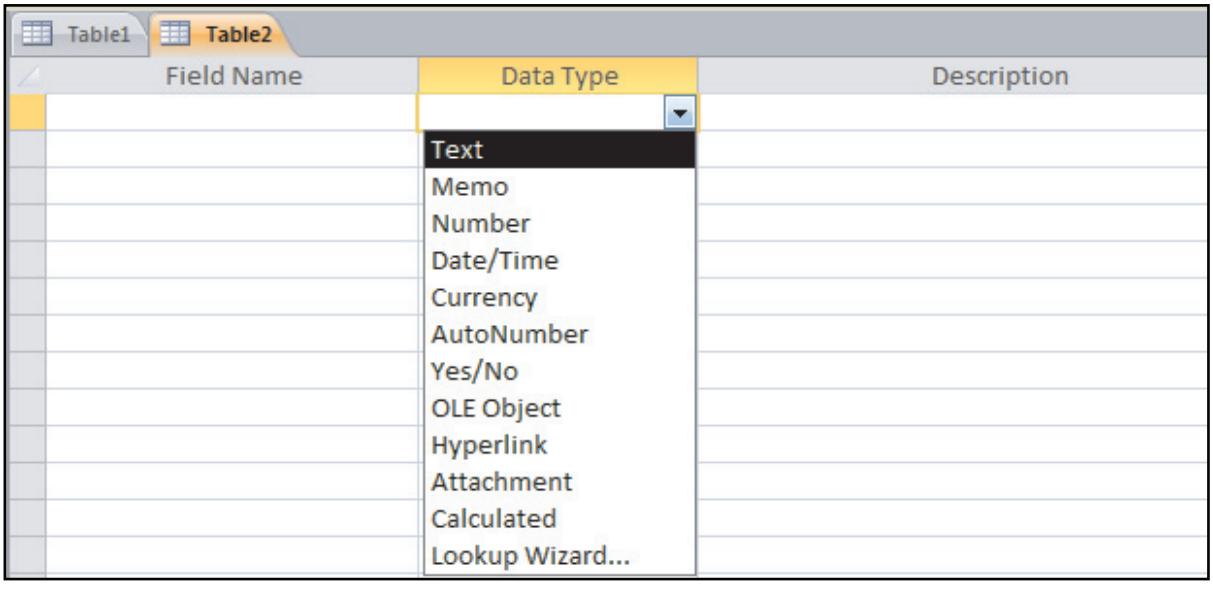


Figure 2.8: Data Types

Each data type has a set of supported properties that can be modified as per requirement. Properties such as Format, Caption, Default Value, Input Mask, Validation Rule, Validation Text, Required, Allow Zero Length, Indexed, and Text Align are common to most of the data types. Table 2.1 lists some of the different data types available in Access 2010.

Data Type	Description	Default Size	Additional Properties
Text	Allows storing short, alpha-numeric values.	255	
Memo	Used to store formatted text and text that is up to 65535 characters.		
Number	Allows storing only numeric values that is not a monetary value.	Long Integer	Decimal Places
Date/Time	Allows storing date and time.		

Concepts

Data Type	Description	Default Size	Additional Properties
Currency	Allows storing values with accuracy upto 15 digits to the left and 4 digits to the right of the decimal point.		Decimal Places
AutoNumber	Used to generate a unique value to make each record unique. Most commonly used for a primary key field.	Long Integer	New Values
Yes/No	Used to store Boolean values. It is a binary field and its value can be set to Yes/No, True/False, or On/Off.	0	
Hyperlink	Stores hyperlinks such as an e-mail address or a Website URL.		
Calculated	Uses the Expression Builder to create a calculated field by combining data of other existing fields.		Result Type
Lookup Wizard	Used to create a lookup field which displays either a static list of values entered while creating the field or values retrieved from a query or another table.		

Table 2.1: Access 2010 Data Types

2.4 Field and Field Properties

A field is an object used to store a piece of data in a table. Usually, all fields of a table are generally created to store data whereas fields on other objects such as forms, queries, and reports can be a combination of existing fields, display fields as well as calculated fields.

2.4.1 Inserting, Updating, Deleting, and Renaming a Field

To create and save a table in Access 2010, the table must have at least one column. When a table is created in the **Datasheet View**, Access 2010 creates one default column named ID and the user can start entering data into its cells. When a user enters data in a new column, that column receives an incremental name. The first column to the right of ID is named `Field1`, the second, `Field2`, and so on. However, these names are insignificant and can be changed later to meaningful names.

To insert and rename a field in the **Design View**, perform the following steps:

1. In the `Table2` created earlier in the **Design View**, click the first cell under the column **Field Name** and type '`Stud _ ID`'.
2. In the **Data Type** column, select **Text** from the drop-down list. The list of supported properties of **Text** data type will be displayed under the table in the **Field Properties** pane.

Concepts

3. In the **Description** column, type '`Unique ID of a student`' as shown in figure 2.9.

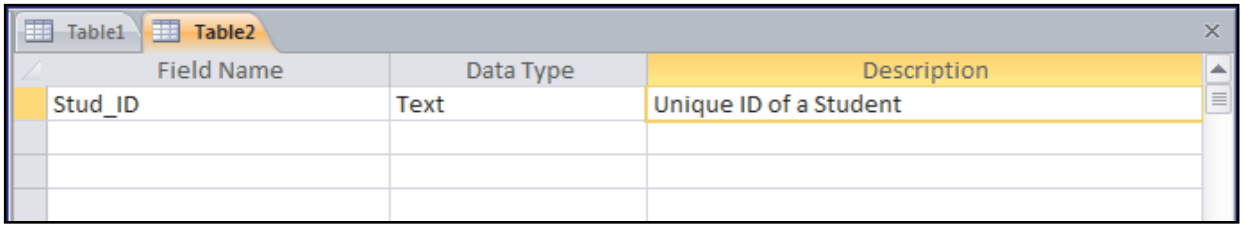


Figure 2.9: Adding a Field to a Table

4. Similarly, add few more fields to the table as shown in figure 2.10.

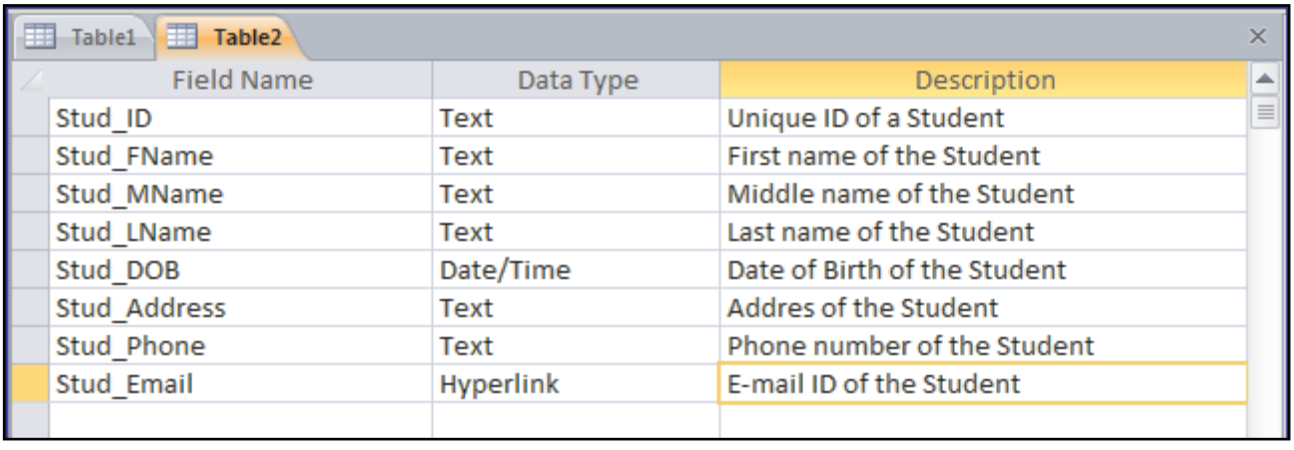


Figure 2.10: Adding More Fields to a Table

5. To view data that goes beyond the column width, increase the column width by positioning the mouse on the right border of a column header. The mouse pointer changes into a horizontal double arrow crossed by a vertical line. Click and drag the arrow to the desired size. Double-clicking the column resizes it to the largest value of the column, provided the largest value is wider than the column header. If the largest value is narrower than the column header, the column width becomes wide enough to display the name of the column.

6. Right-click the `Table2` tab and select **Save** from the menu. The **Save As** dialog box is displayed.

7. Type the value '`Student _ Details`' in the **Table Name** box and click **OK**. A message box asking for primary key will be displayed as shown in figure 2.11.

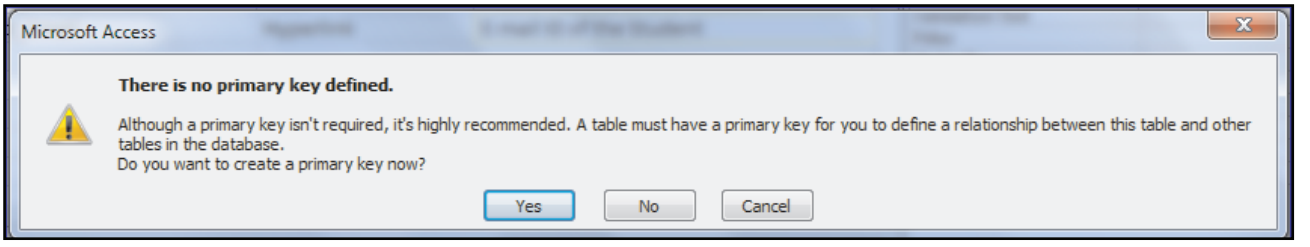


Figure 2.11: Primary Key Missing Message Box

8. Click **No** as no primary key needs to be created at this stage. The table `Table2` will be saved with the name `Student _ Details` and will be added to the Tables pane on the left.

9. Double-click the `Student _ Details` table in the Tables pane. The table will be opened in the **Datasheet View** as shown in figure 2.12 with the cursor placed in the first row and first column of the table.

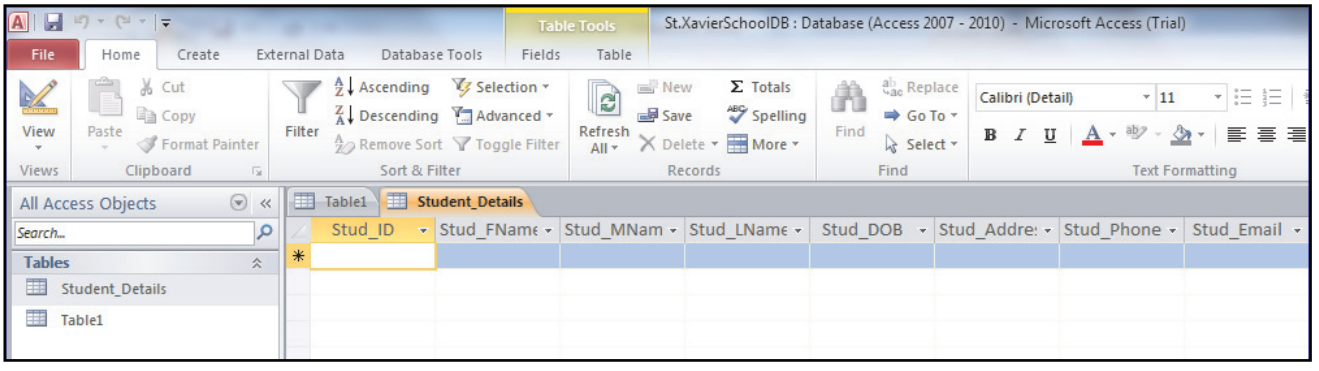


Figure 2.12: Student_Details Table in Datasheet View

Now, the user can start entering data in the table.

To update a field in the **Design View**, perform the following steps:

1. Right-click the `Student _ Details` table in the **Tables** pane and select **Design View** from the menu.

2. Change the **Description** of the `Stud_Email` column to '`E-mail address of the student`' and press **Ctrl+S** or click the **Save** icon in the **Quick Access** toolbar in the top-left corner.

To delete a field in the **Design View**, perform the following steps:

1. If the **Design View** is not open, right-click the `Student _ Details` table in the **Tables** pane and select **Design View** from the menu.

2. Right-click the `Stud _ Email` field and select **Delete Rows** from the menu. A message box asking for confirmation of permanent deletion will be displayed.

3. Click **OK**. The field will be removed from the table.

4. Click the **Save** icon to save the changes to the table.

Concepts

To rename a field in the **Design View**, perform the following steps:

1. If the **Design View** is not open, right-click the `Student _ Details` table in the **Tables** pane and select **Design View** from the menu.
2. Click the `Stud _ Phone` field and change it to `Stud _ Ph _ No`.
3. Click the **Save** icon to save the changes to the table.

To directly insert, update, delete, and rename fields in the **Datasheet View**, right-click the desired field and select the required action from the context menu as shown in figure 2.13.

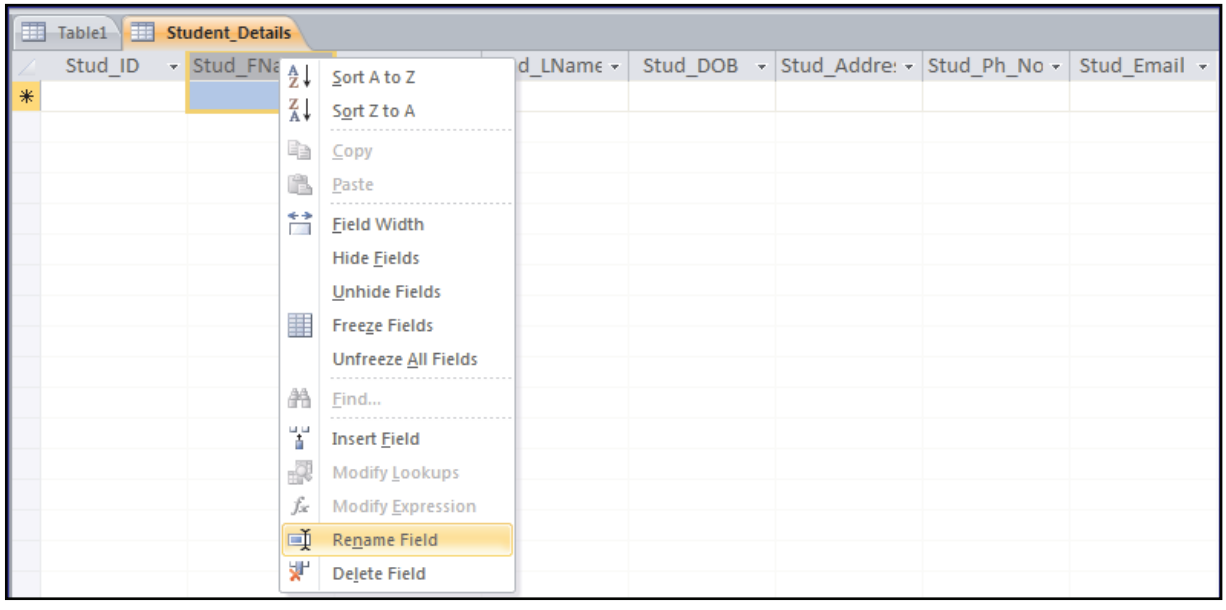


Figure 2.13: Insert, Update, Delete, and Rename a Field in the Datasheet View

2.4.2 Freezing and Unfreezing Fields

To keep a certain section of a datasheet visible while scrolling to another section, a user can freeze one or more of the fields. The fields that are frozen are moved to the leftmost position of the datasheet.

To freeze fields, perform the following steps:

1. Open the `Student _ Detail` table in the **Datasheet View**.
2. To freeze several fields, first the user must move them to make them contiguous. For example, to freeze `Stud _ ID` and `Stud _ DOB`, drag the `Stud _ DOB` field to the left and place it after `Stud _ ID` field.
3. Select the `Stud _ ID` and `Stud _ DOB` fields. To do this, press and hold **Shift** while clicking the fields.
4. Right-click the selected fields and then, click **Freeze Fields**.

Concepts

Now, when the user scrolls through the horizontal scroll bar at the bottom, the fields `Stud_ID` and `Stud_DOB` will remain fixed, while the data of other fields can be scrolled as shown in figure 2.14.

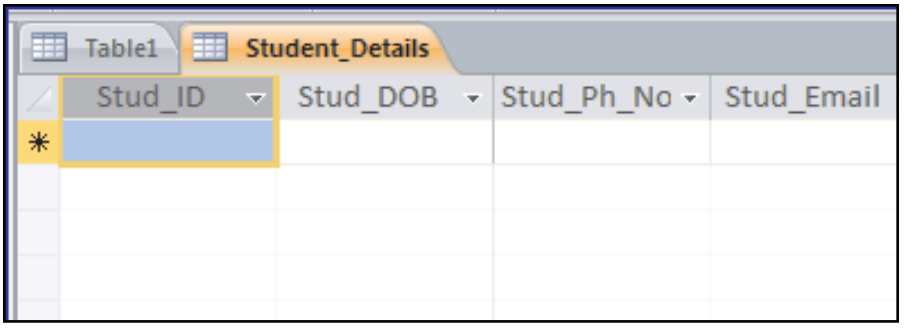


Figure 2.14: Freezing Fields in the Datasheet View

To leave the fields frozen after finishing the task, save the changes while closing the datasheet.

To unfreeze the field/s, right-click the fields, and then, click **Unfreeze All Fields** from the menu.

Note - Multiple fields can be frozen only if they are contiguous, so they must be moved to the left next to each other before applying the freeze setting. After unfreezing the fields, they must be manually moved back to their original positions if required.

2.4.3 Setting and Modifying Field Properties

Each data type in Access 2010 has a set of associated field properties that are displayed at the bottom of the screen when a data type is selected from the **Data Type** drop-down list in the **Design View**. On selecting a property, Access 2010 displays a short description of the selected property on the right side of the window. Figure 2.15 shows the **Field Properties** for the **Text** data type and a brief description of the **Field Size** property.

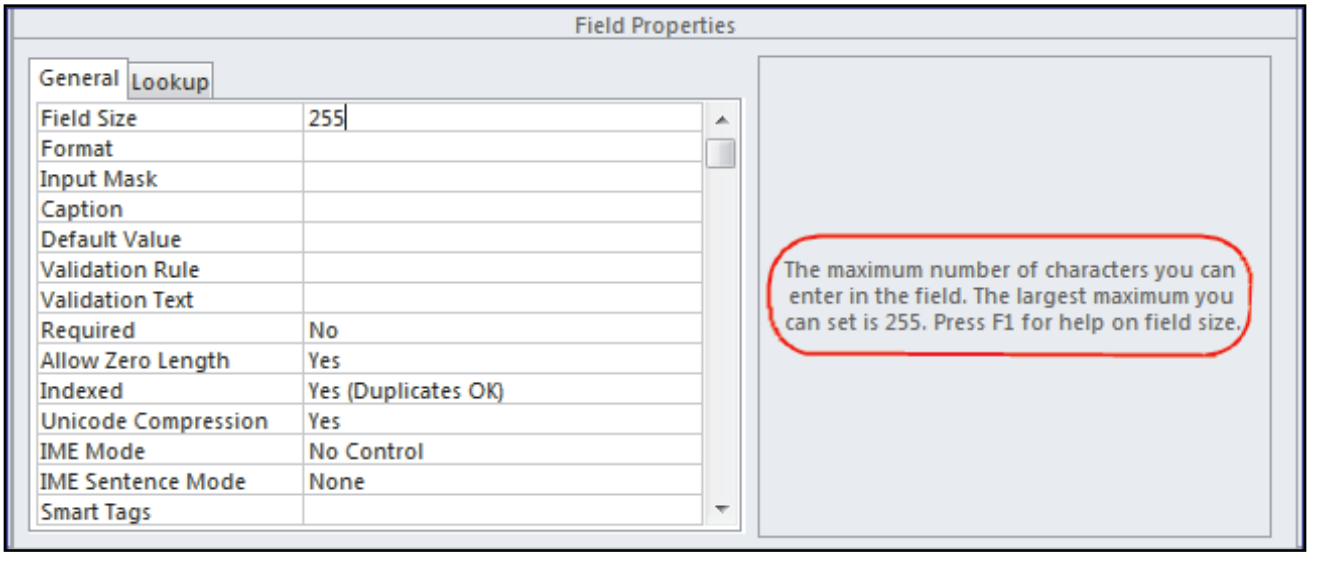


Figure 2.15: Description of the Field Size Property of Text Data Type

Concepts

Table 2.2 lists some of the field properties and their description.

Property	Data Type	Description
Field Size	Text, Number, AutoNumber	Specifies the maximum number of characters allowed in the field.
New Values	AutoNumber	This property applies only to the AutoNumber data type. The options are: Increment and Random.
Expression	Calculated	It shows the equation/expression.
Result Type	Calculated	Used to determine the field type and numeric field size of the result.
Format	Text, Memo, Hyperlink, Calculated, Number, Currency, AutoNumber, Calculate, Yes/No	The format box for the Text, Memo, Hyperlink, and Calculated data types are empty and require custom formats. Number formats are: General number, Euro, Currency, Fixed, and so on. Date/Time fields allow creation of custom formats. The default formats for the Yes/No field are Yes/No, True/False, or On/Off.
Decimal Places	Numbers, Currency	This property setting has no effect if the Format property is blank or is set to General Number.
Input Mask	Text, Numbers, Date/Time, Currency	Input masks allow a user to set a pattern or a template for the data to be entered to give the data a consistent look.
Validation Rule	Text, Memo, Number, Date/Time, Currency, Yes/No, Hyperlink	This property is used to restrict the values that are entered into a field.
Validation Text	Text, Memo, Number, Date/Time, Currency, Yes/No, Hyperlink	This property holds the error message to be displayed when the data entered does not match the Validation Rule.
Required	Text, Memo, Number, Date/Time, Currency, OLE Object, Hyperlink, Attachment	This property specifies if data is required to be entered into the field.
Show Date Picker	Date/Time	The default value of this property is always with dates, so Access 2010 provides a little calendar at the end of any Date/Time field.

Table 2.2: Field Properties

To set a field property, perform the following steps:

1. If the **Design View** is not open, right-click the `Student _ Details` table in the **Tables** pane and select **Design View** from the menu.
2. Click the **Text** data type of the `Stud _ ID` field. The **Field Properties** pane displays the general properties of the **Text** data type.
3. Click the drop-down list next to the **Required** property and set it to **Yes**. This means that if `Stud_ID` column is left blank while entering a record, Access 2010 will display an error message.
4. Click the Save icon to **save** the changes.

Similarly, the other properties of a data type can also be set as per the predefined values.

To modify a field property, perform the following steps:

1. If the **Design View** is not open, right-click the `Student_Details` table in the **Tables** pane and select **Design View** from the menu.
2. Click the **Text** data type of the `Stud_ID` field. The **Field Properties** pane displays the general properties of the **Text** data type.
3. Delete the default value `255` in the **Field Size** box and type `10`. A warning message box is displayed as shown in figure 2.16. Now, the field is restricted to ten characters.

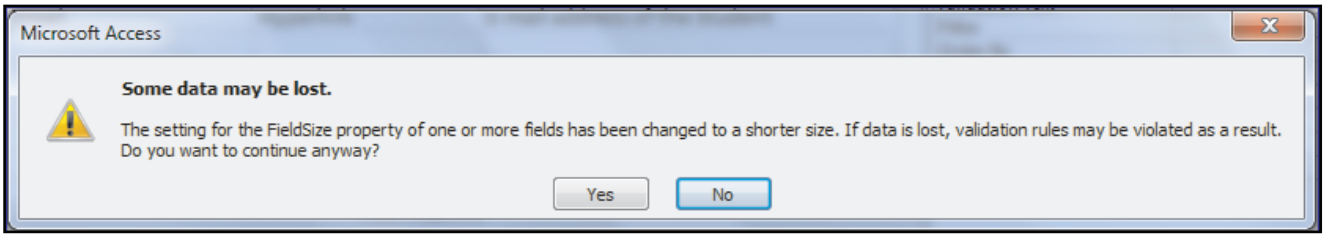


Figure 2.16: Warning Message Box

The message box warns the user that data will be lost if validation rules are violated due to change of property value.

4. Click **Yes** as there is no data in the table yet.
5. Similarly, change the **Field Size** value for `Stud _ FName`, `Stud _ MName`, and `Stud _ LName` fields to `100`, `Stud _ Ph _ No` field to `14`, and `Stud _ Address` field to `200`.
6. Click the **Save** icon to save the changes.

Similarly, the other properties of a data type can also be modified according to requirement provided there is no data in the table or it does not affect the data that is already entered into the table.

Concepts

2.4.4 Defining Input Masks

Input masks are used to set a pattern or a template for the data to be entered to give the data a consistent look. The Input Mask Wizard can be opened pressing the ellipses (...) button appearing at the end of the **Input Mask** property box.

The **Input Mask Wizard** provides several predefined formats for common data entry items, such as Zip Code, Phone Number, and Social Security Number.

The input mask option applies to Text, Numbers, Date/Time and Currency fields, but the Wizard builds a mask only for the Text and Date/Time fields.

To build a custom Input Mask such as Medical Record Numbers, Product codes, and so on, and add to the Wizard, click the **Edit List** button on the first screen of the Wizard. Use **L** for letters; **9** for optional numbers; **0** for required numbers; **A** for alphanumeric characters.

To create an Input Mask, perform the following steps:

1. If the **Design View** is not open, right-click the `Student _ Details` table in the **Tables** pane and select **Design View** from the menu.
2. Click the **Text** data type of the `Stud _ Ph _ No` field. The **Field Properties** pane displays the **General** properties of the **Text** data type.
3. Click the **Input Mask** ellipses '...' button in the **Input Mask** property box of `Stud _ Ph _ No` field to open the **Input Mask Wizard**. If a message box appears to save the table, click **Yes**. The **Input Mask Wizard** dialog box is displayed as shown in figure 2.17.

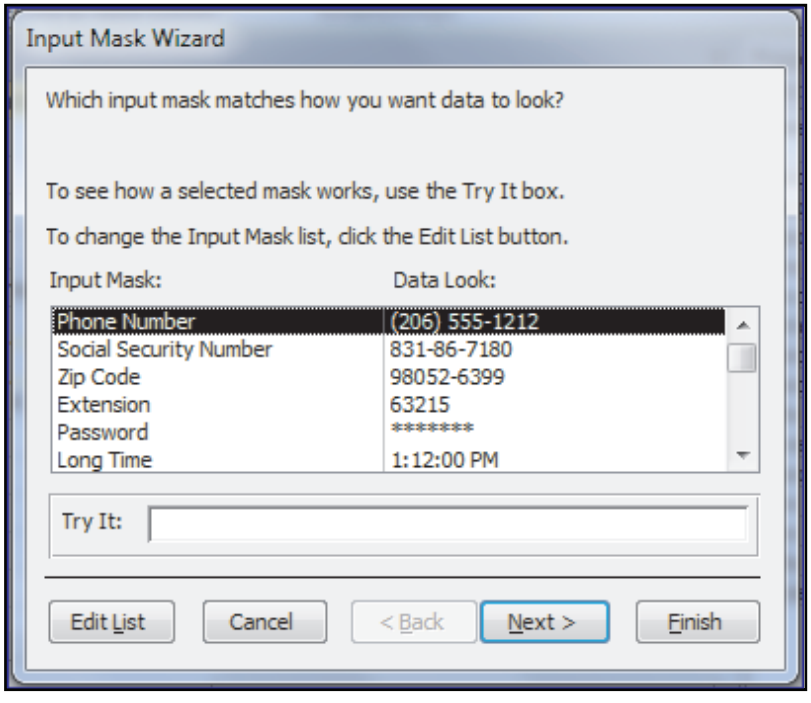


Figure 2.17: Input Mask Wizard

4. Click **Edit List**. The **Customize Input Mask Wizard** dialog box is displayed.

5. Click the **Next Record** ▶ button in the **Record** bar at the bottom and scroll till an empty record is reached. Insert the values for the new phone number pattern as shown in figure 2.18.

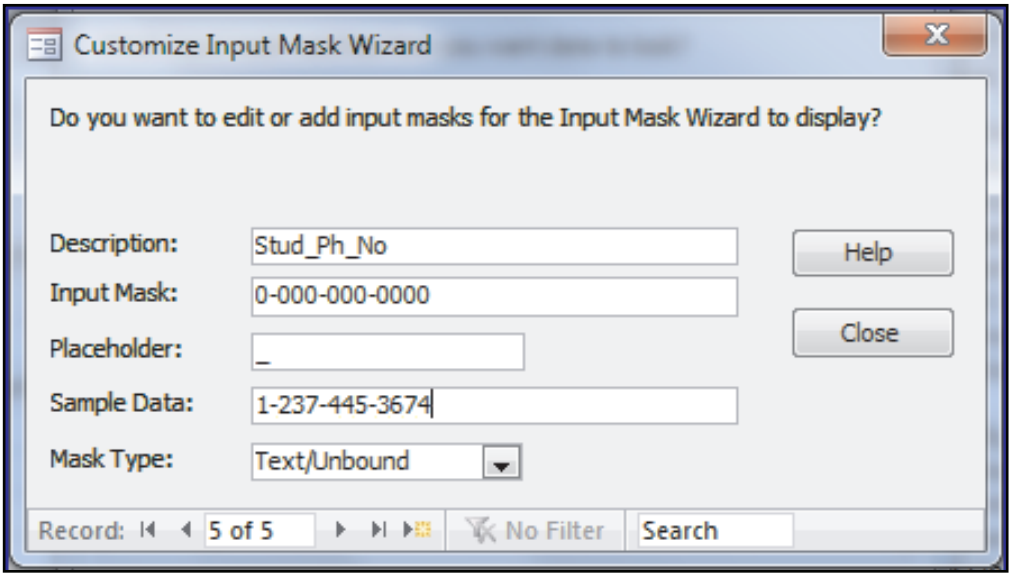


Figure 2.18: Customize Input Mask Wizard

6. Click **Close**. The new input mask `Stud _ Ph _ No` appears in the **Input Mask** list box of the **Input Mask Wizard**.

7. Select the `Stud _ Ph _ No` input mask and click **Next**. A screen showing the name and pattern of the input mask is displayed with a **Try It** box.

8. Click **Next**. A screen is displayed as shown in figure 2.19.

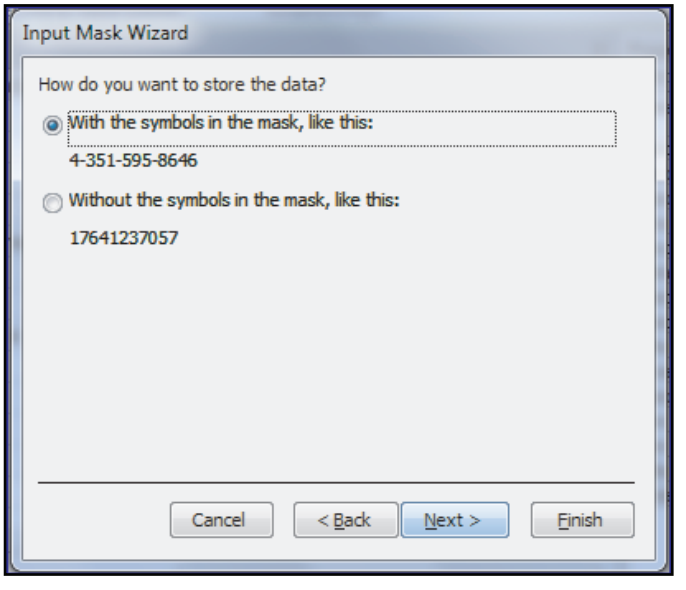


Figure 2.19: Input Mask Wizard Screen

Concepts

9. Select the **With the symbols in the mask, like this:** option button as shown in figure 2.19 and click **Next**.

10. Click **Finish**. The selected input mask is displayed in the **Input Mask** box in the **Field Properties** of the `Stud _ Ph _ No` field.

11. Click **Save** to save the changes to the table design.

12. Double-click the `Student _ Details` table in the **Tables** pane to open it in the **Datasheet View**.

13. Click inside the first cell under the `Stud _ Ph _ No` column. The input mask place holders will appear as shown in figure 2.20.

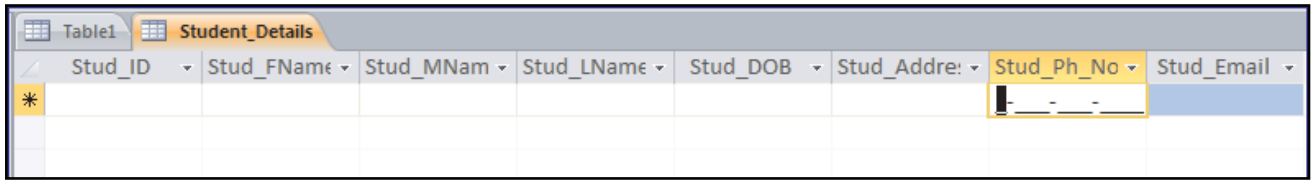


Figure 2.20: Input Mask Applied to Stud_Ph_No Field

14. Similarly, change the Input Mask for the **Date/Time** data type of the `Stud _ DOB` field to **Short Date** and click the **Save** icon to save the changes.

2.4.5 Creating Multivalued Fields

Multivalued fields in Access 2010 allow a user to select and store multiple values in a single field. For example, if several developers are developing the same project, one can store names of all the developers in the same field.

The multivalued fields feature leads to confusion since it seems as if it violates the concept of normalization according to which only one value should be stored per field. Hence, it is preferable to avoid using multivalued fields. The reason for this perception is that Access 2010 does not physically store the values in a single field. To the user, it appears to be under the same field, but internally, Access 2010 stores the data in normalized, but hidden, tables. By doing so, it handles normalization by itself by separating and storing the data and then, combining it in the user interface. The multivalued field is like a many-to-many relationship that does not need to be created manually nor can it be seen physically.

Some guidelines to decide on the use of a multivalued field are as follows:

- ➔ This feature should not be used with long lists.
- ➔ Multivalued field can be used with SharePoint Services.
- ➔ Keep upgradation in mind while using a multivalued field since SQL Server converts a multivalued field to a Memo field.

Consider the `Student_Details` table. A user wants to enter details in this table about the languages known to the student.

Concepts

For that, a multivalued field can be created that allows the user to enter more than one language in the same column.

To create a multivalued field, perform the following steps:

1. Open the `Student _ Detail` stable in **Design View**.
2. Click the blank row after the `Stud _ Email` field.
3. Click **Modify Lookups** in the **Tools** group on the **Design** tab of **Table Tools**. The **Lookup Wizard** is displayed.

Note - The Lookup Wizard can also be opened by selecting the **Lookup Wizard** option from the **Data Type** drop-down list.

4. In the first screen, select the '**I will type in the values that I want**' option button as shown in figure 2.21.

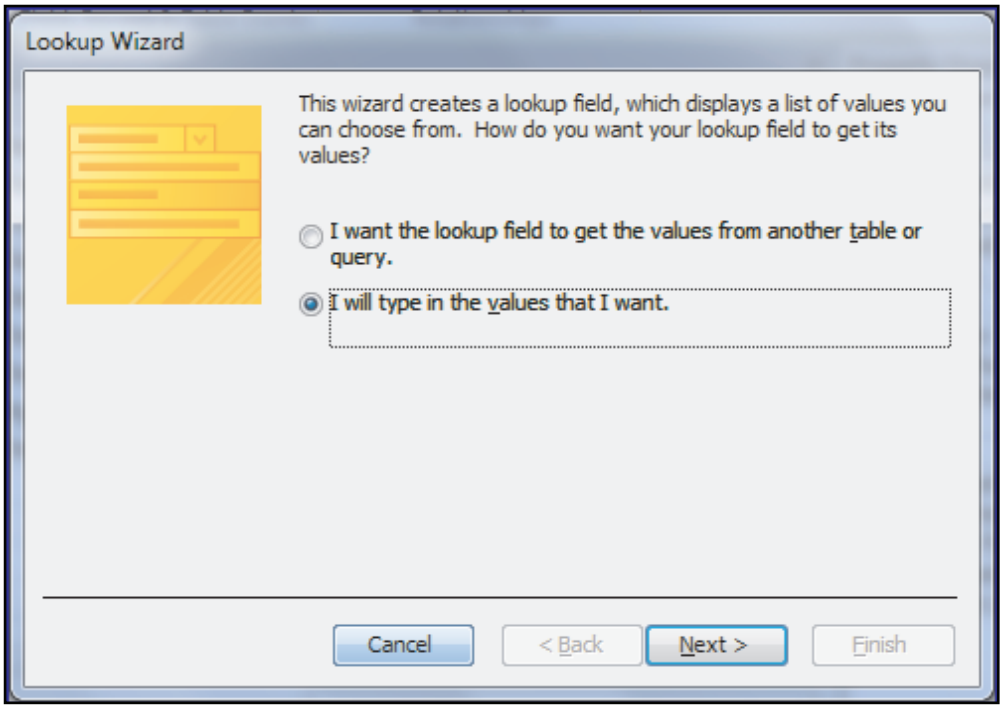


Figure 2.21: Lookup Wizard

The other option button can be used when data is to be retrieved from another table.

Concepts

5. Click **Next**. Leave the number of columns as 1 and enter the values in **Col1** column as shown in figure 2.22.

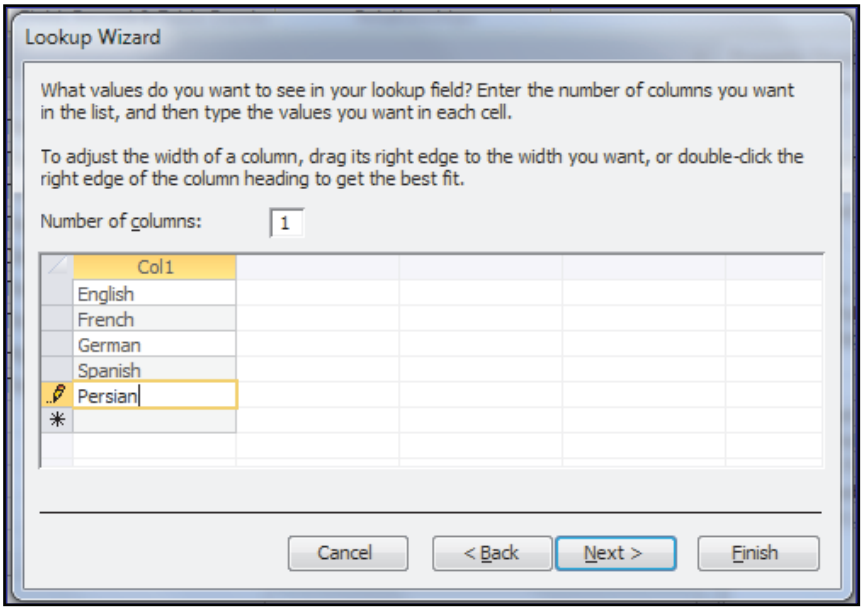


Figure 2.22: Specifying the Lookup Field Values

6. Click **Next**. Specify the lookup field name as '`Stud _ Lang`' and select the **Allow Multiple Values** check box as shown in figure 2.23.

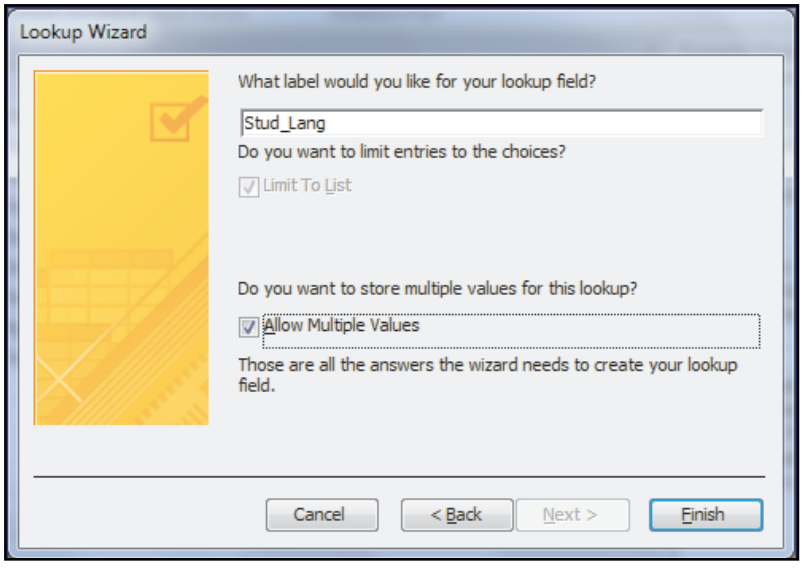


Figure 2.23: Specifying the Name and Criteria of Lookup Field

7. Click **Finish**. The `Stud _ Lang` field will be added to the `Student _ Details` table in the **Design View**.

8. Type '`Languages known to the student`' in the **Description** of the `Stud _ Lang` field.

Concepts

9. Click the **Save** icon to save the changes.
10. Double-click the `Student _ Details` table in the **Tables** pane to open it in the **Datasheet View**.
11. Scroll to the `Stud _ Lang` column and click the arrow next to the first cell. A drop-down list with check boxes against each item is displayed as shown in figure 2.24.

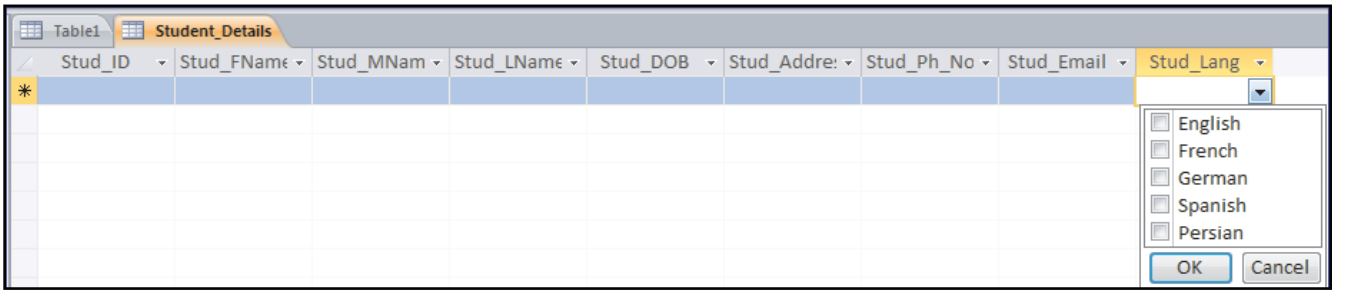


Figure 2.24: Lookup Field of a Table in the Datasheet View

The user can check the desired options which will be added to the field value with a comma as separator.

2.5 Data Entry Validations

Field validation allows validation of data entered by the user by applying necessary constraints over the field. It also displays messages in case of an invalid entry. A user can create simple or complex expressions to validate fields and apply rules over them. Different expressions can be applied over queries, tables, reports, macros, and forms for validation.

2.5.1 Entering and Inserting Data

A user can enter data into a table directly from the **Datasheet View**, using a query, or by creating forms. To enter data in the **Datasheet View**, open the `Student_Details` table in the **Datasheet View** and type the data as shown in figure 2.25.

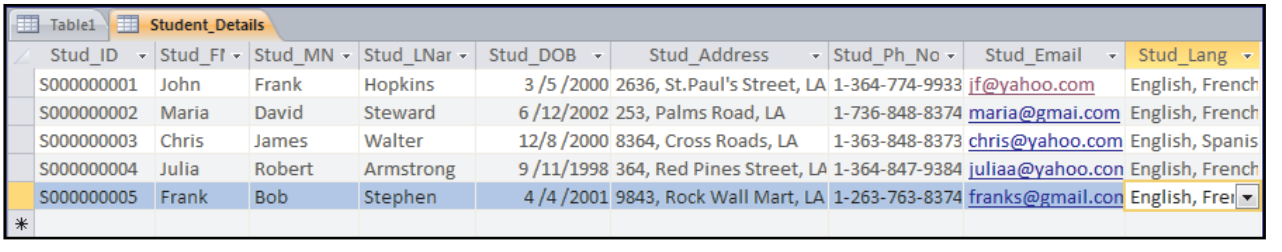


Figure 2.25: Entering Data in a Table in the Datasheet View

When the cursor is moved from one cell to another, data in the cell gets automatically saved. As can be seen, the `Stud_DOB` field is one of the fields that required data validation since the date of birth of a student needs to be restricted to a particular range. Suppose it is required that the date of birth should not be less than 1/5/1997, that is January 05, 1997. Then, the `Stud_DOB` field must be validated for the data entered.

2.5.2 Setting Validation Rules

The validation rules can be applied by setting the Validation Rule property of the particular field. Validations can be applied by using operators as well as wildcards as follows:

➔ **Using Operators**

Relational operators such as >, <, and so on can be used along with Logical operators such as **AND**, **OR**, **NOT**, and so on to create validation expressions.

For example, the expression '<>0' indicates that the value cannot be equal to zero.

➔ **Using validation with Dates**

Validations can be applied to dates to restrict the date within a range. For example, the expression '>=#01/01/2000' indicates that the date must be on or after January 01, 2000.

➔ **Using keywords and wildcards**

Access 2010 allows using keyword operators such as 'between' and 'like' to create validations. For example, the expression '*ay', indicates that the text must end with 'ay'. Also, the expression 'Between 2 and 20' indicates that the value must be between 2 and 20.

2.5.3 Field Level Validation

A field validation rule is used to check the value entered in a field. For example, in a Date field, a validation rule '>=#01/05/1997#' forces the user to enter a value on or after January 05, 1997. If a value earlier than that is entered, Access 2010 will prevent the user from leaving the current field until the problem is fixed.

To apply field validation to the Stud_DOB field, perform the following steps:

1. Open the Student _ Details table in **Design View**.
2. Click the data type column of the Stud _ DOB field. The **Field Properties** of the **Date/Time** data type will be displayed.
3. Click the **Validation Rule** box and click the ellipses '...' button to open the **Expression Builder** dialog box.
4. Type the expression '>=#1/5/1997#' in the **Expression Builder** box as shown in figure 2.26.

Concepts

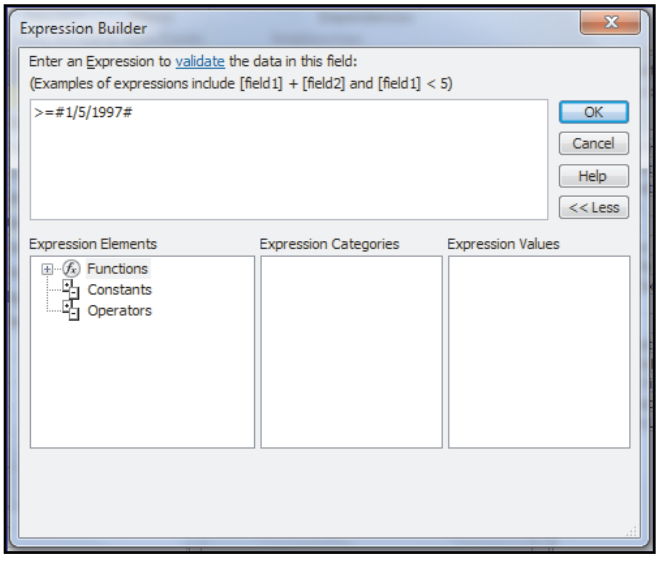


Figure 2.26: Creating Validation Rule for Stud_DOB Field

5. Click **OK**. The expression will be added to the **Validation Rule** property.

6. Type '`Date of Birth must be greater than January 05, 1997`' in the **Validation Text** property. This message will be displayed to the user if the date of birth is invalid.

7. Click the **Save** icon and open the `Student _ Details` in the **Datasheet View**.

8. Type another record `S000000006` and then, type the date of birth as '`01/05/1996`'. A message box showing the **Validation Text** will be displayed to the user as shown in figure 2.27.

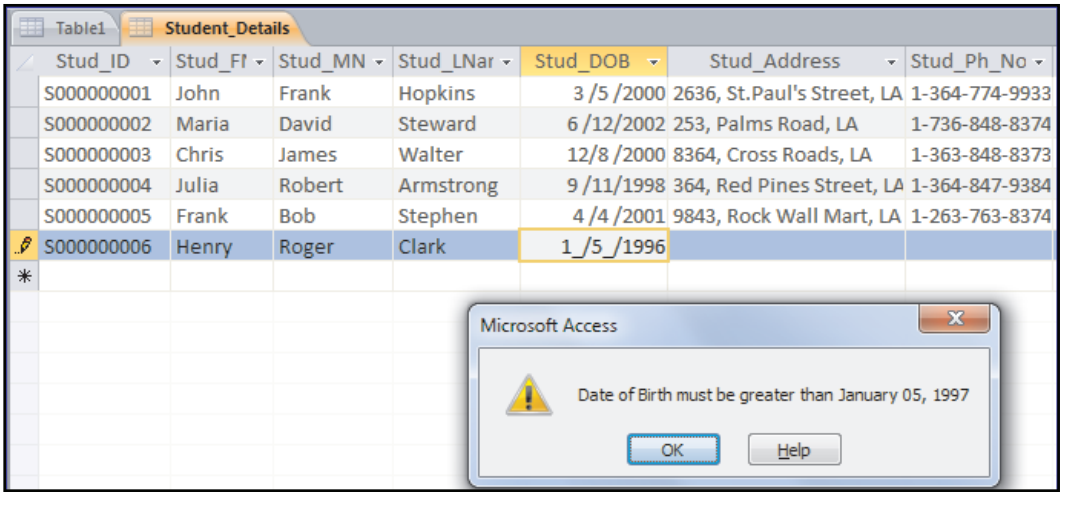


Figure 2.27: Validation Text Displayed on Invalid Entry

9. Click **OK**.

10. Try to move the cursor to another cell by using the arrow key or the mouse pointer. The **Validation Text** message will appear again. Access 2010 prevents the user from proceeding further until a valid date value is specified that fulfils the validation rule.

Concepts

11. Change the date of birth to '01/05/1998' and move the cursor to the Stud _ Address column. Access 2010 will allow the user to move to the next field since the date is valid.

12. Add the address, phone number, and languages to complete the record entry.

Table 2.3 lists some examples of validation rules.

Validation Rule	Description
<>0	The value must be non-zero.
>=0	The value must be greater than or equal to zero.
>=#01/01/2007# AND <#01/01/2008#	The date must belong to the year 2007.
M Or F	The value must be M for male or F for female.
LIKE "[A-Z]*@[A-Z].com"	The e-mail address must have @ and .com as per the given format.
[SupplyDate]<=[OrderDate]+20	The supply date should be within 20 days after the order date.
IN ("Mon","Tue","Wed")	The value must be either Mon, Tue, or Wed.
BETWEEN 10 AND 100	The value must be between 10 and 100.
LIKE "Am*"	The value must start with 'Am'.
January OR February	The value must be either January or February.

Table 2.3: Examples of Validation Rules

2.5.4 Table Level Validation

A record validation rule is used to control when a record or a row in a table can be saved. Unlike a field validation rule, a record validation rule is used to refer to other fields of the same table. This means that record validation rule checks the values of one field against the values in another.

For example, suppose a user wants to check that the age of student should not be more than 16 years. If the condition is not satisfied, the student cannot be admitted to the school. For this, a user can define a record validation rule such as Right $ ([DateOfAdmission] , 4) - Right$([Stud_DOB],4) <=16. A record validation rule to check input to one or more fields is applied when the focus leaves the record. Usually, a record validation rule compares the values of two or more fields.

To apply record validations to the Student_Details table, perform the following steps:

1. Open the Student _ Details table in **Design View**.

2. Create a column named DateOfAdmission and set the data type to Date/Time. Enter dates for this column in the **Datasheet View** of the table ensuring that the age should not exceed 16 years.

3. Open the **Design View** of the table and click the **Validation Rule** box in the **Property Sheet** pane on the right side that displays the Table properties and click the ellipses '...' button.

Concepts

The **Expression Builder** dialog box is displayed.

4. Type the expression '`Right$([DateOfAdmission],4)-Right$([Stud _ DOB],4)<=16`' in the **Expression Builder** box and click **OK**.

Explanation of the expression:

Here, `Right$()` is a function that returns the right most characters of a string according to the length specified.

For example, `Right$([DateOfAdmission], 4)` will return the last four characters of the `DateOfAdmission` field. This means, the user can retrieve the year part from the date.

The subtraction of `Stud_DOB` from `DateOfAdmission` will return the difference of the year parts that will give the age of the student.

The expression '`<=16`', checks if the value returned after subtraction is less than or equal to `16`. If yes, the user will be allowed to move ahead. If not, then a message box with the validation text will be displayed to the user.

5. Type '`Age cannot be greater than 16`' in the **Validation Text** box as shown in figure 2.28.

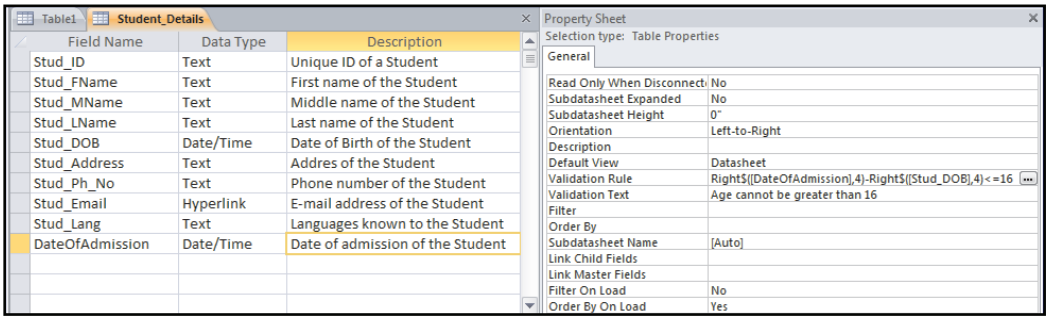


Figure 2.28: Setting Table Level Validation

6. Click the **Save** icon and open the `Student _ Details` table in the **Datasheet View**.

7. Enter a new record `S000000007`. In the `Stud _ DOB` type '`01/05/1997`' and in `DateOfAdmission` field, type `05/07/2014`. A message box displaying the validation text is displayed as shown in figure 2.29.

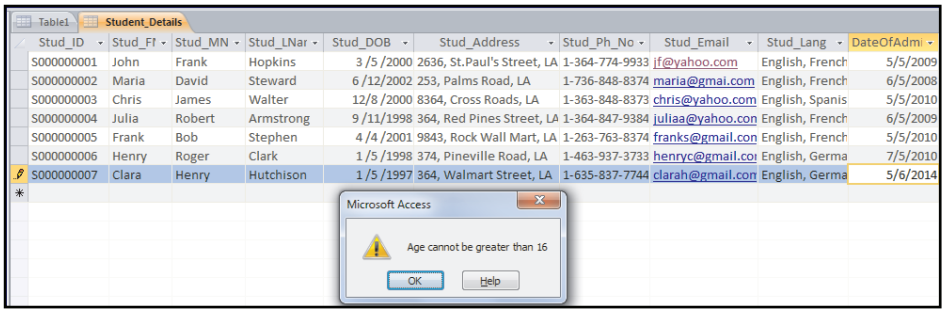


Figure 2.29: Creating Table Level Validation

Concepts

When the DateOfAdmission is specified, Access 2010 internally calculates the difference of DateOfAdmission and Stud_DOB fields and the result comes to 17. This leads to failure of validation rule and Access 2010 displays the validation text.

8. Click **OK**. Change the value to 5/6/2010 and save the table.

2.5.5 Modifying and Updating Data

To update data in a table, perform the following steps:

1. Open the Student _ Details table in the **Datasheet View**.
2. Modify the required column. For example, change the Stud _ LName value for the student with Stud _ ID as 'S000000006' to 'Clovis' and move the cursor to another cell using the arrow key or the mouse to save the change.
3. Similarly, change the phone number of the student with Stud _ ID as 'S000000003' to '1-264-475-3646'.

2.5.6 Deleting Data

Access 2010 allows deletion of data in individual columns as well as deletion of entire row. However, if the column is marked as Required, Access 2010 will not allow leaving it blank.

To delete an entire record from the table, perform the following steps:

1. Open the Student _ Details table in the **Datasheet View**.
2. Select the record to be deleted. For example, select the record with Stud _ ID as 'S000000007'.
3. Right-click and select **Delete Record** from the menu as shown in figure 2.30.

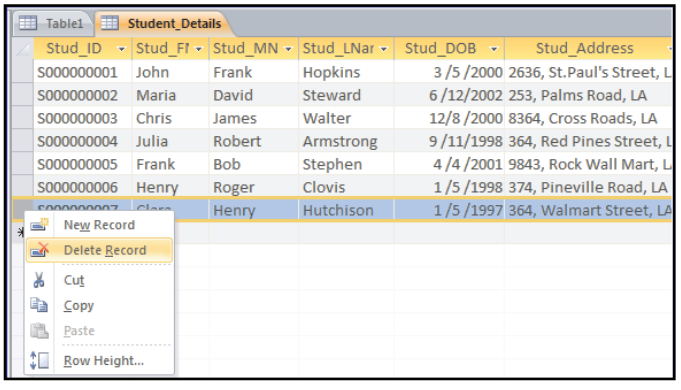


Figure 2.30: Deleting a Record from the Table

4. Click **Yes** in the warning message asking for confirmation of deletion. The record will be deleted from the table.

Note - A record can also be deleted by selecting the record and clicking **Delete** option in the **Records** group of the **Home** tab.

2.6 Data Access

At times, a user may require viewing only a certain section of data that meets a particular criterion. For example, a user may want to view data only about those students whose name starts with 'H' or whose date of birth is greater than January 01, 2000. To do such an analysis in the **Datasheet View**, the **Ribbon** includes a group named **Sort & Filter** provided in the **Home** tab. Also, the **Datasheet View** shows columns with a down arrow on the right side which also provides sort and filter options as shown in figure 2.31.

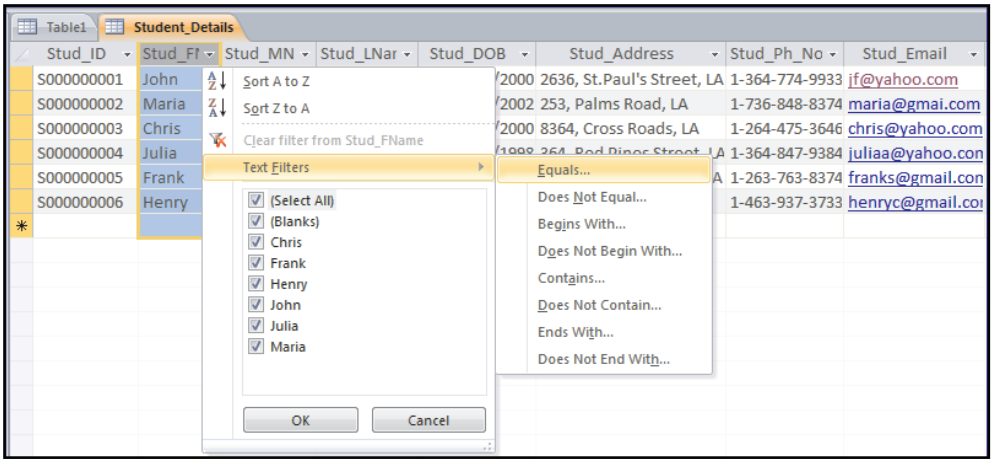


Figure 2.31: Sort and Filter Options

2.6.1 Sorting Data

To sort the data in the ascending order of `Stud_FName` column, perform the following steps:

1. Open the `Student _ Details` table in **Datasheet View**.
2. Click in any cell of the `Stud _ FName` column and select **Ascending** from the **Sort & Filter** group of the **Home** tab. The data will be sorted in the ascending order of the `Stud _ FName` column as shown in figure 2.32.

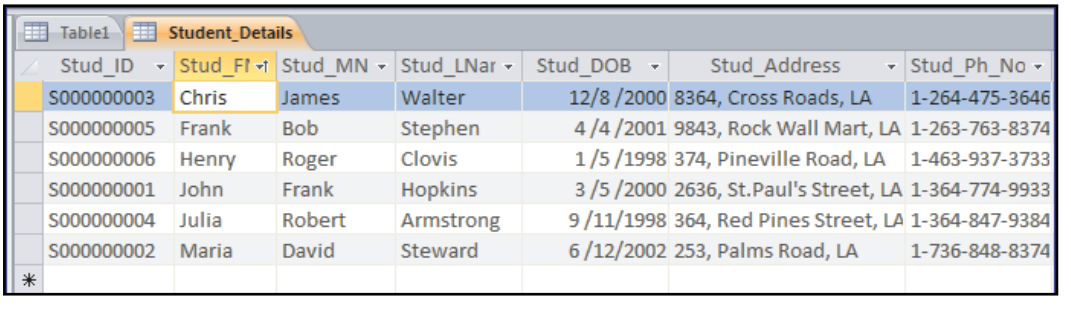


Figure 2.32: Records Sorted in Ascending Order of Stud_FName Column

Concepts

Note - To sort the records in the ascending order of `Stud_FName` column by using the down arrow, click the down arrow next to the column header and select **Sort A to Z** from the menu.

2.6.2 Finding and Replacing Data

In some cases, a user may have to make a lot of changes in the data with the least amount of effort possible. In such cases, the find and replace feature can be used. For example, to change 'LA' to 'Chicago' in the entire table using Access 2010, the user can use **Find and Replace** tool.

To use **Find and Replace** tool, perform the following steps:

1. Open the `Student _ Details` table in **Datasheet View**.
2. Click the **Find** option in the **Find** group of the **Home** `tab`. The **Find and Replace** dialog box is displayed.
3. Type '`LA`' in the **Find What** text box and click the **Replace** tab.
4. Type '`Chicago`' in the **Replace With** text box.
5. Select **Current document** from the **Look In** drop-down list, **Any Part of Field** from the **Match** drop-down list, **All** from the **Search** drop-down list, and select the **Match Case** check box as shown in figure 2.33.

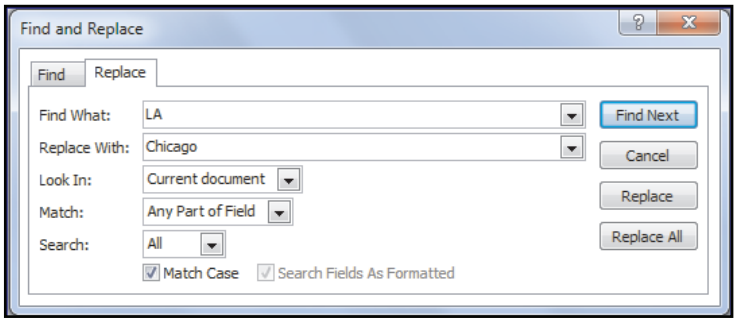


Figure 2.33: Find and Replace Dialog Box

6. Click **Replace All**. A confirmation message box will be displayed.
7. Click **Yes**. The text '`LA`' in the `Stud _ Address` column will be changed to '`Chicago`' as shown in figure 2.34.

Table1 | Student_Details

Stud_ID	Stud_FI	Stud_MN	Stud_LNar	Stud_DOB	Stud_Address	Stud_Ph_No
S000000003	Chris	James	Walter	12/8 /2000	8364, Cross Roads, Chicago	1-264-475-3646
S000000005	Frank	Bob	Stephen	4 /4 /2001	9843, Rock Wall Mart, Chicago	1-263-763-8374
S000000006	Henry	Roger	Clovis	1 /5 /1998	374, Pineville Road, Chicago	1-463-937-3733
S000000001	John	Frank	Hopkins	3 /5 /2000	2636, St.Paul's Street, Chicago	1-364-774-9933
S000000004	Julia	Robert	Armstrong	9 /11/1998	364, Red Pines Street, Chicago	1-364-847-9384
S000000002	Maria	David	Steward	6 /12/2002	253, Palms Road, Chicago	1-736-848-8374

Figure 2.34: Result of Find and Replace Tool

8. Close the **Find and Replace** dialog box.

Note - To replace the text one by one, a user can click **Find Next** and **Replace** buttons instead of **Replace All**.

2.6.3 Filtering Data

Consider a situation when you want a list of all female students. In such a case, one can use the filter functionality of Access 2010.

To filter data according to the gender, perform the following steps:

1. Open the `Student _ Details` table in the **Design View**.
2. Create a new column named '`Stud _ Gender`' with data type set to **Text** by inserting a new row before the `Stud _ DOB` field.
3. Set the **Field Size** to 1.
4. Click the **Save** icon to save the table and open it in the **Datasheet View**.
5. Type the gender of the students in the table as shown in figure 2.35.

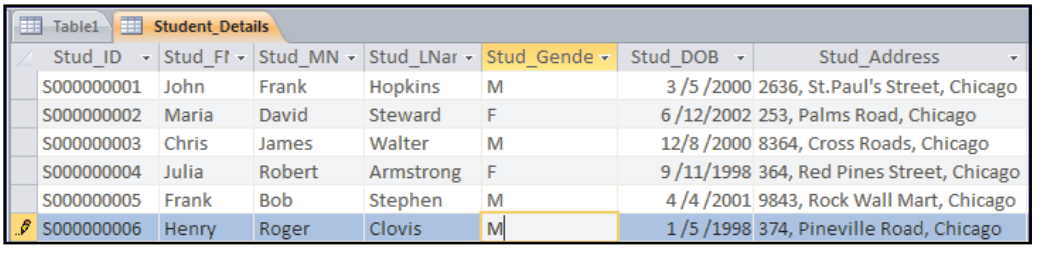


Figure 2.35: Inserting Data in the Stud_Gender Column

6. Right-click the '`F`' data in any cell and select '`Equals"F"`' option from the menu as shown in figure 2.36.

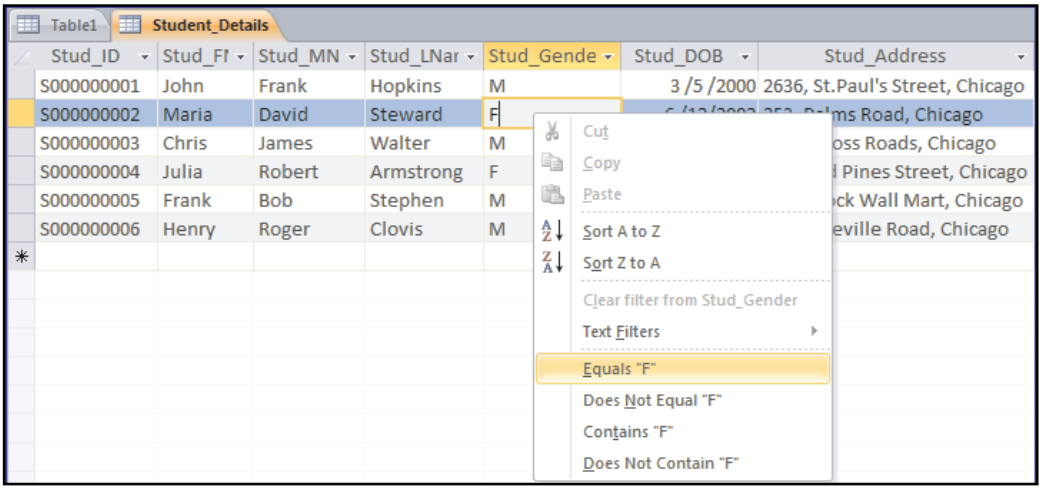


Figure 2.36: Using Filters

Concepts

The filtered table is shown in figure 2.37.

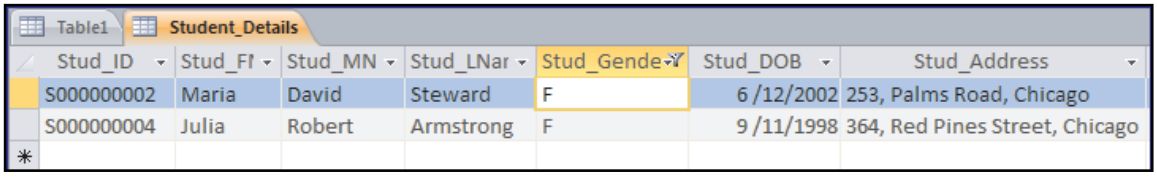


Figure 2.37: Result of Filtering

Notice the filter icon next to the `Stud_Gender` column header.

7. To remove the filter, click the filter icon and select the **Select All** check box from the menu. The table will display all the records.

Similarly, one can filter dates by using **Date Filters**.

Note - Filters can also be removed by clicking **Toggle Filter** option in the **Sort & Filter** group of the **Home** tab. The other filter options such as **Does Not Equal**, **Contains**, and so on can also be used from **Text Filters** according to requirement.

2.7 Establishing Relationships

A major advantage of an RDBMS is to establish among tables relationships along with constraints. In the real world where there is so much information, it is not possible to store all the data in one table. Hence, it is mandatory for the data to be divided into several tables. Later, the tables can be connected using a common field.

2.7.1 Defining a Primary Key

The primary key of a relational table is a field that helps to uniquely identify each record. A primary key may be created from a single attribute or combination of more than one attribute. For example, in the `Student_Details` table containing records of students of a school, the student ID represented by the `Stud_ID` field can be selected as the primary key. The reason is that the student Id of each student will always remain unique. However, a student's first name, last name, or address cannot be selected as primary key since more than one student can have the same first name, last name, or address.

A primary key field of one table can be included in other tables to refer back to the parent table that contains the primary key. In the other tables, the primary key field is referred to as foreign key.

Access 2010 provides internally generated unique identifier in the form of **AutoNumber** data type. The **AutoNumber** data type is used to automatically increment the field every time a new record is created. To create an **AutoNumber** field, select the **AutoNumber** data type from the **Data Type** drop-down list while creating a table in **Design View**.

The characteristics of a good candidate for primary key are as follows:

➔ It helps to uniquely identify each row.

➔ It cannot be empty or null.

➔ The values of a primary key rarely (never) change.

Any field that is lacking in one or more of the characteristics for a good candidate key becomes a poor choice for a primary key. Table 2.4 lists some field that would be poor candidates for a primary key.

Poor Candidate for Primary Key	Reason
Name	Not unique. Also, likely to be changed.
Phone number	May be changed.
E-mail address	May be changed.
Social Security Number (SSN)	Private information not allowed to be shared by government. Some people may not have a SSN.
Address	Mixed information and hard to maintain.

Table 2.4: Poor Candidates for Primary Key

To create a primary key, perform the following steps:

1. Open the `Student _ Details` table in the **Design View**.
2. Select the first row consisting of the `Stud _ ID` column.
3. Click the **Primary Key** option in the **Tools** group of the **Design** tab as shown in figure 2.38.

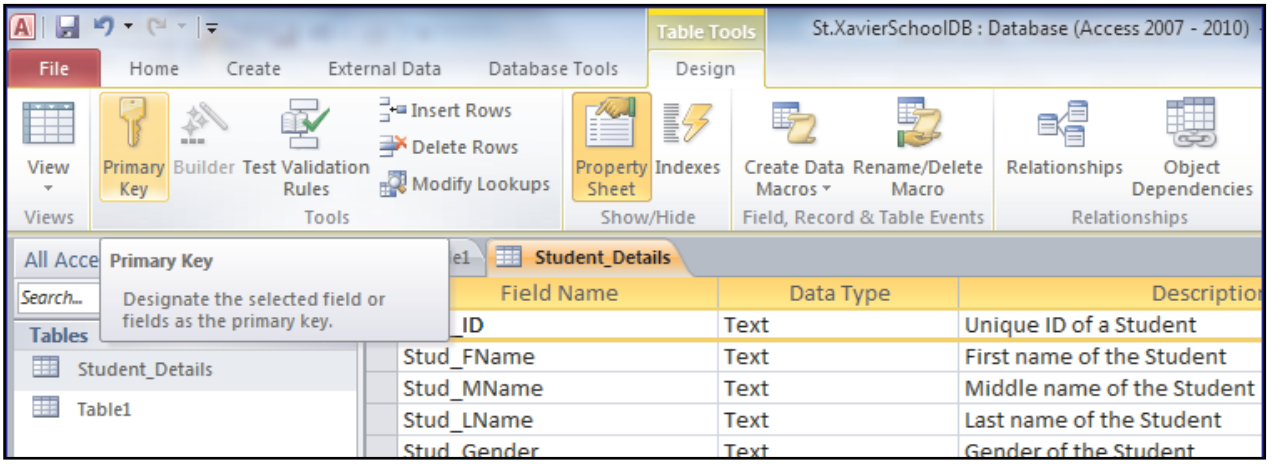


Figure 2.38: Setting the Primary Key

Concepts

A key indicator showing the primary key is added to the left of the Stud_ID field as shown in figure 2.39.

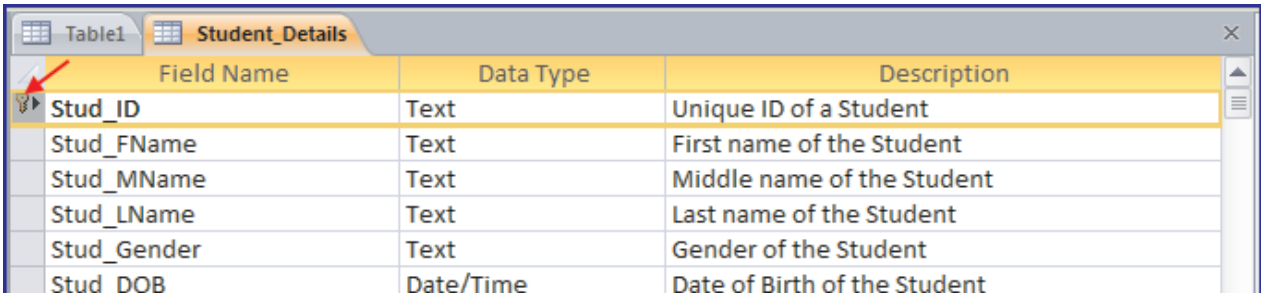


Figure 2.39: Primary Key Indicator Added to the Field

Note - Primary key can also be set by right-clicking the field name and selecting **Primary Key** option from the menu.

4. To remove the primary key, click the highlighted **Primary Key** option in the **Tools** group of the **Design** tab. The primary key indicator will be removed from the Stud _ ID column.

Note - Primary key can also be removed by right-clicking the field name and selecting **Primary Key** option from the menu. If the table is part of a relationship, Access warns the user to first delete the relationships.

2.7.2 Defining and Modifying a Multifield Primary Key

In some cases, a user may want to use more than one field as a primary key for a table. For example, in an Order_Details table storing details for orders may use two fields as primary key namely, Order_ID and Product_ID. Such a primary key that uses more than one field is also referred to as a composite key.

To create a multifield primary key, perform the following steps:

1. Open the Student _ Details table in the **Design View**.
2. Select the Stud _ FName column. Hold down the **Ctrl** key and select the Stud _ LName as well as Stud _ DOB columns to select all of them together. The selected columns will get a dark gray shade on the left corner.
3. Click the **Primary Key** option in the **Tools** group of the **Design** tab. A primary key indicator is added to left of all the three fields as shown in figure 2.40.

Concepts

Table1 | Student_Details

Field Name	Data Type	Description
Stud_ID	Text	Unique ID of a Student
Stud_FName	Text	First name of the Student
Stud_MName	Text	Middle name of the Student
Stud_LName	Text	Last name of the Student
Stud_Gender	Text	Gender of the Student
Stud_DOB	Date/Time	Date of Birth of the Student
Stud_Address	Text	Addres of the Student
Stud_Ph_No	Text	Phone number of the Student
Stud_Email	Hyperlink	E-mail address of the Student
Stud_Lang	Text	Languages known to the Student
DateOfAdmission	Date/Time	Date of admission of the Student

Figure 2.40: Multifield Primary Key

4. To remove the primary key, click the highlighted **Primary Key** option in the **Tools** group of the **Design** tab. The primary key indicator will be removed from all the selected columns.

2.7.3 Understanding Table Relationship

At times, it is useful to create relationships among tables as they allow a user to add and view related data quickly and easily as well as make data maintenance easier. However, creating relationship also brings up issues such as data integrity. This means, while creating, modifying, or deleting relationships, one needs to take care of data existing in the dependent tables.

While using tables that are part of a relationship, it is very important ensure accuracy of data from one table to the other. Data integrity or referential integrity can be used to verify that two tables are related through one or more than one field on each table that is used as the primary key and the foreign key.

Two fields having the same data type can only be used to establish a relationship between two tables. Also, the tables that are part of the relationship must belong to the same database.

2.7.4 Types of Relationships

The different types of relationships between tables are as follows:

➔ **One-to-one:**

This type of relationship indicates that a record in the primary table is associated to one and only one record in the child table. This relationship is rarely used because the fields that get added to the child table are usually placed in the primary table itself since they will not be repeated for another record.

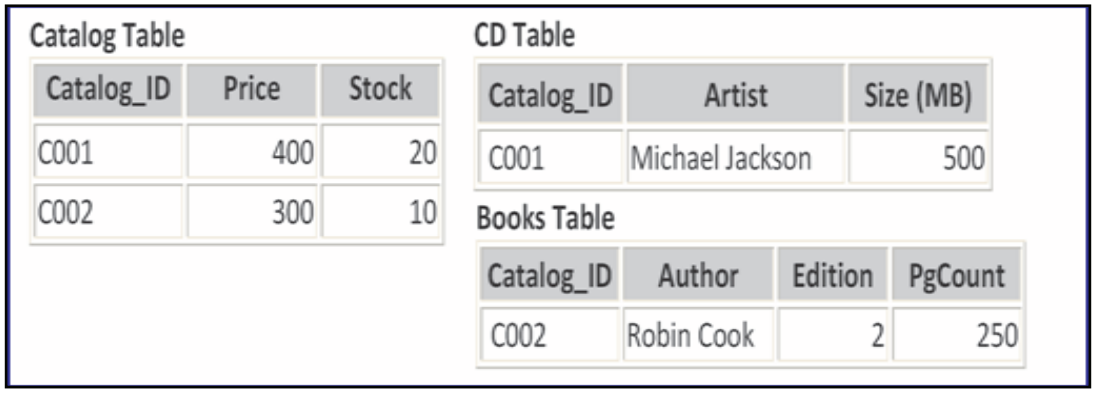
For example, consider a table named `Catalog` used to store details about the items sold by a store. The table stores information such as `Catalog` number, `Size`, `Price`, and other such data.

Concepts

At times, it might be required to store some more specific detail about an individual item such as `Author Name` and `Edition` for a book or `Artist Name` and `Number of Tracks` for a CD. One can create a table structure to fit all the data in the same table, but later when some new field comes up, the table structure will not work.

Instead, a user can create another table to store the item-specific detail and link it to the primary table with a one-to-one relationship on the primary key. This will lead to creation of one master table called `Catalog` and one or more child tables such as `Books`, `CD`, and so on.

Figure 2.41 shows a one-to-one relationship created between the `Catalog` and `Books` tables and the `Catalog` and `CD` tables.



Catalog Table

Catalog_ID	Price	Stock
C001	400	20
C002	300	10

CD Table

Catalog_ID	Artist	Size (MB)
C001	Michael Jackson	500

Books Table

Catalog_ID	Author	Edition	PgCount
C002	Robin Cook	2	250

Figure 2.41: One-to-One Relationship

By splitting item-specific data into separate tables, the one-to-one relationship has saved the doubling of the number of fields in the `Catalog` table. Also, it helps to split the database into more discrete entities. This table structure helps to get all the general information about a particular item from the `Catalog` table and item-specific information from the respective sub tables by using the primary key `Catalog_ID`.

➔ **One-to-many:**

This type of relationship indicates that a record in the primary table is associated to more than one record in the child table. However, each record of the child table is associated with only one record of the primary table.

For example:

Consider two tables namely, `Emp_Detail` and `Sale_Detail` as shown in figure 2.42.

Concepts

Emp_Detail Table

Emp_ID	EmpName
E001	Clark Harrison
E002	Roger Burt

Sale_Detail Table

Sale_ID	Emp_ID	Item_Name	Item_Qty
S001	E001	Screws	1000
S002	E001	Nuts	1000
S003	E002	Nuts	2000
S004	E001	Clamps	500

Figure 2.42: One-to-Many Relationship

The two tables have the `Emp_ID` field in common and can be linked together using the `Emp_ID` field. After linking the tables, the details of sales made by each salesman can be found by providing the `Emp_ID`. Another advantage of linking is that the two separate set of details can be modified individually.

➔ **Many-to-many:**

This type of relationship indicates that a record in the primary table is associated to more than one record in the child table and vice versa. This type of relationship is also very common. For example: one student can enroll for many courses and a course can have more than one student.

Similarly, one order can contain many items, and one item can appear in many orders.

This relationship is a little more complex than the one-to-many because, apart from the two primary tables, it is required to create a third table to join the two tables.

For example, consider two tables `Course` and `Student` that store the general information about the courses and students of a school. However, it is not possible to store information about the courses that a student attempts in the Student table because a student can enroll for more than one course. Similarly, one course can have several students enrolled to it. Combining the data will lead to redundancy or repetition of data.

In such a case, one needs to create a third table that holds the foreign key of both the primary tables as shown in figure 2.43.

Course Table

Course_ID	Title
C001	PHP
C002	ASP .NET
C003	JAVA

Student Table

Stud_ID	Stud_Name	Stud_DOB
S001	Mark	12/12/1999
S002	Larry	4/5/2001
S003	Clara	6/8/2000
S004	Julia	7/5/2000

Stud_Course Table

Stud_ID	Course_ID
S001	C001
S001	C003
S002	C002
S003	C002
S003	C003
S004	C001
S004	C003

Figure 2.43: Many-to-Many Relationship

From figure 2.43, it is clear that one student enrolls for more than one course and one course consists of many students. This table structure removes all the redundant data from the `Stud_Course` table that holds only two columns for the IDs of students and courses that they have enrolled into. Now, any changes made to the structure of the `Student` or `Course` table will not affect the `Stud_Course` table. Also, the `Stud_Course` table can be modified independent of the two parent tables.

2.7.5 Creating Relationships

Access 2010 allows a user to link tables by creating relationships between them. After creating a valid relationship between two tables, one must ensure that when data changes in the parent table, this change gets reflected in the child table.

For example, if an employee changes her last name after getting married, the change in one table should be reflected in all the other related tables, such as the one used to process her transactions such as salary and leave report without having to make the change in each table. Similarly, when the data is deleted from the primary table, the change must be reflected in the child tables also. Such kind of changes from the parent to the child are called cascading changes.

To ensure data integrity, Access 2010 provides three check boxes in the **Edit Relationships** dialog box as shown in figure 2.44.

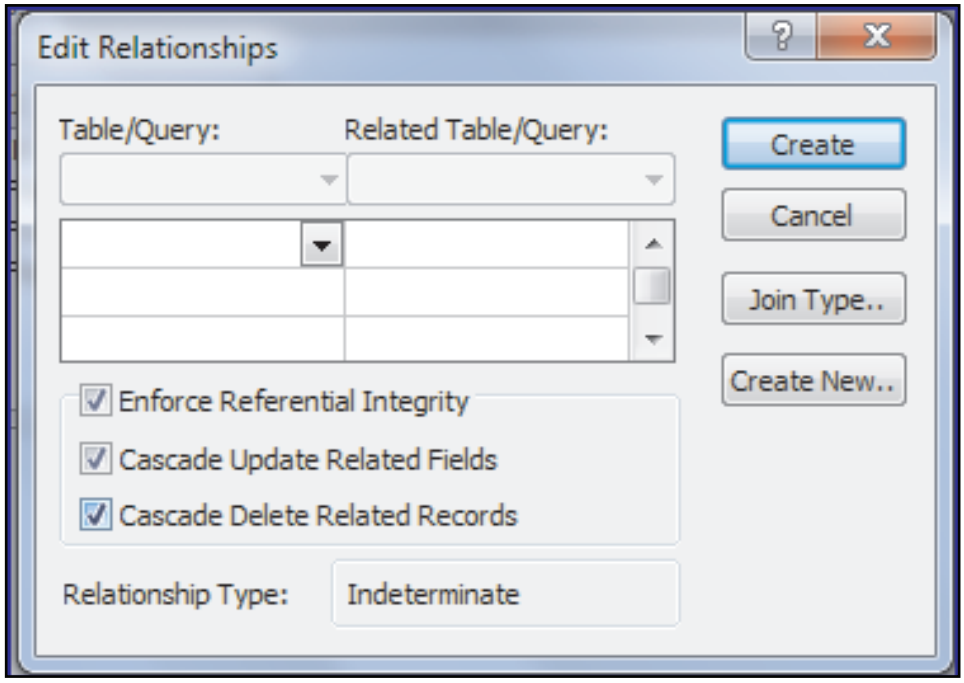


Figure 2.44: Edit Relationships Dialog Box

To make Access 2010 monitor the data flow, click the **Enforce Referential Integrity** check box. This will enable the other two check boxes.

➔ Creating One-to-Many Relationship

To create a one-to-many relationship, perform the following steps:

1. Open the `Student _ Details` table in **Design View** and delete the `Stud _ Address` column.
2. Save and close the table.
3. Right-click the **Table1** tab and click **Save**.
4. Type the name as '`Student _ Address`'.
5. Open the `Student _ Address` table in the **Design View**.
6. Create columns in the table as shown in figure 2.45.

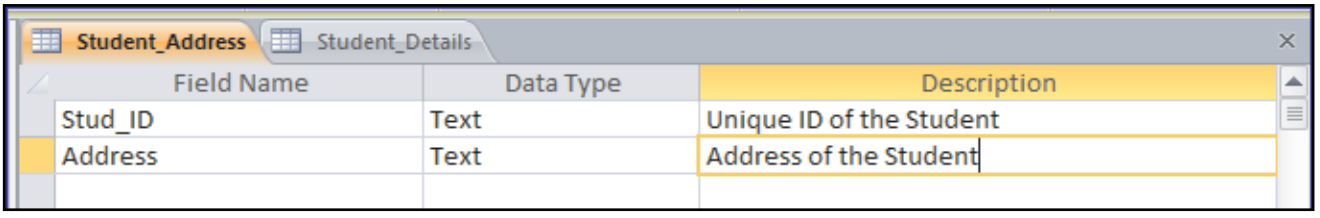


Figure 2.45: Student_Identification Table

7. Change the **Field Size** of `Stud _ ID` to `10`, `Address` to `100`.
8. Click the **Save** icon to save the changes.
9. Open the `Student _ Address` table in the **Datasheet View** and add two records for `Stud _ ID S000000001`, one for `S000000002`, and two records for `S000000003` with the addresses.
10. Save and close the table.
11. Click **Database Tools** tab and click **Relationships** option from the **Relationships** group. The **Show Table** dialog box is displayed.
12. Double-click both the tables: `Student _ Details` and `Student _ Address` to add them to the **Relationships** pane. Alternatively, one can select the table name and click **Add**.
13. Click **Close** on the **Show Table** box to close it.
14. Select the `Stud _ ID` field of the `Student _ Details` table and hold and drag the mouse over the `Stud _ ID` of the `Student _ Address` table. This will open the **Edit Relationships** box showing both the tables and the fields selected.
15. Select the **Enforce Referential Integrity** check box. This will enable the other two check boxes.
16. Select the **Cascade Update Related Fields** and **Cascade Delete Related Records** check boxes as shown in figure 2.46.

Concepts

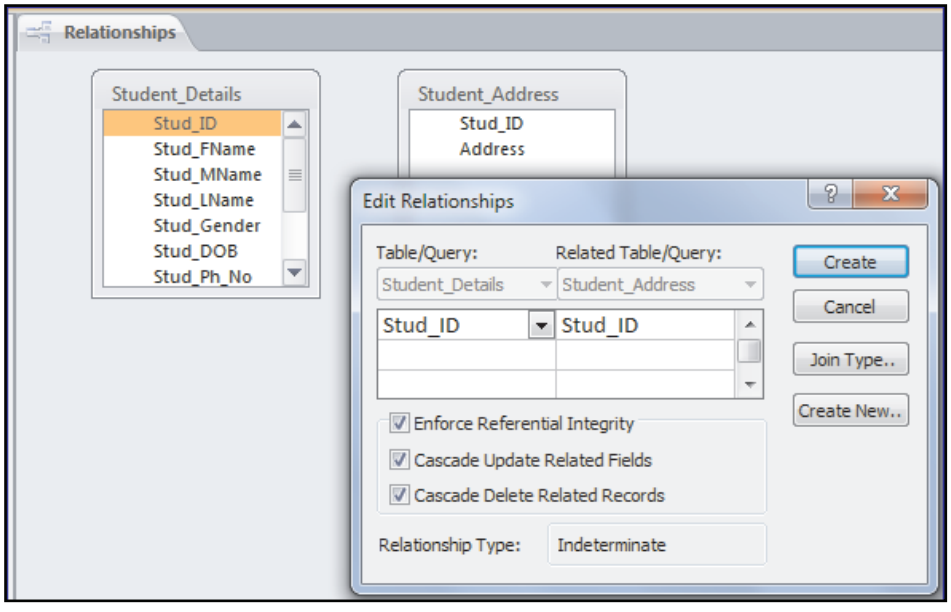


Figure 2.46: Edit Relationships Between Two Tables

17. Click **Create**. The relationship will be displayed as shown in figure 2.47.

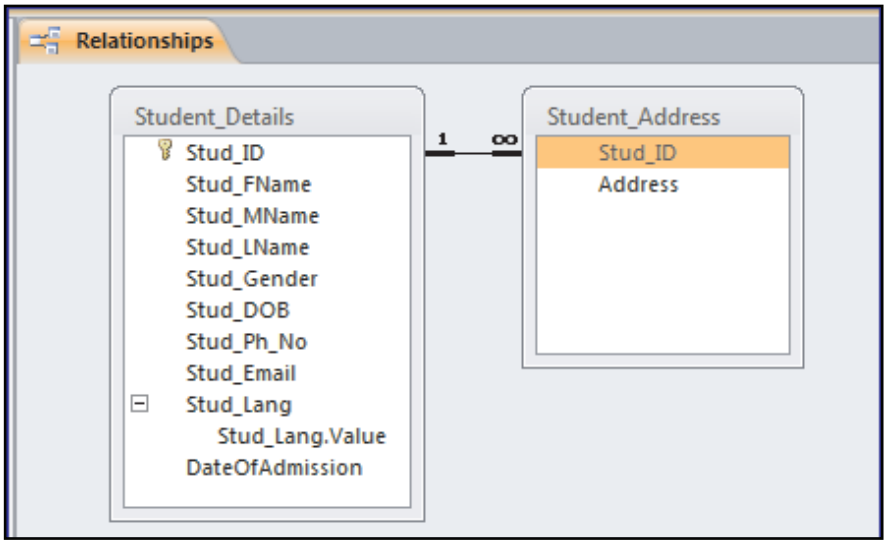


Figure 2.47: Relationship between Student_Details and Student_Identificaiton Tables

The symbol 1-∞ indicates a one-to-many relationship.

➔ Creating Many-to-Many Relationship

For a many-to-many relationship, a junction table containing at least three fields must be created. The first field, which is not much useful here, is set as the primary key, similar to almost every table. The other fields hold the foreign key from other tables.

Concepts

To create a many-to-many relationship, perform the following steps:

1. Click **Home** → **Table Design** from the **Tables** group to create a new table.
2. Create the fields and save the **Table** with the name `Course _ Details` as shown in figure 2.48.

Course_Details

Field Name	Data Type	Description
Course_ID	Text	Unique Id of the course
Course_Title	Text	Title of the additional courses
Course_Duration	Number	Duration in weeks

Figure 2.48: Course_Details Table

3. Set `Course _ ID` as the primary key and change the **Field Size** property of `Course _ ID` to `4`, `Course _ Title` to `100`, and `Course _ Duration` to `Integer`.
4. Add data to the table as shown in figure 2.49.

Course_Details

Course_ID	Course_Title	Course_Dura
C001	Social Sciences	10
C002	Internet	5
C003	Mind Sport	10

Figure 2.49: Data Added to Course_Details Table

5. Create another table named `Stud _ Course _ Details` as shown in figure 2.50.

Stud_Course_Details

Field Name	Data Type	Description
Stud_Course_ID	AutoNumber	Unique ID of Student and Course transaction
Stud_ID	Text	Unique ID of student
Course_ID	Text	Unique ID of course

Figure 2.50: Stud_Course_Details Table

6. Set the **Data Type** of `Stud _ ID` to **Lookup Wizard**.
7. In the first screen, select the '**I want Lookup field to get values from another table or query**' option button.
8. In the second screen of the **Lookup Wizard**, select the `Student _ Details` table and click **Next**.
9. Select the `Stud _ ID` column and add it to the right list box using the **>** button and click **Next**.

Concepts

10. Select `Stud _ ID` from the **Ascending** drop-down list and click **Next**.
11. Click **Next**. Type the name as `Stud _ ID` and select the **Enable Referential Integrity** check box.
12. Select the **Cascade Delete** option button and click **Finish**.
13. Similarly, for the `Course _ ID`, create a **Lookup Field** based on the `Course _ ID` column of the `Course _ Details` table.
14. Click **Database Tools** → **Relationships** → **All Relationships** from the **Relationships** group. The **Relationships** pane is displayed as shown in figure 2.51.

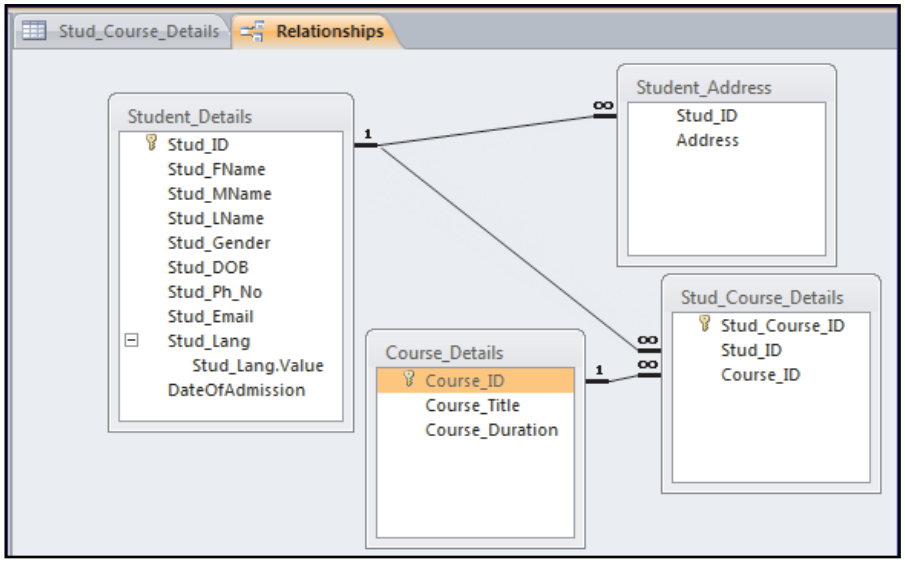


Figure 2.51: Many-to-Many Relationship

15. Save and close the **Relationships** pane.
16. Open the `Stud _ Course _ Details` table in the **Datasheet View** and add data to it as shown in figure 2.52.

Concepts

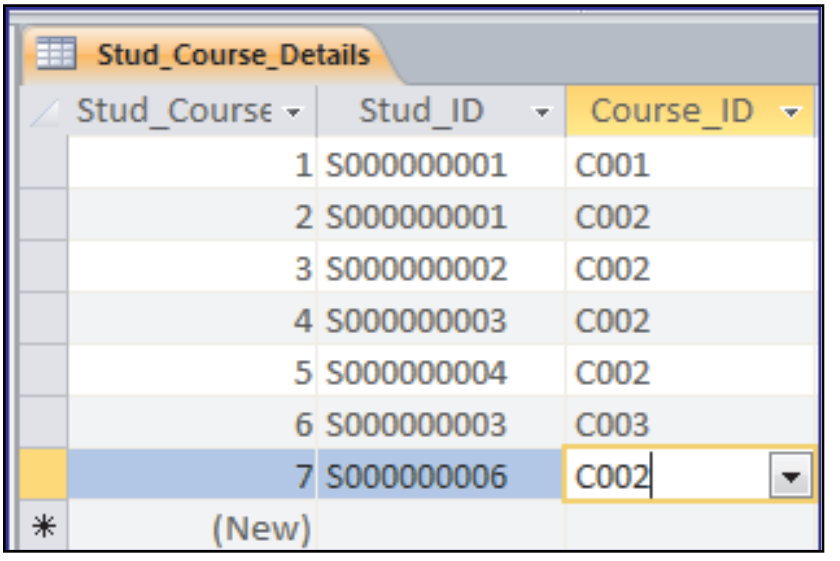


Figure 2.52: Data Added to Stud_Course_Details Table

Observe that Access 2010 provides the drop-down list of existing students and courses for the user to select. This is because the `Stud_ID` and `Course_ID` fields are created as lookup fields. Also, the `Stud_Course_ID` is auto generated.

17. Open the `Student _ Details` and `Course _ Details` tables in the **Datasheet View**. The tables will have '+' symbols on the left corner of each record. On expanding the '+' symbol, the related records of that record in the `Stud _ Course _ Details` table will be displayed as shown in figure 2.53.

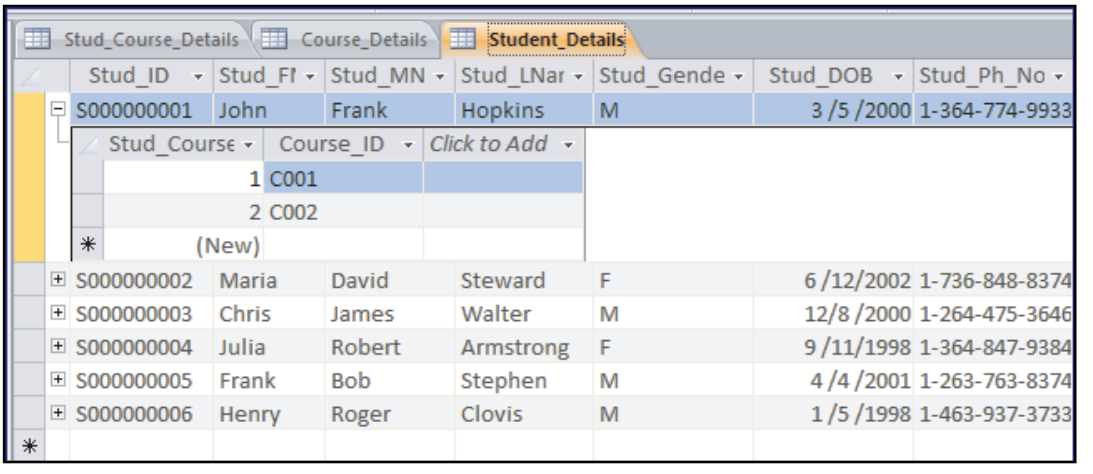


Figure 2.53: Course Details Related to a Student

In some cases, the **Insert Subdatasheet** dialog box will open. Click the `Stud_Course_Details` table and then, click **OK**. The related data will be displayed as shown in figure 2.53.

18. Save and close the `Stud _ Course _ Details` table.

Concepts

19. Open `Student _ Details` table in the **Datasheet View** and delete the record of student with `Stud _ ID` value as `S000000006`. A warning message box is displayed as shown in figure 2.54.

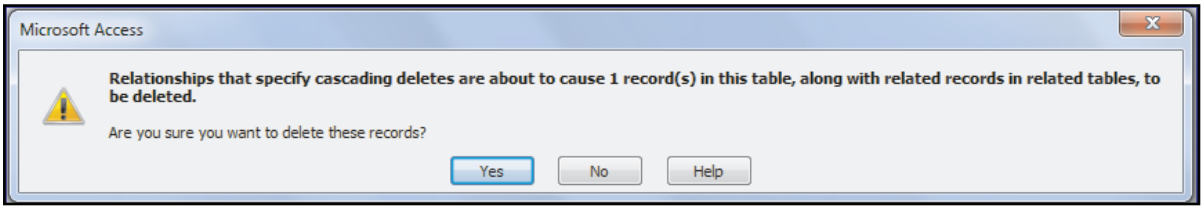


Figure 2.54: Cascading Deletes Warning Box

20. Click **Yes**. The record `S000000006` will be deleted from the `Student _ Details` table. The corresponding record in the `Stud _ Course _ Details` table will also be deleted.

2.8 Importing Data

Access 2010 allows importing data form database in several ways. The **Import & Link** features offer a good control and better flexibility over the data that is imported, and the way it is imported into the destination database. When data is imported from another database, Access 2010 generates a copy of the objects or data in the destination database. However, it does not alter the source database.

To add records from one database to an existing table in another database, it is preferable to import the records to a new table and then, create an append query. Records cannot be appended to an existing table during an import operation.

2.8.1 Importing Data to a New Table

Consider a situation wherein the school needs to make a separate copy of the `Student_Details` table for management purposes. The import feature can be used to copy the table with its data and definition to another database.

To import data to a new table, perform the following steps:

1. Create a new database named `St.XavierSchoolDB _ Mgmt`.
2. Click **External Data** → **Access** option in the **Import & Link** group. The **Get External Data** dialog box is displayed as shown in figure 2.55.

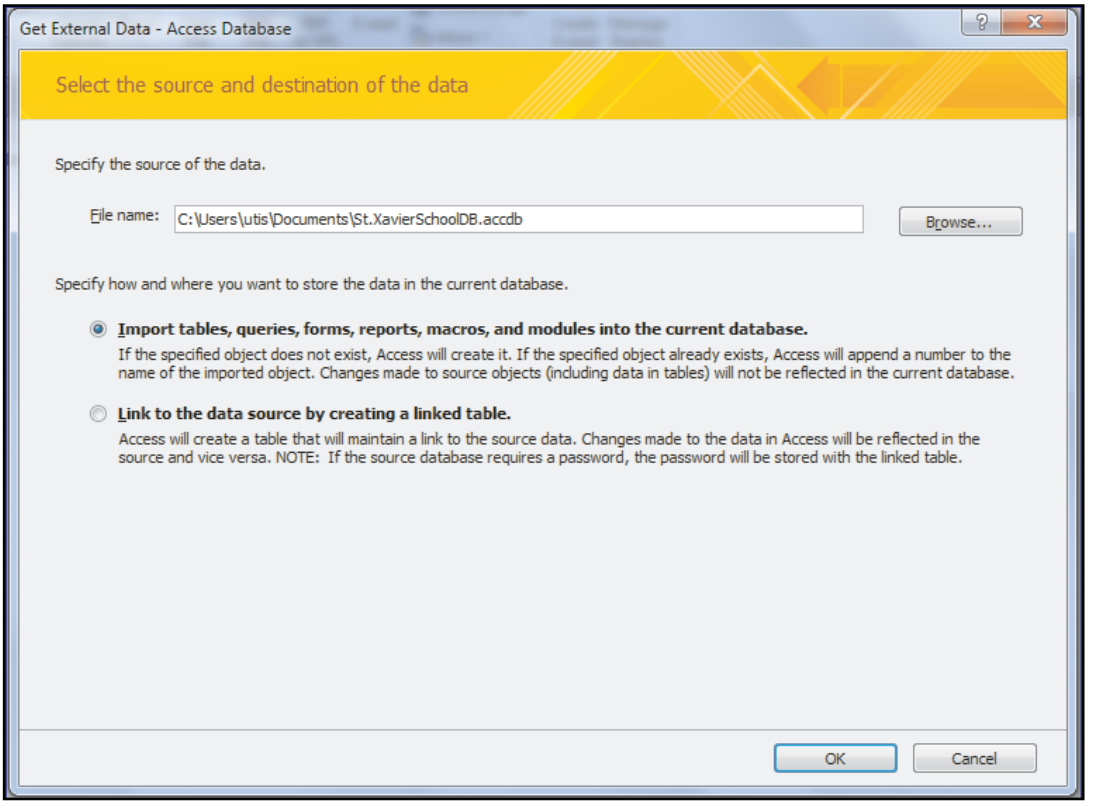


Figure 2.55: Get External Data Dialog Box

3. In the **File name** text box, type the name of the source database or click **Browse** to display the **File Open** dialog box and browse to the location of the `St.XavierSchoolDB.accdbfile`. For an **.mdb** or **.accdb** file, Access 2010 allows importing tables, forms, queries, macros, reports, and modules. For an **.mde** or **.accde** file, Access 2010 allows importing only tables.

4. Select the first option button named **Import tables, queries, forms, reports, macros, and modules into the current database** and click **OK**. The **Import Objects** dialog box is displayed.

5. Select the `Student _ Details` table in the **Tables** tab and click **Options**. The different import options are displayed at the bottom. Table 2.5 lists the description of the various options.

OPTION	DESCRIPTION
Relationships check box	Allows importing relationships between the selected tables.
Menus and Toolbars check box	Allows importing custom menus and toolbars from the source database. The menus and toolbars are displayed on the Add-Ins tab.
Import/Export Specs check box	Allows importing saved import or export specifications.
Nav Pane Groups check box	Allows importing custom Navigation pane groups.

Concepts

OPTION	DESCRIPTION
Definition and Data option button	Allows importing the structure and data of all the selected tables.
Definition Only option	Allows importing only the fields from the selected tables and not the data.
As Queries option button	Allows importing the selected queries as queries. Here, it is required to import all the tables used in the queries.
As Tables option button	Allows importing the queries as tables. Here, it is not required to import the tables.

Table 2.5: Options of the Import Objects Dialog Box

A user can also select other objects such as Queries, Forms, Reports, Macros, and Modules by selecting the respective tabs.

6. Select the **Relationships** check box from the **Import** section and then, select **Data and Definition** option button from the **Import Tables** section as shown in figure 2.56.

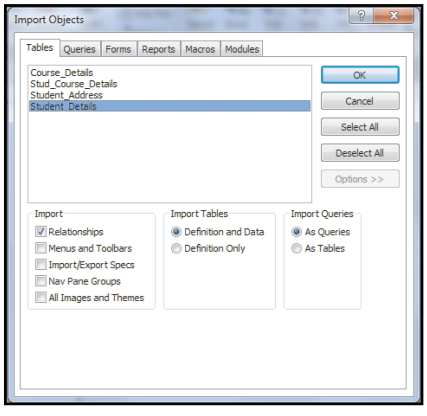


Figure 2.56: Import Objects Dialog Box

The import operation does not modify or overwrite the existing tables or objects. If the destination database consists of an object with the same name as the source object, Access 2010 adds suffix such as 1, 2, 3, and so on to the name of the imported object. For example, if the destination database already has a table named `Student_Details`, the imported table will be named as `Student_Details1`. If the `Student_Details1` already exists, then the new table will be named as `Student_Details2`, and so on. To append records to and existing table in the destination database, the append query must be used instead of an import operation.

7. Click **OK**. Access 2010 will copy the data and open the **Save Import Steps** screen. It will display error messages if any problem is encountered. This screen allows a user to save the details of the import operation for future use.

8. Select the **Save import steps** check box to save the import operation details. This helps to avoid repeat the same steps later without having to use the wizard.

Concepts

9. In the **Save As** box that is activated, specify the name of the import as `Import-St.XavierSchoolDB` and in the **Description** box type '`Import details for Student _ Details table`' as shown in figure 2.57.

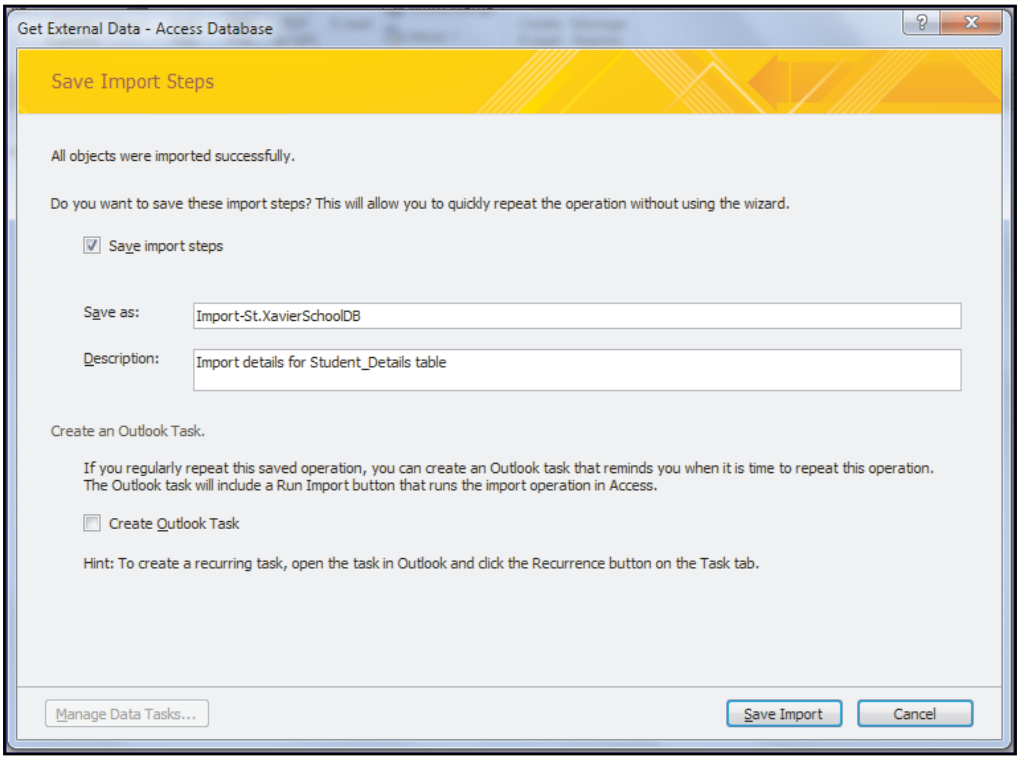


Figure 2.57: Saving the Import Operation Details

10. Click **Save Import**. The `Student _ Details` table is imported to the new database along with the data and definition. A user can now review the imported tables and other objects to verify that they were imported correctly.

11. To view the saved imports, click **Saved Imports** option in the **Import & Link** group. The **Manage Data Tasks** dialog box is displayed with the list of **Saved Imports** as shown in figure 2.58.

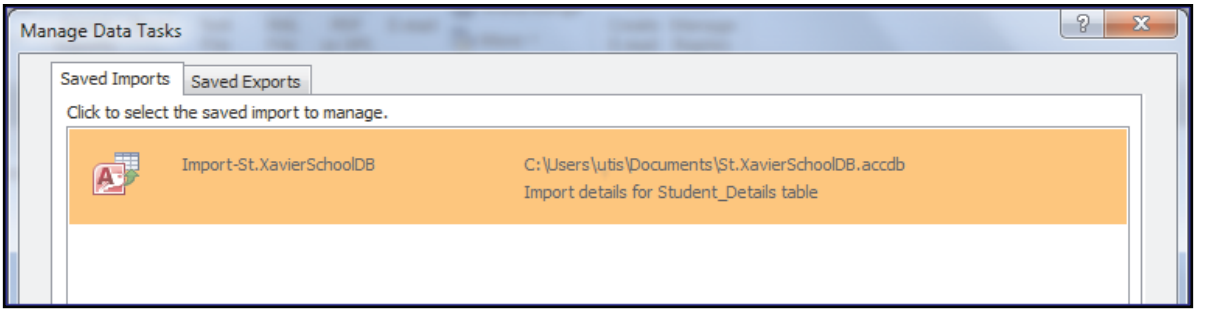


Figure 2.58: List of Saved Imports

A user can select the existing saved import to run the import operation again by clicking **Run**.

Concepts

2.8.2 Appending Records to a New Table

The append query can be used to fetch data from one or more tables and add it to another table.

Consider that the user copied the `Student_Address` table to the new database and later, a new record was added to the original table. To add the new rows to the destination table, a user can use the append query.

To use the append query to add new records, perform the following steps:

1. Copy the `Student _ Address` table to `St.XavierSchoolDB _ Mgmt` database using the Import operation.
2. Open the `St.XavierSchoolDB` database and add a record for student id '`S000000004`' in the `Student _ Address` table.
3. Click **Create** → **Query Design**.
4. Select `Student _ Address` table in the **Show Table** box and click **Add**.
5. Click **Close** to close the **Show Table** box.
6. On the **Design** tab, in the **Query Type** group, click **Append**. The **Append** dialog box opens.
7. In the **Append** dialog box, click the arrow in the **Table Name** box, and then, select `Student _ Address` table from the drop-down list.
8. Select the **Another Database** option button and browse to the path of `St.XavierSchoolDB _ Mgmt` database.
9. Click **OK**. In the design grid at the bottom, the **Append To** row appears and the **Show** row disappears.
10. In the design grid, specify the fields for the query and the criteria as shown in figure 2.59.

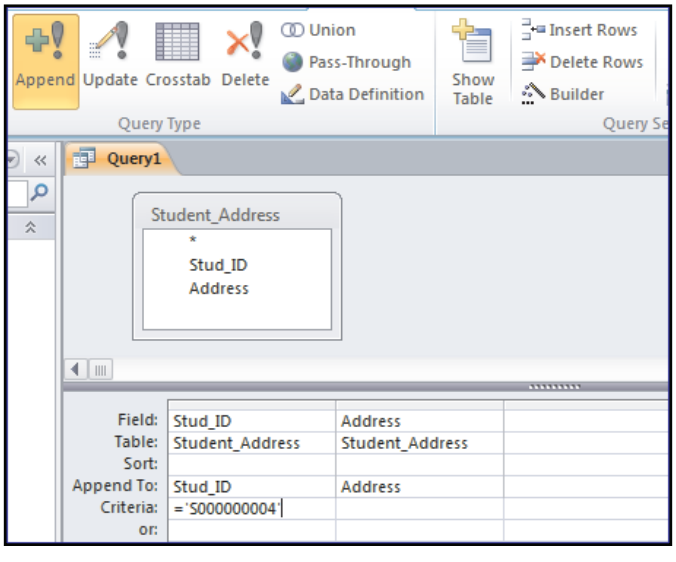


Figure 2.59: Append Query

Concepts

The criteria '=S000000004' will append the new record to the selected destination table and the data will be copied to the fields selected in the **Append To** row.

11. On the **Design** tab, in the **Results** group, click **Run**. A warning box for confirmation will appear as shown in figure 2.60.

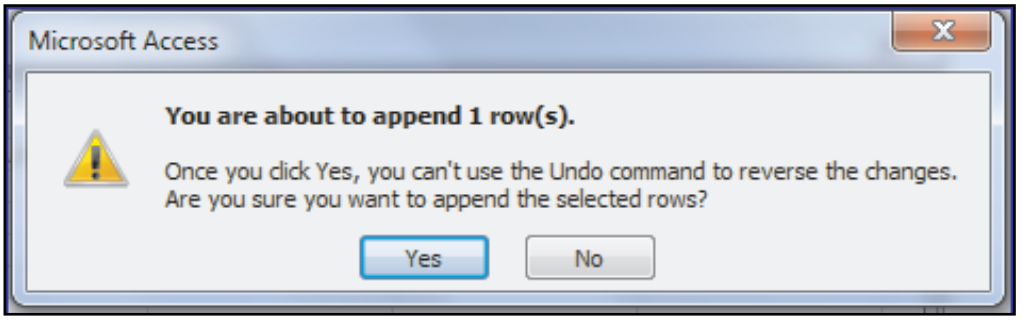


Figure 2.60: Append Query Warning Message

12. Click **Yes**.
13. Press **Ctrl+S** to save the query.
14. Type `Stud _ Address Query` as the name in the **Save As** dialog box. The query will be added to the **Tables** navigation pane in the **Queries** section.
15. Close the `St.XavierSchoolDB` database and open the `St.XavierSchoolDB _ Mgmt` database.
16. Double-click the `Student _ Address` table in the **Tables** navigation pane to open it in the **Datasheet View**. The new record for student `S000000004` will be shown added to the table.

Note - An append query cannot contain a multivalued field.

2.8.3 Importing Data as a Linked Table

Linking allows a user to connect to data in another database without having to import it. This helps to view and modify the latest data in both the source and destination databases. Also, it avoids the need to create and maintain two copies of the same data. Access 2010 allows linking only tables in another Access 2010 database. Link to forms, queries, macros, reports, or modules cannot be created.

On linking a table in a database, Access 2010 creates a new table called a linked table. This table maintains a link to the source table's fields and records. Any changes made to the data in the source table are automatically reflected in the linked table and vice versa. However, changes cannot be made to the structure of a linked table in the destination database.

To create a link to a table, perform the following steps:

1. Close the source database, that is, `St.XavierSchoolDB` and ensure that the database is not open on any other shared machine.
2. Open the destination database `St.XavierSchoolDB _ Mgmt`.

Concepts

3. On the **External Data** tab, in the **Import & Link** group, click **Access** option.
4. In the **File name** text box, type the name of the source database `St.XavierSchoolDB` or click **Browse** to display the **File Open** dialog box.
5. Select **Link to the data source by creating a linked table** option button, and then, click **OK**.
6. In the **Link Tables** dialog box, select the `Course _ Details` table. A user can change or remove the selection by clicking the table name again.
7. Click **OK**. Access 2010 creates the linked table as shown in figure 2.61.

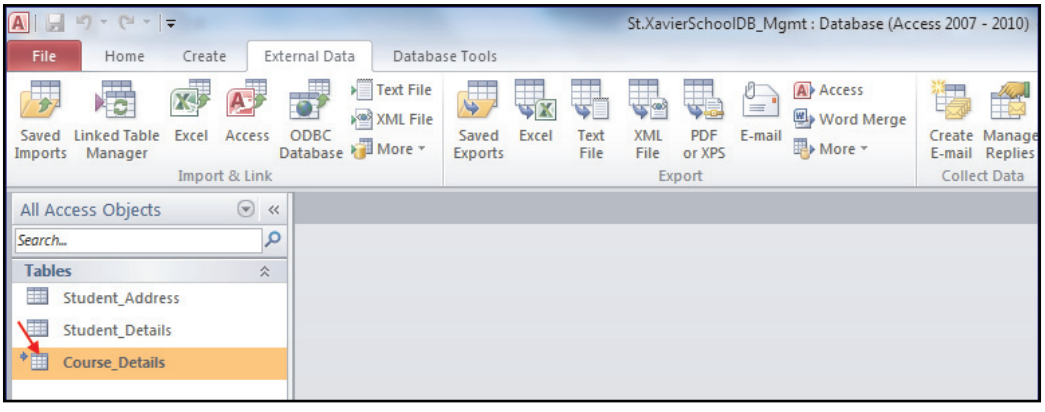


Figure 2.61: Creating a Linked Table

8. Double-click the `Course _ Details` table to open it in the **Datasheet View** to verify that the data looks correct.
9. Add a new record in the table for course C004. Give the `Course _ Title` as '`Human Biology`' and `Course _ Duration` as 15.
10. Save and close the database.
11. Open the original `St.XavierSchoolDB` database and double-click the `Course _ Details` table to open it in the **Datasheet View**. The new record for course `C004` will be added to the original table. This is because the table was linked to the destination database. So any change made to the table in the source or destination copy will be reflected in the other table also. Thus, linking a table offers a better advantage and more flexibility over the import operation.

2.9 Check Your Progress

1. The _____ data type uses the Expression Builder to create a calculated field by combining data of other existing fields.

(A)	Number	(C)	Look up
(B)	Calculated	(D)	AutoNumber

2. Which of the following statements about a table are true?

(A)	A table is a two-dimensional structure consisting of fields and records.
(B)	A field in a table is a category or an attribute used to store data about a particular object.
(C)	A complete or incomplete set of values belonging to each category of a particular item in the table is called a record of a table.
(D)	A table name should not reflect the type of the information that is to be stored into it.

3. Match the following relationship types with the corresponding descriptions.

Relationship Type		Description	
(A)	One-to-Many	1)	Indicates that a record in the primary table is associated to one and only one record in the child table
(B)	One-to-One	2)	Indicates that a record in the primary table is associated to more than one record in the child table and vice- versa
(C)	Many-to-Many	3)	Indicates that a record in the primary table is associated to more than one record in the child table

4. Which of the following options is used to unfreeze the fields in a table?

(A)	Select the fields and select **Unfreeze All Fields** in the **Home** tab.
(B)	Double-click the fields, and then, click **Unfreeze All Fields** in the **Unfreeze** dialog box.
(C)	Right-click the fields, and then, click **Unfreeze All Fields**.
(D)	Right-click the fields, and then, click **Unfreeze**.

Concepts

5. Identify the valid methods of creating a table.

(A)	Click **Create** → **Table Design** in the **Tables** group.
(B)	Click **Create** → **Table** in the **Tables** group.
(C)	Click **Table** → **Create Table** in the **Add & Delete** group.
(D)	Click **Table** → **Create Table Design** in the **Add & Delete** group.

6. Identify the correct sequence to set the column named `Prod _ ID` of the `Product` table as the primary key.

(A)	Select the first row consisting of the `Prod_ID` column.
(B)	Open the `Product` table in the **Design View**.
(C)	Click the **Save** icon to save the table.
(D)	Click the **Primary Key** option in the **Tools** group of the **Design** tab.

2.9.1 Answers

1.	B
2.	A, B, C
3.	A-3, B-1, C-2
4.	C
5.	A, B
6.	B, A, D, C

Summary

- A table is a collection of values arranged as a two-dimensional list according to their categories.
- A data type is a property of a field that determines the kind of data that the field will store.
- Input masks are used to set a pattern or a template for the data to be entered and give the data a consistent look.
- Multivalued fields in Access 2010 allow a user to select and store multiple values in a single field.
- The primary key of a relational table is a field that helps to uniquely identify each record.
- A many-to-many relationship indicates that a record in the primary table is associated to more than one record in the child table and vice versa.
- The Import & Link features offer a good control and better flexibility over the data that is imported, and the way it is imported into the destination database.

Try it Yourself

M/S. Siemens and Sons are a renowned chartered accountant consultancy firm in **Los Angeles**, **USA**. The management of the firm wishes to automate the functioning of the company by storing data in the computer instead of maintaining manual record sheets of the clients. The company needs to store the personal details, bank details, and similar other information about the customers. The manager of the company has asked you as part of the IT department to create a database for the company with the following facilities:

1. The personal details and bank details of a customer should be stored in separate tables.
2. A user should be able to edit data common to several records without much effort.
3. A user should be able to view the records based on some criteria such as Bank name or First name.
4. Restrictions must be placed on the format of customer's date of birth and Account number.
5. On selecting a customer in the personal details table, the associated bank details should also be displayed.
6. The bank details of the customers should be copied to another database for management purpose.
7. The personal details of the customers should be copied to another database such that changes made to it from will be automatically reflected in the source or destination copy.

Session - 2 (Workshop)

Working with Tables and Data

In this workshop, you will learn to:

- ➔ Create tables
- ➔ Create a lookup field
- ➔ Build relationships between tables

2.1 Creating Tables, Lookup Field, and Relationships

You will view and practice how to create tables and Lookup field. You will also view and practice how to create relationships between tables.

➔ Creating Tables

➔ Creating a Lookup Field

➔ Creating Relationships between Tables

Note - Please refer to the respective lab deliverable for demonstrations and guided simulations.

Session - 3

Working with Queries

Welcome to the Session, **Working with Queries**.

This session explains the concept of a query and its various types. The session also explains designing and modifying a query as well as adding criteria and functions to it. Further, the session explains querying multiple record sources and summarizing query results. The session also describes unions and subqueries as well as creation of calculated fields. Lastly, the session explains sorting and filtering of data within a query.

In this Session, you will learn to:

- Explain a query, its purpose, and its types
- Explain designing, modifying, and using query results
- Explain adding input, criteria, and functions to a query
- Explain querying multiple record sources and summarizing query results
- Describe Unions and Subqueries
- Explain creation of Calculated fields through Zoom box and Expression builder
- Explain sorting and filtering of data within a query

3.1 Introduction

Consider a situation where a company has created a database to store data about the product it manufactures. Now, when a user wishes to view, add, modify, or delete data, every time it is required to open the table and scroll to the particular record/s to manipulate it. This can be very difficult and time consuming especially when the table consists of a large number of records. For this reason, it is required to have a way by which the user can directly filter the required records or modify them without having to open the table again and again. This can be achieved by creating a query.

3.2 Queries

One of the major advantages of Access 2010 over Microsoft Excel is the ability to create queries and reports. Access 2010 queries allow a user to view, insert, modify, delete, and calculate fields and records from one or more tables of a database.

3.2.1 Overview of a Query

A query is a request made to one or more tables of a database for viewing data, results of action on data, or both. A query can be used to add, modify, or delete data, combine data from multiple tables, and even to perform calculations. Simply put, a query is a way to assemble information present in one or more tables.

Structured Query Language (SQL) is a standard query language used to create queries to view and modify data in a database. There are different types of SQL statements such as Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), and Transaction Control Language (TCL) statements.

➔ **Data Definition Language**

The DDL statements are used to make structural changes to the database such as creation, modification, and deletion of tables.

Table 3.1 lists the various DDL statements.

STATEMENT TYPE	DESCRIPTION	SYNTAX	EXAMPLE
CREATE TABLE	Creates a new table.	CREATE TABLE <Table name>(<FieldName1><Type> [(<Size>)] [NOT NULL], >(<FieldName2><Type> [(<Size>)] [NOT NULL] [,…]); where, <Table name>: Specifies the name of the table to be created. <FieldName1>, <FieldName2>: Specifies the name of the fields of the table. <Type>: Specifies the data type of the field. <Size>: Specifies the size of each field.	CREATE TABLE Customer (Cust_Id INTEGER PRIMARY KEY, Cust_Name TEXT(100) NOT NULL, Cust_Address TEXT(150));
ALTER TABLE	Modifies a table.	ALTER TABLE <TableName> ADD|DROP|ALTER COLUMN <FieldName1><Type> [(<Size>)] [NOT NULL], COLUMN (<FieldName2><Type> [(<Size>)] [NOT NULL] …; where, <TableName>: Specifies the name of the table to be modified. ADD: Keyword used for adding a column. DROP: Keyword used for removing a specified column. ALTER: Keyword used for modifying a specified column. COLUMN: Keyword used to specify the field. <FieldName1>, <FieldName2>: Specifies the name of the fields of the table. <Type>: Specifies the data type of the field. <Size>: Specifies the size of each field.	ALTER TABLE Customer ALTER COLUMN Cust_Name TEXT (200) NOT NULL;

Concepts

STATEMENT TYPE	DESCRIPTION	SYNTAX	EXAMPLE
DROP TABLE	Deletes a table.	DROP TABLE <TableName>; where, <Table name>: Specifies the name of the table to be deleted.	DROP TABLE Customer;

Table 3.1: DDL Statements

➔ **Data Manipulation Language**

Data Manipulation Language (DML) is used to fetch, add, modify, or delete records from a table. Table 3.2 lists the various DML statements.

STATEMENT TYPE	DESCRIPTION	SYNTAX	EXAMPLE
SELECT	Retrieves data from a table as a recordset.	SELECT [ALL|DISTINCT] * | [<Table Name>.]<FieldName1> [AS Alias1], [<TableName>.]<FieldName2> [AS Alias2] [,…] FROM <TableName>; where, <Table name>: Specifies the name of the table. *: Specifies that all the fields of the table must be retrieved. <TableName>.<FieldName1>, <Table Name>.<FieldName2>: Specifies names of the fields of the table. <Alias1>, <Alias2>: Specifies the name of the column to be assigned as field header instead of the original field name.	SELECT Cust_ID, Cust_Name, Cust_Address FROM Customer; SELECT * FROM Customer

Concepts

STATEMENT TYPE	DESCRIPTION	SYNTAX	EXAMPLE
INSERT	Adds a new record to a table.	INSERT INTO <TableName> [(<FieldName1>, <FieldName2>, …)] VALUES (VALUE1, VALUE2, …); where, <TableName>: Specifies the name of the table to add the records. <FieldName1>, <FieldName2>: Specifies the name of the fields in which data is to be inserted. VALUES: Keyword used to specify values. <VALUE1>, <VALUE2>: Specifies the values to be added to the fields.	INSERT INTO Customer (Cust_ID, Cust_Name, Cust_ Address) VALUES (3, "Brian", "2833, Palms Road, LA, USA");
UPDATE	Modifies the value of a field in a table. The SET keyword is used to specify the new value.	UPDATE <TableName> SET <FieldName1> = <Value1> [,FieldName2> = <Value2>, …] [WHERE condition]; where, <TableName>: Specifies the name of the table. <FieldName1>, <FieldName2>: Specifies the name of the fields. <Value1>, <Value2>: Specifies the new values of the fields. WHERE: Keyword used to specify a criteria.	UPDATE Customer SET Cust_Address = "3273, St. Paul's Street, LA, USA" WHERE Cust_ID=3;
DELETE	Removes the records from the table.	DELETE [*] FROM <TableName> [WHERE condition]; where, <TableName>: Specifies the name of the table. WHERE: Keyword used to specify a criteria.	DELETE * FROM Customer WHERE Cust_ID=3;

Table 3.2: DML Statements

Concepts

3.2.2 Purpose of a Query

Queries used to fetch data from a table or to perform calculations are termed as select queries. Queries used to add, modify, or delete data are termed as action queries. A query can also be used to provide data to a form or report.

In a well-designed and normalized database, the data to be presented through a form or report is usually stored in multiple tables. A query helps a user to combine the data required to generate the form or report.

For example, for two tables named Product and Orders, a user can combine information about Product Id, Product Name, Order Id, Order Quantity, Payable amount, and so on by using a query. Thus, data from different but related tables can be combined using a query without the need of duplication of data in any of the individual tables.

When a select query is created, Access 2010 displays the results known as recordset in a datasheet. A user can work with the recordset in the same way as a datasheet. That is, one can add or modify data, and Access 2010 will write the changes to the source tables used in the query.

The recordset can also be used as a source for other queries, reports, and forms. For example, to generate a report of the products that need replacement, one can create a query that returns the required data, and then, generate a report based on the resulting recordset.

3.2.3 Types of Query Wizard

Access 2010 provides two ways for creating select queries — the Query Wizard and the Query Designer.

Irrespective of the tool chosen, the following steps must be performed while creating a select query:

1. Select a record source for the query. A record source can be one or more tables, queries, or both.
2. Select the fields from the record source that need to be used in the query.
3. Decide on the sorting, filtering, or other criteria for the query if required.
4. Decide on the input parameter of the query if any.
5. After adding the fields and selection criteria if any, run the query to verify whether it gives the correct results.
6. Refine the query by adding or removing fields or the selection criteria if required.

   The query wizards can be viewed by clicking **Query Wizard** option in the **Queries** group of the create tab.

Concepts

The **New Query** dialog box will be displayed with the list of available query wizards as shown in figure 3.1.

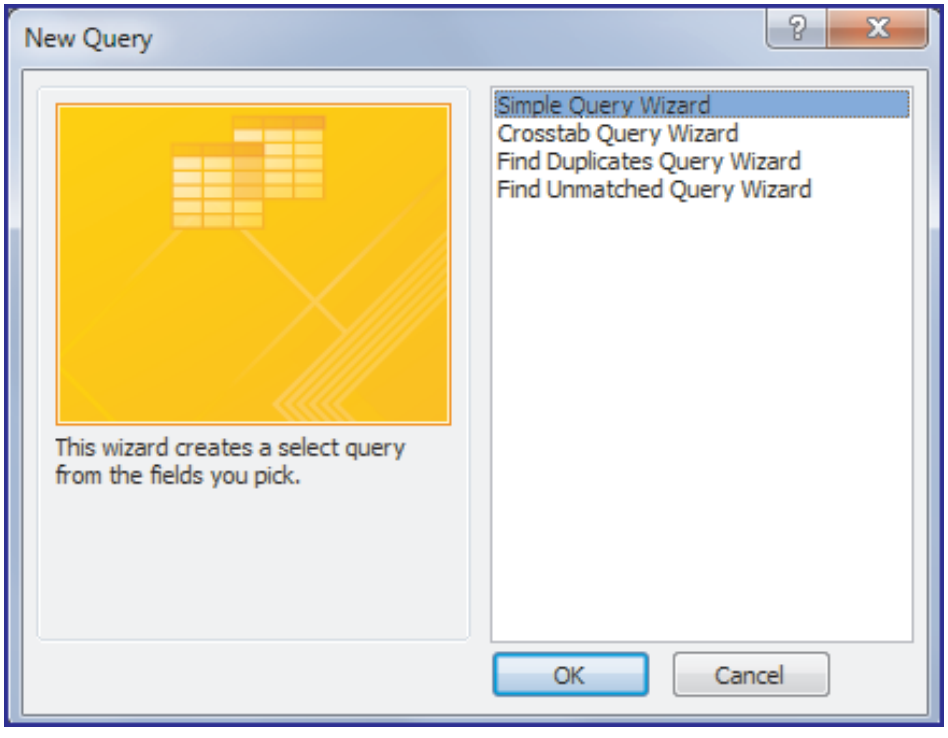


Figure 3.1: Types of Query Wizards

Note - To open the **New Query** dialog box, the database must have at least one table.

The different types of Query Wizards are as follows:

➔ **Simple Query Wizard**

One way of creating a query is to use the built-in **Simple Query Wizard**. This wizard is an attempt by Microsoft to simplify query creation. This wizard allows a user to step through the query design process by providing logical steps to create a query easily. However, in practice once a user learns creating a query, it becomes easy to bypass the Simple Query Wizard for other more efficient ways of creating queries.

➔ **Crosstab Query Wizard**

The **Crosstab Query Wizard** is basically the easiest and fastest for creating a crosstab query which is a type of select query. The wizard does most of the work. However, there are few options that it does not provide.

The wizard is easy to use, since the user has to simply answer a set of questions that are guided by the wizard itself. The wizard helps to group dates automatically from a date/time field into intervals such as quarters or months.

Concepts

The wizard can be used to create simple crosstab queries and then, refine the query using **Design View**.

However, the wizard does not allow to use more than one query or table as a record source, expressions for creating fields, prompts for parameters, and provides a fixed set of values to be used as column headings.

➔ **Find Duplicates Query Wizard**

The **Find Duplicates Query Wizard** allows a user to locate duplicate records in one or more tables. A duplicate record is a record that lists the same person or thing as another record.

However, all records containing similar information may not be duplicates. For example, records of three orders containing identical items but created on different dates cannot be considered as duplicate records. Similarly, all duplicate records may not contain completely matching information. For example, two records of the same customer may contain different addresses and phone numbers. However, a record with an outdated address can be considered as a duplicate record.

It is important to remove duplicate records because if multiple records for one customer can make it difficult to verify the records and also the order may be sent to an outdated address. Thus, duplicate records can compromise the integrity and accuracy of the data in the database.

The **Find Duplicates Query Wizard** makes it easy to search for potential duplicate records. However, Access 2010 does not delete the records nor indicates which one is outdated. That has to be done by the user. If the user is familiar with the data in the database, removal of the listed duplicate records can be accomplished easily.

➔ **Find Unmatched Query Wizard**

When two tables are associated with a one-to-many relationship, a user can use the find unmatched records query to search for records in the table representing the 'one' side that do not match with any records in the table representing the 'many' side. For example, one can search for records for two tables Customer and Orders; a user can identify the customers whose record does not exist in the Orders table by using the **Find Unmatched Query Wizard**.

3.2.4 Designing a Query

Access 2010 provides different types of queries to manipulate the data in a table such as Select, Make Table, Append, Update, Crosstab, and Delete as shown in figure 3.2.

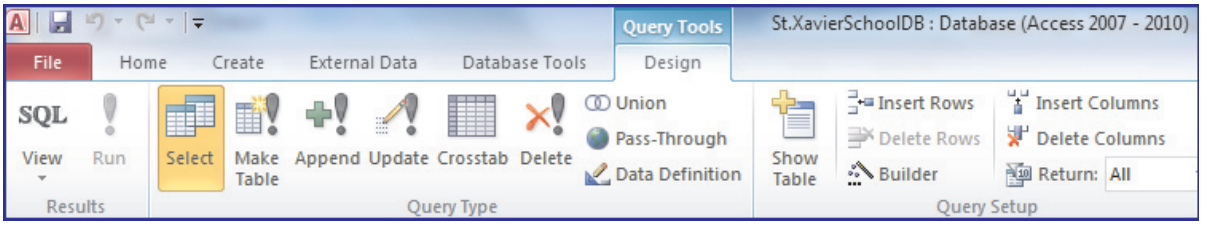


Figure 3.2: Query Types

Concepts

The query types are available in the **Query Type** group of the **Design** tab of **Query Tools**. A select query can be created either by using the Query Wizard, Query Designer, or by using the SQL View. The different views of a query can be selected from the **View** option of the **Results** group of the **Design** tab of **Query Tools**.

Consider a company named **ToyHeaven Pvt. Ltd.** located in **Chicago** that manufactures and sells toys. The management of the company wishes to automate the transactions of the company to make the functioning more quick and efficient. The solution is to create a database and add tables pertaining to the various entities in the system. Also, the user can create queries to retrieve and modify the data quickly and efficiently.

To create a simple select query using the Query Wizard, perform the following steps:

1. Create a new database named `ToyHeavenDB` and create a `Product` table as shown in figure 3.3.

Product

Field Name	Data Type	Description
Prod_ID	Text	Unique ID of the product
Prod_Name	Text	Name of the product
Prod_Category	Text	Category of product
Prod_Price	Number	Price of the product
Prod_Stock	Number	Quantity of product on hand

Figure 3.3: Product Table Design

2. Set the **Field Size** property of `Prod _ ID` to `5`, `Prod _ Name` to `100`, and `Prod _ Category` to `20`. Also, set the **Required** property of `Prod _ ID`, `Prod _ Price`, and `Prod _ Stock` to `Yes`.
3. Set `Prod _ ID` as the primary key.
4. Add data to the `Product` table as shown in figure 3.4.

Product

Prod_ID	Prod_Name	Prod_Catego	Prod_Price	Prod_Stock
P0001	Hotwheels	Car	400	20
P0002	Blocks	Construction	200	100
P0003	Barbie	Doll	500	200
P0004	Silly Putty	Creative	200	250
P0005	Lego Mindstorms	Educational	200	150
P0006	Speak and Spell	Educational	200	150
P0007	Capsela	Construction	250	100
P0008	Construx	Construction	300	150
P0009	Dinky Cars	Car	100	200
P0010	Fulla	Doll	250	100
P0011	G.I.Joe	Action Figure	100	200
P0012	He-Man	Action Figure	120	150

Figure 3.4: Product Table Data

5. Click **Create** → **Query Wizard** from the **Queries** group. The **New Query** dialog box is displayed.
6. Select **Simple Query Wizard** and click **OK**.
7. In the next screen, select the **Product** table from **Tables/Queries** drop-down list and use the **>** button to move the fields `Prod _ Id`, `Prod _ Name`, and `Prod _ Price` from the `Available Fields` to the `Selected Fields` as shown in figure 3.5.

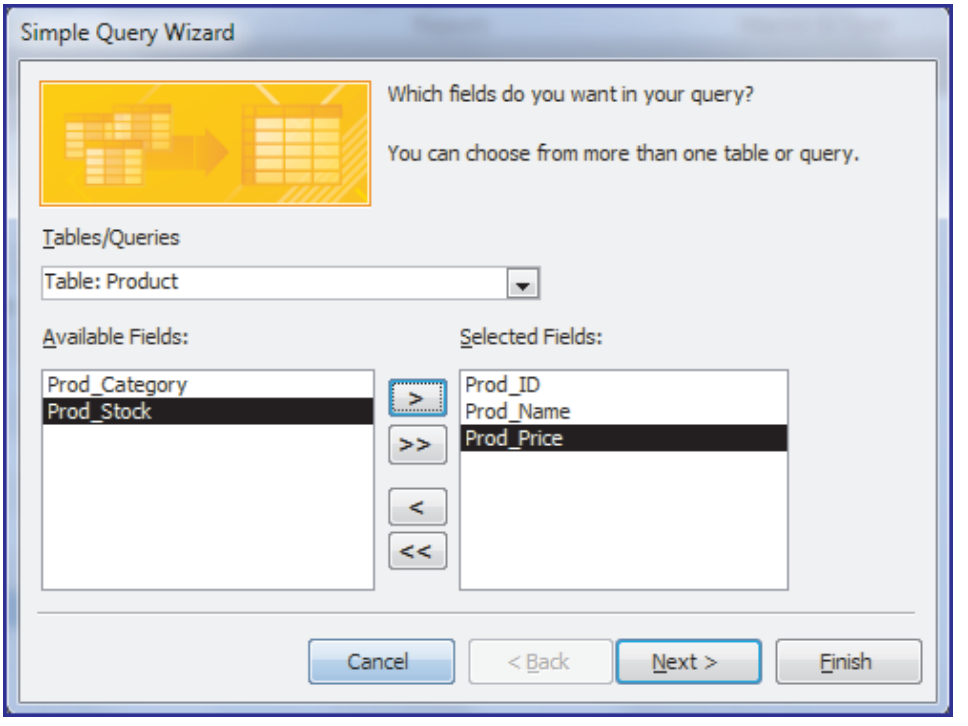


Figure 3.5: Selecting Fields for Display

8. Click **Next** and select the **Detail (shows every field of every record)** option button.
9. Click **Next**, and then, specify the title for the query as '`Display Products Query`'. Also, ensure that the **Open the query to view information** option button is selected.

Concepts

10. Click **Finish**. Access 2010 will execute the query and display the result as a recordset in the **Datasheet View** as shown in figure 3.6. Also, the name of the query is added to the **Queries** section in the navigation pane on the left side.

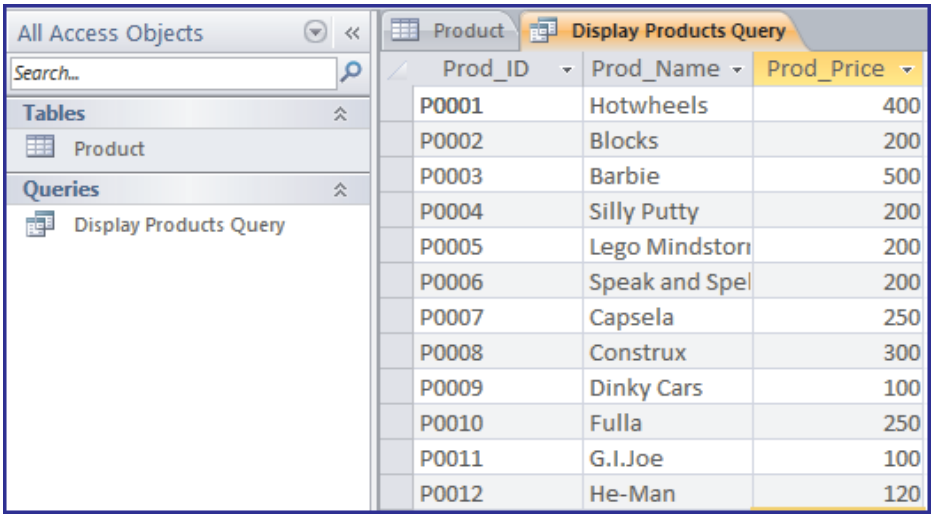


Figure 3.6: Result of Select Query on Product Table

Notice that the table shows only three columns namely, `Prod_ID`, `Prod_Name`, and `Prod_Price` that were selected in the Query Wizard during query creation.

11. Right-click the `Display Products Query` tab and select `Close` to close the query result.

To create a simple select query using the Query Designer, perform the following steps:

1. Click **Create** → **Query Design** from the **Queries** group. The **Show Table** dialog box is displayed.
2. Select the `Product` table and click **Add**. The `Product` table will be added to the query pane.
3. Click **Close** to close the **Show Table** dialog box. The **Query Design** window will appear as shown in figure 3.7.

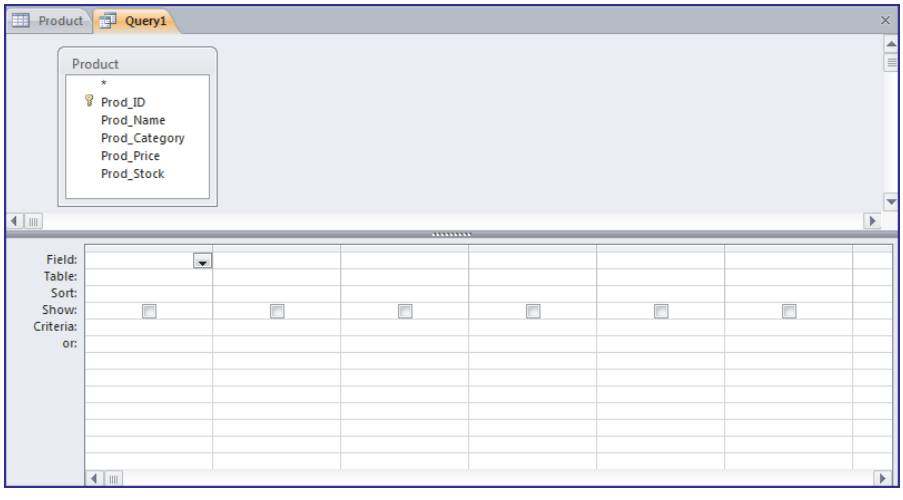


Figure 3.7: Query Designer

The top wide area of the Query window displays the objects such as tables, queries, or other objects used to create the query. The lower portion of the query window displays boxes used to perform different tasks related to the query. A splitter bar separates the upper and lower portions of the query window. The bar can be dragged to resize the sections according to requirement.

The lower portion consists of rows such as **Field**, **Table**, **Sort**, **Show**, **Criteria**, and **or**. As the name suggests, the **Field** row is used to specify the fields, **Table** to specify the table whose column is to be selected, **Sort** is used to specify the sorting if required, **Show** check boxes are selected for the columns that need to be displayed, and **Criteria** and **or** are used to specify a condition to filter the query output.

The tables or queries added to the upper section of the query window can be used to select fields for the query as follows:

- ➔ To select a single field, click the field in the list.
- ➔ To select multiple contiguous fields, select the first field in the range, press **Shift**, and then, click the last field in the range.
- ➔ To select random fields, click one of the fields, press **Ctrl** and then, click each of the other required fields.
- ➔ To select all the fields, click the line with * symbol at the top of the list.

After selecting the fields, the user can add them to the lower portion of the window by dragging and dropping them in the lower section.

Instead of dragging the fields, one can either double-click the required field, or the line with the * symbol to add all fields to the query.

Alternatively, in the lower section of the query window, click an empty Field box to display the drop-down list and then, select the desired field from the list.

4. Select the fields `Prod _ Id`, `Prod _ Name`, `Prod _ Category`, and other fields of the `Product` table one by one in the individual columns of the **Field** row as shown in figure 3.8.

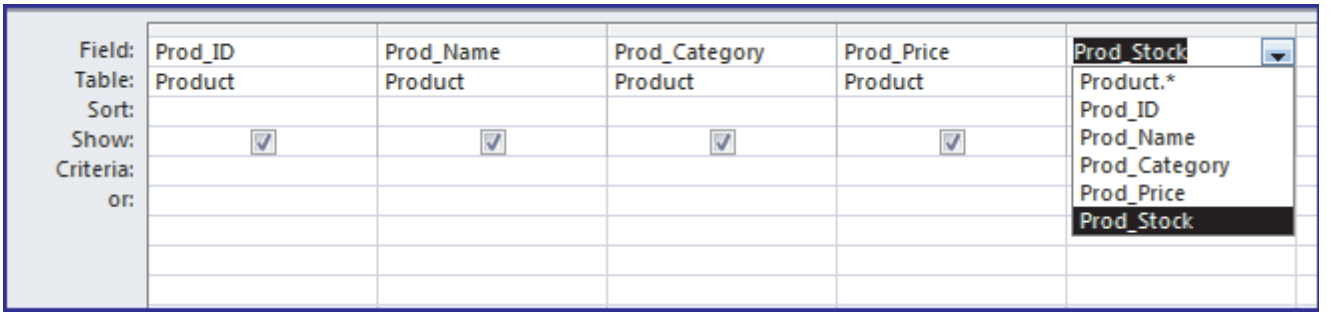


Figure 3.8: Query Window

Concepts

5. Clear the **Show** check boxes of the columns `Prod _ Category` and `Prod _ Stock`.

6. Right-click the **Query1** tab and select **Save** from the menu.

7. Type '`Display Product Query1`' in the **Query Name** box in the **Save As** dialog box and click **OK**.

8. Click **Run** in the **Results** group of the **Design** tab of **Query Tools**. Access 2010 will execute the query and display the results as a recordset in the **Datasheet View** as shown in figure 3.9.

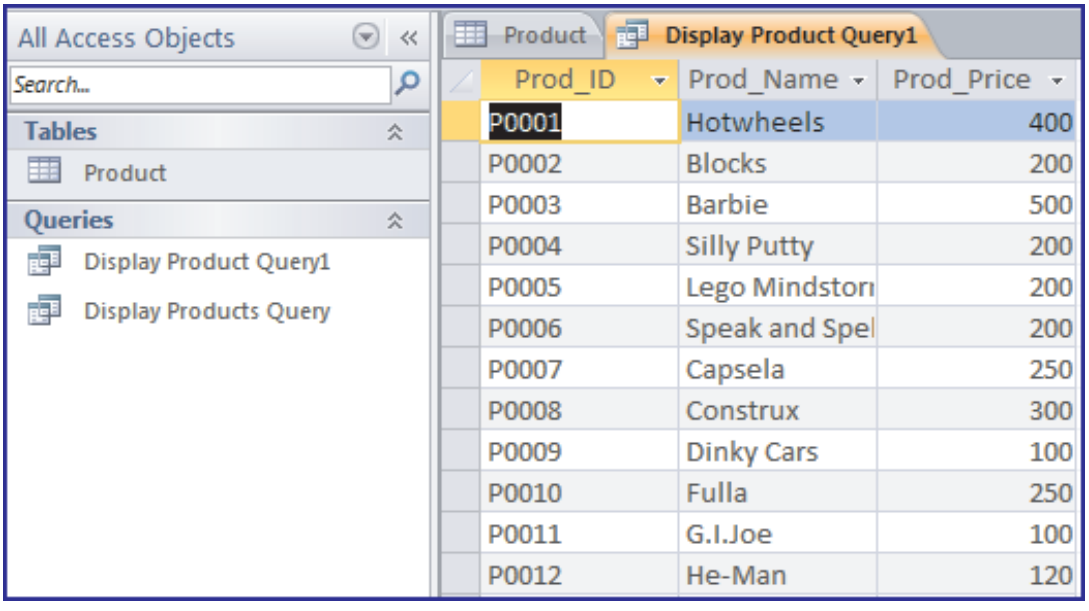


Figure 3.9: Select Query Output

9. Close the `Display Products Query1` window.

10. Use the **View** option of the **Results** pane to change the view of the query as shown in figure 3.10.

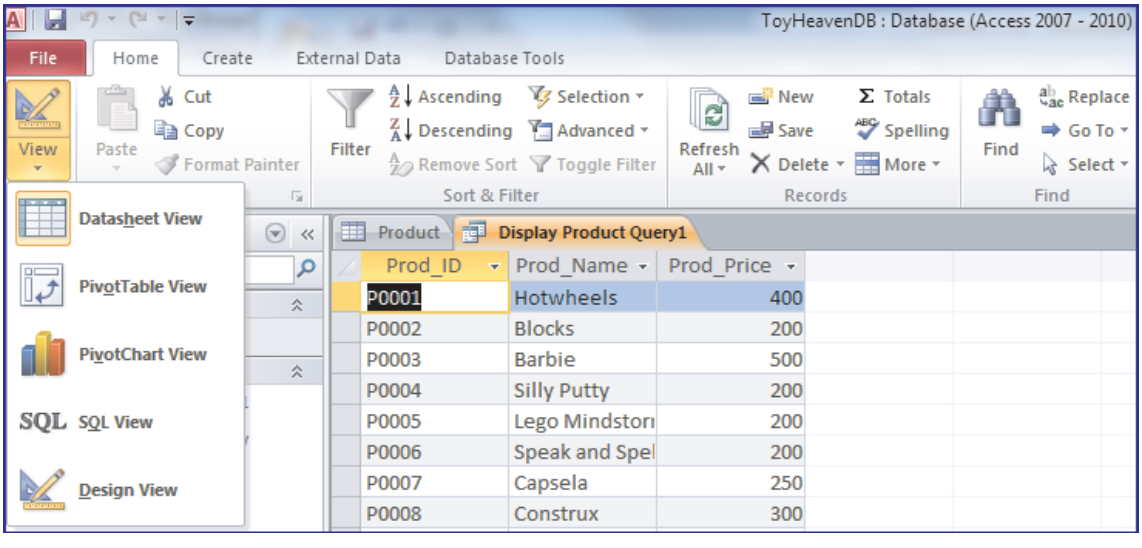


Figure 3.10: Query Views

Concepts

To write the query manually, a user can select the **SQL View** and type the `SELECT` query statement. To execute it, a user can change the view to **Datasheet View**.

3.2.5 Creating Update and Delete Queries

Access 2010 allows creating Update and Delete queries to manipulate a table.

To create a simple update query to change the name of product `P0012` by using the Query Designer, perform the following steps:

1. Click **Create** → **Query Design** from the **Queries** group. The **Show Table** dialog box is displayed.
2. Select the **Product** table and click **Add**. The **Product** table will be added to the query pane.
3. Click **Close** to close the **Show Table** dialog box.
4. Select **Update** option in the **Query Type** group, specify the update criteria for the `Prod _ ID` column, and save the query as `Update Product Query` as shown in figure 3.11.

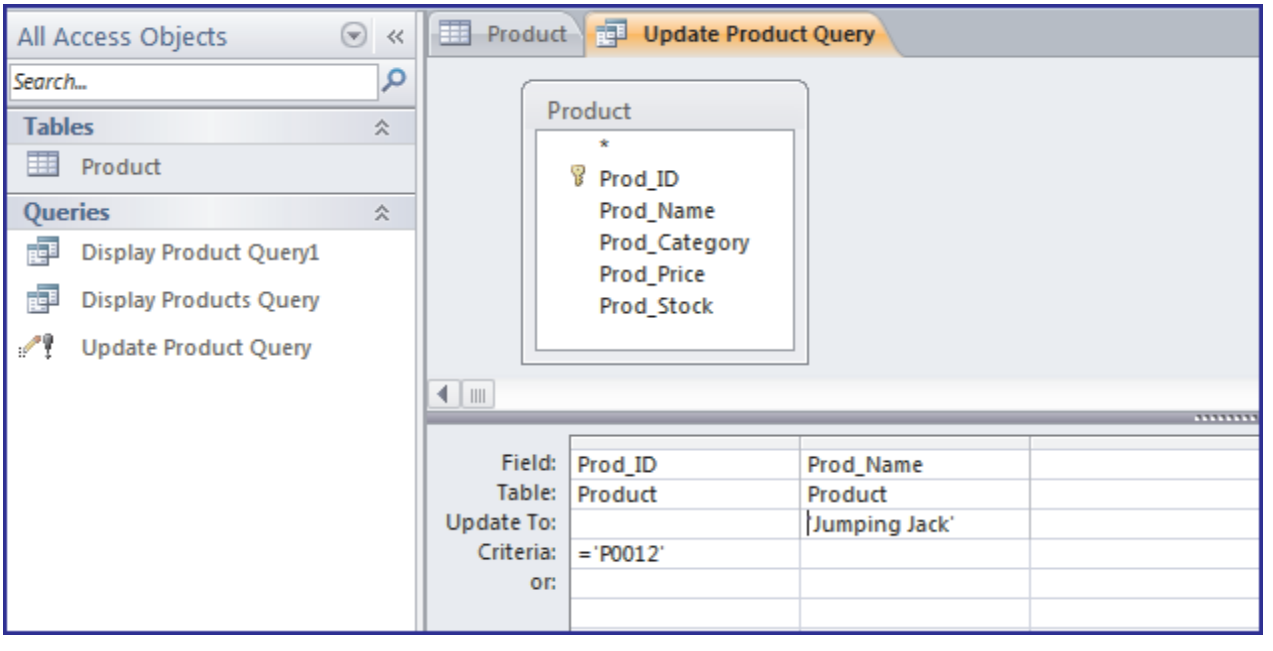


Figure 3.11: Update Query

5. Click **Run** to execute the query. A warning message appears for confirming the action.
6. Click **Yes**.

Concepts

7. Double-click the Product table in the navigation pane to open it in the **Datasheet View**. The table will show record `P0012` with the updated value '`Jumping Jack`' for `Prod _ Name` field as shown in figure 3.12.

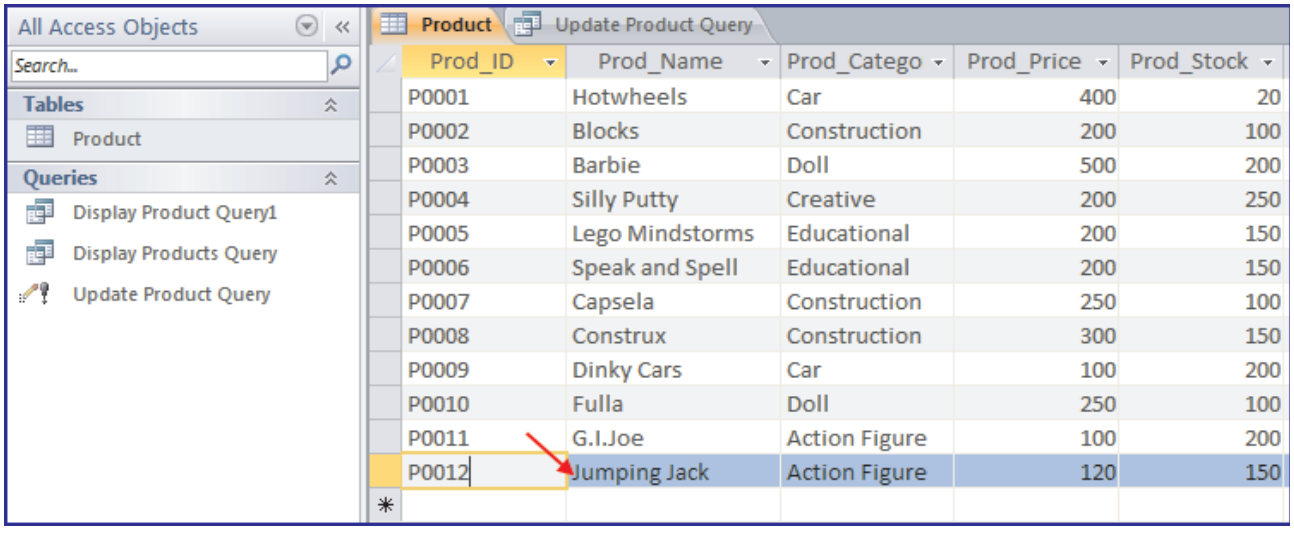


Figure 3.12: Result of Update Query

8. Close the `Update Product Query` window.

To create a simple delete query to delete details of a product by using the Query Designer, perform the following steps:

1. Add a new record '`P0013`' to the table in the Datasheet View as shown in figure 3.13.

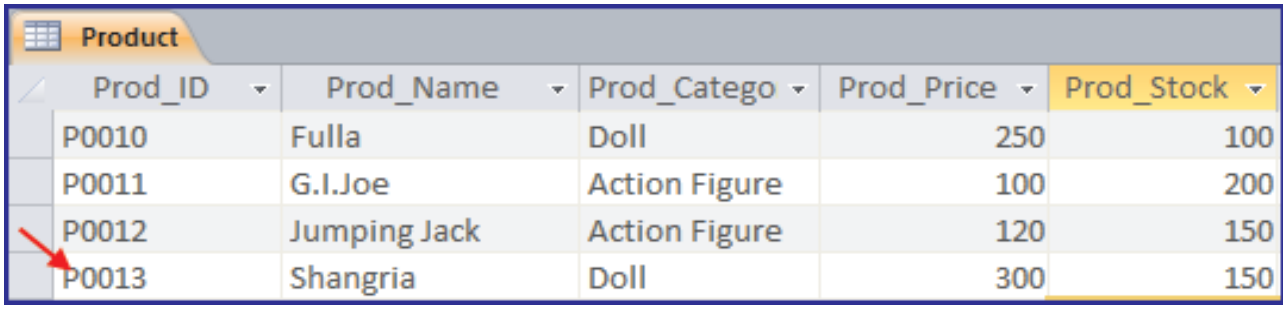


Figure 3.13: Record P0013 Added to the Product Table

2. Click **Create** → **Query Design** from the **Queries** group. The **Show Table** dialog box is displayed.
3. Select the `Product` table and click **Add**. The `Product` table will be added to the query pane.
4. Click **Close** to close the **Show Table** dialog box.

5. Select **Delete** option in the **Query Type** group, specify the delete criteria for the `Prod _ ID` column and save the query as `Delete Product Query` as shown in figure 3.14.

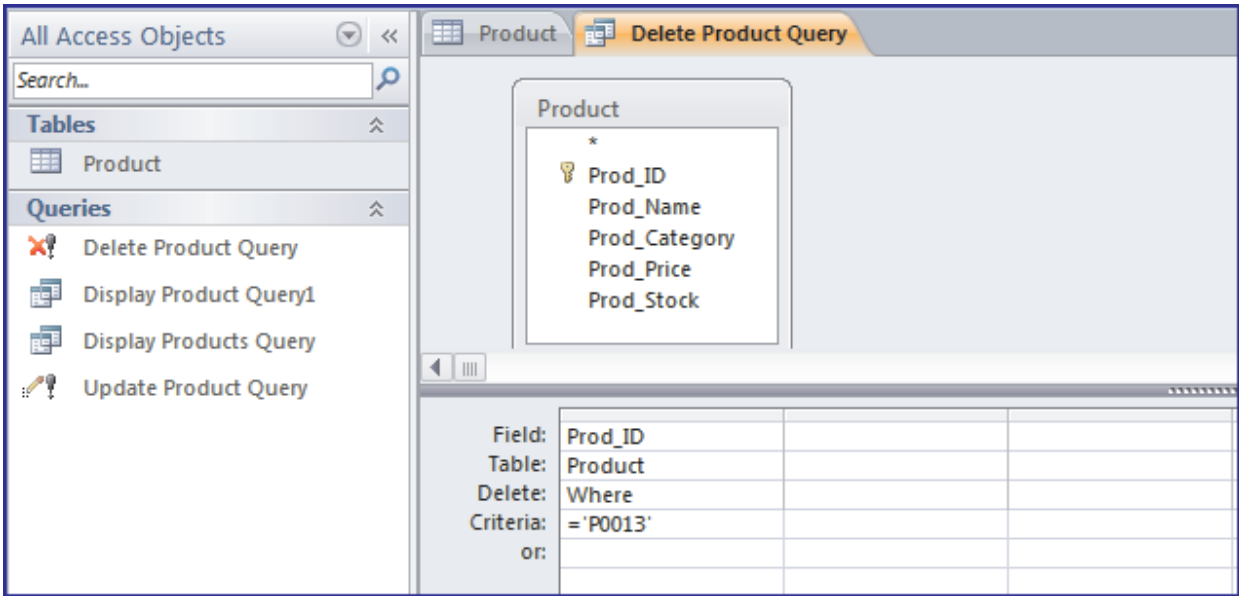


Figure 3.14: Delete Query

6. Click **Run** to execute the query. A warning message appears for confirming the action.
7. Click **Yes**.
8. Double-click the `Product` table in the navigation pane to open it in the **Datasheet View**. The table will show the deleted record with `#Deleted` text as shown in figure 3.15.

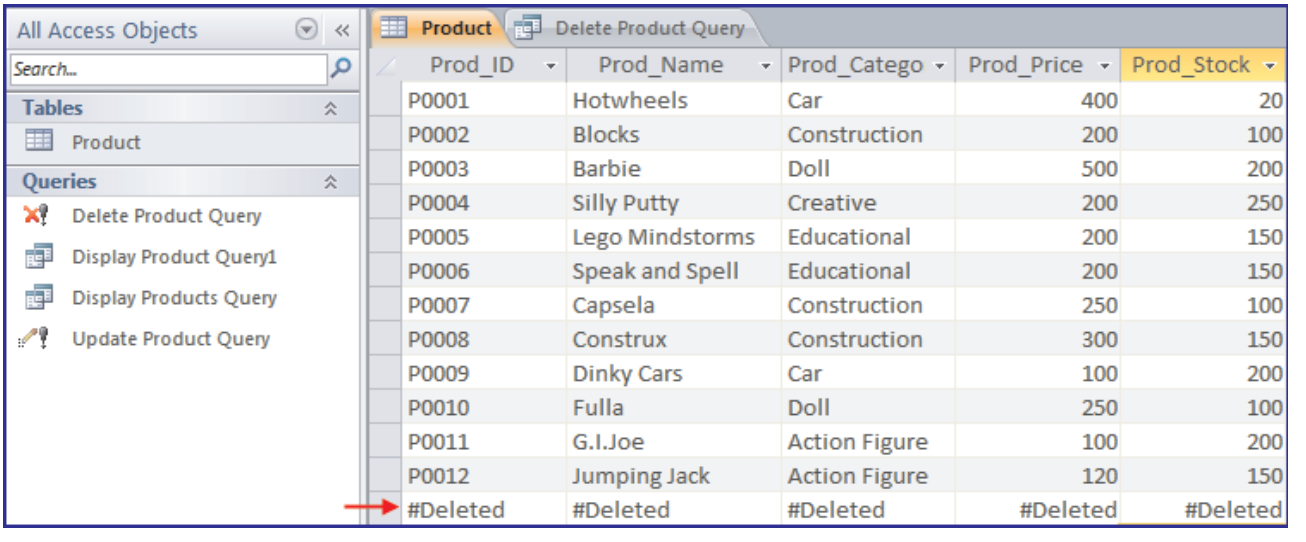


Figure 3.15: Result of Delete Query

Concepts

9. Save and close the `Delete Product Query` window.

3.2.6 Modifying a Query

A user can modify a query by opening it in the **Design View** or the **SQL View**. To modify the `Display Products Query1` to include the `Prod_Category` field, perform the following steps:

1. Right-click `Display Products Query1` in the navigation pane and select **Design View**.
2. Select the `Prod _ Category` field in the **Field** row and select the **Show** check box of the `Prod _ Category` column.
3. Click **Run** in the **Results** group of the **Design** tab to execute the query. The `Prod _ Category` column will be included in the output as shown in figure 3.16.

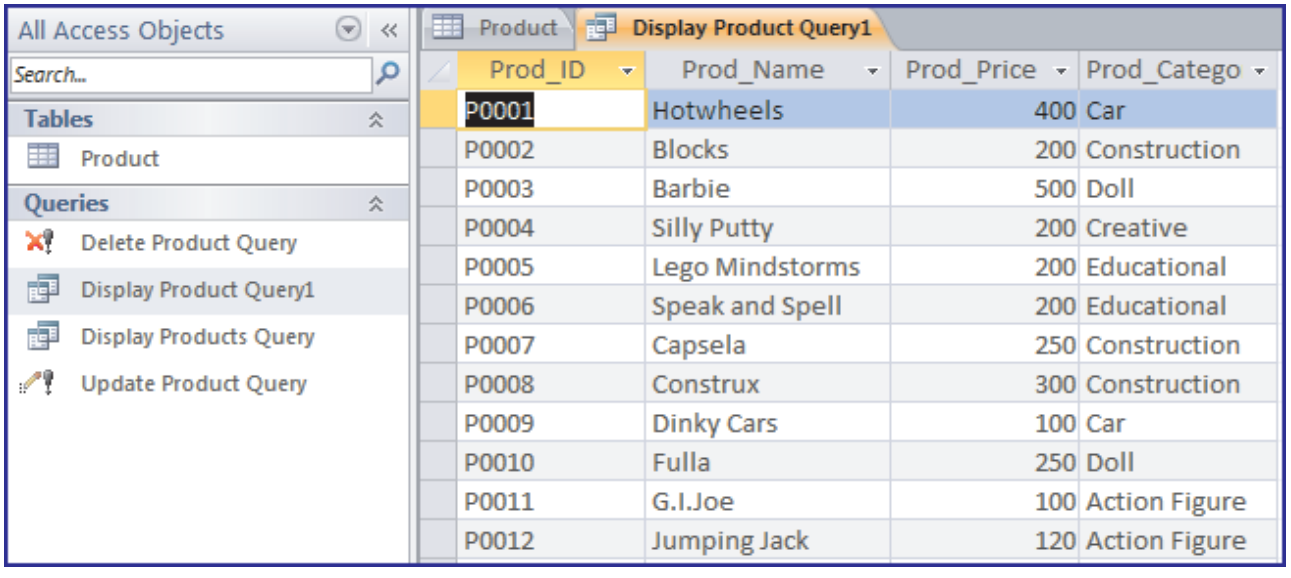


Figure 3.16: Modified Query Output

4. Save and close the `Display Product Query1` tab.

3.2.7 Use Query Results

The result of one query can be used as a source for another query, form, report, and so on. To use the query '`Display Products Query`' in another query, perform the following steps:

1. Click **Create** → **Query Design** from the **Queries** group. The **Show Table** dialog box is displayed.
2. Select the **Queries** tab. The list of existing queries is displayed.
3. Select the '`Display Products Query`' and click **Add**.
4. Click **Close** to close the **Show Table** dialog box. The query is added as a source in the top section of the query window.

Concepts

5. Select the fields `Prod _ ID` and `Prod _ Price` together using the **Ctrl** key and drag and drop them in the lower portion of the query window.

6. Save the query as '`Product Price Query`' as shown in figure 3.17.

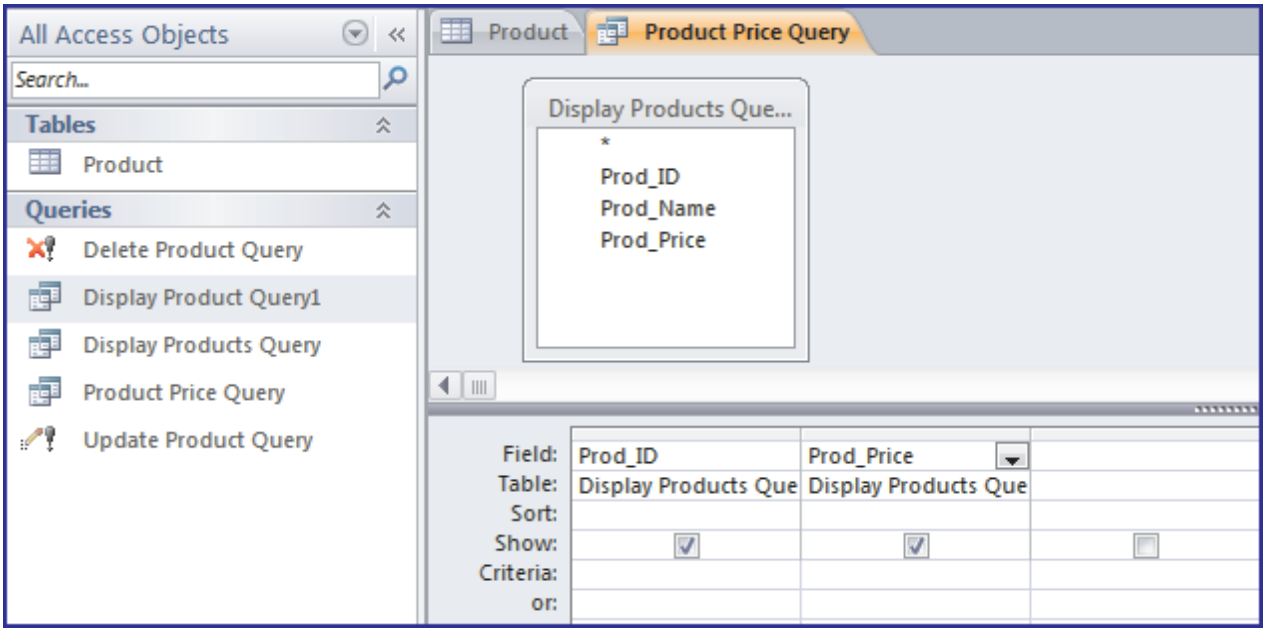


Figure 3.17: Using Query as a Data Source

Notice that the **Table** row shows `Display Products Query` as a data source.

7. Click **Run** in the **Design** tab. The result of the query will be displayed as shown in figure 3.18.

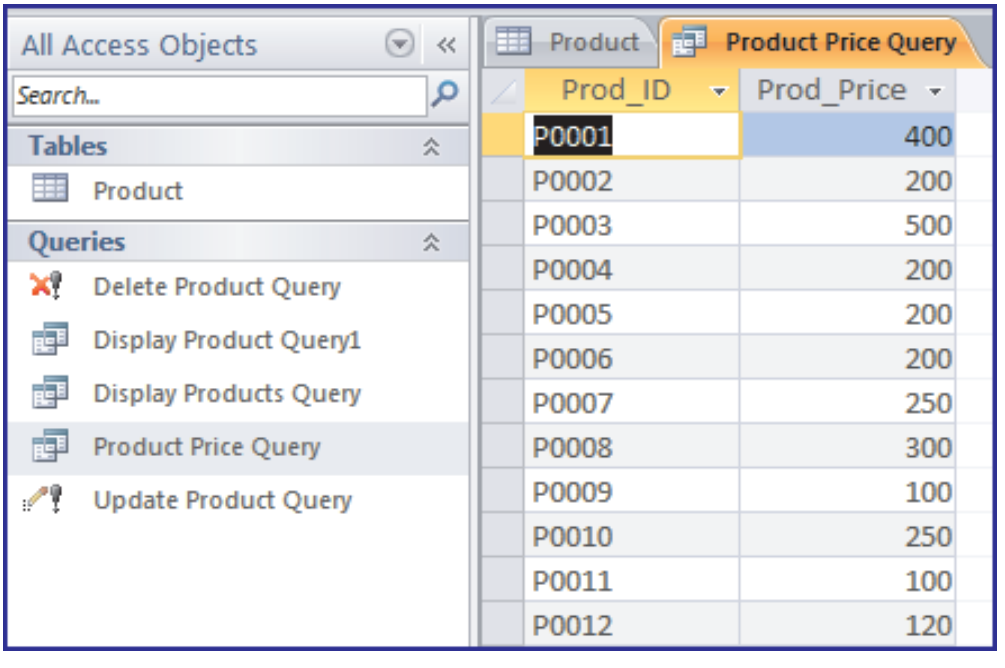


Figure 3.18: Output of Product Price Query

Concepts

Thus, a query can be further optimized to retrieve only relevant data without having to query the table again.

8. Save and close the `Product Price Query` tab.

3.3 Advanced Queries

Access 2010 allows a user to query a table by passing parameters, specifying criteria and functions, use multiple record sources, and summarize query results according to requirement.

3.3.1 Providing Input to a Query

Access 2010 allows creating a query that asks for input every time it is executed. This can be done by creating a parameterized query. A user can also create a form to collect parameters for a query.

To create a parameterized query for the Product table that returns records based on the value specified by the user, perform the following steps:

1. Create a select query in the **Design View** for the `Product` table and select all the fields in the **Field** row.
2. In the **Criteria** row of the field for which the parameter is to be supplied, type the text that is to be displayed in the parameter dialog box, enclosed in square brackets. In the current example, type `[Product Category]` in the **Criteria** of `Prod_Category` field and save the query as '`Parameterized Query`' as shown in figure 3.19.

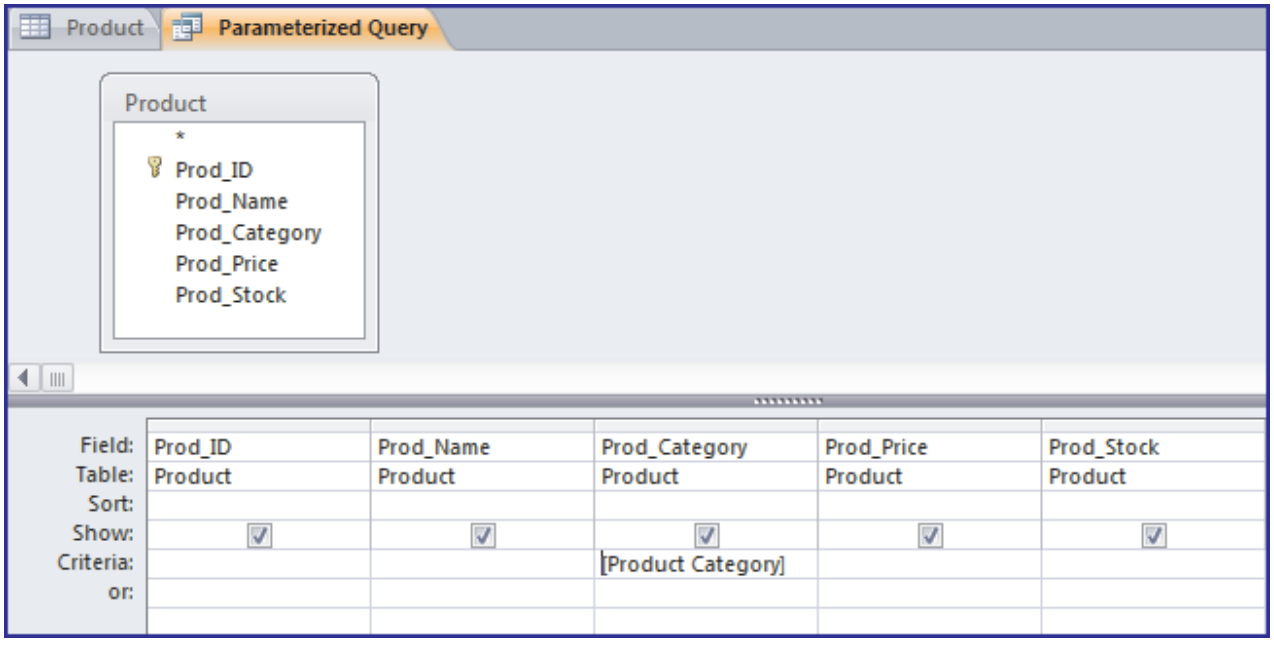


Figure 3.19: Parameterized Query

Parameters can be added to the Select, Crosstab, Append, Make-table, and Update queries.

3. Click **Run**. The **Enter Parameter Value** dialog box is displayed.
4. Type `Car` in the `Product Category` box as shown in figure 3.20.

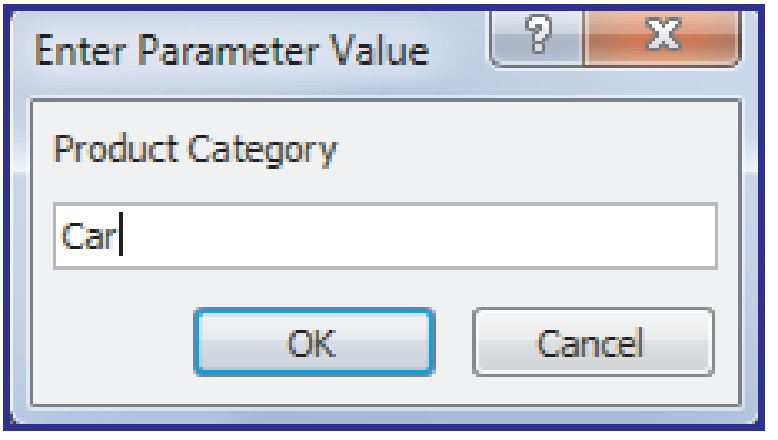
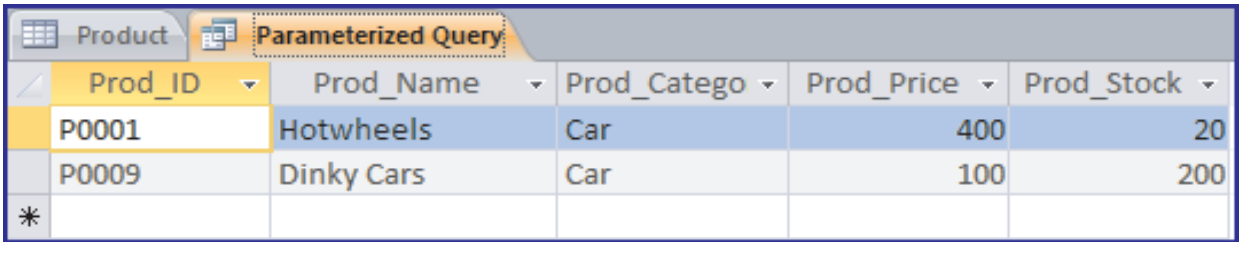


Figure 3.20: Enter Parameter Value Dialog Box

5. Click **OK**. Access 2010 will execute the query and display the output as a recordset as shown in figure 3.21.



Prod_ID	Prod_Name	Prod_Catego	Prod_Price	Prod_Stock
P0001	Hotwheels	Car	400	20
P0009	Dinky Cars	Car	100	200
*				

Figure 3.21: Result of Parameterized Query

Notice that the output includes only those records that belong to the category `Car`.

6. Save and close the `Parameterized Query` tab.

3.3.2 Adding Criteria to a Query

At times, a user might want to limit the result of a query based on some condition. For this purpose, query criteria can be used. A query criterion is a value or an expression that is used to compare with the field values to decide whether the record should be included or not. For example, the expression ="`Car`" compares each value of a field with the word 'Car' and includes only those records in the result in that match the criteria.

Concepts

Table 3.3 lists some examples of query criteria.

CRITERION	DESCRIPTION
`>10 and <20`	This criterion can be used with a Number field such as Age, Price, and so on. It indicates that the value must be greater than 10 and less than 20.
`>=#1/5/2000#`	This criterion can be used with a Date/Time field, such as `DateOfBirth`, `OrderDate`, and so on. It indicates that the date must be greater than or equal to `May 01, 2000`.
`Is Null`	This criterion can be used with any type of field to display all records with value `Null`.
`Like Y*`	This criterion returns records for strings in the field that begin with the letter '`Y`'.

Table 3.3: Examples of Query Criteria

Field references such as `[DateOfBirth]` and functions such as '**Sum**' can also be used to specify a criterion.

To add a criteria to a query, perform the following steps:

1. Click **Create** → **Query Design**. The **Show Table** dialog box is displayed.
2. Select the `Product` table and click **Add**.
3. Click **Close** to close the **Show Table** dialog box.
4. Add all the fields one by one to the lower section of the query window.
5. Specify the criteria for the `Prod _ Category` field as '`=Educational`' and save the query with the name '`Product Criteria Query`' as shown in figure 3.22.

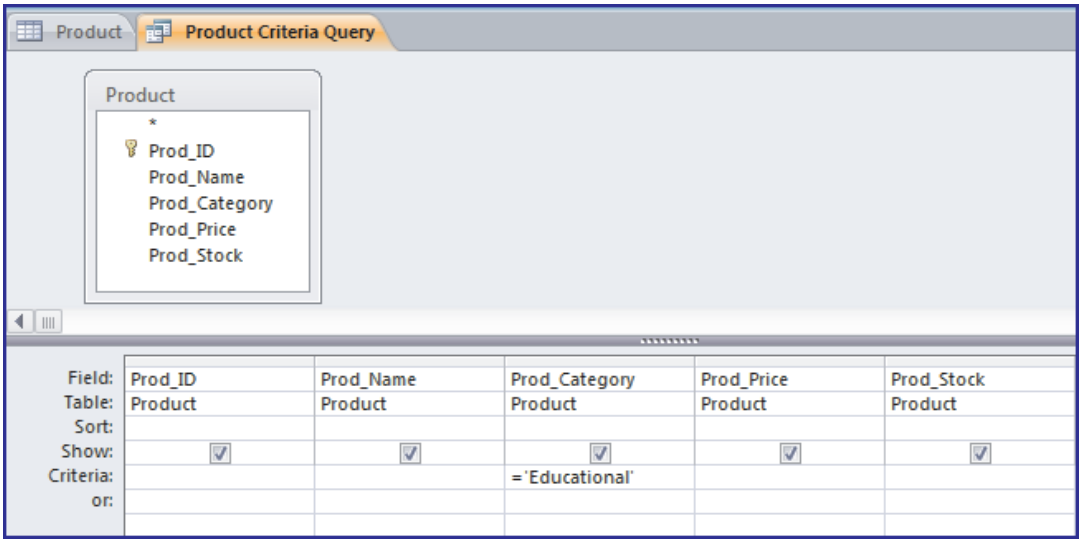


Figure 3.22: Specifying Query Criteria

Concepts

6. Click **Run** in the **Design** tab to execute the query. The filtered data showing only records with category 'Educational' will be displayed as shown in figure 3.23.

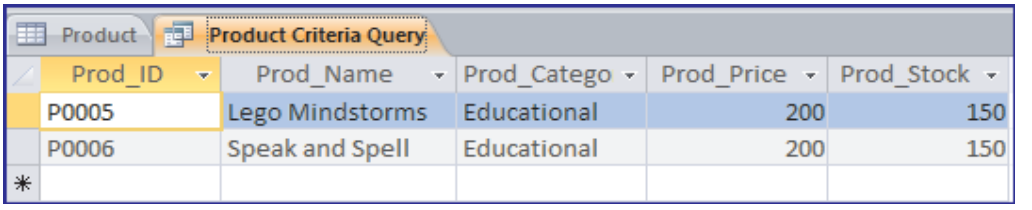


Figure 3.23: Result of Query Criteria

7. Save and close the Product Criteria Query.

At times, a single criterion may not suffice and the user may require specifying one or more alternate criteria. In such a case, one can use both the **Criteria** and the **or** rows in the design grid.

To provide alternate criteria, perform the following steps:

1. Open the Product Criteria Query in the **Design View**.
2. In the **or** row of the Prod _ Category field type '=Creative' as shown in figure 3.24.

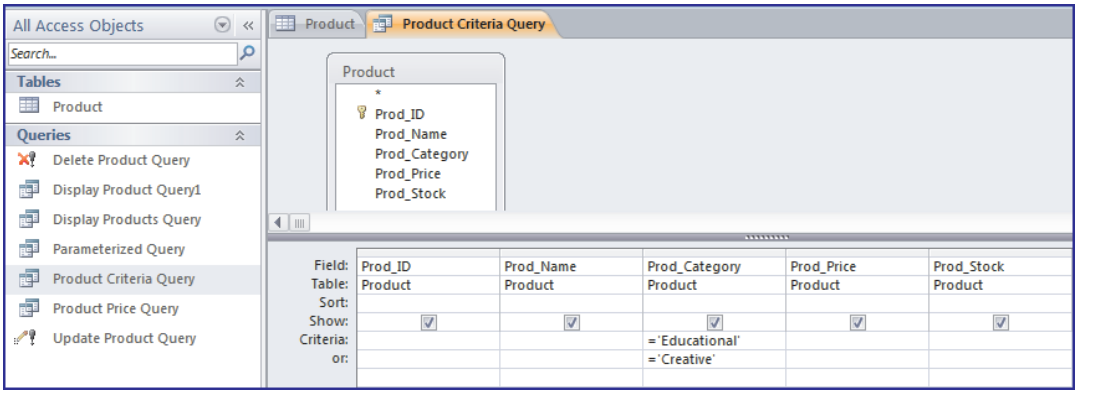


Figure 3.24: Specifying Alternate Query Criteria

The query is modified to include the Educational as well as Creative category products.

3. Click **Run** to execute the query. The result including the Creative category records is displayed as shown in figure 3.25.

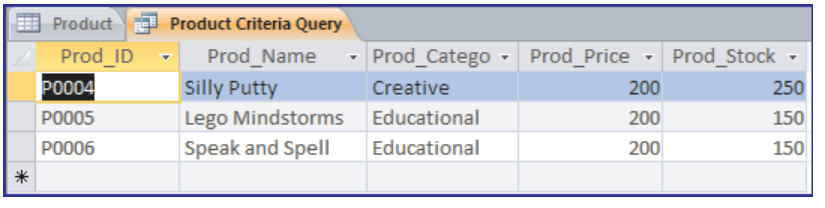


Figure 3.25: Result of Alternate Query Criteria

4. Save and close the Product Criteria Query.

Concepts

Similarly, a user can create a criterion such as '>200 and <400' on the Prod_Price field to get the list of all products with price between 200 and 400.

3.3.3 Adding Functions to a Query

Another commonly used method of altering the selection criteria is the use of built-in functions. The use of functions simplifies the counting and summarizing tasks for the user. For example, to know the number of products available in a particular category, one can use the Count function in the query and the total will be displayed in the query results.

Table 3.4 lists the commonly used functions and their description.

FUNCTION	DESCRIPTION
Sum	Returns the addition of the values in the field.
Avg	Returns the average of the values in a field.
Min	Returns the smallest value in the field.
Max	Returns the largest value in the field.
Count	Returns the total number of values in a field.

Table 3.4: List of Commonly Used Functions

To include a function in a query to calculate the average price of products belonging to the category Educational, perform the following steps:

1. Create a new query on the Product table and save it as Function Query.
2. Click **View** and select **SQL View** from the list. This will open the query in **SQL View** as shown in figure 3.26.

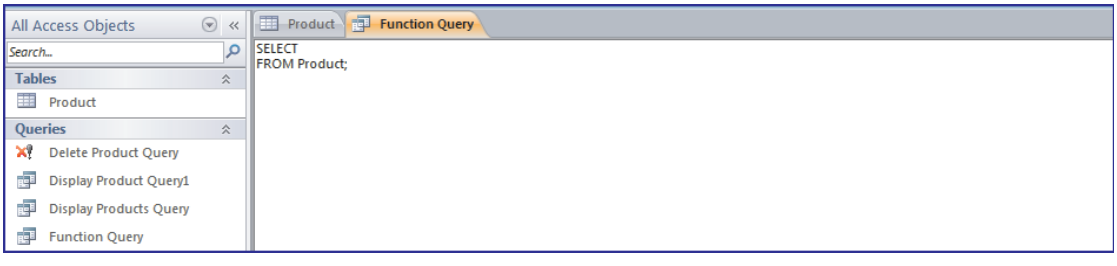


Figure 3.26: SQL View of a Query

3. Modify the query in the query window as follows:

```
SELECT AVG(Prod_Price) as Average_Price
FROM Product
WHERE Prod_Category ='Educational';
```

The query uses the AVG() function to generate an average of the Prod_Price column. The average is calculated for only those records where Prod_Category is 'Educational'.

Concepts

The result will be displayed in a column with title 'Average_Price' which is an alias name given to the result column.

4. Click **Run** to execute the query.

   The output of the query is displayed as shown in figure 3.27.

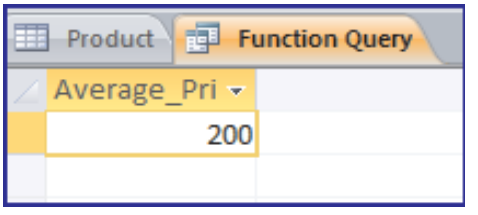


Figure 3.27: Result of AVG() Function in a Query

5. Open the query in the **Design View**. The query window will be displayed as shown in figure 3.28.

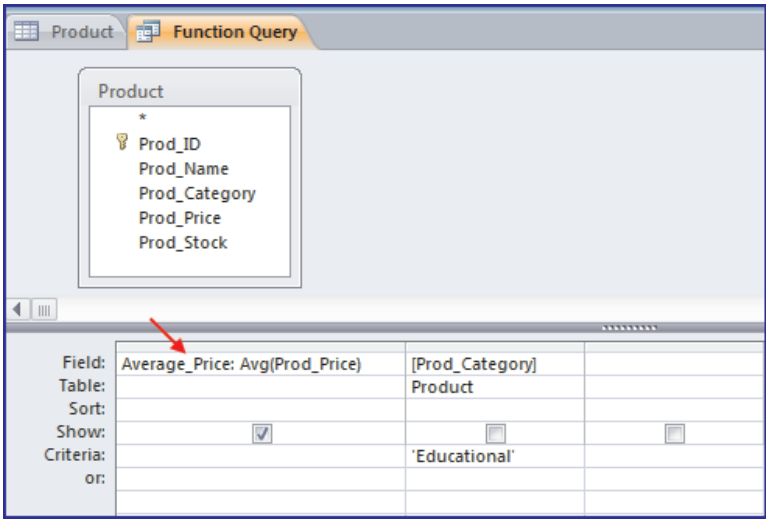


Figure 3.28: Design View of Function Query

6. Save and close the Function Query.

3.3.4 Summarizing Query Results

With a huge amount of data, a user may want to generate different types of statistics to get an idea about the functioning of a system. For example, a user may want to know the number of products available in a particular category or a product with the highest price. To assist a user in getting such statistics, Access 2010 provides the summary query that can be used to generate different types of statistics from the available data.

For creating a summary query, one can use the **Query Wizard** or **Design View**. While using the **Design View**, summary query can be created by using the Totals row that is added to the query window by clicking **Totals** in the **Show/Hide** group.

A summary query can be created with all or any of the fields used in a query.

Concepts

However, the purpose of a summary query is not to review data but to summarize data. Therefore, one should select a column having records that can be grouped into categories.

To support the summary query, SQL provides the **GROUP BY** clause. This clause is used to group records into categories while summarizing and displaying the result. For example, to get total number of students according to gender, one needs to select the column that stores the gender of a student.

Table 3.5 lists the options available in the **Totals** drop-down list.

FUNCTION	DESCRIPTION
Group By	Used to group common values in a field.
Sum	Returns the summation of the values in the field.
Avg	Returns the average of the values in a field.
Min	Returns the smallest value in the field.
Max	Returns the largest value in the field.
Count	Returns the count of the number of occurrences of a category in a field.
StDev	Returns the standard deviation of the values in a field.
Var	Returns the statistical variance of the numeric values of a group. If there is no value in a group, the function returns NULL.
First	Returns the first occurrence of the value in the category.
Last	Returns the last occurrence of the value in the category.
Expression	Used to create an expression to generate a summary.
Where	If a query has an empty (or null) record, instead of using the criteria row of the referential column, one can add the same column one more time, set its **Total** to **Where** and then, in its **Criteria** box, enter the criteria 'Is Not NULL' in that column. Also, one needs to clear its **Show** check box so as to not display the column in the query result.

Table 3.5: Totals Drop-down List Options

One can use the various summary functions as follows:

➔ COUNT

To create a summary query to find the total number of products available in each category by using the Count function, perform the following steps:

1. Create a new query on the Product table and save it as Product Count Query.
2. Add the Prod _ Category column in the lower portion of the query window.
3. Click **Totals** in the **Show/Hide** group of the **Design** tab. A new **Total** row will be added to the lower section of the query window.
4. Add the Prod _ Category column again in the lower portion.

Concepts

5. In the **Total** row of the second `Prod _ Category` column, select `Count` from the drop-down list as shown in figure 3.29.

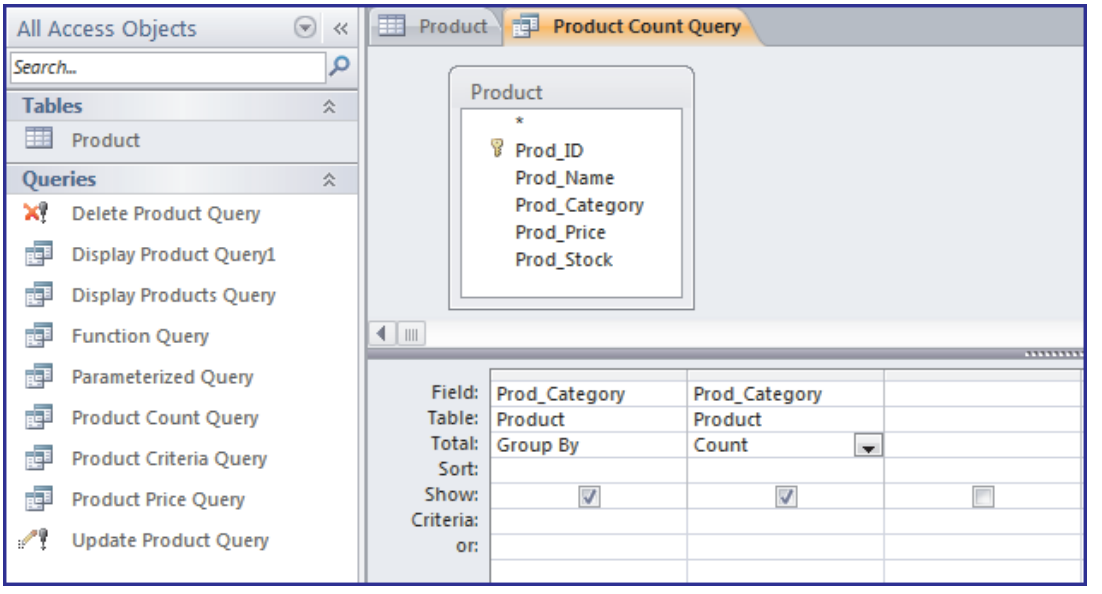


Figure 3.29: Design View of Product Count Query

Notice the Group By clause added to the first `Prod_Category` column. The **Group By** clause is used to group the common values of `Prod_Category` column while calculating the count and display it only as one record in the result.

6. Click **Run** to execute the query. The category wise count of products is displayed as shown in figure 3.30.

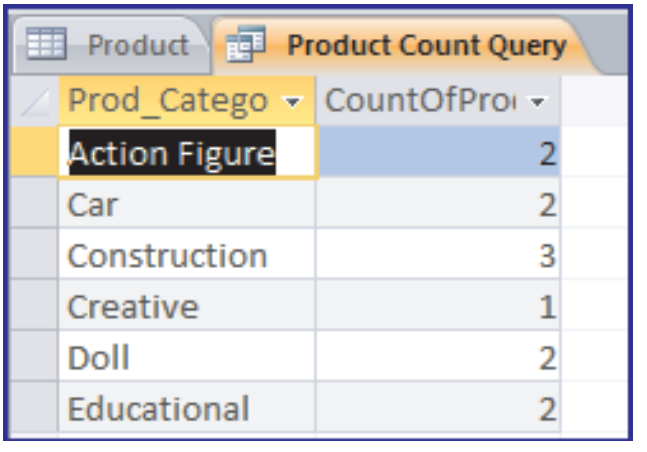


Figure 3.30: Result of the Product Count Query

7. Save and close the `Product Count Query`.

Concepts

➔ SUM

To create a summary query to find the sum of stock of all products available in each category by using the Sum function, perform the following steps:

1. Create a new query on the Product table and save it as Stock Summary Query.
2. Add the Prod _ Category and Product _ Stock columns in the lower portion of the query window.
3. Click **Totals** in the **Show/Hide** group of the **Design** tab. A new **Total** row will be added to the lower section of the query window.
4. In the **Total** row of the Prod _ Stock column, select Sum from the drop-down list as shown in figure 3.31.

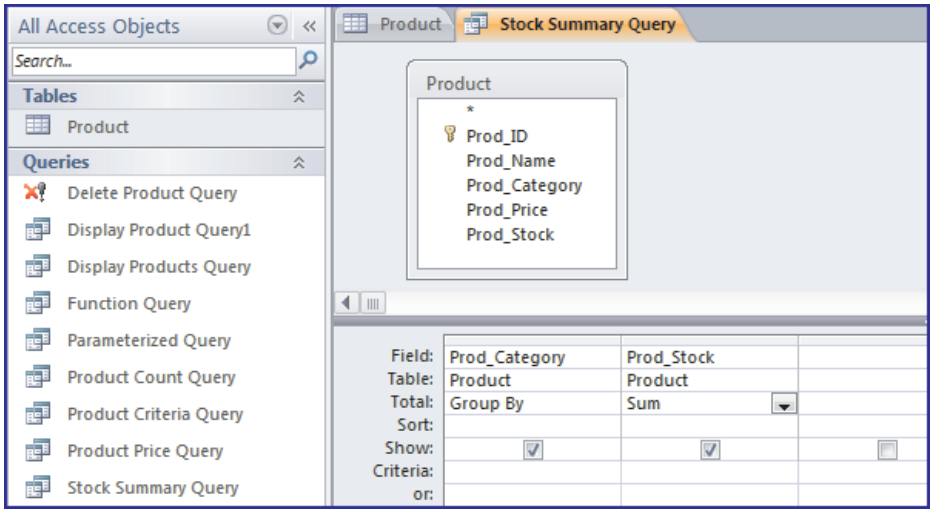


Figure 3.31: Stock Summary Query

The **Group By** clause is used to group the common values of Prod_Category column while calculating the sum and display it only as one record in the result.

5. Click **Run** to execute the query. The category wise sum of stock is displayed as shown in figure 3.32.

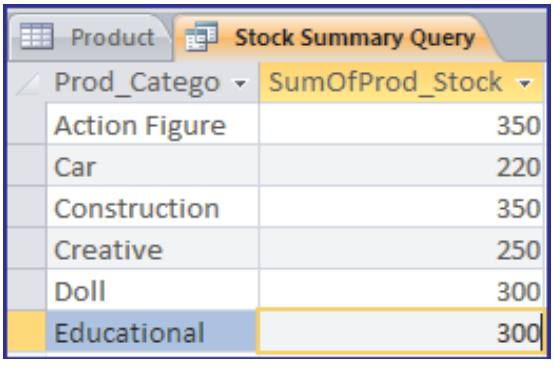


Figure 3.32: Result of Stock Summary Query

Concepts

6. Save and close the Stock Summary Query.

➔ MIN and MAX

A user can create a summary query to find the minimum and maximum price of the product in each category. This can be done by using the Min and Max functions as follows:

1. Create a new query on the Product table and save it as Product Rate Query.
2. Add the Prod _ Category column in the lower portion of the query window.
3. Add the Prod _ Price column twice in the query window.
4. Click **Totals** in the **Show/Hide** group of the **Design** tab. A new **Total** row will be added to the lower section of the query window.
5. In the **Total** row of the first Prod _ Price column, select **Min** and in the second Prod _ Price column, select **Max** from the drop-down list as shown in figure 3.33.

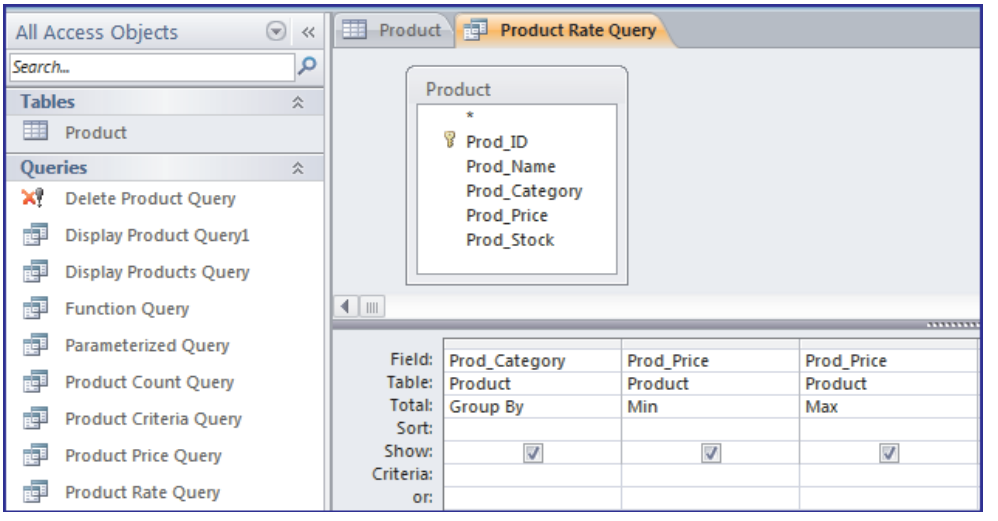
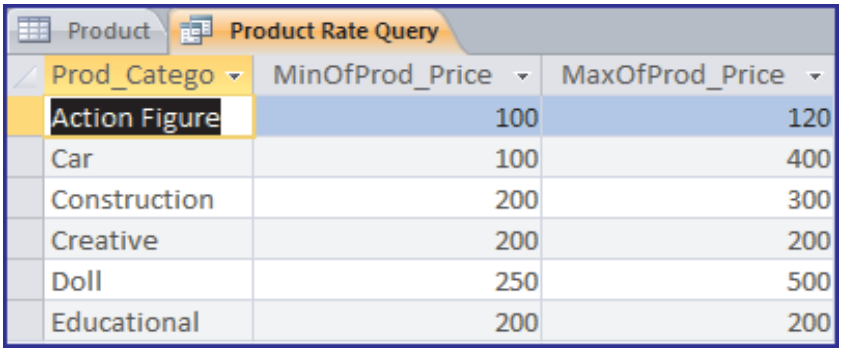


Figure 3.33: Product Rate Query

6. Click **Run** to execute the query. The category wise minimum and maximum rate of product is displayed as shown in figure 3.34.



Prod_Catego	MinOfProd_Price	MaxOfProd_Price
Action Figure	100	120
Car	100	400
Construction	200	300
Creative	200	200
Doll	250	500
Educational	200	200

Figure 3.34: Result of Product Rate Query

Concepts

Notice that the Creative category has same value for both **Min** and **Max** price. This is because there is only one record in the Creative category whereas both products in the Educational category have the same prices.

7. Save and close the Product Rate Query.

To further optimize a query, a user can use the HAVING clause as follows:

➔ HAVING clause

The HAVING clause is used to specify which of the grouped records should be displayed in a SELECT statement along with a GROUP BY clause. The GROUP BY clause groups the records based on a category, the HAVING clause filters the result based on the HAVING clause criteria. Then, it displays only those records that fulfill the conditions specified in the HAVING clause.

The syntax of HAVING clause is as follows:

Syntax:

```
SELECT <Field1>, <Field2>[, …]
FROM <TableName>
WHERE condition
GROUP BY <GroupField1>,<GroupField2> [,…]
[HAVING group condition]
```

where,

<Field1>, <Field2>: Specifies the fields to be included for display.

<TableName>: Specifies the name of the table.

condition: The condition to be applied to the selected fields for grouping.

<GroupField1>, <GroupField2>: Specifies names of fields based on which grouping is to be done.

group condition: The condition to determine which of the grouped records to display.

Note - The HAVING clause is optional. HAVING works similar to WHERE which determines the records to be selected. After the selected records are grouped using GROUP BY, the HAVING clause decides which records should be displayed.

A HAVING clause allows up to 40 expressions linked together using logical operators such as **And** and **Or**.

To create a summary query to find the category of product with total stock greater than 300 and the category name starting with 'C', perform the following steps:

1. Create a new query on the Product table as and save it as Having Clause Query.
2. Click **View** → **SQL View** to open the query in the **SQL View**.

Concepts

3. Type the following query in the **SQL View**:

```
SELECT Prod_Category, Sum(Prod_Stock) as Total_Stock
FROM Product
GROUP BY Prod_Category
HAVING Sum(Prod_Stock) > 300 AND Prod_Category LIKE "C*";
```

The `Sum(Prod_Stock)` function generates a sum of the stock value of products belonging to each category using the `GROUP BY` clause. Later the `HAVING` clause applies the two conditions: `Sum(Prod_Stock)>300` and `Prod_Category LIKE 'C*'` to display only those groups that have total stock greater than 300 as well as category name starting with letter 'C'.

4. Click **Run** to execute the query. The result is displayed in figure 3.35.

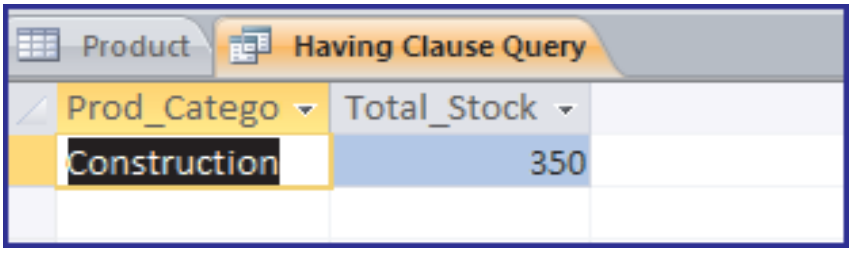


Figure 3.35: Result of HavingClause Query

The query returns only one record for `Construction` category since, it satisfies both conditions specified in the `HAVING` clause.

5. Save and close the `Having Clause Query`.

3.4 Unions and Subqueries

When data is distributed across multiple tables, it is required to combine the data to generate the required information. To do this, first the related tables must be linked through relationships. Later, queries can be created to retrieve data from the related tables to generate required information.

3.4.1 Querying Multiple Record Source

Data from more than one table can be combined using a query across multiple record sources. For example, to get the list of all orders related to each product, one can create a query that retrieves data from both the tables `Product` and `Product_Order`.

Concepts

To do this, perform the following steps:

1. Create a table named **Product _Order** as shown in figure 3.36.

Product | Product_Order

Field Name	Data Type	Description
Prod_ID	Text	Unique Id of product
Order_Qty	Number	Quantity of product ordered
Order_Date	Date/Time	Date of Order
Order_Status	Yes/No	Delivery status of order

Figure 3.36: Product_Order Table

2. Change the **Field Size** property of **Text** data type to `5`, **Format** property of **Date/Time** to Short `Date`, and **Format** of **Yes/No** data type to `Yes/No`.

3. Enter data into the **Product_Order** table as shown in figure 3.37.

Product | Product_Order

Prod_ID	Order_Qty	Order_Date	Order_Statu
P0001	10	1/12/2012	☑
P0002	50	5/1/2012	☑
P0003	80	5/31/2012	☑
P0004	100	5/8/2012	☑
P0005	60	6/29/2012	☑
P0006	90	5/12/2012	☑
P0007	80	7/23/2012	☑
P0008	200	8/30/2012	☐
P0009	120	7/16/2012	☑
P0010	120	6/13/2012	☑
P0011	50	12/8/2012	☑
P0012	130	3/13/2012	☑

Figure 3.37: Product Order Details

4. Save and close the `Product _ Order` as well as `Product` table.

Concepts

5. Create a one-to-many relationship between the `Product` and `Product _ Order` table with referential integrity as shown in figure 3.38.

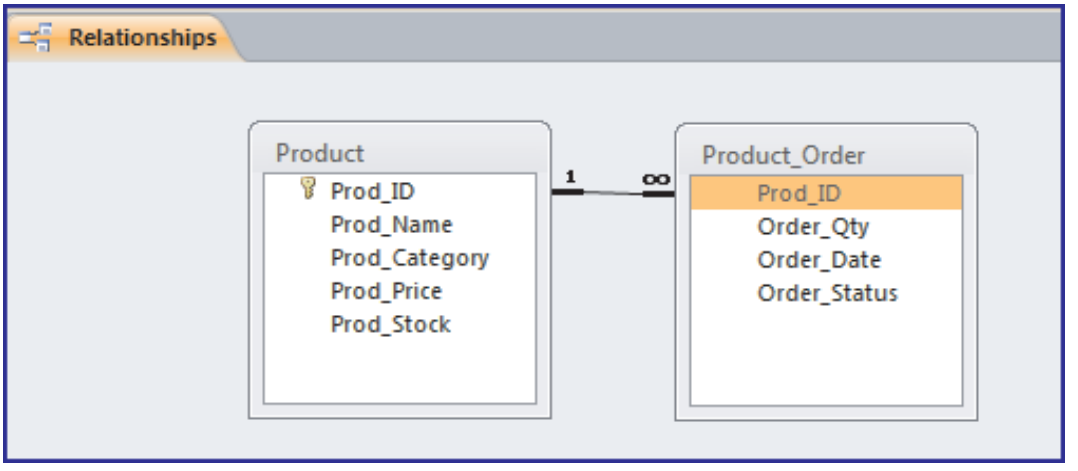


Figure 3.38: Relationship between Product and Product_Order Tables

6. Save and close the **Relationships** pane.
7. Click **Create** → **Query Wizard**. The **New Query** dialog box is displayed.
8. Select **Simple Query Wizard** and click **OK**.
9. In the next screen, select the `Prod _ ID`, `Prod _ Name`, and `Prod _ Stock` fields from the Product table and `Order _ Qty`, `Order _ Date`, and `Order _ Status` fields from the `Product _ Order` table as shown in figure 3.39.

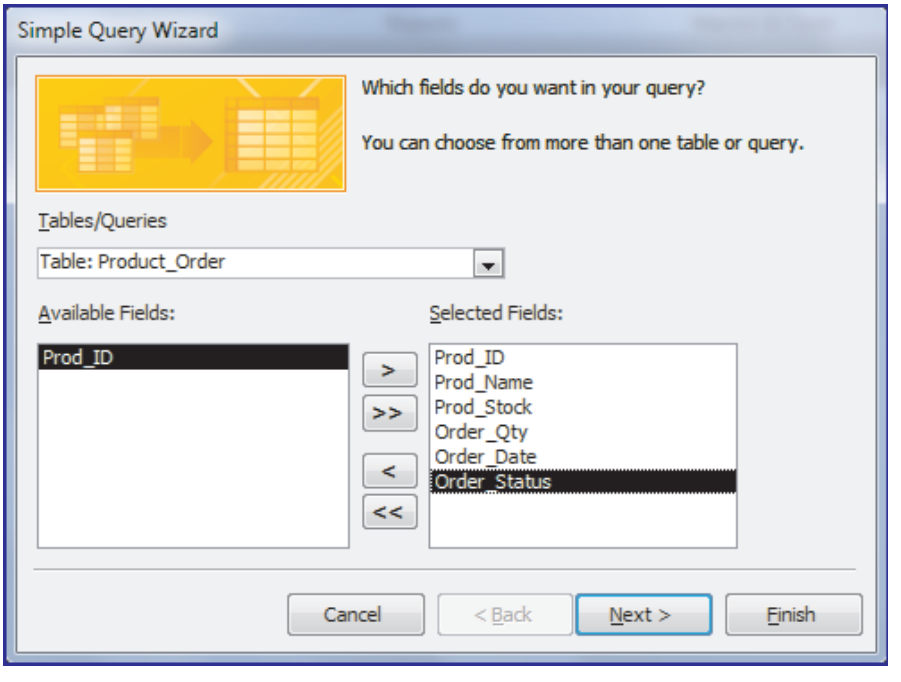


Figure 3.39: Adding Fields to the Query

Concepts

10. Click **Next**. Select the **Details (shows every field of every record)** option button.

11. Click **Next**.

12. Type the name as '`Multiple Table Query`' and click **Finish**. The query is executed and data combined from both the tables is displayed as shown in figure 3.40.

Multiple Table Query

Prod_ID	Prod_Name	Prod_Stock	Order_Qty	Order_Date	Order_Statu
P0001	Hotwheels	20	10	1/12/2012	☑
P0002	Blocks	100	50	5/1/2012	☑
P0003	Barbie	200	80	5/31/2012	☑
P0004	Silly Putty	250	100	5/8/2012	☑
P0005	Lego Mindstorms	150	60	6/29/2012	☑
P0006	Speak and Spell	150	90	5/12/2012	☑
P0007	Capsela	100	80	7/23/2012	☑
P0008	Construx	150	200	8/30/2012	☐
P0009	Dinky Cars	200	120	7/16/2012	☑
P0010	Fulla	100	120	6/13/2012	☑
P0011	G.I.Joe	200	50	12/8/2012	☑
P0012	Jumping Jack	150	130	3/13/2012	☑

Figure 3.40: Result of Multiple Table Query

13. Open the query in **Design View** as shown in figure 3.41.

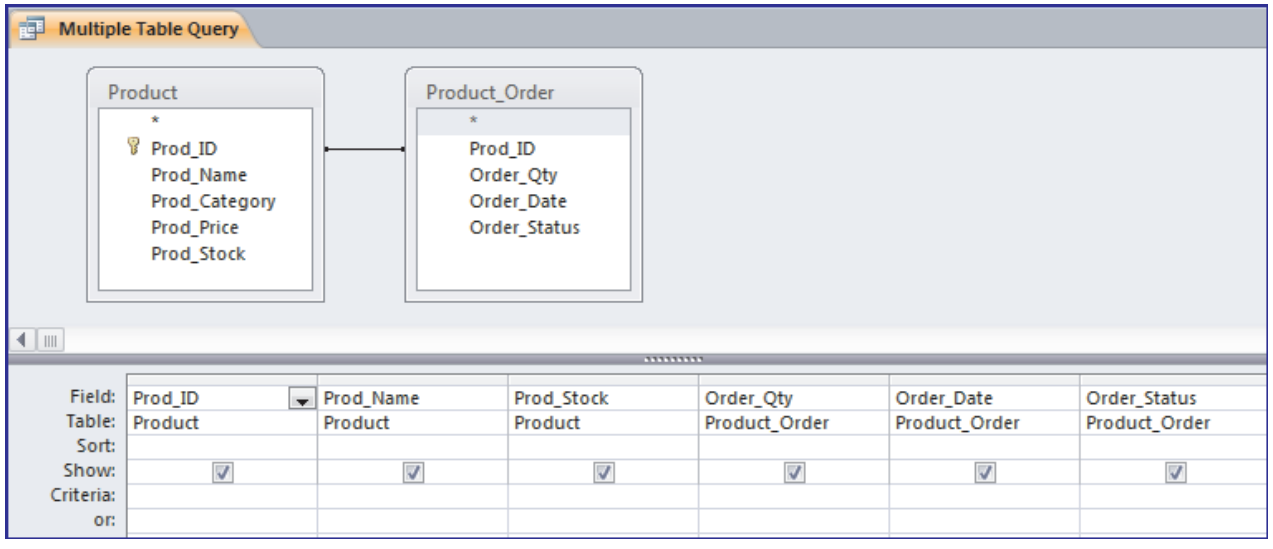


Figure 3.41: Design View of Multiple Table Query

Notice that Access 2010 automatically created the query in the **Design View**.

Concepts

14. Open the query in the SQL View as shown in figure 3.42.

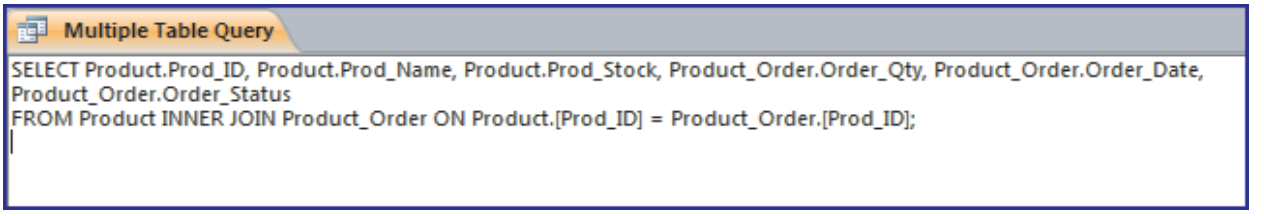


Figure 3.42: SQL View of Multiple Table Query

Notice the `INNER JOIN` clause in the query. This is used to combine the records of `Product` and `Product_Order` tables that have same the values in the `Prod_ID` column in both tables.

Joins allow a user to combine data from more than one table. There are different types of joins such as `INNER JOIN`, `LEFT OUTER JOIN`, `RIGHT OUTER JOIN`, `FULL OUTER JOIN`, and `SELF JOIN`. The joins can be used as follows:

➔ `INNER JOIN`

By using `INNER JOIN`, one can retrieve records that are common in both the tables participating in the join as shown in figure 3.43.

For example, for tables A and B, an `INNER JOIN` will retrieve all records that are present in A as well as B.

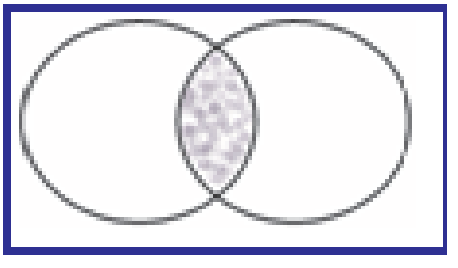


Figure 3.43: INNER JOIN

➔ `LEFT OUTER JOIN`

This type of join displays all the rows from the left table and only the matching rows from the right table as shown in figure 3.44. For example, for tables A and B, a `LEFT OUTER JOIN` will retrieve all records that are present in A and the matching records in B.

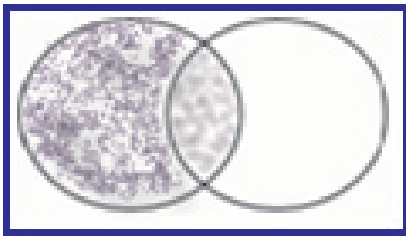


Figure 3.44: LEFT OUTER JOIN

Concepts

➔ `RIGHT OUTER JOIN`

This type of join displays all the rows from the right table and only the matching rows from the left table as shown in figure 3.45. For example, for tables A and B, a `RIGHT OUTER JOIN` will retrieve all records that are present in B and the matching records in A.

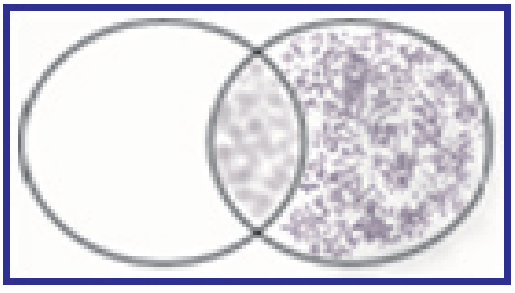


Figure 3.45: RIGHT OUTER JOIN

➔ `FULL OUTER JOIN`

The `FULL OUTER JOIN` returns all rows from both the tables including those that do not match. It combines the results of both left and right outer joins. For the records that do not have matching rows on either side of the join, `NULL` will be displayed in the fields for such records as shown in figure 3.46.

For example, for tables A and B, a `FULL OUTER JOIN` will retrieve all records that are present in A as well as B including the records that do not match in either of the tables. For the fields in the records that do not have matching record in A or B, `NULL` will be displayed.

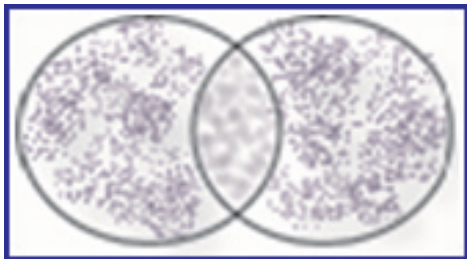


Figure 3.46: FULL OUTER JOIN

➔ `SELF JOIN`

This type of join connects a table with itself. For example, for a table A, one can retrieve all records that are present in A that match a particular condition by creating an alias A1.

3.4.2 Creating Union Queries

A union query is used to combine the result sets of multiple similar select queries. For example, suppose a database has two tables namely `Employee` and `Customer` that store information about the employees and customers of the company. Also, no relationship exists between the two tables.

Concepts

However, a user wants to view the contact information of employees as well as customers at the same time. In such a case, one can create a select query on the individual tables that retrieves only the contact information, but still that would be in two separate places. To merge the results of the two queries, one can use the union query.

While creating a union query, the individual select queries must have the same number of fields, in the same sequence, and with the same or compatible data types. When a union query is executed, data from all the fields of both the queries is merged into one output field.

The Number and Text data types can be used for a union query. The syntax of union query must be written directly in SQL. However, the user can create the individual select queries in the **Design View** and then, combine them in the **SQL View**.

The syntax of a union query is as follows:

Syntax:

```
SELECT <Field1>, <Field2>[,…]
FROM <Table1>, <Table2> [,…]
UNION [ALL]
SELECT <Field_A>, <Field_B> [,...]
FROM <Table_A>, <Table_B> [,…]
```

where,

`<Field1>, <Field2>`: Specifies the fields to be selected from the first table.

`<Table1>, <Table2>`: Specifies the tables to be used for the first query.

`UNION`: Keyword used to create a union.

`<Field_A>, <Field_B>`: Specifies the fields to be selected from the second table

`<Table_A>, <Table_B>`: Specifies the tables to be used for the second query.

In a union query, each select statement has a `SELECT` and `FROM` clause. It can also have the WHERE clause.

When the `ALL` keyword is used, duplicate rows will not be removed from the combined set generated by the union. This improves the query performance, because Access 2010 does not need to check the output for duplicate rows.

The `ALL` keyword can be used when one or more of the following conditions are true:

- ➔ It is apparent that select queries will not generate any duplicate rows.
- ➔ The presence of duplicate rows does not affect the output.
- ➔ The user wants to view the duplicate rows.

Concepts

To create a union query, perform the following steps:

1. Create two tables namely, `Employee` and `Customer` as shown in figure 3.47.

Customer

Field Name	Data Type	Description
Cust_ID	Text	Unique Id of Customer
Cust_Name	Text	Name of customer
Cust_Address	Text	Address of customer
Cust_DOB	Date/Time	Date of birth of customer
Cust_Phone	Text	Phone number of customer
Cust_Email	Hyperlink	E-mail address of customer

Employee

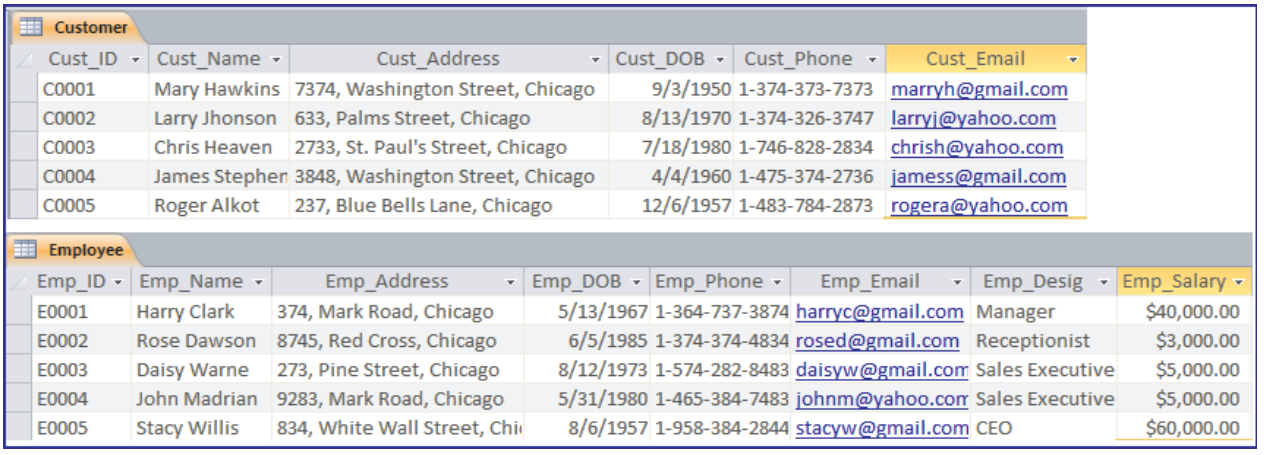
Field Name	Data Type	Description
Emp_ID	Text	Unique ID of the employee
Emp_Name	Text	Name of the employee
Emp_Address	Text	Address of the employee
Emp_DOB	Date/Time	Date of birth of employee
Emp_Phone	Text	Phone number of the employee
Emp_Email	Hyperlink	E-mail of the employee
Emp_Desig	Text	Designation of the employee
Emp_Salary	Currency	Salary of the employee

Figure 3.47: Customer and Employee Tables

2. Change the **Field Size** of `Cust _ ID` and `Emp _ ID` to `5`, `Cust _ Name` and `Emp _ Name to 100`, `Cust _ Address` and `Emp _ Address` to `150`, and `Cust _ Phone` and `Emp _ Phone` to `14`.
3. Change the **Date/Time Format** of `Cust _ DOB` and `Emp _ DOB` to `Short Date` and **Format** of **Currency** data type to `Currency`.
4. Set `Cust _ ID` and `Emp _ ID` as primary keys in the respective tables and set the **Required** property of both to `Yes`.

Concepts

5. Add data to the `Customer` and `Employee` tables as shown in figure 3.48.



Customer

Cust_ID	Cust_Name	Cust_Address	Cust_DOB	Cust_Phone	Cust_Email
C0001	Mary Hawkins	7374, Washington Street, Chicago	9/3/1950	1-374-373-7373	marryh@gmail.com
C0002	Larry Jhonson	633, Palms Street, Chicago	8/13/1970	1-374-326-3747	larryj@yahoo.com
C0003	Chris Heaven	2733, St. Paul's Street, Chicago	7/18/1980	1-746-828-2834	chrish@yahoo.com
C0004	James Stephen	3848, Washington Street, Chicago	4/4/1960	1-475-374-2736	jamess@gmail.com
C0005	Roger Alkot	237, Blue Bells Lane, Chicago	12/6/1957	1-483-784-2873	rogera@yahoo.com

Employee

Emp_ID	Emp_Name	Emp_Address	Emp_DOB	Emp_Phone	Emp_Email	Emp_Desig	Emp_Salary
E0001	Harry Clark	374, Mark Road, Chicago	5/13/1967	1-364-737-3874	harryc@gmail.com	Manager	$40,000.00
E0002	Rose Dawson	8745, Red Cross, Chicago	6/5/1985	1-374-374-4834	rosed@gmail.com	Receptionist	$3,000.00
E0003	Daisy Warne	273, Pine Street, Chicago	8/12/1973	1-574-282-8483	daisyw@gmail.com	Sales Executive	$5,000.00
E0004	John Madrian	9283, Mark Road, Chicago	5/31/1980	1-465-384-7483	johnm@yahoo.com	Sales Executive	$5,000.00
E0005	Stacy Willis	834, White Wall Street, Chi	8/6/1957	1-958-384-2844	stacyw@gmail.com	CEO	$60,000.00

Figure 3.48: Data Entered into Customer and Employee Tables

6. Click **Create** → **Query Design**. The **Show Table** dialog box is displayed.
7. Click **Close** to close the **Show Table** dialog box.
8. Save the query as `Union Query`.
9. Click **Union** option in the **Query Type** group to open the `Union Query` in the **SQL View**.
10. Type the following query in the SQL window.

```
SELECT Cust_ID, Cust_Name, Cust_Address, Cust_DOB, Cust_Phone,
Cust_Email

FROM Customer

UNION ALL SELECT Emp_ID, Emp_Name, Emp_Address, Emp_DOB, Emp_Phone,
Emp_Email

FROM Employee;
```

11. Click **Run** to execute the query. The combined records of both the tables will be displayed together.

    The names of the result fields of a union query are derived from the first `SELECT` clause by default. Therefore, the output shows the field names of the `Customer` table which was the first query.

12. To give general name to the column, modify the `Union Query` in the **SQL View** as follows:

```
SELECT Cust_ID as ID, Cust_Name as Name, Cust_Address as Address, Cust_DOB
as DOB, Cust_Phone as Phone, Cust_Email as Email

FROM Customer

UNION ALL SELECT Emp_ID, Emp_Name, Emp_Address, Emp_DOB, Emp_Phone,
Emp_Email

FROM Employee;
```

Concepts

13. Click **Run** to execute the query. The fields will now have the alias names defined in the `Union Query` as shown in figure 3.49.

Union Query

ID	Name	Address	DOB	Phone	Email
C0001	Mary Hawkins	7374, Washington Street, Chic	9/3/1950	1-374-373-7373	marryh@gmail.cor
C0002	Larry Jhonson	633, Palms Street, Chicago	8/13/1970	1-374-326-3747	larryj@yahoo.com
C0003	Chris Heaven	2733, St. Paul's Street, Chicag	7/18/1980	1-746-828-2834	chrish@yahoo.con
C0004	James Stephen	3848, Washington Street, Chic	4/4/1960	1-475-374-2736	jamess@gmail.cor
C0005	Roger Alkot	237, Blue Bells Lane, Chicago	12/6/1957	1-483-784-2873	rogera@yahoo.cor
E0001	Harry Clark	374, Mark Road, Chicago	5/13/1967	1-364-737-3874	harryc@gmail.com
E0002	Rose Dawson	8745, Red Cross, Chicago	6/5/1985	1-374-374-4834	rosed@gmail.com
E0003	Daisy Warne	273, Pine Street, Chicago	8/12/1973	1-574-282-8483	daisyw@gmail.cor
E0004	John Madrian	9283, Mark Road, Chicago	5/31/1980	1-465-384-7483	johnm@yahoo.cor
E0005	Stacy Willis	834, White Wall Street, Chicag	8/6/1957	1-958-384-2844	stacyw@gmail.con

Figure 3.49: Result of Union Query

14. Save and close the `Union Query`.

3.4.3 Creating Crosstab Queries

A Crosstab query allows a user to summarize information similar to spreadsheet. For example, to analyze the marks of students in various subjects, select query would not be a good choice because one has to scroll to the specific student to read the details. In crosstab query, one needs to provide at least one **row heading**, one **column heading**, and a **value**.

To view the marks secured by each student in different courses, one needs to change the student names to column or field heading, and course name into row heading, and assign value to marks.

Suppose the manager of **ToyHeaven Pvt. Ltd.** wants to view the stock of all products in different categories.

To create a crosstab query to know the stock of different products in different categories, perform the following steps:

1. Click **Create** → **Query Design**. The **Show Table** dialog box is displayed.
2. Double-click the `Product` table to add it to the query pane and close the **Show Table** dialog box.
3. Add the fields `Prod _ Name`, `Prod _ Category`, and `Prod _ Stock` in the lower section of the query window and save the query as **Crosstab Query**.
4. Click **Crosstab** in the **Query Type** group of the **Design** tab. Two new rows namely, **Total** and **Crosstab** are added in the lower section.
5. Select **Row Heading** for `Prod _ Name` field in the **Crosstab** row, **Column Heading** for `Prod _ Category`, and **Value** for `Prod _ Stock`.

Concepts

6. Select **Sum** from the **Total** row drop-down of the `Prod _ Stock` field since, **Group By** cannot be specified for a **Value** field as shown in figure 3.50.

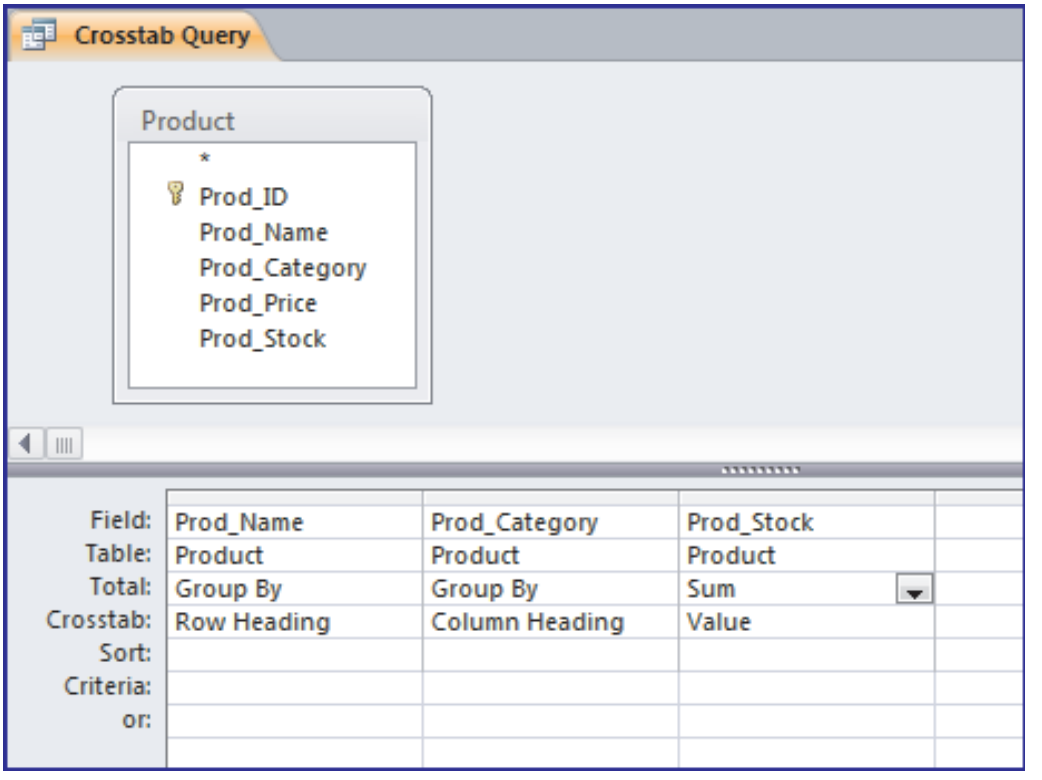


Figure 3.50: Crosstab Query

7. Click **Run** to execute the query. The result of query is displayed as shown in figure 3.51.

Crosstab Query

Prod_Name	Action Figure	Car	Construction	Creative	Doll	Educational
Barbie					200	
Blocks			100			
Capsela			100			
Construx			150			
Dinky Cars		200				
Fulla					100	
G.I.Joe	200					
Hotwheels		20				
Jumping Jack	150					
Lego Mindstorms						150
Silly Putty				250		
Speak and Spell						150

Figure 3.51: Result of Crosstab Query

Concepts

Notice that the product names are now the column headings and categories are now the row headings. The user gets a clearer picture of the data without having to scroll to individual product and category to view the details.

8. Save and close the `Crosstab Query`.

3.4.4 Creating Subqueries

A subquery is an SQL `SELECT` statement nested inside another SQL statement. Subqueries can be a part of a `SELECT`, `INSERT`, `DELETE`, or `UPDATE` statements.

Suppose a user wants to know the frequency with which the customers place orders for products. The data can be retrieved by creating a query across the `Customer` and `Orders` tables by performing the following steps:

1. Create an `Orders` table as shown in figure 3.52.

Orders

Field Name	Data Type	Description
Order_ID	Text	Unique ID of an order
Cust_ID	Text	Unique Id of a customer
Order_Date	Date/Time	Date of order placement
Order_Amt	Currency	Total payable amount of the order
Order_Delivery	Yes/No	Delivery status of an order
Order_Payment	Yes/No	Payment status of an order

Figure 3.52: Orders Table

2. Change the **Field Size** of `Order _ ID` and `Cust _ ID` fields to `5`, **Format** of **Date/Time** to `Short Date`, and Format of data type of `Order _ Delivery` and `Order _ Payment` fields to `Yes/No`.
3. Set the **Required** property of `Order _ ID` and `Cust _ ID` to Yes and make `Order _ ID` the primary key.
4. Set the **Format** of **Currency** data type to `Currency`.
5. Create a one-to-many relationship between the `Customer` and `Orders` tables on the `Cust _ ID` field along with referential integrity as shown in figure 3.53.

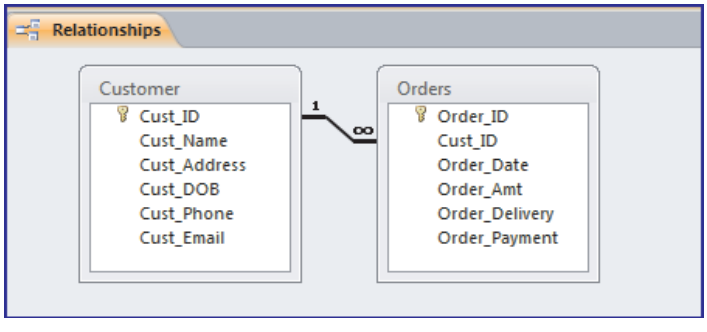


Figure 3.53: Creating Relationship between Customer and Orders Tables

Concepts

6. Save and close the **Relationships** pane.

7. Open the `Orders` table in **Datasheet View** and add the data as shown in figure 3.54.

Orders

Order_ID	Cust_ID	Order_Date	Order_Amt	Order_Deliv	Order_Paym
O0001	C0002	4/4/2012	$4,000.00	☑	☑
O0002	C0001	5/5/2012	$2,000.00	☑	☑
O0003	C0002	5/31/2012	$1,000.00	☑	☑
O0004	C0003	6/4/2012	$5,000.00	☑	☑
O0005	C0004	6/6/2012	$2,000.00	☑	☑
O0006	C0001	6/8/2012	$3,500.00	☑	☑
O0007	C0002	7/10/2012	$5,500.00	☑	☑
O0008	C0005	8/8/2012	$1,000.00	☑	☑
O0009	C0004	8/30/2012	$3,000.00	☑	☑
O0010	C0005	9/2/2012	$1,000.00	☑	☐

Figure 3.54: Data Added to Orders Table

8. Close the `Orders` table and click **Create** → **Query Design** to open the **Show Table** dialog box.

9. Double-click the `Customer` and `Orders` table to add them to the query window and close the **Show Table** dialog box.

10. Save the query as `Subquery Query`.

11. Open the `Subquery Query` in the **SQL View** and type the following query in the SQL window:

```
SELECT Orders.Cust_ID, Orders.Order_Date, (SELECT MAX(Order_Date)
FROM Orders AS Orders1
WHERE Orders1.Order_Date <Orders.Order_Date
AND Orders1.Cust_ID = Orders.Cust_ID) AS PreviousOrderDate,
[Order_Date]-[PreviousOrderDate] AS OrderInterval
FROM Customer INNER JOIN Orders
ON Customer.Cust_ID = Orders.Cust_ID
ORDER BY Orders.Cust_ID, Orders.Order_Date DESC;
```

The subquery '`SELECT MAX(Order_Date) FROM Orders AS Orders1 WHERE Orders1.Order_Date <Orders.Order_Date AND Orders1.Cust_ID = Orders.Cust_ID`' uses a SELF JOIN in which the `Orders` table queries itself using an alias `Orders1`. It selects maximum date from records of the `Orders` table where the `Order_Date` in `Orders1` is less than the `Order_Date` of Orders table for all customers having `Cust_ID` value of `Orders1` table same as the `Orders` table. These values are stored as the `PreviousOrderDate` column.

Concepts

Next, the `PreviousOrderDate` values are subtracted from the corresponding customer's `Order_Date` value and stored as the `Order_Interval` column based on the matching values of `Customer.Cust_ID` with `Orders.CUST_ID` and sorted in the descending order of the `Cust_ID` and `Order_Date` fields of the `Orders` table.

12. Click **Run** to execute the query. The details of each order and its previous order with the order interval are displayed as shown in figure 3.55.

Subquery Query

Cust_ID	Order_Date	PreviousOrderDate	OrderInterval
C0001	6/8/2012	5/5/2012	34
C0001	5/5/2012		
C0002	7/10/2012	5/31/2012	40
C0002	5/31/2012	4/4/2012	57
C0002	4/4/2012		
C0003	6/4/2012		
C0004	8/30/2012	6/6/2012	85
C0004	6/6/2012		
C0005	9/2/2012	8/8/2012	25
C0005	8/8/2012		

Figure 3.55: Result of Subquery Query

When the query is executed from the **SQL View**, Access 2010 automatically generates the **Design View** of the query.

13. Open the `Subquery Query` in the **Design View** as shown in figure 3.56.

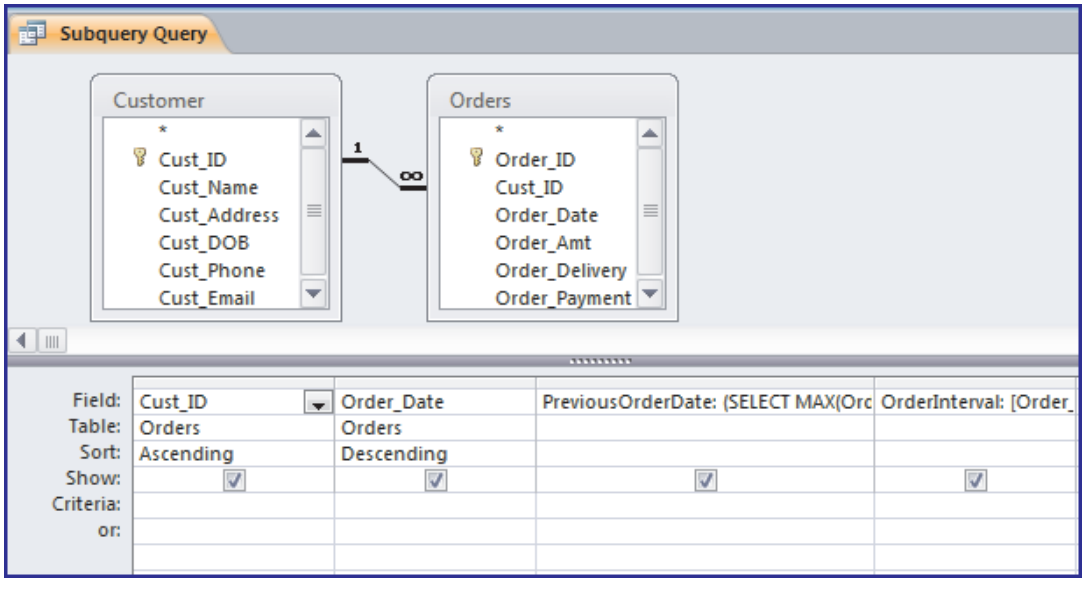


Figure 3.56: Result of Subquery Query

Concepts

Note that the `Customer` table does not appear anywhere in the **Table** row. This is because it is only used to create a join with the `Orders` table on the column `Cust_ID` and not used anywhere in the selection list.

14. Save and close the `Subquery Query`.

3.4.5 Types of Subqueries

Subqueries can be created to retrieve different types of analytical data as follows:

- Display a value from the next or previous record in a table.
- Display the TOP n values of a field.
- Display values that do not exist in a particular set of values.
- Display a value as a percentage of a total.
- Calculate or filter values from tables that are not part of the query.

Many other tasks can be done by using subqueries as follows:

- **Using the TOP n Records**

  Suppose a user wants to retrieve the list of products that have product price less than any of the top five product prices. The data can be retrieved by creating a subquery on the `Product` table by performing the following steps:

  1. Create query on the `Product` table in the **Design View** and save it as `Top 5 Subquery`.
  2. Open the query in the **SQL View.**
  3. Type the following `SELECT` statement in the SQL window:

```
SELECT Prod_ID, Prod_Price
FROM PRODUCT
WHERE Prod_Price< ANY
(SELECT TOP 5 Prod_Price
FROM Product
ORDER BY Prod_Price DESC)
ORDER BY Prod_Price DESC;
```

The `SELECT` statement uses a subquery '`SELECT TOP 5 Prod_Price FROM Product ORDER BY Prod_Price DESC`' that returns a list of the top five prices in the `Product` table as shown in figure 3.57.

Prod_Price
500
400
300
250
250

Figure 3.57: Result of Nested Query

It can be seen that the values 500 and 250 are repeated since, there is more than one instance of these values.

Now, in the outer query, the nested query gets replaced by the values returned from its execution as follows:

'`SELECT Prod_ID, Prod_Price FROM PRODUCT WHERE Prod_Price < ANY(500, 500, 400, 300, 250, 250) ORDER BY Prod_Price DESC;`'.

When the outer query gets evaluated, Access 2010 looks for products with `Prod_Price` value less than `ANY` of the values listed in the nested query result and displays them in the descending order of `Prod_Price` values as shown in figure 3.58.

Top 5 Subquery

Prod_ID	Prod_Price
P0001	400
P0008	300
P0010	250
P0007	250
P0006	200
P0005	200
P0004	200
P0002	200
P0012	120
P0011	100
P0009	100

Figure 3.58: Result of Top 5 Subquery

Concepts

Notice that the value 250 is displayed in the result even though it was mentioned in the top five prices. This is because of the use of ANY. 250 is not less than 250 but is less than 300, 400, and 500. Therefore, it is displayed. Same is the case with 300 and 400 that also get displayed in the result because they are less than 500.

4. Save and close the Top 5 Subquery query.

➔ Select the Unmatched Records

Suppose a user wants to view the customers whose records exist in the Orders table and the order amount is greater than 4000, one can create a subquery on the Customer and Orders table as follows:

```
SELECT Cust_ID, Cust_Name
FROM Customer
WHERE NOT EXISTS
   (SELECT Orders.Order_ID
   FROM Orders
   WHERE Orders.Cust_ID = Customer.Cust_ID
     AND Orders.Order_Amt> 4000);
```

The query 'SELECT Orders.Order_ID FROM Orders WHERE Orders.Cust_ID = Customer.Cust_ID AND Orders.Order_Amt > 4000' returns the list of all Order IDs that match the Customer ID value from the customer table and where the Order_Amt is greater than 4000.

Next, the outer query 'SELECT Cust_ID, Cust_Name FROM Customer WHERE NOT EXISTS' uses the list of values to display only those customers whose orders do not exist in the list returned by the inner query.

➔ Delete Duplicate Records

To delete duplicate records from the Customer table with similar value for Cust_Name, first create a duplicate record number C0006 and give the name as 'Roger Alkot' similar to record C00005. Now, use the following query to delete the duplicate record:

```
DELETE FROM Customer
WHERE Cust_ID NOT IN (SELECT Min(Cust_ID) AS MinOfStudID FROM Customer AS
Duplicate
WHERE Duplicate.Cust_Name = Customer.Cust_Name);
```

Concepts

3.5 Manipulate Fields

Access 2010 allows manipulation of fields in a query to add, remove, sort, show, and hide the fields. It also provides the ability to create custom fields according to requirement.

3.5.1 Generate Calculated Fields Using Zoom Box and Expression Builder

Calculated fields allow a user to display custom data in the tables. When a calculated field is created, a new value is added to each row that contains the calculated result based on values of other fields of the same row. To create a calculated field, an expression made of field names in the table and operators must be created. A calculated field can be created by using the Zoom box or the Expression Builder.

For example, suppose the company **ToyHeaven Pvt. Ltd.** organizes free medical check-ups once every year for all the employees above the age of 40. The manager of **ToyHeaven Pvt. Ltd.** wants to view the age of all the employees to decide who are eligible for a free medical check-up.

One way to do this is to create a query with a field named age containing a calculated field based on the current date and date of birth of each employee. This can be done using the **Zoom box** or **Expression Builder** as follows:

➔ **Using Zoom Box**

To create a calculated field using Zoom box to generate the age of an employee, perform the following steps:

1. Click **Create** → **Query Design** to open the **Show Table** dialog box.
2. Double-click the `Employee` table to add it to the query pane and close the **Show Table** dialog box.
3. Save the query as '`Calculated Field Query`'.
4. Add the fields `Emp _ ID`, `Emp _ Name`, and `Emp _ DOB` in the lower portion of the query pane.

Concepts

1. In the fourth column of the query pane, right-click inside the field box and select **Zoom** from the menu as shown in figure 3.59.

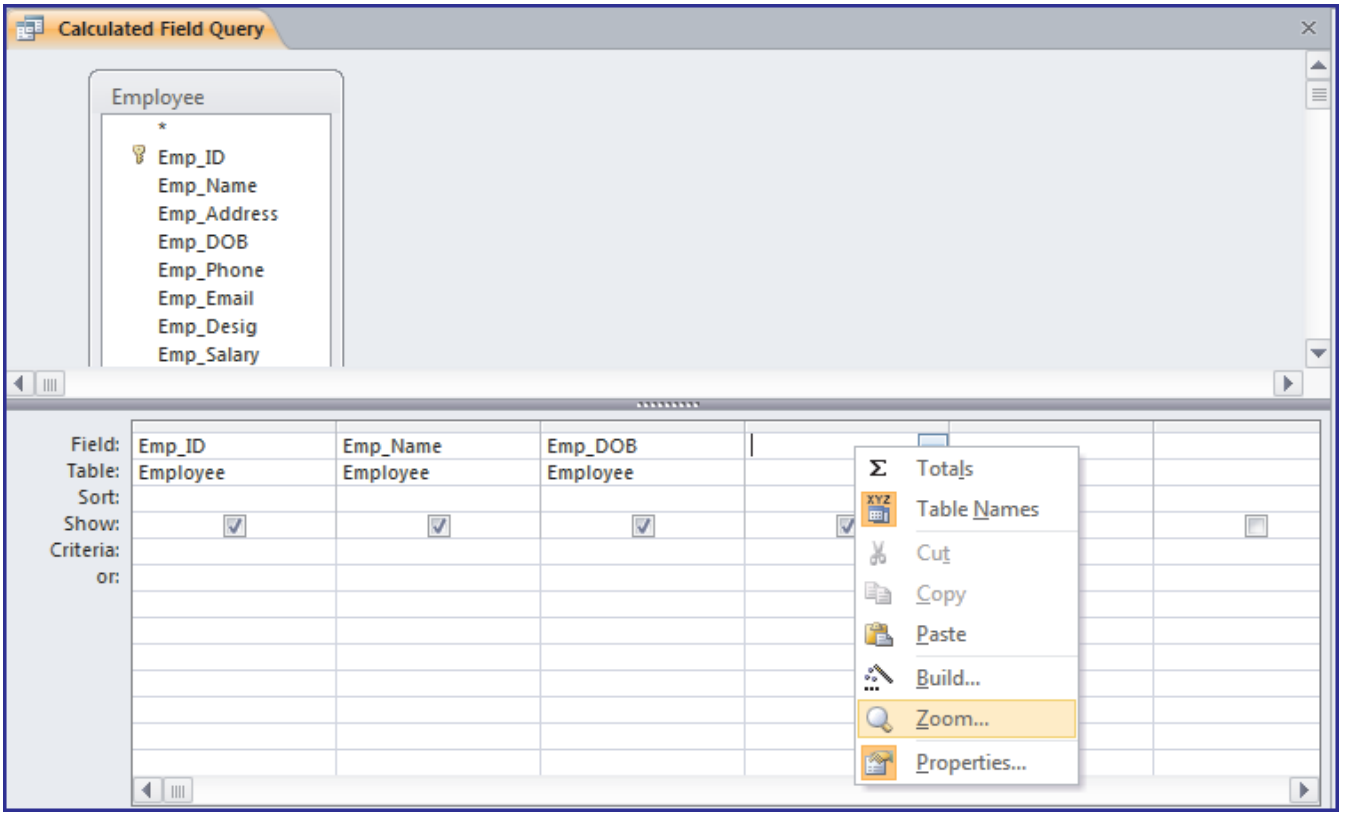


Figure 3.59: Zoom Option

6. Type the expression 'Age: Right$(Date(),4)-Right$([Emp _ DOB],4)' in the **Zoom** dialog box as shown in figure 3.60.

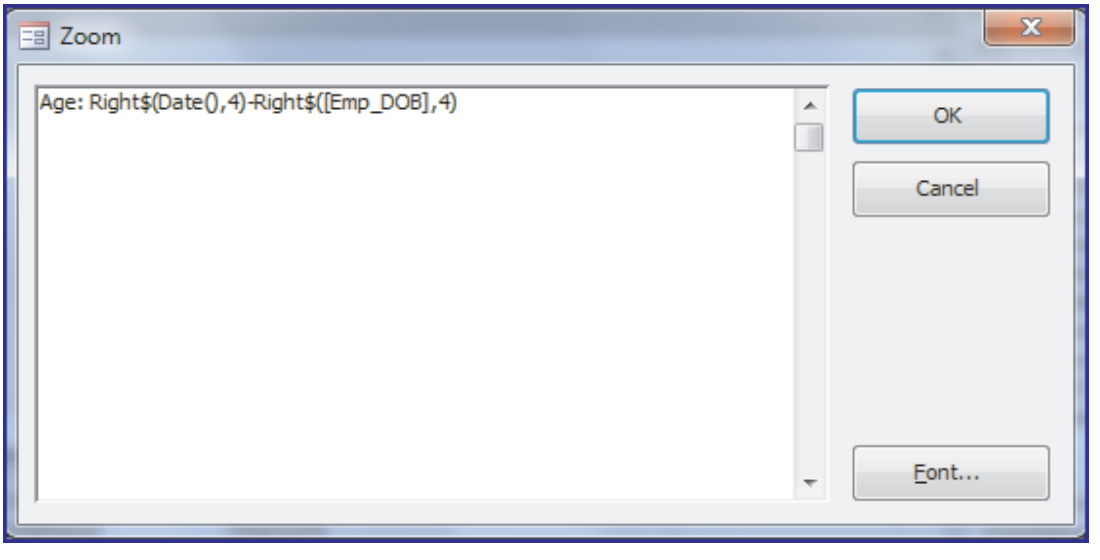


Figure 3.60: Zoom Dialog Box

Concepts

The function `Right$()` takes two parameters namely, the string and the number of characters. It returns the number of characters from the right of the string specified. In the expression, the `Right$()` function returns the right most four characters that represent the year of the date. The years are then subtracted to get the age of the employee.

A user can also use the `DATEPART(<interval>,<date>)` function to retrieve the year of the dates as follows:

```
Age: DatePart("yyyy",Date())-DatePart("yyyy",[Emp_DOB])
```

The interval specifies the part of date that is to be retrieved. Valid values for interval are shown in table 3.6.

Interval	Explanation
yyyy	year
q	quarter
m	month
y	day of year
d	day
w	weekday
ww	week
h	hour
n	minute
s	second

Table 3.6: Interval Values for DatePart() Function

7. Click **OK**. The expression will be added to the **Field** row of the fourth field.
8. Click **Run** to execute the query. The Age column consisting of the calculated age is displayed as shown in figure 3.61.

Calculated Field Query

Emp_ID	Emp_Name	Emp_DOB	Age
E0001	Harry Clark	5/13/1967	45
E0002	Rose Dawson	6/5/1985	27
E0003	Daisy Warne	8/12/1973	39
E0004	John Madrian	5/31/1980	32
E0005	Stacy Willis	8/6/1957	55

Figure 3.61: Calculated Field

9. Save and close the `Calculated Field Query`.

➔ Using Expression Builder

To create a calculated field using **Expression builder** to generate the age of an employee, perform the following steps:

1. Click **Create** → **Query Design** to open the **Show Table** dialog box.
2. Double-click the `Employee` table to add it to the query pane and close the **Show Table** dialog box.
3. Save the query as '`Calculated Field Query1`'.
4. Add the fields `Emp _ ID`, `Emp _ Name`, and `Emp _ DOB` in the lower portion of the query pane.
5. Click inside **Field** row of the fourth column and click **Builder** option in the **Query Setup** group of the **Design** tab. The **Expression Builder** dialog box is displayed.

Note - Expression Builder dialog box can also be opened by right-clicking inside the field box and selecting **Build** from the menu.

6. Type the expression '`Age: Right$(Date(),4)-Right$([Emp _ DOB],4)`' in the dialog box as shown in figure 3.62.

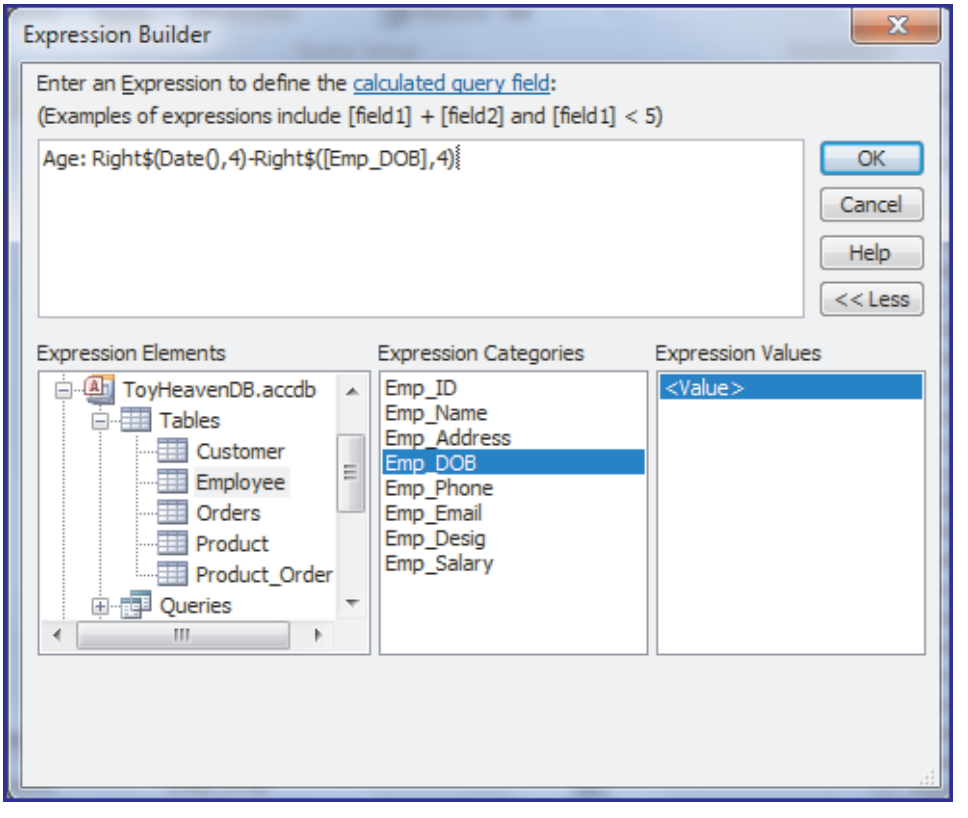


Figure 3.62: Expression Builder Dialog Box

Concepts

Alternatively, the `Emp_DOB` field can also be selected from the **Expression Elements** list as shown in figure 3.62. The operators and functions can be selected from the **Expression Values** list as shown in figure 3.63.

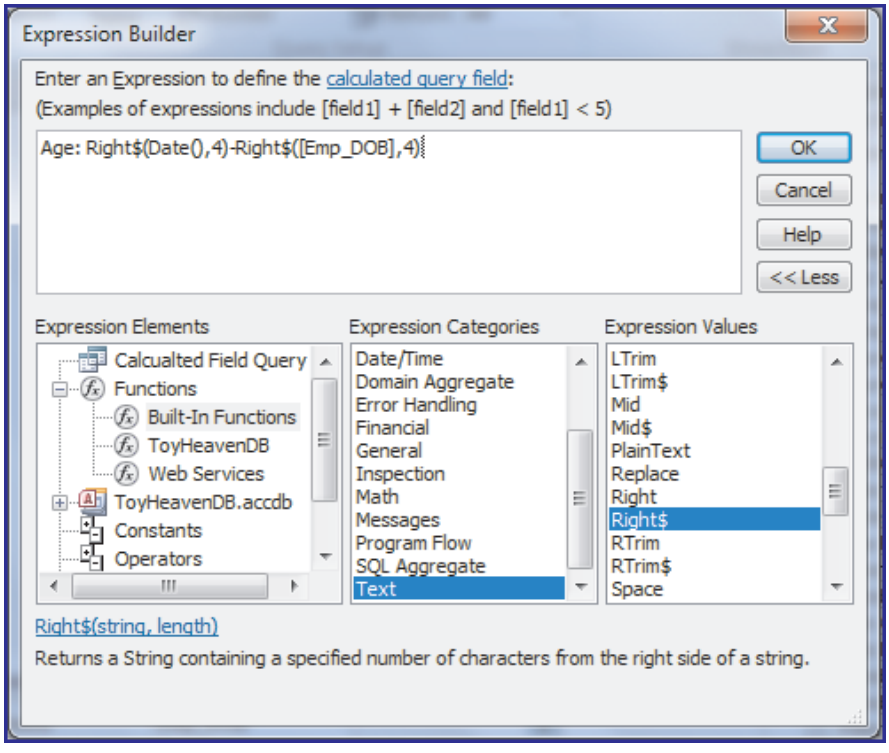
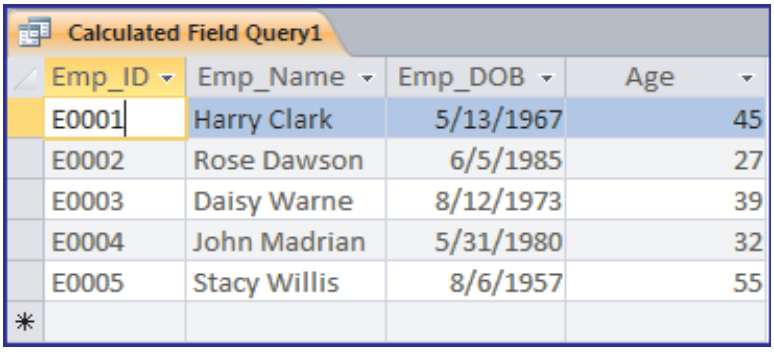


Figure 3.63: Using Expression Elements List

7. Click **OK**. The expression is added to the **Field** row of the fourth column.

8. Click **Run** to execute the query. Click **Run** to execute the query. The `Age` column consisting of the calculated age is displayed as shown in figure 3.64.



Emp_ID	Emp_Name	Emp_DOB	Age
E0001	Harry Clark	5/13/1967	45
E0002	Rose Dawson	6/5/1985	27
E0003	Daisy Warne	8/12/1973	39
E0004	John Madrian	5/31/1980	32
E0005	Stacy Willis	8/6/1957	55

Figure 3.64: Result of Calculated Field Query1

9. Save and close the `Calculated Field Query1`.

Concepts

3.5.2 Adding a Field

Suppose the manager of **ToyHeaven Pvt. Ltd.** requires viewing the product details such as product ID, name, price, as well as the stock. Instead of creating a new query, user can modify the existing `Display Products Query` to include the `Prod_Stock` field.

To add a field to a query in the design view, perform the following steps:

1. Open the `Display Products Query` in the **Design View**.
2. In the **Field** row of the fourth column in the lower section of the query pane, select `Prod _ Stock` from the drop-down list as shown in figure 3.65.

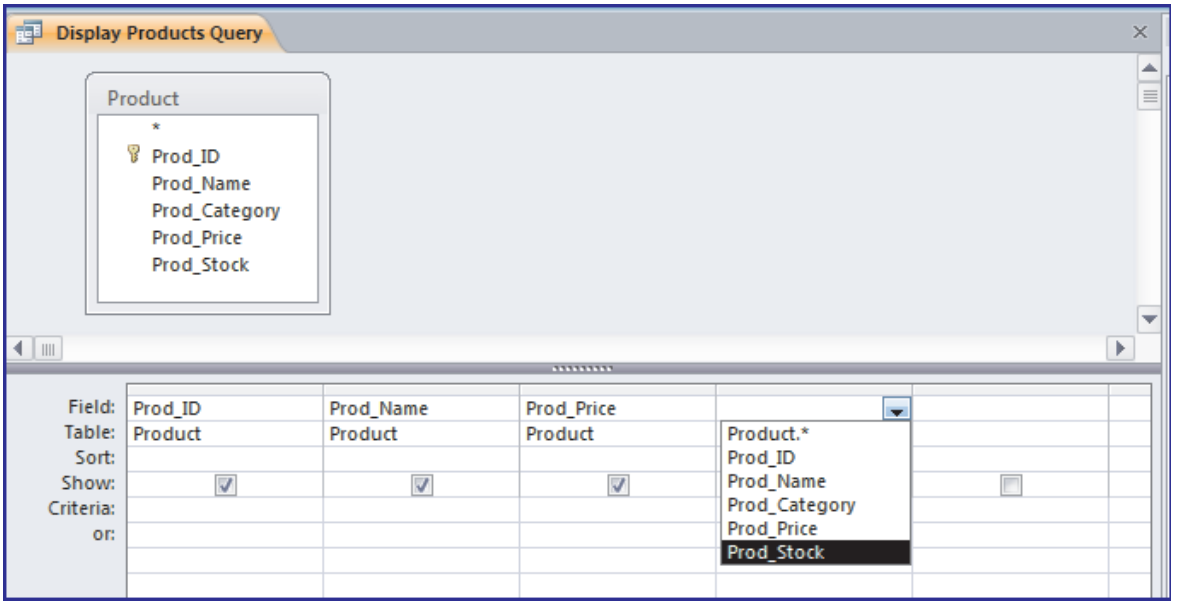


Figure 3.65: Adding a Field to Display Products Query

3. Click **Run** to execute the query. The result will show the `Prod _ Stock` column also.
4. Save and close the `Display Products Query`.

3.5.3 Removing a Field

While removing a field, the user must ensure that the field is not used in any calculated field, or is not part of any relationship.

To delete a column, do any of the following:

1. Select the column in the **Design View** and press **DELETE**.
2. Right-click the column header and select **Cut** from the menu.
3. To delete multiple columns, select the columns and press **DELETE**.

Concepts

4. To remove a column from an SQL statement, the user needs to simply delete it. For example: To remove the `Emp _ DOB` field from the query '`Select Emp _ ID, Emp _ Name, Emp _ DOB FROM Employee`'; change it to '`Select Emp _ ID, Emp _ Name FROM Employee`'.

3.5.4 Re-arranging Field Using Adding Filter, Sort, and Show Options

Access 2010 allows specifying filter criteria, sorting options and show/hide facility for a field in a query. For example, filter and sort can be used to view the personal details and salary of employees whose date of birth is on or after `January 01, 1970`, salary greater than `$1000`, and arrange them in the descending order of the salary.

To specify a filter, sort and show/hide options to a query, perform the following steps:

1. Create a new query in **Design View** on the `Employee` table and save it as `Filter and Sort Query`.
2. Add the `Emp _ ID`, `Emp _ Name`, `Emp _ Address`, `Emp _ DOB`, `Emp _ Desig`, and `Emp _ Salary` fields to the lower section of the query window.
3. Specify the criteria in the `Prod _ Date` column as '`>=#1/1/1970#`' and in the `Emp _ Salary` column as '`>1000`'.
4. In the **Sort** row of `Prod _ DOB`, select **Ascending** and in the `Prod _ Salary` select **Descending**.
5. In the **Show** row, clear the check boxes of `Emp _ Address` and `Emp _ Desig` columns to hide them from the query results as shown in figure 3.66.

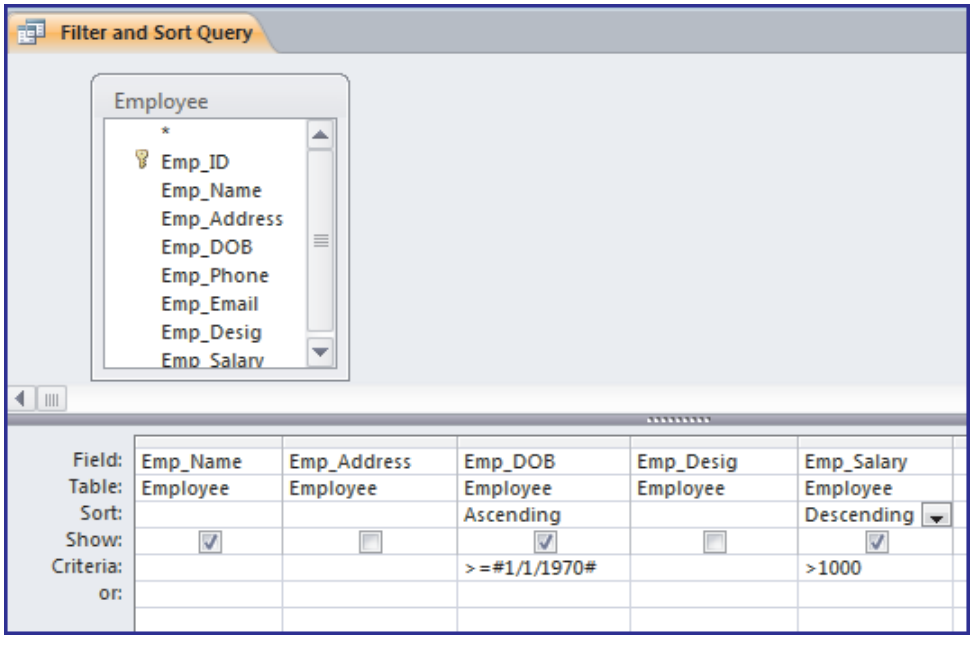


Figure 3.66: Specifying Filter, Sort, and Show/Hide Options

Concepts

6. Click **Run** to execute the query. The result of query is as shown in figure 3.67.

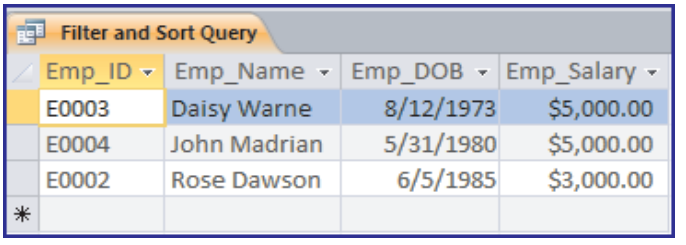


Figure 3.67: Result of Filter and Sort Query

Note that the query does not include records of employees `E0001` and `E0005` even though their salary is greater than `$1000` because the criterion for date of birth is not fulfilled.

7. Save and close the `Filter and Sort Query`.

3.6 Check Your Progress

1. The _____ statements are used to make structural changes to the database such as creation, modification, and deletion of tables.

(A)	Data Manipulation Language
(B)	Data Control Language
(C)	Data Definition Language
(D)	Transaction Control Language

2. Which of the following statements about a Union query are true?

(A)	A union query is used to combine the result sets of multiple similar select queries.
(B)	For a Union query, a relationship must exist between the two tables.
(C)	While creating a union query, the individual select queries must have the same number of fields, in the same sequence, and with the same or compatible data types.
(D)	The Number and Text data types cannot be used for a union query.

3. Match the following statement type with the corresponding description.

Statement Type		Description	
(A)	`DROP TABLE`	1)	Adds a new record to a table.
(B)	`INSERT`	2)	Removes the records from the table.
(C)	`ALTER TABLE`	3)	Deletes a table.
(D)	`DELETE`	4)	Modifies a table.

Concepts

4. Which of the following queries can be created only in the SQL View?

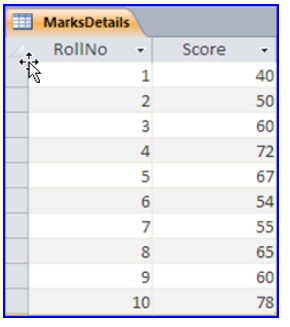
(A)	Update Query
(B)	Delete Query
(C)	Crosstab Query
(D)	Union Query

5. Identify the query that can be used to retrieve the top three scores in a class.

(Assumption: The name of the table is `MarksDetails` with fields `RollNo` and `Score` with data type `Number`)

(A)	`SELECT TOP 3 Score FROM MarksDetails ORDER BY Score ASC;`
(B)	`SELECT TOP 3 Score FROM MarksDetails ORDER BY Score DESC;`
(C)	`SELECT TOP 3 Score FROM MarksDetails;`
(D)	`SELECT TOP 3 Score FROM MarksDetails GROUP BY Score DESC;`

6. Consider the following table and query as shown in figure 3.68.


MarksDetails

RollNo	Score
1	40
2	50
3	60
4	72
5	67
6	54
7	55
8	65
9	60
10	78

Figure 3.68: Markdetails Table

```
SELECT RollNo
FROM MarksDetails
WHERE Score*100/80 > 80;
```

What will be the output of the query?

Concepts

(A)	RollNo 3 5 8 9
(B)	RollNo 10 8 5 4
(C)	RollNo 4 5 8 10
(D)	RollNo 9 8 5 3

Concepts

3.6.1 Answers

1.	C
2.	A - True, B - False, C -True, D -False
3.	A – 3, B – 1, C – 4, D - 2
4.	D
5.	B
6.	C

Summary

- A query is a request made to one or more tables of a database for viewing data, results of action on data, or both.
- The DDL statements are used to make structural changes to the database such as creation, modification, and deletion of tables.
- Data Manipulation Language (DML) is used to fetch, add, modify, or delete records from a table.
- The HAVING clause is used to specify which of the grouped records should be displayed in a SELECT statement along with a GROUP BY clause.
- A union query is used to combine the result sets of multiple similar select queries.
- A subquery is an SQL SELECT statement nested inside another SQL statement.
- INNER JOIN allows a user to retrieve records that are common in both the tables participating in the join.

Try it Yourself

CompuTech Pvt. Ltd. is a famous computer parts manufacturing company in **Chicago**. The company stores all the data about its products in a database named CompuTechDB. There are several categories of products such as Keyboard, Mouse, Monitor, Processor, Motherboard, Printer, and Scanner. The details of products, customer, sales, and employees are stored in the respective tables. Each product has an ID, Name, Category, Description, Price, and Stock. Similarly, Customer table has information such as Customer ID, Name, Address, City, Date of Birth, Phone, and E-mail. Sales table has information such as Sales Id, Product ID, Employee ID, Sales Amount, and Sales Date. The Employee Table has details such as Employee Id, Name, Designation, Date of Birth, Date of Joining, and Salary. The CEO of the firm has asked the manager to generate some data for analyzing the company's functioning and details about the existing products as follows:

1. List the Id, name, and price of all products of category `Monitor` and `Printer`.
2. List the products with sales amount greater than `400`.
3. Use Expression Builder to calculate the age of all the employees and display the list of employees with age less than `40`.
4. Display the list of all customers with date of birth greater than `January 01, 1972` residing in the city of `Chicago`.
5. Use Crosstab query to display the price of different products in different categories.
6. Use the Zoom tool to create a calculated field that displays the number of years of employment of each employee.
7. Display the list of the top six sales along with the employee Id and name.

Concepts

Session - 3 (Workshop)

Working with Queries

In this workshop, you will learn to:

- ➔ Create a query using Query Wizard
- ➔ Create a query using Query Design
- ➔ Create an update query, union, and subquery

3.1 Creating Queries using Query Wizard and Query Design

You will view and practice how to create queries using Query Wizard and Query Design. You will also view and practice how to create update, union, and subquery queries.

- ➔ Creating a Query using Query Wizard
- ➔ Creating a Query using Query Design
- ➔ Creating an Update query, Union query, and Subquery

Note - Please refer to the respective lab deliverable for demonstrations and guided simulations.

Labs

Session - 4

Working with Forms

Welcome to the Session, **Working with Forms**.

This session describes an overview of forms, explores various form creation approaches, subforms, and other features of forms. The session also describes how to add and format various controls on the form. Finally, the session further describes the sorting and filtering of data in a form.

In this Session, you will learn to:

- Describe an overview of forms
- Explain different approaches to create various types of forms
- Describe subforms and their creation
- Describe the other features of forms
- Explain adding and formatting controls on a form
- Explain sorting and filtering data in a form

4.1 Introduction

Consider a scenario of an online ticket reservation system for flights. In this scenario, customers need not have to wait in long queues for booking their tickets. Customers can quickly book their tickets to a destination of their choice without even leaving their homes or standing in queues. Nowadays, forms are a means of communication or data entry through which people can book their tickets online, make online utility payments, and so on.

Forms in Access 2010 provide a simple way to create, modify, and navigate the records in a database. A form also provides several advantages over a database table, in terms of user interactivity and appearance. Forms make the database look attractive with different themes, colors, fonts, and charts. Forms also provide navigation through hyperlinks, buttons, Web controls, and tab controls.

4.2 Overview of Forms

Some of the advantages of a form over a table are as follows:

- It provides a detailed view of records by displaying a single record at a time. Hence, a user can view all the field values of a record at the same time, thereby avoiding the confusion in identifying the corresponding data of each record.

- It reduces the chances of invalid data entry by allowing users to use controls for entering appropriate data.

- It provides different options for customizing the way in which the data is displayed. Access 2010 allows a user to add graphics and other objects to enhance the appearance of a form.

4.3 Tools to Create Forms

There are different types of forms in Access 2010. Various tools are available to create these forms. These tools are present in the **Forms** group of the **Create** tab.

4.3.1 Types of Form Tools

The forms tools listed in the **Forms** group are as follows:

- **Form**

  In this simple **Form** tool, all fields from the underlying table are displayed on the form. Users can start using the new form directly or modify it using the **Design** or **Layout** view to meet their requirements. Consider that Access 2010 finds an individual table that includes one-to-many relationship with a query or table used for building a form. In such a case, Access 2010 adds a datasheet to the form that is based on the related query or table.

Concepts

➔ Form Design

Choosing this option creates a blank form in the **Design** view. In the **Design** view, users can implement advanced design changes to a form such as adding custom controls. For customizing a form, the **Property Sheet** option in the **Tools** group provides access to a huge number of properties that can be set in the form.

➔ Navigation

The **Navigation** option allows users to build Web-based database applications by creating navigation tabs such as **vertical**, **horizontal**, **right**, or **left**. When the users click these tabs the child forms are loaded.

➔ Form Wizard

When users want to create a form in an easy manner and at the same time select fields as per their needs, then, in such a scenario, they can use the **Form Wizard**. This wizard also allows sorting and grouping of data. Data can be used from more than one table if relationships have been already specified in these tables.

➔ Blank Form

Users can create the form using the **Blank Form** tool if they want to add just a few fields on the form. Using **Blank Form** will make it easier to achieve this.

➔ Pop-Up Form

Access 2010 allows the user to create a pop-up form for displaying any important information. The information can include a calendar for checking a date or help information that can aid the user in filling the form.

➔ More Forms

The **More Forms** icon has a drop-down menu that will allow the users to select a **Datasheet**, **PivotTable**, **Modal Dialog**, **Form Wizard**, and so on. The following are the menu items provided by the **More Forms** option:

- **Multiple Items** - When a user creates a form using the simple Form tool, it displays one record at a time. If the users want to display more items on the form, then, they can use the multiple items tool.
- **Datasheet** - This form displays a single record in a tabular form or several records in a typical datasheet view. It is used as a subform that displays one-to-many relationship.
- **Split Form** - This type of a form displays the data in two formats namely, a datasheet view and a form view. Both the views work in synchronization with each other and are connected to an identical data source. Users can edit the data in any of the forms, as the same will be reflected in the other part of the form.

- **Modal Dialog** - This is a special form that allows user to gather some data or can deliver some important data to the users. Users need to be aware of **Form Design** in order to build a Modal form.
- **Pivot Chart** - This chart is a graphical representation of the data in a Pivot table. It has the ability to move the data fields from one part to another, filter data, hide items, and many more.
- **Pivot Table** - This form sums up hundreds of records in a short tabular format that allow users to modify the layout of the table. It also enables users to see different views of their data.

4.4 Creating Forms

Access 2010 helps users to create a form automatically and display the form in the form layout view. To create forms in Access 2010, the user can select tools such as **Blank Form**, **Form Design**, or **Form Wizard**.

Figure 4.1 displays the available form options used to create a form.

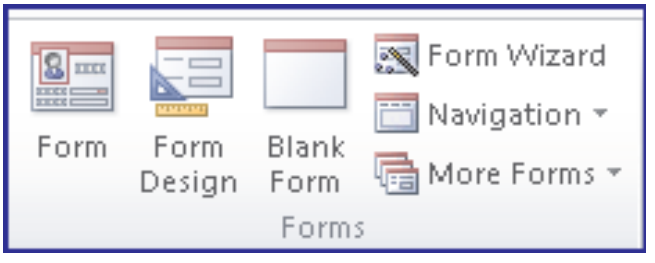


Figure 4.1: Form Options

The users can add buttons, labels, graphs, and images in the forms and enhance the process of working and entering data in an accurate and easy way.

4.4.1 Creating Simple Forms Using the Form Wizard

Consider the scenario of an international school having students from many different countries. To manage the student data in the school, a database is created with a few tables. One of the tables is named `Student`. You need to create a form based on this table.

The steps for creating a form using the **Form Wizard** are as follows:

1. In Access 2010, click the **Create** tab. The **Forms** group will be displayed. In the **Forms** group, click the **Form Wizard** icon. The **Form Wizard** dialog box will be displayed.

Concepts

Figure 4.2 displays the **Form Wizard** dialog box.

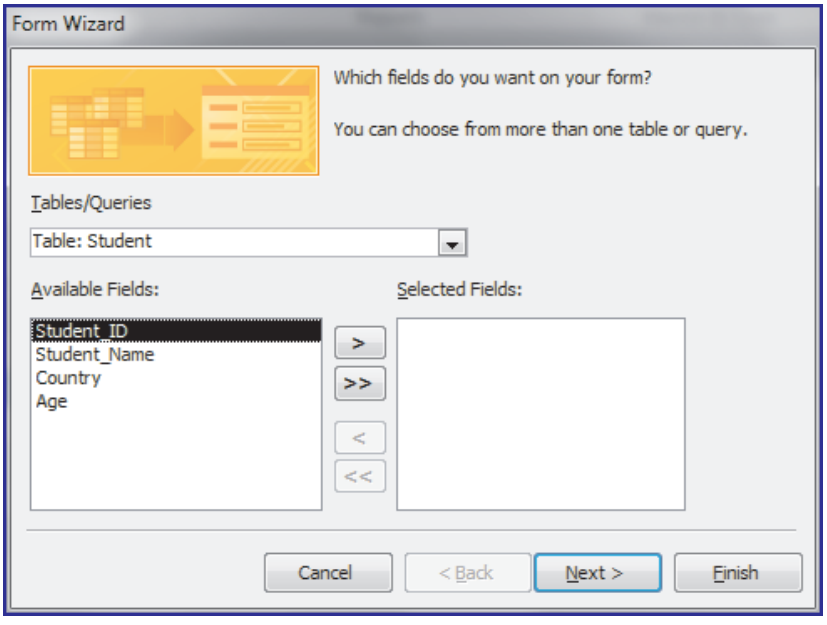


Figure 4.2: Form Wizard

In the **Table/Queries** section, select the table. On the left, the **Available Fields** will display the list of columns in the table.

2. To select the fields that should appear on the form, click the specific field in the **Available Fields** and click **>**. The selected field will appear on the right.
3. To select all fields, click **>>**. This will select all data fields from the table.
4. Click **Next>**. The **Form Wizard** will display the layout view options to be applied to the forms.

Figure 4.3 displays the different types of layout.

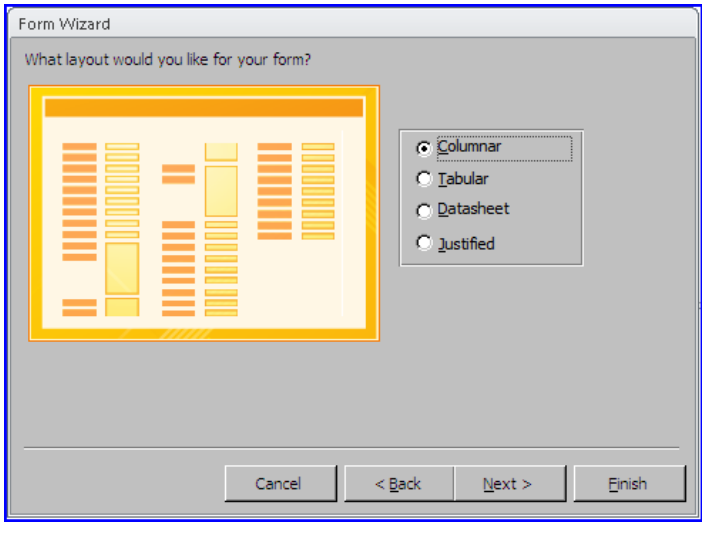


Figure 4.3: Types of Layout

Concepts

5. Click the **Columnar** option.
6. Click **Next**. The **Form Wizard** will ask the title to be displayed on the form. You can change the title if required. It also displays the option to open or modify the form.
7. Select the **Open the form to view or enter the information**.
8. Click **Finish**.

Figure 4.4 displays the `Student_Details` form that is created for the `Student` table.

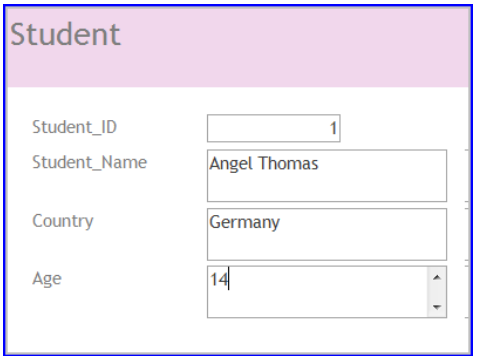


Figure 4.4: Student_Details Form

4.4.2 Creating Forms Using Blank Form

The **Blank Form** tool will create a new blank form and allow the users to format and modify the form as per their requirements. This is the fastest means to create a form, mainly if the user has to add a less number of fields on the form.

The steps to create a form using the **Blank Form** option are as follows:

1. In the **Forms** group, click the **Blank Form** icon. A blank form is opened in the **Layout** view and the **Field List** pane is displayed.
2. In the **Field List** task pane, click **Show all tables**. This will display the list of available tables.

Figure 4.5 displays the **Field List** pane.

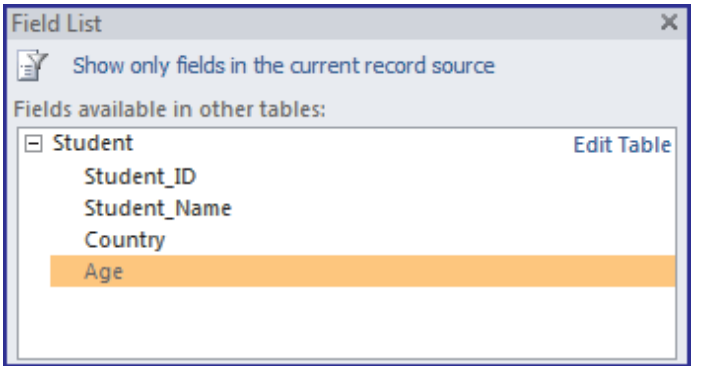


Figure 4.5: Field List Pane

Concepts

3. In the **Field List** task pane, expand the table that contains the fields that you want to see on the form.
4. To insert the field on the form, drag or double-click the field.
5. To add a number of fields at one time, hold the **Ctrl** key, select the fields that you want to add and then, drag them on the form.

   Figure 4.6 displays the form created using the **Blank Form** tool.

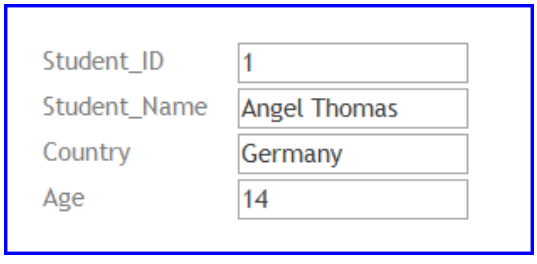


Figure 4.6: Form Created Using Blank Form

4.4.3 Creating Navigation Forms

In many Web applications, the main menu items are displayed at the top of the Web page with sub menu items. Access 2010 has a feature called **Navigation Forms**, which allows users to build standard user interfaces that can be used in Web applications.

The steps to create a form using **Navigation** option are as follows:

1. In the **Create** tab, click the **Navigation** icon. Six different layouts will be displayed as shown in figure 4.7.

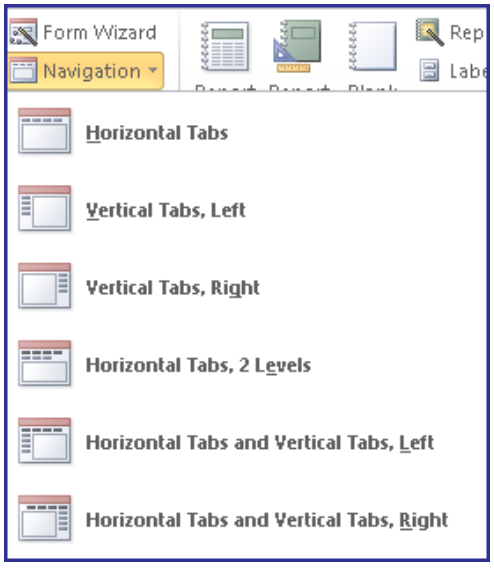


Figure 4.7: Different Layouts for Navigation

2. To select the layout that you want to display on the form, click **Horizontal Tabs and Vertical Tabs**, **Left** option.

3. To create the top-level tabs, click **Add New** of the navigation form and rename the text to `Student`.

   Figure 4.8 displays the outcome of using the **Navigation Form** tool.

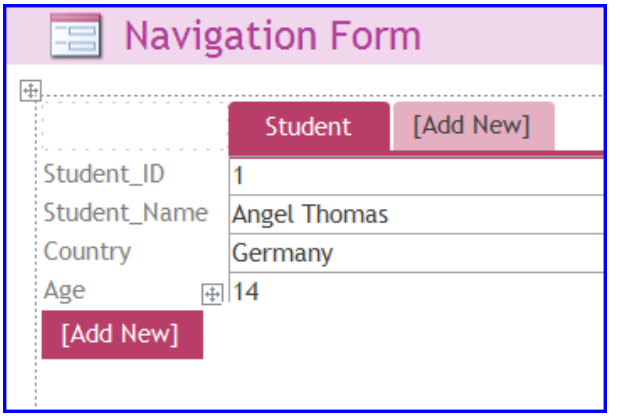


Figure 4.8: Creating a Navigation Form

Users can also apply styles to the navigation form through the layout view. Select the **Format** tab, click the **Quick Styles** drop-down button, and then, select the styles to apply them to the tabs on the form. **Quick Styles** change the visual appearance of the forms.

Figure 4.9 displays the available **Quick Styles**.

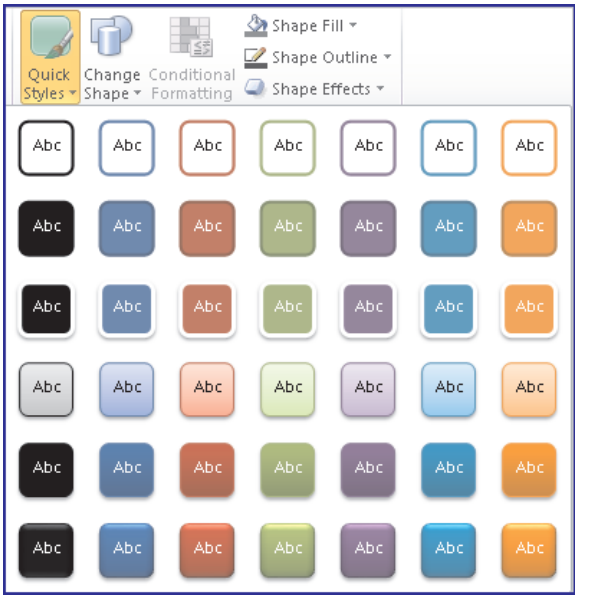


Figure 4.9: Quick Styles

Concepts

4.4.4 Creating Forms Using Form Design Tool

Users can also create forms in Access 2010 using the **Form Design** tool. Users can add buttons, text boxes, labels, and images in the form from the **Controls** tab. The difference between using **Blank Form** and **Form Design** tool is that in the latter, you can also use advanced features such as adding ActiveX controls.

To create a form, click the **Create** tab, and then, click the **Form Design** icon. A blank form is opened in the design view and the **Property Sheet** pane is displayed. **Controls** tab will be displayed on the **Ribbon**. Users can select the controls and drag the controls on the form. Figure 4.10 displays the **Design view** of the form created using the **Form Design** tool.

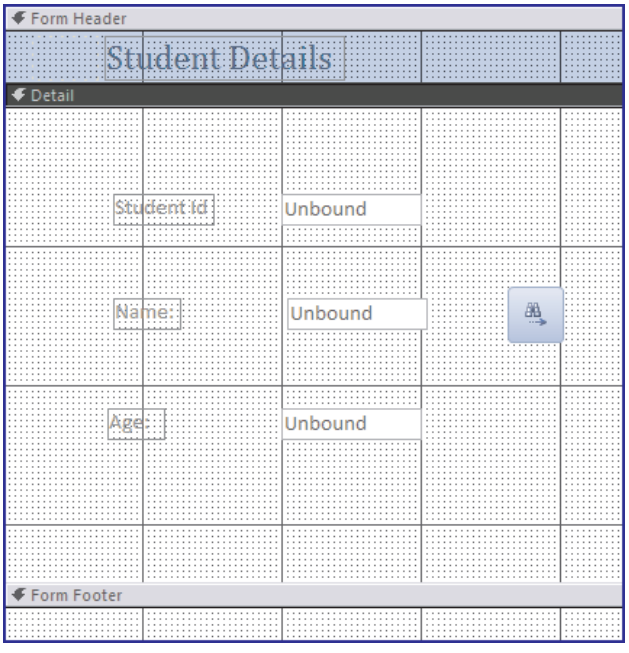


Figure 4.10: Design View of the Form

4.4.5 Creating a Pop-Up Form

The following are the steps to create a pop-up form:

1. Open a form in the **Design View**.
2. On the **Design** tab, in the **Tools** group, click **Property Sheet**. The **Property Sheet** pane is displayed.
3. Click the **All** tab.
4. Click the **Pop Up** property.
5. Select **Yes**.

The user can then add controls by clicking the appropriate control from the **Controls** group of the **Design** tab, and dragging it on the **Design View** of the pop-up form.

Concepts

4.5 Using Layout View for Creating SubForms

Subforms are the forms that are added in other forms, also called as main forms. The main form is a primary form. A form/subform is also called as a parent/child or a master/detail form.

A subform is useful only when the users want to display data from queries or tables having a one-to-many relationship. In a one-to-many relationship, there is an association between the two tables. For example, consider a `Customers` table and an `Orders` table. A single customer can place many orders. This is a one-to-many relationship. Here, for this scenario, users can create a form that displays the customer data, and also contains a subform that displays each and every customer order.

Users can add subforms to an existing form by using the **SubForm Wizard**. They can either choose to create a new form or use the existing form to create subforms.

The steps to create a subform using **SubForm Wizard** are as follows:

1. Open an existing form in Access 2010.
2. Right-click the existing form and then, click **Design View**.
3. Select the **Design** tab, in the **Controls** group click the drop-down arrow. This will display the **Controls** gallery.
4. Click the **Subform/Subreport** button. The cursor will change into a plus sign.
5. Click the form to place the subform. The **SubForm Wizard** will be displayed.

   Figure 4.11 displays the **SubForm Wizard**.

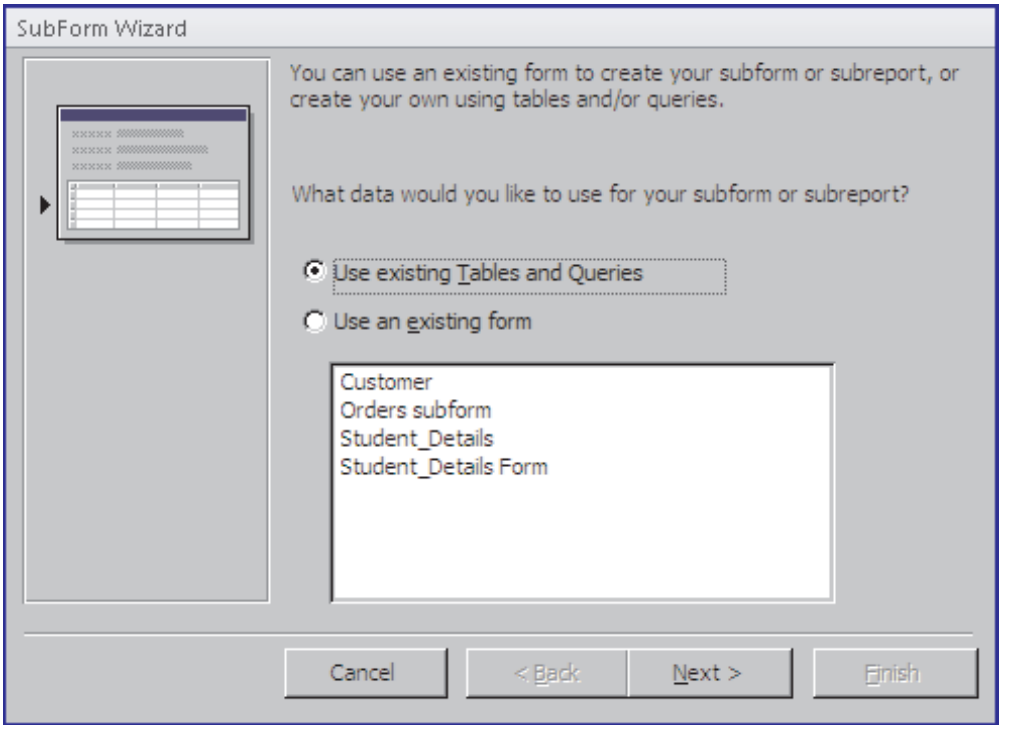


Figure 4.11: SubForm Wizard

Concepts

You can create the subform by using either the existing tables, queries, or an existing form.

6. Select **Use existing Tables and Queries** and then, click **Next**.
7. Select the table for the subform and also select the fields that you want to add in the form and click **Next**.
8. Type the name of the subform that you want to appear on the form.
9. Click **Finish**.

   After clicking **Finish**, Access 2010 adds a subform control to the form.

4.5.1 Using Layout View to Change a Form

Layout View provides enhanced design layouts having groups of controls. These controls allow the user to rearrange columns, fields, rows, and whole layouts. Users can also add or remove formatting in the **Layout View**. In this view, users can actually view the data while making changes in the form. This feature allows user to size the controls more accurately.

The steps for using the **Layout View** to change a form are as follows:

1. Open the form.
2. In the navigation pane, double-click the form.
3. Right-click the document tab.
4. Click **Layout View**.

   By using the **Layout View**, users can perform the following:

➔ **Resize controls in a column**: Users can select a single label and then, drag it to the required size of the form.

➔ **Change the text alignment, font color, and font type**: Users can select the label control and click the Format tab. The Format tab contains the option to change the alignment of text, font type, and font color.

➔ **Format multiple labels at one time**: Users can format multiple labels at the same time by pressing the **Ctrl** key and then, applying the formatting as per their needs.

The following are the types of layout:

➔ **Control Layout**

Control Layout is an important feature provided by Access 2010 for arranging and organizing various controls such as labels and text boxes in a form. **Control Layout** provides a consistent appearance of each record displayed in the form.

Concepts

The different types of **Control layouts** are as follows:

➔ **Tabular Layout**

Tabular Layout arranges the controls in the form of rows and columns. In **Tabular** control layout, each field name and its corresponding value are displayed vertically.

Figure 4.12 displays a form in **Tabular Layout**.

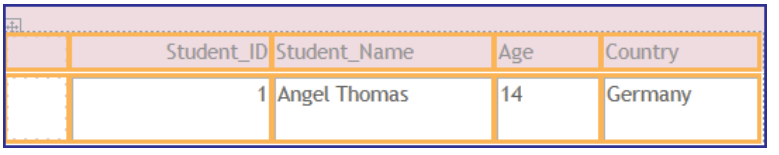


Figure 4.12: Tabular Layout

➔ **Stacked Layout**

In **Stacked Layout**, each field name and its corresponding value are displayed horizontally. Each field name is placed on the left side of the corresponding value.

Figure 4.13 displays a form in **Stacked Layout**.

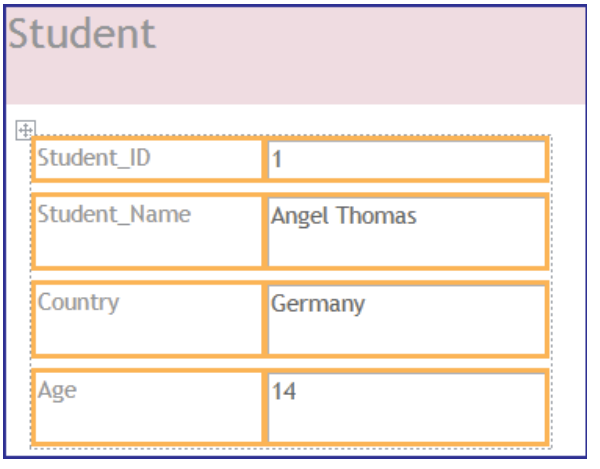


Figure 4.13: Stacked Layout

A **Control Layout** needs to be created for arranging the controls within a form. It is created either in **Layout View** or **Design View** of the form.

To create a new **Control Layout**, perform the following steps:

1. Open a form in the **Layout View**.
2. Select appropriate controls in the form.
3. On the **Arrange** tab, in the **Control Layout** group, click an appropriate control layout.

Concepts

The arrangement of controls can be switched from **Tabular Layout** to **Stacked Layout** or vice-versa by selecting the appropriate controls and clicking the appropriate control layout command from the **Control Layout** group of the **Arrange** tab.

The **Control Layout** can be removed according to the users' requirements by clicking the **Remove** command from the **Control Layout** group of the **Arrange** tab.

4.6 Using Form Design Tab View to Apply Header Footer and Themes

Users may sometimes feel the need to include information such as page number, date/time, and other useful information on every form. Access 2010 allows including such information either at the top or at the bottom of the form by using the **Page Header** and **Page Footer** options respectively. Page header and footer are especially helpful while printing the form.

To add a page header and footer to a form, perform the following steps:

1. Open the form in **Design View**.
2. On the **Arrange** tab, in the **Show/Hide** group, click **Page Header/Footer**. The **Page Header** and **Page Footer** sections are added to the form.
3. On the **Design** tab, in the **Controls** group, click the appropriate control and drag and drop it to the **Page Header** section.

Similarly, controls can be added to the **Page Footer** section of the form.

Access 2010 also allows the user to remove page header and footer section from the form if it is not required. To remove the **Page Header** and **Page Footer** sections, right-click the **Detail** section of the form, and click **Page Header/Footer**.

Users can also apply themes to change the appearance of the forms. To add a theme to a form, perform the following steps:

1. Open the form in **Design View**.
2. On the **Design** tab, in the **Themes** group, click the **Themes** drop-down button.

Figure 4.14 displays the different types of themes.

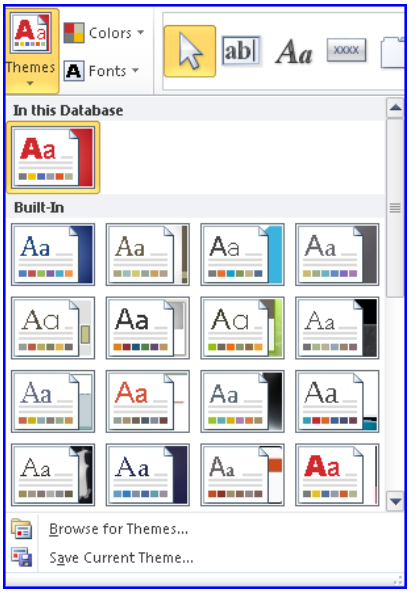


Figure 4.14: Different Types of Themes

3. Select the theme that you want to apply to the form.

4.7 Other Features of Forms

In Access 2010, some of the **Ribbon** tabs are visible only when the users want to use them. Similarly, users can see the **Arrange** and the **Format** tab only when they have created the form or are working on the form.

➔ **Arrange Tab**

The **Arrange** tab allows the arrangement of tables, rows and columns, merge/split, move, and position in the forms. Figure 4.15 displays the **Arrange** tab.

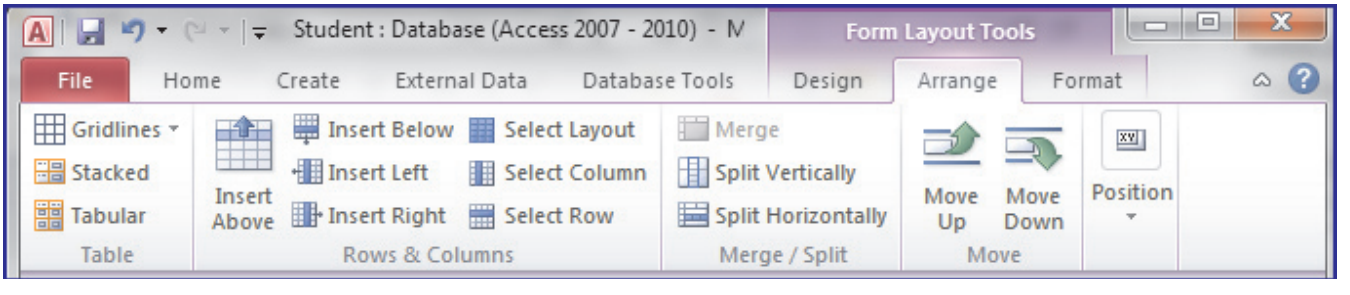


Figure 4.15: Arrange Tab

➔ **Format Tab**

The **Format** tab allows the formatting of the controls in the form.

Concepts

The **Format** tab contains the following tabs:

- **Selection**: Selects the labels or controls by name.
- **Font**: Applies the font effects such as italics, bold, text alignment, font color, and fill colors.
- **Number**: Applies the formatting to the numeric data fields that includes percentage values, currencies, and decimal numbers.
- **Background**: Adds the line styles, gridlines, and background images to the form layout.
- **Control Formatting**: Allows a user to apply conditional formatting, or shape outline, shape fill, and change the shape.

Figure 4.16 displays the **Format** tab.

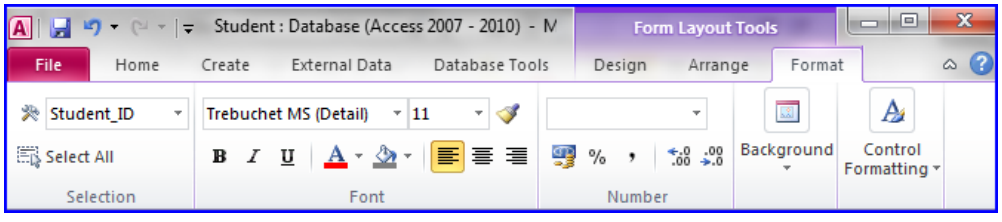


Figure 4.16: Format Tab

4.8 Overview of Controls

Controls are the objects on a form that are used to decorate, display data, and perform actions.

Users can add various objects called as Windows controls on the form. These are the controls that database users use to interact with the product. These controls are available on the **Ribbon** when the form is in **Design View**. Users can click the **Design** tab to access the controls.

There are several techniques used to add the controls to the form. Some of these are as follows:

➔ Click the control and drag it on the form. Similarly, users can add different controls as per his requirements.

➔ Suppose the user wants to add the same control several times, the user can double-click it and then, click the form as many times. To terminate the control, the user can click the select button.

4.8.1 Use of Form Controls

Controls are used to add, display, and edit data on the form. For example, the text box control is used to display or enter data on the form. Other controls include check box, command button, combo box, and so on. Controls allow users to work with and view data in a database application. The commonly used controls are text box, command button, check boxes, and labels.

4.8.2 Types of Controls

There are two types of controls that are used in the forms. They are as follows:

➔ **Bound Controls**

The controls that are linked to the data fields in a query or a table are called as Bound controls. These types of controls can either display data from the data fields or place the data into the data field retrieved from the tables or queries.

➔ **Unbound Controls**

The controls that are not linked to any source in a query or a table are called as Unbound controls. These types of controls are used to display data or instructions.

Different types of controls are used to display different types of data. For example, a text box is used to display numbers and data. An object box is used to display images.

Labels are used to display descriptive text while the squares and line controls are used to separate or organize data.

4.9 Adding, Moving, and Removing Controls

Access 2010 includes the formatting tools that allow you to modify the forms as per the user requirements. Users can add logos, modify the form layouts, add command buttons, and add images.

Suppose the user wants to rearrange and resize the control, then, they can make use of the **Arrange** tab.

To resize the form controls, perform the following steps:

1. Switch to the **Layout View** of the form.
2. Select the control that you want to resize, and hover the mouse over the edge of the control. The cursor will become a double-sided arrow.
3. Click and drag the edge of the control to resize it, and release the mouse when the control appears in the desired size.

Users can move the controls by dragging them it on the form.

4.10 Formatting and Arranging Controls

Access 2010 applies a default font for text and other controls to display them in the forms. Users can change the font of the form to make it look more attractive. They can also set the font for individual controls. Users can change the font type, font size, and text alignment of the controls in the form.

The steps to change the font of the control on a form are as follows:

1. Select the control to apply the font.

2. Click **Format** on the **Ribbon**.

3. Click the **Font** combo box and select the font that you want to apply.

When the users apply a font to any control in the form, it does not affect the base tables. They remain unaffected.

The steps to change the font size of the control on a form are as follows:

1. Select the control to apply the font size.

2. Click **Format** tab on the **Ribbon**.

3. Click the **Font size** combo box and select the desired font size that you want to apply.

There are many ways to change the text alignment. Access 2010 allows the users to change the text alignment by using the various buttons in the **Format** tab. This tab contains the **Font** section that displays the Align Text Left, Center, and Align Text Right buttons. Users can apply the alignment as per their requirements.

4.10.1 Creating Conditional Formatting on Controls

Access 2010 allows users to use conditional formatting to focus attention on the specific areas of the report. For example, in a school management system, one can highlight the scores of the students who are performing well in the subjects.

The following are the steps used to perform **Conditional Formatting**:

1. In the **Format** tab, click **Conditional Formatting**. The **Conditional Formatting Rules Manager** dialog box will be displayed. Figure 4.17 displays the **Conditional Formatting Rules Manager** dialog box.

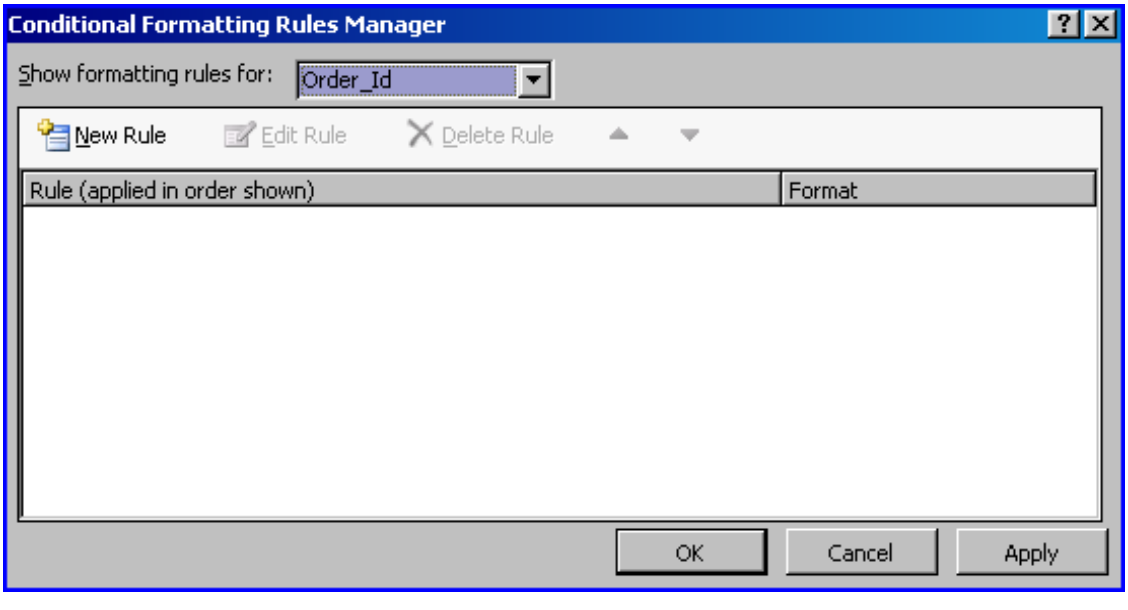


Figure 4.17: Conditional Formatting Rules Manager

2. Click **New Rule**. The **New Formatting Rule** dialog box is displayed. Figure 4.18 displays the **New Formatting Rule** dialog box.

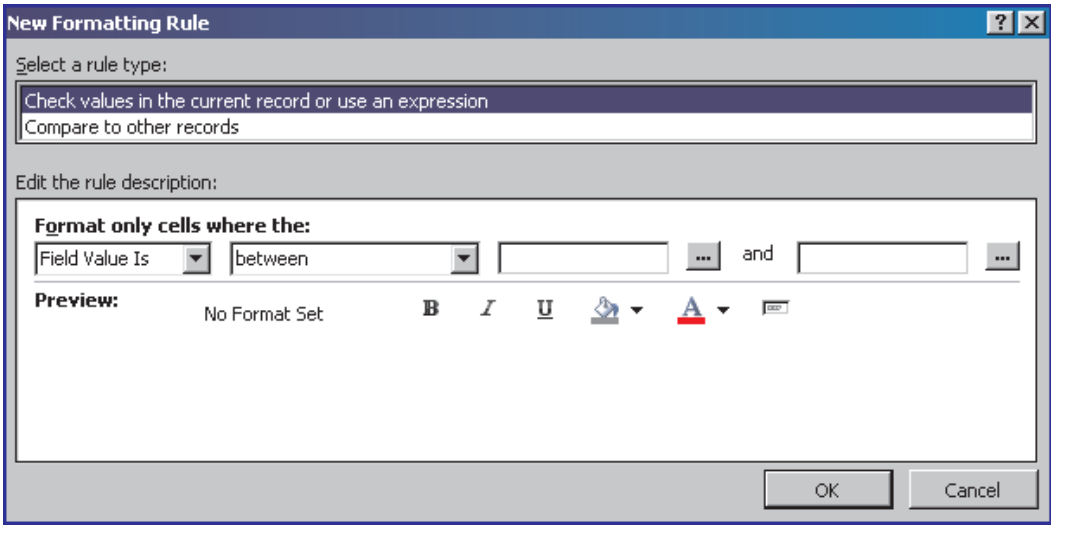


Figure 4.18: New Formatting Rule

3. Select the kind of rule that you want to create.
4. In the **Select a rule type** list section, select the **Check values in the current record or use an expression**.
5. In the rule options, select the **Field Value Is**, **equal to**, and **0**.
6. In the formatting options, select the foreground color as grey.
7. Click **OK** to add the new rule. The dialog box is displays with the new rule.

4.11 Sorting and Filtering Data within a Form

Forms display data in an unsorted order. A user can sort the data in forms in the same way as they can sort data in the datasheet. Sorting a form is easy for users as they can use the **Home → Sort & Filter** section for sorting the records.

Filtering minimizes the total number of records so that user can only view those records that they want to see, based on a specific condition.

Steps to apply filtering of records in a form are as follows:

1. Select **Home → Sort & Filter → Advanced → Filter By Form**.

   Access 2010 will change the form view to Search mode, and the form will look exactly the same except all other fields will be blank.
2. Move to the field where you want to apply the filter. A drop-down arrow will appear in the field.
3. Click the drop-down arrow, and choose the value to be included in the results.

Concepts

4. If you want to apply filters to more than one field, then, click the **Or** tab and fill out more filter settings.

5. Right-click the blank space on the form, and then, select **Apply Filter/Sort**.

   Access 2010 will switch back to the normal form and apply the filter settings. The outcome will show filtered records.

4.12 Check Your Progress

1. _____ make the database look attractive with different themes, colors, fonts, and charts.

(A)	Table	(C)	Forms
(B)	Column	(D)	Query

2. In the _____ tool, all fields from the underlying table are displayed on the form.

(A)	Sort	(C)	Navigation
(B)	Simple Form	(D)	Blank Form

3. Which of the following layouts provides for arranging and organizing various controls such as labels and text boxes in a form?

(A)	Format	(C)	Control
(B)	Design	(D)	Navigation

4. Brandol is working on an online banking Web site form. This form allows user to sum up hundreds of records in a short tabular format that allow users to modify the layout of the table. Which of the following features allows him to accomplish this task?

(A)	Pivot Chart	(C)	Modal Dialog
(B)	Pivot Table	(D)	Split Form

5. Which of the following features allow users to build standard user interfaces for Web applications?

(A)	Pivot Table	(C)	Quick Styles
(B)	Navigation	(D)	SubForm

Concepts

4.12.1 Answers

1.	C
2.	B
3.	C
4.	B
5.	B

Summary

- A form provides several advantages over a database table, in terms of user interactivity and appearance.
- Access 2010 provides ways to create forms easily.
- Quick Styles can be used to change the visual appearance of the forms.
- Subforms are the forms that are added in other forms also called as main forms.
- Access 2010 allows including information such as page number, logo, and so on either at the top or at the bottom of the form by using the Page Header and Page Footer options respectively.
- Controls are the objects on a form that are used to decorate, display data, and perform actions.
- Sorting a form is easy for users in Access 2010 as they can use the Sort & Filter section for sorting the records.

Try it Yourself

1. Zen Associates is an advertising agency which is headquartered in Hong Kong. They want to upgrade the latest technologies that are coming up in the market. They want to create a database using Access 2010 for their advertising agency. The Web site will display the list of products for sale such as books, computers, vehicles, cameras, laptops, musical instruments, and so on.

   Assume that you have to create a database named ZenProductsDB having tables with structures as shown in figure 4.19:

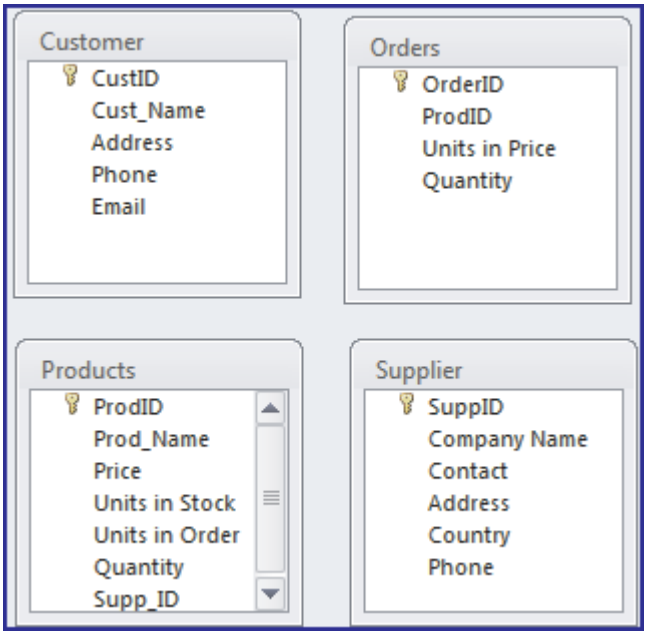
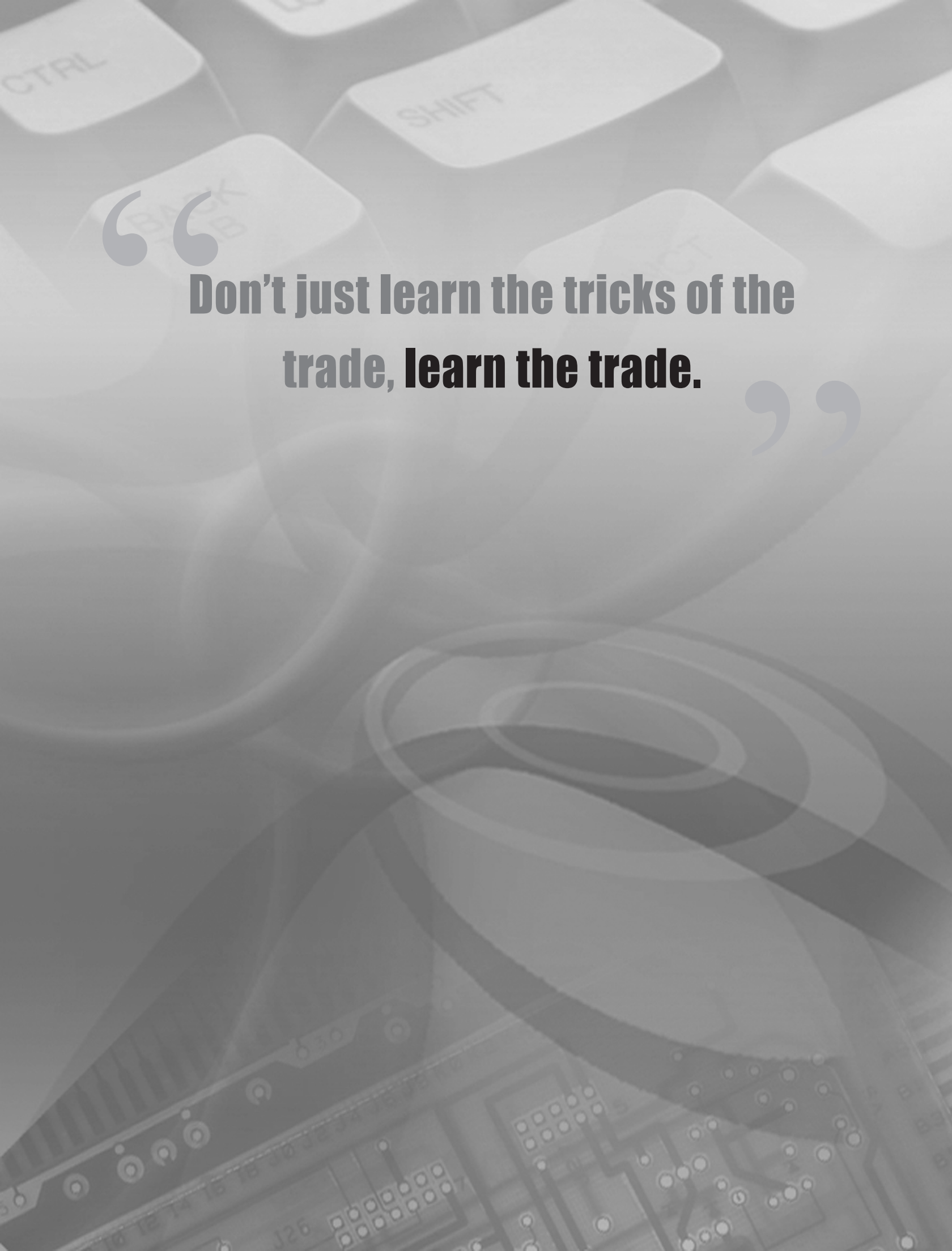


Figure 4.19: ZenProductsDB

Perform the following tasks:

1. Create a form to display the list of products that are sold by the Zen Associates.
2. Sort the form by alphabetical order of product names.
3. Create a form that will show only a list of supplier in a particular country in 2012.

Concepts


“Don’t just learn the tricks of the trade, learn the trade.”

Session - 4 (Workshop)

Working with Forms

In this workshop, you will learn to:

- ➔ Create forms and form controls in Access 2010
- ➔ Add functionality to forms and form controls in Access 2010
- ➔ Create subforms in Access 2010
- ➔ Customize a form with data-bound combo boxes
- ➔ Customize a form with lookup fields and validation rules

4.1 Working with Forms

You will view and practice how to create forms and form controls in Access 2010. You will also view and practice how to add functionality to forms and form controls in Access 2010. You will also view and practice how to create subforms in Access 2010. You will also view and practice how to customize a form with data bound combo boxes. You will also view and practice how to customize a form with lookup fields and validation rules.

- Creating Forms and Form Controls
- Adding Functionality to Forms and Form Controls in Access 2010
- Creating Subforms in Access 2010
- Customizing a Form with Data Bound Combo Boxes
- Customizing a Form with Lookup Fields and Validation Rules

Note - Please refer to the respective lab deliverable for demonstrations and guided simulations.

Labs

Session - 5

Working with Reports

Welcome to the Session, **Working with Reports**.

This session describes an overview of reports in Access 2010. The session explores the creation of reports, advanced options of reports such as using expressions and functions, and other features of reports. It also describes how to work with and format controls on the reports. Finally, the session describes sorting and filtering of data in a report.

In this Session, you will learn to:

- Explain the overview of reports
- Explain the creation of reports
- Describe the advanced options of reports
- Explain the use of Label Wizards
- Explain working with report design, arrange, and format tabs
- Describe the sorting and filtering of records in a report

5.1. Introduction

Reports are the summarized final output of a database. A report is a combination of the raw facts in the database with sufficient information that gives meaning to these facts and presents visual results. For example, users can use reports instead of showing tabular data. A report is the best way to collect, organize, and display data that can also be formatted appropriately. A report also summarizes data in an efficient manner.

5.2. Overview of Reports

Reports allow users to create a standard formatted report that automatically stores the information from the tables and queries. In many ways, forms and reports are similar as they are used to display the table records. The major difference between forms and reports is that a report can also summarize the data. For example, users want to create a report for customers that display the list of the orders made by each and every customer. At the same time, users also want the total amount of all orders placed by each individual customer. To generate this kind of output, one would need to use a report as a form cannot display the summarized data.

The **Report** tool is the simplest way to create a report. The **Report** tool creates a report that is connected or bound to a single data source, query, or a table.

The following are the steps that are used to create a report:

1. Select the table that to be used as a data source for the report in the **Navigation** Pane. Assume that the `Student` table created in the earlier session is selected.
2. Click the **Create** tab.
3. Click **Report** in the **Reports** group.

Figure 5.1 displays the simple report that is generated.

Student — 24 September 2012 15:33:33

Student_ID	Student_Name	Country	Age
1	Angel Thomas	Germany	14
2	Peter Thompson	Norway	13
3	Catherine Gomes	Canada	15
4	Edwin Theordo	Iran	15
5	Filomeena Baretto	United States	14
6	Carol Cruz	United Kingdom	16
7	Delmon Stallone	Mexico	17
8	Savio Patrick	Egypt	15

Page 1 of 1

Figure 5.1: Simple Report

Concepts

Access 2010 creates a report that contains all data fields in the data source. The report is automatically created with the tabular layout. The report is opened in a **Layout View** that gives users a chance to change it. For example, users can resize the fields by clicking and dragging an individual control.

5.3. Creating Simple Report Using Blank Report Tool

Users can make use of the Blank Report tool when only a few records are to be selected to display on the report. The **Blank Report** tool allows users to create a report from scratch. This is an easy way to create reports mainly when there are only a few data fields for displaying records. Following are the steps to create a report using the **Blank Report** tool:

1. Select the **Create** tab in the **Reports** group and then, click **Blank Report**. The **Layout View** displays the blank report and the **Field List** pane is displayed.
2. Click **Show all tables** in the **Field List** pane.
3. Click the plus sign (+) next to the tables containing the fields that you want to view on the report.
4. Drag the fields on the report at one time, or press the **Ctrl** key, select several fields, and then, drag them on the report.

   Figure 5.2 displays the report created using the **Blank Report** tool.

Student_ID	Student_Name	Country	Age
1	Angel Thomas	Germany	14
2	Peter Thompson	Norway	13
3	Catherine Gomes	Canada	15
4	Edwin Theordo	Iran	15
5	Filomeena Baretto	United States	14
6	Carol Cruz	United Kingdom	16
7	Delmon Stallone	Mexico	17
8	Savio Patrick	Egypt	15
9	Delon Donald	Nigeria	16
10	Tom Cruise	Colombia	16
11	George Dragon	Germany	15

Figure 5.2: Report Created Using Blank Report Tool

Users can also add logos, page numbers, title, or the date and time on the report.

5.4. Creating Simple Report Using Report Wizard

Users can use the **Report Wizard** to generate a report by selecting the data fields that they want to display on the report. Users can specify the grouping and sorting of data. The steps for creating a report using the **Report Wizard** are as follows:

1. In Access 2010, click the **Create** tab. The **Reports** group will be displayed.
2. In the **Reports** group, click the **Report Wizard**. The **Report Wizard** dialog box will be displayed.

   Figure 5.3 displays the **Report Wizard**.

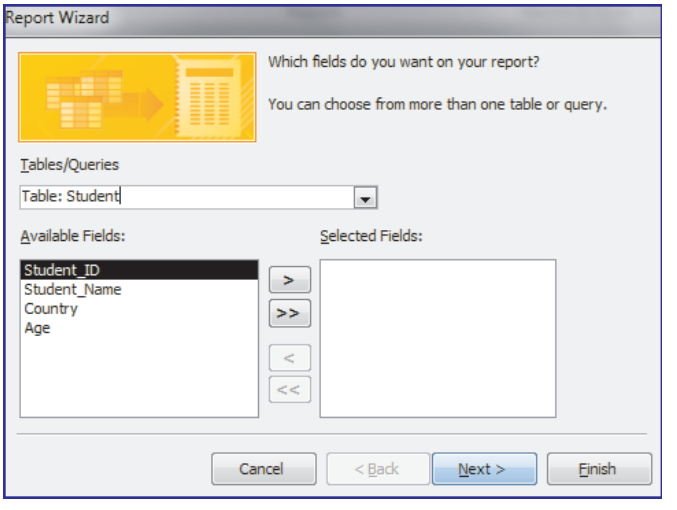


Figure 5.3: Report Wizard

   In the **Table/Queries** section, select the table shown on the left. The **Available Fields** will display the list of columns in the table.

3. To select the fields, click the field in the **Available Fields** and then, click **>**. Select the data fields to display on the report.
4. To select all fields, click **>>**. This will select all data fields from the table.

Concepts

Figure 5.4 displays the **Report Wizard** with all data fields selected.

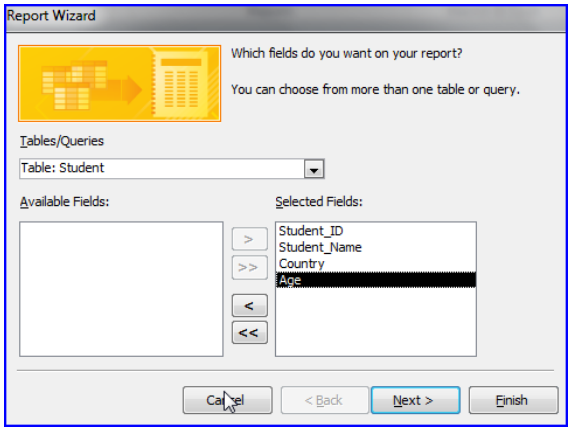


Figure 5.4: Report Wizard with Data Fields

5. Click **Next>**.

The **Report Wizard** will display the grouping levels to be applied to the reports. Figure 5.5 displays the grouping levels in the **Report Wizard**.

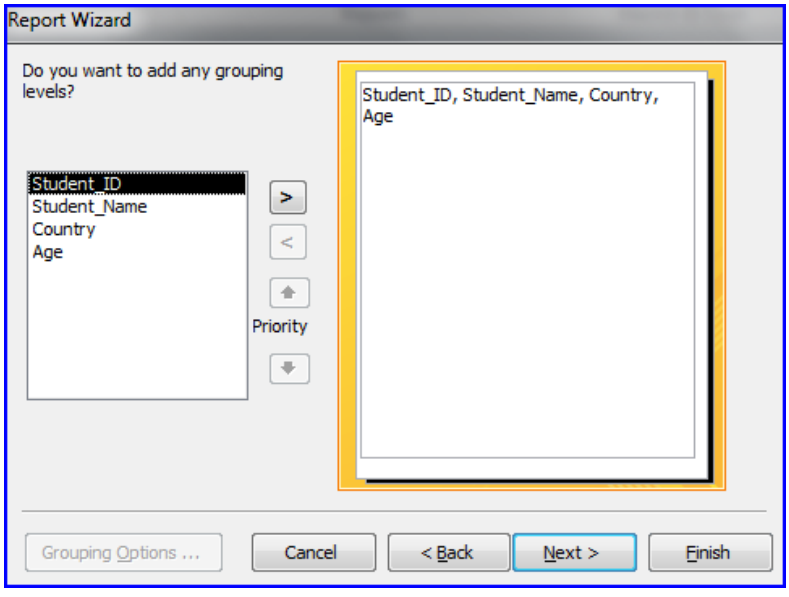


Figure 5.5: Grouping Levels

6. Click **Next**.

7. You can select the order, either ascending or descending, for the data fields for the report and then, click **Next**. This will allow you to select the **Layout** and **Orientation** of the report.

Concepts

Figure 5.6 displays the **Layout** and **Orientation** option in the **Report Wizard**.

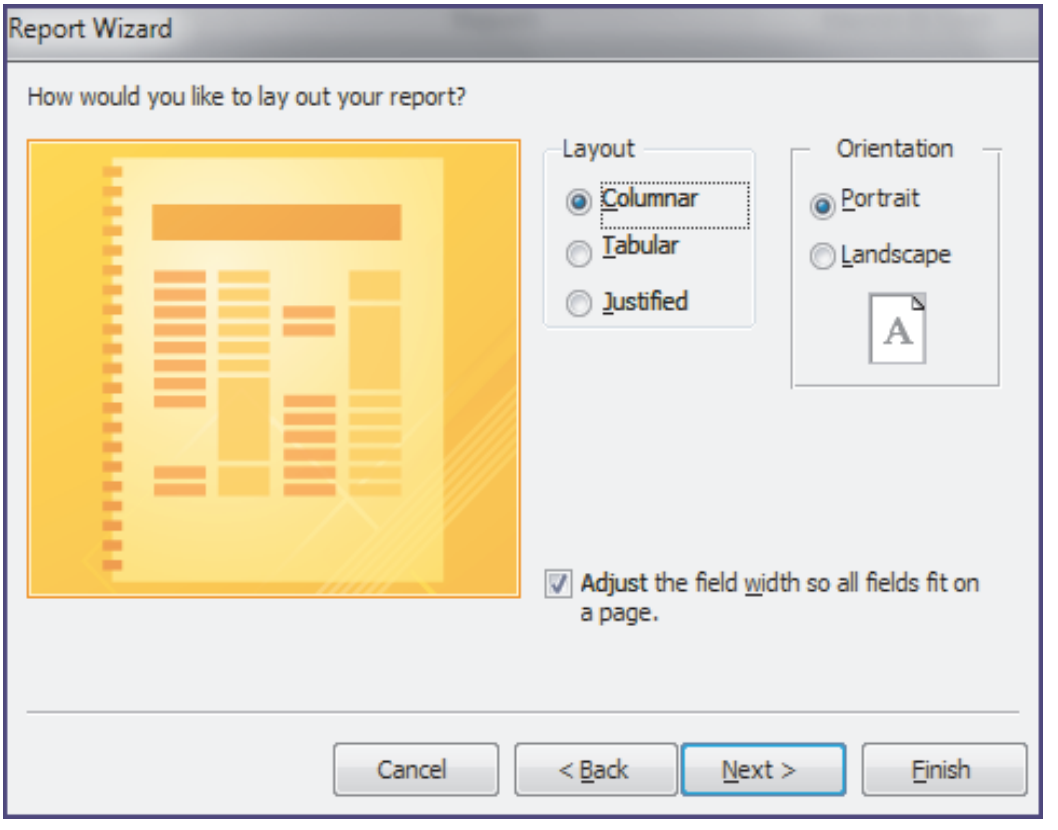


Figure 5.6: Layout and Orientation

8. Click **Next**.
9. Name the title of the Report and then, click **Next**.
10. Click **Finish**.

Concepts

Figure 5.7 displays the report created using the **Report Wizard**.

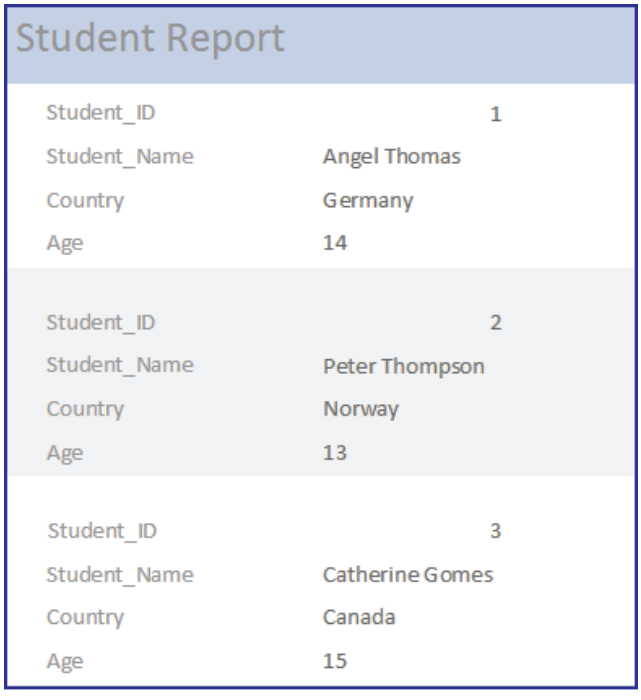


Figure 5.7: Student Report

The report will be displayed in the print format. Users can also zoom in to view the details in the report.

5.5. Advanced Options for Reports

Access 2010 has various advanced options to create and modify reports.

5.5.1 Report Properties

In Access 2010, the report properties are used to customize the report and to add various functions in the report.

To access the **Property Sheet**, perform the following steps:

1. Open the Access 2010 database.
2. From the **Navigation Pane**, select an appropriate report.
3. Open the report in the **Design View**.
4. On the **Design** tab, in the **Tools** group, click **Property Sheet**.

Figure 5.8 displays the **Property Sheet**.

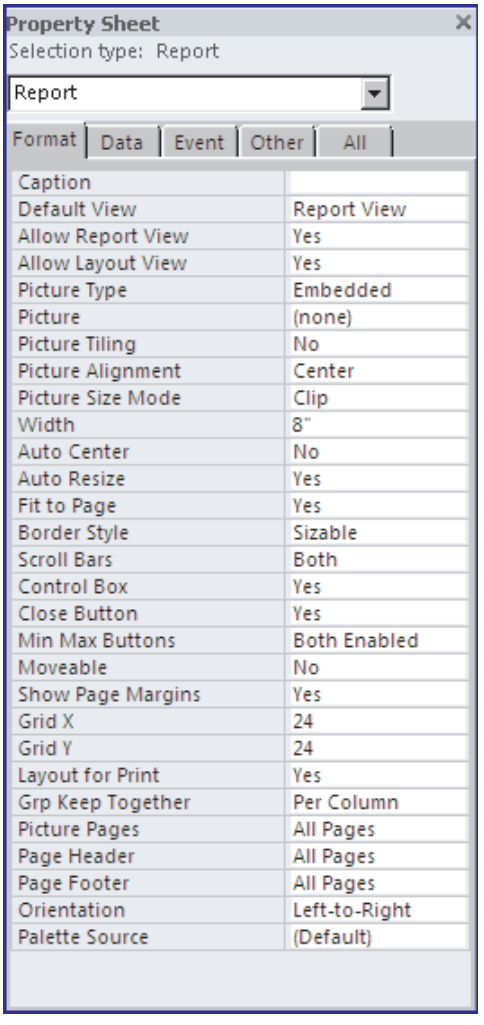


Figure 5.8: Property Sheet

In the **Property Sheet**, the properties of the report are divided into five tabs namely, **Format**, **Data**, **Event**, **Other**, and **All**. The **All** tab displays all the properties of the report.

The properties available in the **Property Sheet** enable the user to specify important characteristics such as a validation rule, sort order for the table data, and so on.

Some of the important properties are as follows:

- **Filter on Load**: Allows applying the filter property to the report, each time it is opened. To set this property, select **Yes** from the drop-down list.
- **Default View**: Allows determining the type of the view of the report. The **Report View** and the **Print Preview** are the only views that can be set as the default view.
- **Record Locks**: Allows securing the records from being modified by other users.

Concepts

- **Date Grouping**: Allows specifying the group for date-time values. The group can be selected from the drop-down list as **Use System Settings** or **US Defaults**.

- **Picture Size Mode**: Allows controlling the size of the image, which is to be applied in the report. The different modes available in this property are **Clip**, **Stretch**, **Zoom**, **Stretch Horizontal**, and **Stretch Vertical**.

- **OnActivate**: Allows executing a macro or a piece of code when a report is activated. This property is available under the **Event** tab.

5.5.2 Using Expressions

An expression is similar to a formula used to generate a result. It contains a number of elements that can be used alone or in combination. The elements used in expressions are as follows:

- **Identifiers**: Allows specifying the names of table fields, controls, reports, or properties of those fields or controls.

- **Operators**: Are notations that specify specific operations to be performed on some values. The types of operators are Arithmetic, Comparison, and Logical.

- **Functions**: Allows performing a certain task and return a value. Functions such as SUM or AVG are used in expressions to return a value.

- **Constants**: Indicates the values that do not change and are not calculated by an expression. The constant can be a string, numeric value, or a boolean value.

The most common use of expressions in Access 2010 is to perform operations such as calculating total, percentage, and average.

- **Examples of Expressions**

  An expression always begins with the = sign.

Concepts

Table 5.1 lists some of the expressions that are used in the reports.

Expression	Purpose
`=[RequiredDate]-[DeliveryDate]`	Allows calculating difference between the values of two text boxes namely, RequiredDate and DeliveryDate.
`=[Order]>0`	Allows setting validation that the value of the text box named Order should be greater than 0.
`=Trim([«stringexpr»])`	Allows removing any extra spaces between words in the text box.
`="Page" & [Page]`	Allows displaying page numbers on each page of the report.
`=IIf([«expr»],[«truepart»],[«falsepart»])`	Allows using the 'IIf' function in conditional expressions.
`= Avg («expr»)`	Allows displaying the average of the values of table fields or controls.
`=Count([«expr»])`	Allows displaying the number of records in the column.
`= Sum («expr»)`	Allows finding the total of the values in a column.

Table 5.1: Examples of Expressions

5.5.3 Domain Aggregate Functions

The Domain aggregate functions allow calculating sum or counting the values in the table selectively. A 'domain' consists of fields in tables or controls in reports. Table 5.2 lists some of the domain aggregate functions that are used in the reports.

Expression	Purpose	Example
`= DSum («expr», «domain», «criteria»)`	Returns the sum total of the values in the field of the table.	`=DSum("Salary","Employees","City='New York'")` The example returns the sum of the salary of employees residing in New York city.
`= DCount («expr», «domain», «criteria»)`	Returns the number of values in the field of the tables.	`=DCount("OrderID","Orders","Year(OrderDate)=1996")` The example counts the orders placed in the year 1996.

Table 5.2: Domain Aggregate Functions

Concepts

5.5.4 Date Functions

Date functions allow users to filter the date fields inserted in the report.

Table 5.3 lists some of the commonly used date functions.

Expression	Purpose
`=Date()`	Displays the current date of the report.
`= DateAdd («interval», «number», «date»)`	Displays the date after adding the specified number of intervals to the specified date.
`= Month («number»)`	Displays the month from the specified date.
`= Year («number»)`	Displays the year of the specified date.

Table 5.3: Date Functions

5.5.5 Customizing Reports

Reports are made user-interactive, by adding controls from the **Controls** group in Access 2010. Some of the control-related operations that are used to customize reports are adding a hyperlink, adding a background picture, applying conditional formatting to the fields, or inserting buttons and text boxes.

➔ Adding a Hyperlink

A Hyperlink can be added to a report to enable users to navigate to a specific location or file on the computer. It can also be used to get connected to the Internet and visit a specified Web site. When a hyperlink is added to a report, it is placed in the upper-left corner of the Detail section by default. A user can place a hyperlink at any position and any section in the report.

To add a hyperlink, perform the following steps:

1. Open the report in the **Design View**.
2. On the **Design** tab, in the **Controls** group, click **Insert Hyperlink** icon.

The **Insert Hyperlink** dialog box is displayed as shown in figure 5.9.

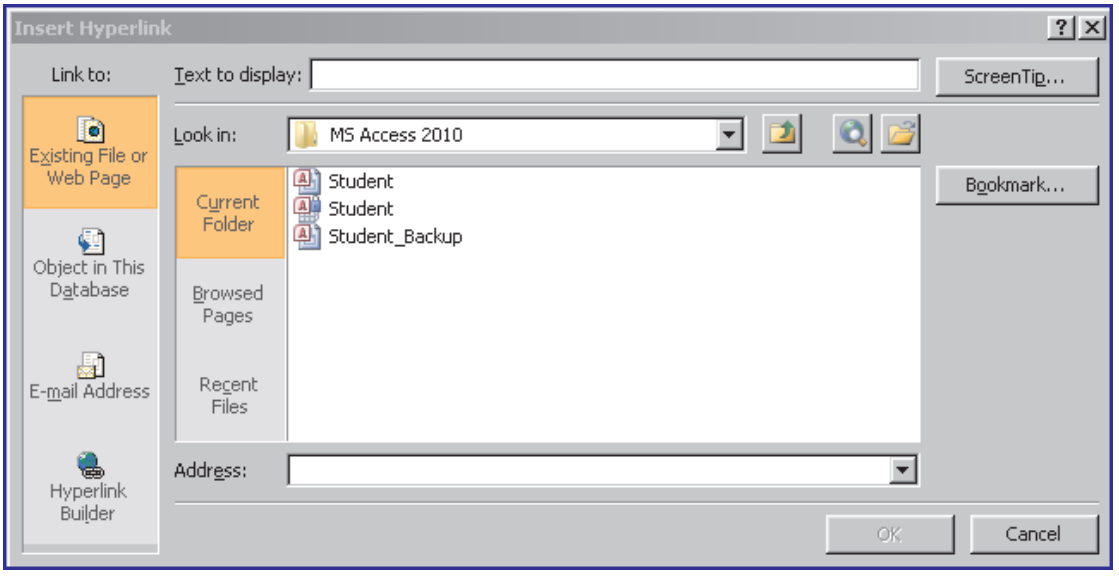


Figure 5.9: Insert Hyperlink Dialog Box

3. Select the text that has to be displayed from **Current Folder** or provide the link of any Web address from **Browsed Pages**.
4. Click **OK** to insert the hyperlink in the report.

➔ Adding a Background Picture

Pictures enhance the appearance of a report. Access 2010 allows adding a picture to a report as a background to make it more attractive and descriptive.

It is possible to set the picture mode to different sizes such as **Clip**, **Stretch**, **Zoom**, **Stretch Horizontal**, and **Stretch Vertical**. The **Clip** option keeps the original size of the image, whereas the **Stretch** option stretches the image in both horizontal and vertical directions. The **Zoom** option magnifies the image. The **Stretch Horizontal** and **Stretch Vertical** option stretches the image horizontally and vertically respectively.

When a background picture is added to a report, it is applied to the entire page. The user can set the pages on which to print the background picture by using the **All Pages**, **First Page**, or **No Pages** options.

To add a background picture, perform the following steps:

1. Open the report in the **Design View**.
2. On the **Design** tab, in the **Tools** group, click **Property Sheet**. The **Property Sheet** is displayed.
3. Select **Report** from the **Selection type** drop-down list.
4. On the **Format** tab, in the **Picture** property, enter the path and file name of the picture.

Concepts

5. Set the **Picture Size Mode** property to the appropriate size mode.
6. Set the **Picture Pages** property to select the appropriate option to decide on which pages to print the background picture.

Note - Access 2010 includes support for all common picture types such as `.bmp`, `.jpg`, `.gif`, and `.png`.

5.5.6 Creating Multi-table Reports

Users can create multi-table reports in Access 2010 using the **Blank Report** tool. While creating multi-table reports, users have to check for the type of relationship that exists between the two tables.

To create a multi-table report, perform the following steps:

1. Open the Access 2010 database.
2. Select the **Create** tab, click the **Blank Report** icon. The **Blank Report** will be displayed.
3. Select the first table in the **Fields List** pane.
4. Drag the necessary data fields from the first table on to the report.
5. Select the second table in the **Fields List** pane.
6. Drag the necessary data fields from the second table on to the report. The **Specify Relationship** dialog box appears. Figure 5.10 displays the **Specify Relationship** dialog box.

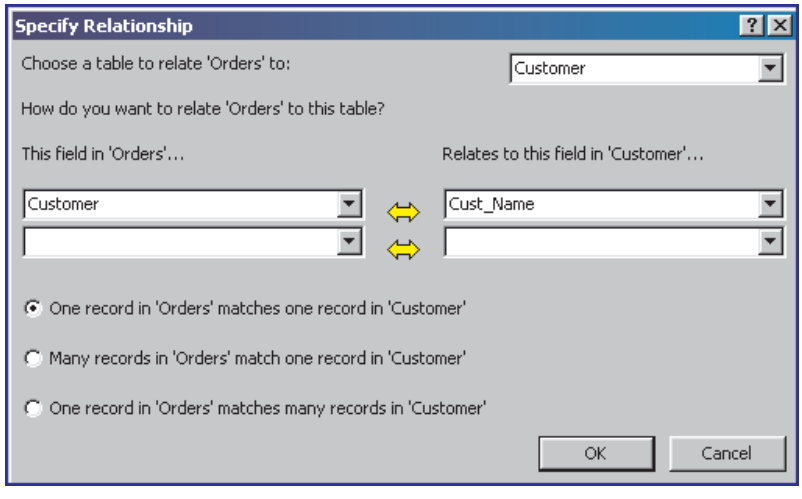


Figure 5.10: Specify Relationship

5.6. Other Features of Reports

There are various other features that are used with reports.

Concepts

5.6.1 Creating Labels Using Label Wizard

Users can display data from the fields of a table in the form of labels by using the **Label Wizard**.

To create a label report using the **Label Wizard**, perform the following steps:

1. Select the **Create** tab, in the **Reports** group, click **Labels**. The **Label Wizard** will be displayed. Figure 5.11 displays the **Label Wizard**.

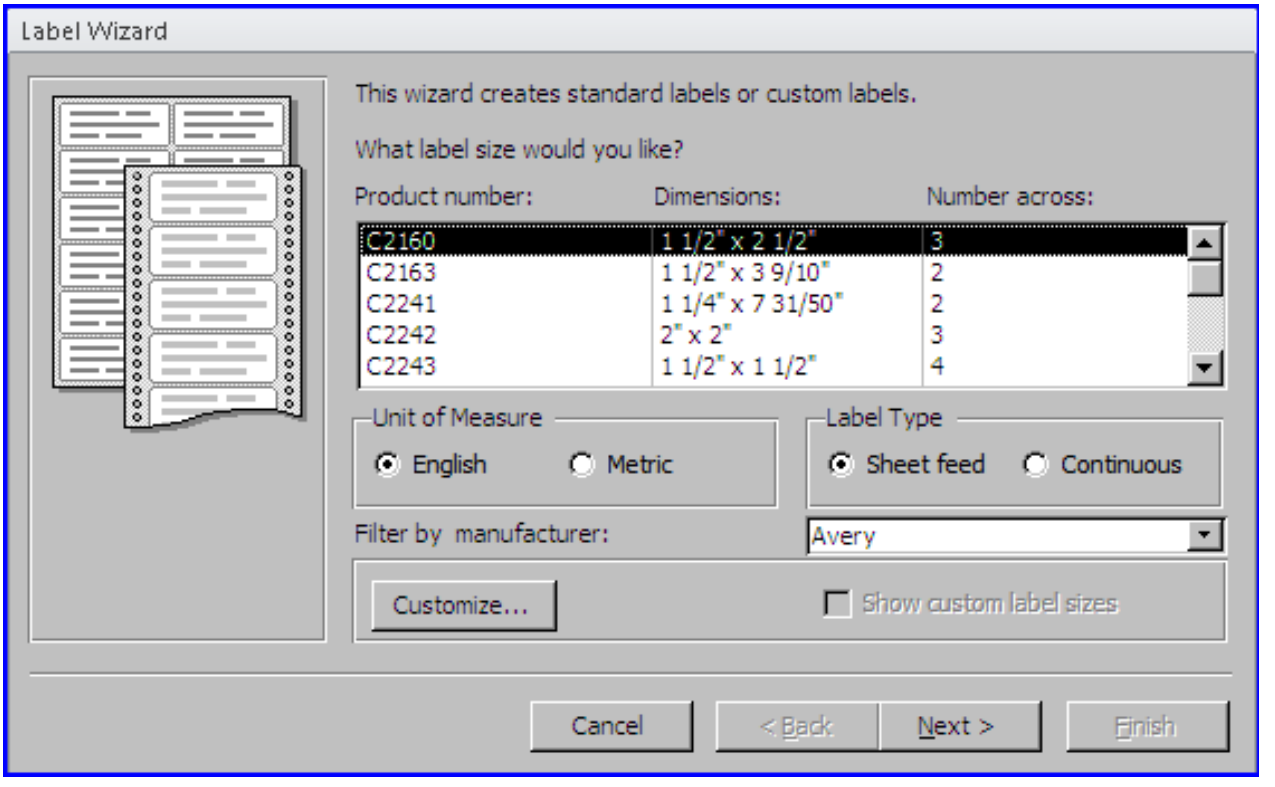


Figure 5.11: Label Wizard

2. Select the **Label Type** as **Sheet feed** or **Continuous** and then, click **Next>**. The Sample label will be displayed. You can change the text appearance of the label.

Concepts

Figure 5.12 displays the sample text appearance in the **Label Wizard**.

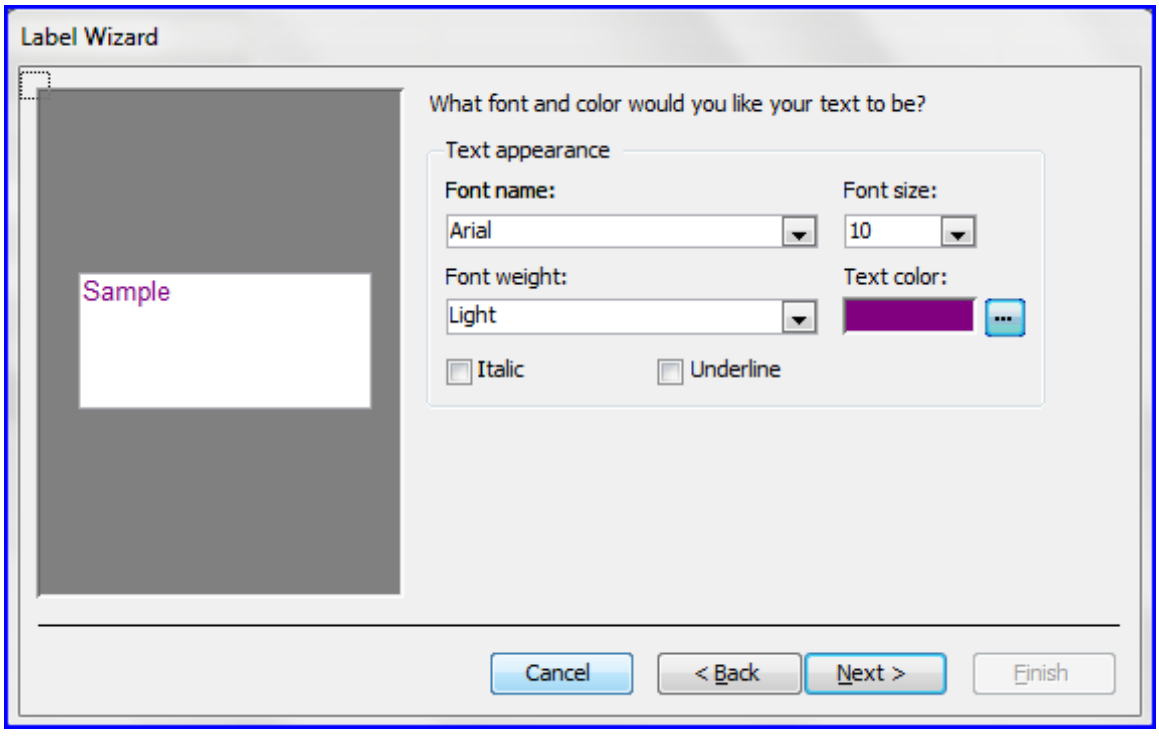


Figure 5.12: Text Appearance

3. Select the **Font name**, **Font size**, **Font weight**, and **Text color** and then, click **Next>**. Figure 5.13 displays the **Prototype label**.

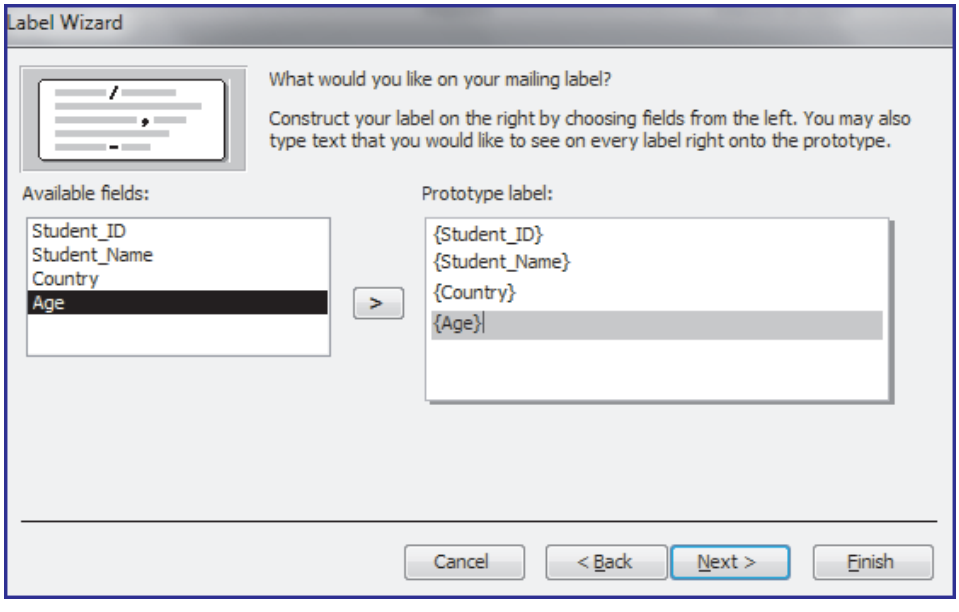


Figure 5.13: Prototype label

4. Add the fields from the **Available fields** in the **Prototype label** and then, click **Next>**.

Concepts

Figure 5.14 displays the sorting of fields. Select the field if you want to Sort in the **Sort by** pane.

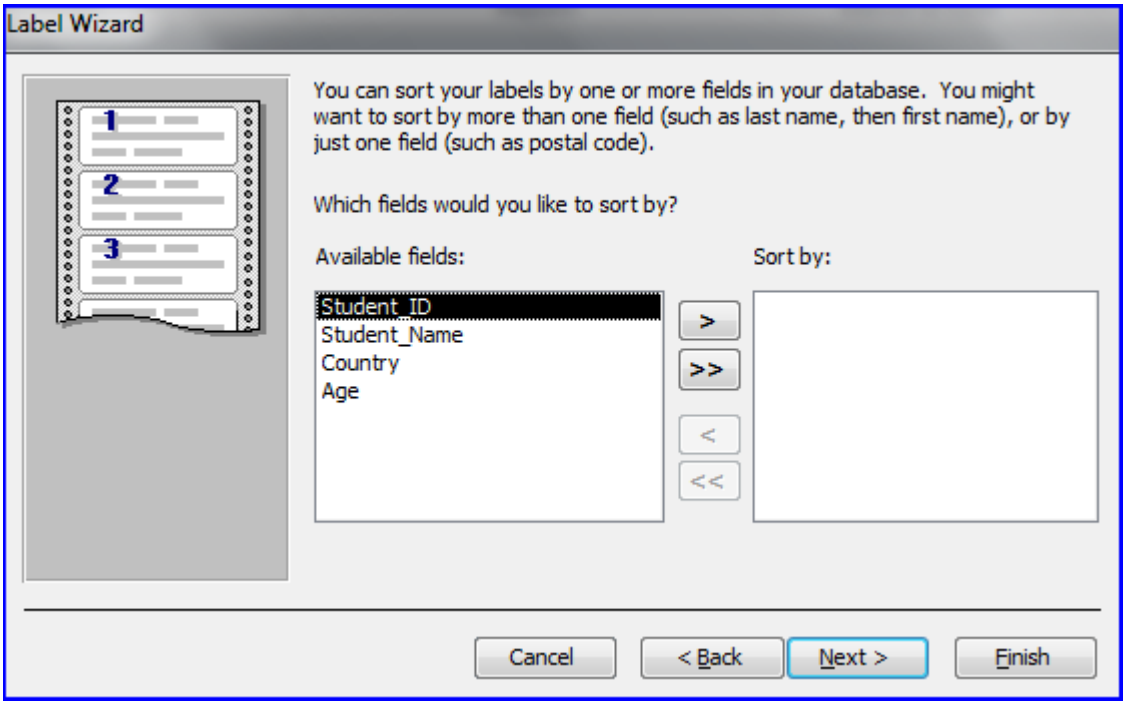


Figure 5.14: Selecting Fields to Sort

5. Click **Next>**. Rename the report. Figure 5.15 displays the name to be displayed in the report.

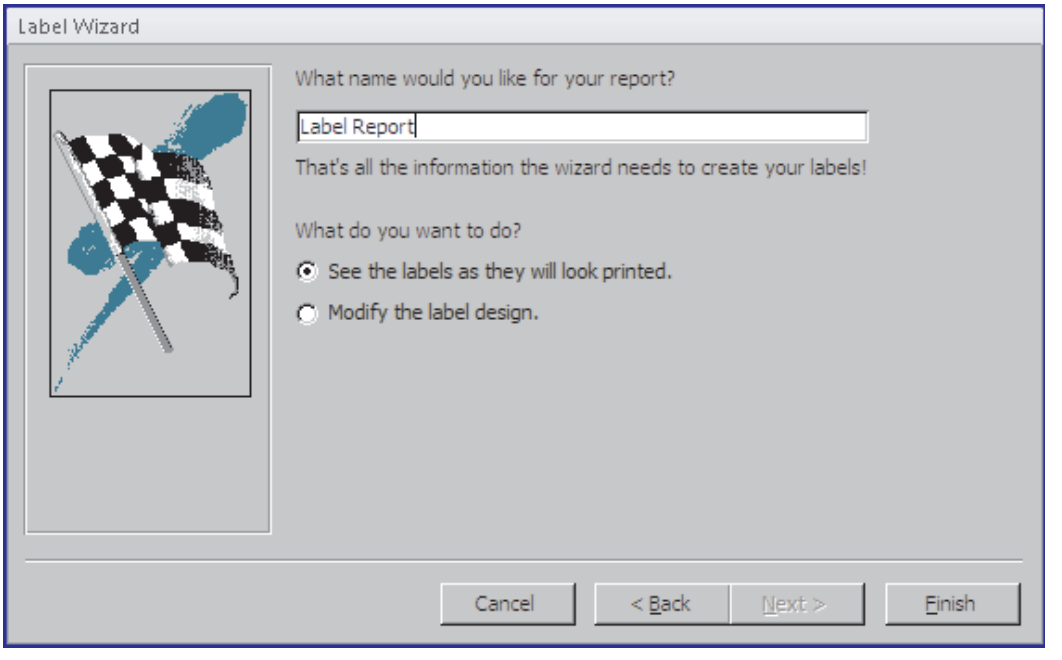


Figure 5.15: Naming the Report

6. Click **Finish**.

Concepts

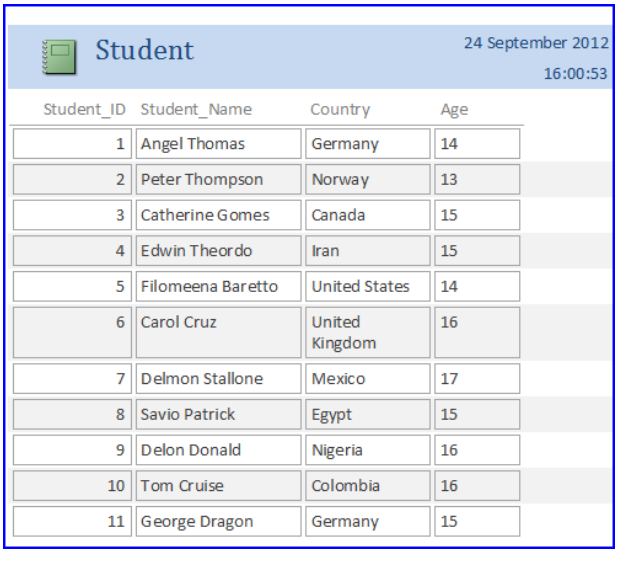
Figure 5.16 displays the **Label Report** that will appear in the **Print Preview**.

Figure 5.16: Label Report

5.6.2 Creating Print Layout

Before users view the report on a printed paper, they can use the **Print Preview** option to preview the report and make corrections, if any. When a user creates a report using the **Report Wizard**, it automatically enables the data to be viewed in a **Print Preview** format. In the **Navigation** Pane, users can right-click the report and then, click **Print Preview**.

Figure 5.17 displays the **Print Preview** report.



Student

24 September 2012
16:00:53

Student_ID	Student_Name	Country	Age
1	Angel Thomas	Germany	14
2	Peter Thompson	Norway	13
3	Catherine Gomes	Canada	15
4	Edwin Theordo	Iran	15
5	Filomeena Baretto	United States	14
6	Carol Cruz	United Kingdom	16
7	Delmon Stallone	Mexico	17
8	Savio Patrick	Egypt	15
9	Delon Donald	Nigeria	16
10	Tom Cruise	Colombia	16
11	George Dragon	Germany	15

Figure 5.17: Print Preview Report

Concepts

In the **Print Preview**, the report appears with margins and the data is displayed in the similar manner as it would appear on a paper.

5.7. Working with Options in Report Design, Arrange, and Format Tabs

The different sections of a report are **Report Header**, **Page Header**, **Detail**, **Page Footer**, and **Report Footer**. Sometimes, the user might not require all the sections of the report to be displayed. In such a case, the sections that are not required can be removed. This can be done by hiding the unwanted sections as follows:

1. Open the report in the **Design View**.
2. Select the appropriate section of the report.
3. Right-click the section and then, select **Properties**. The **Property Sheet** window is displayed.
4. Click the **Format** tab.
5. Click the **Visible** property.
6. Select **No**.

Access 2010 provides a variety of controls, which allows users to design or modify the report layout. Various controls such as a text box, logo, option group, and check box can be included within the report. The controls are added to a report for providing user interactivity.

Figure 5.18 displays the controls available in the **Controls** group.

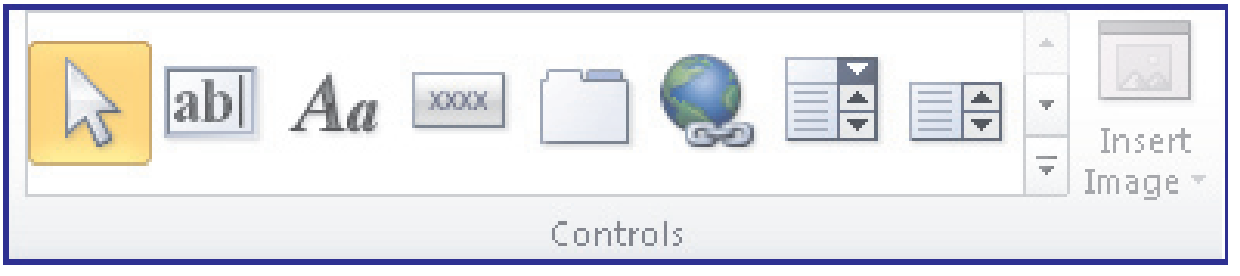


Figure 5.18: Controls Group

To add a control to a report, perform the following steps:

1. Open the report in the **Design View**.
2. On the **Design** tab, in the **Controls** group, select the appropriate control.
3. Drop the control on the desired location in the report.

To add a theme to a report, perform the following steps:

1. Open the report in the **Design View**.
2. On the **Design** tab, in the **Themes** group, select the theme to apply to the report.

Concepts

Figure 5.19 displays the report with the theme.

Student

24 September 2012
16:04:37

Student_ID	Student_Name	Country	Age
1	Angel Thomas	Germany	14
2	Peter Thompson	Norway	13
3	Catherine Gomes	Canada	15
4	Edwin Theordo	Iran	15
5	Filomeena Baretto	United States	14
6	Carol Cruz	United Kingdom	16
7	Delmon Stallone	Mexico	17
8	Savio Patrick	Egypt	15
9	Delon Donald	Nigeria	16
10	Tom Cruise	Colombia	16
11	George Dragon	Germany	15

Figure 5.19: Report with the Theme

➔ Applying Conditional Formatting

The conditional formatting feature is used for customizing reports. Using this feature, the font of any specified table or field can be formatted.

To apply conditional formatting, perform the following steps:

1. Open the Access 2010 database.
2. From the **Navigation Pane**, select an appropriate report.
3. Open the report in the **Design View**.
4. Select the field for which the conditional format has to be applied.
5. On the **Format** tab, in the **Control Formatting** group, click **Conditional Formatting**.

Concepts

Figure 5.20 displays the **Conditional Formatting Rules Manager** dialog box.

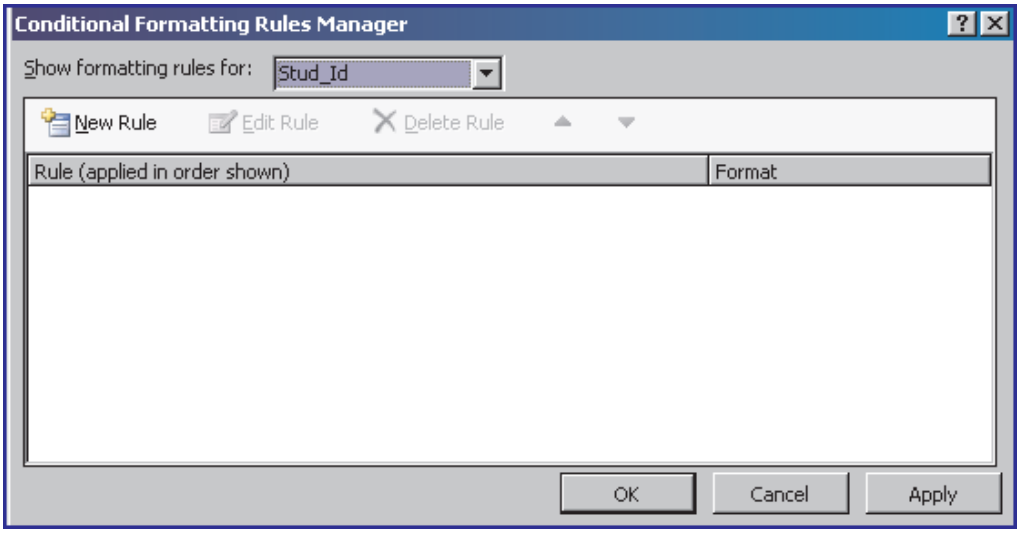


Figure 5.20: Conditional Formatting Rules Manager

6. Select the appropriate condition that is to be applied to the field.
7. Select the appropriate format that is to be applied when the given condition is met.
8. Click **OK**.

Access 2010 provides the **Arrange** tab for reports that allow users to change the layout of the report. Users can apply **Stacked** or **Tabular Layout** to the report as per their needs. Users can insert rows and columns with the **Insert Above**, **Insert Below**, **Insert Right**, and **Insert Left** options provided by Access 2010 in the **Rows & Columns** section. Further, they can also use the **Move Up** and **Move Down** to move the rows and columns in the report. In the **Position** group, users can use the **Control Margins** and **Control Padding** feature.

Access 2010 provides the **Format** tab for reports that allows the formatting of the report. The **Format** tab contains the **Selection**, **Font**, **Number**, **Background**, and **Control Formatting** groups. Users can apply the different styles of formatting by using **Quick Styles**.

5.8. Sorting and Filtering Records in a Report

Filtering is a feature that allows users to view only the data that they want to display. Users can use filters to display only some records in a report, form, or a datasheet. Applying filters allow users to restrict the data in the report without making any change in the tables.

There are various types of filters that users can easily apply and remove when not required. Access 2010 has some common filters that are used to filter records.

To apply a filter to the records in a report, perform the following steps:

1. Open the report in the Access 2010.
2. Select the **Home** tab in the **Sort & Filter** group, click **Filter**, and then, click the `Age, 15`.

Concepts

Figure 5.21 displays the creation of the filter for the Age field.

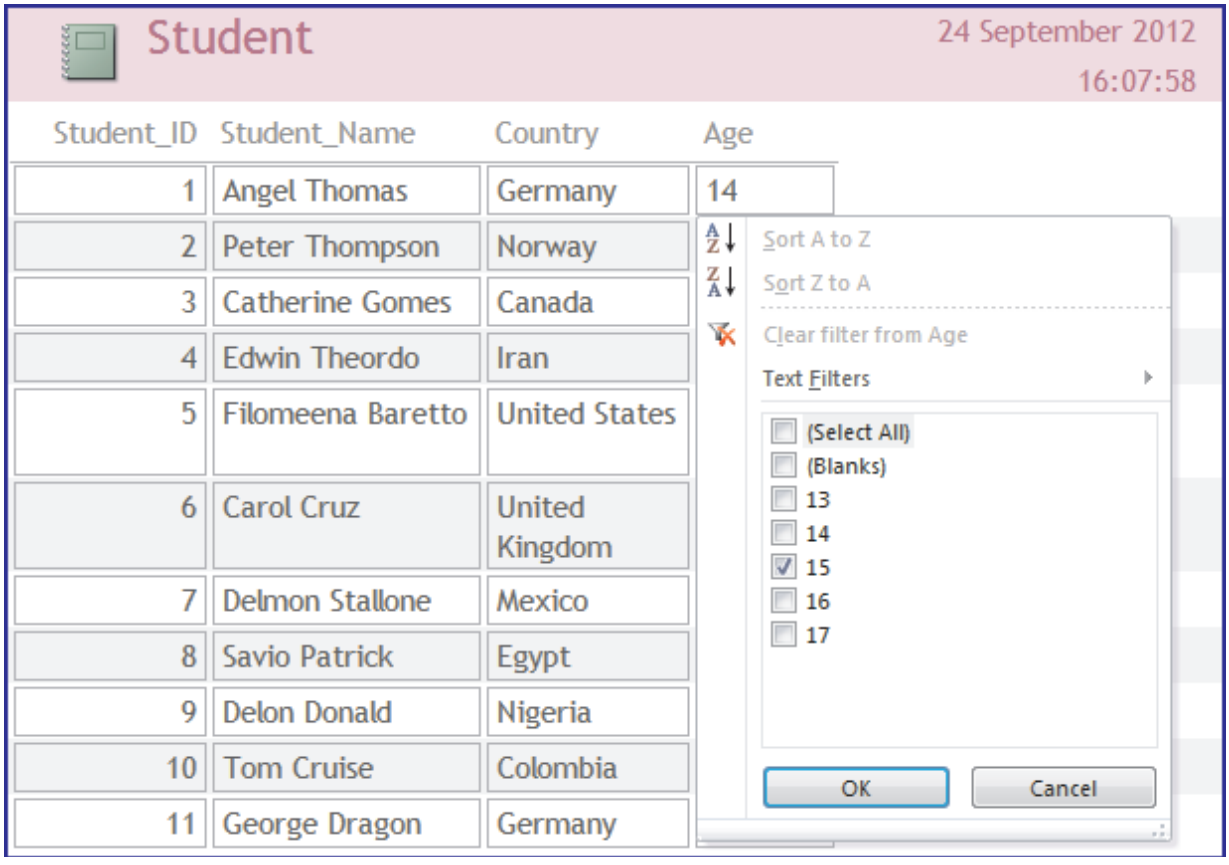


Figure 5.21: Creating a Filter for Age

3. Click **OK**.

Figure 5.22 displays the filtered records for the Age, 15.

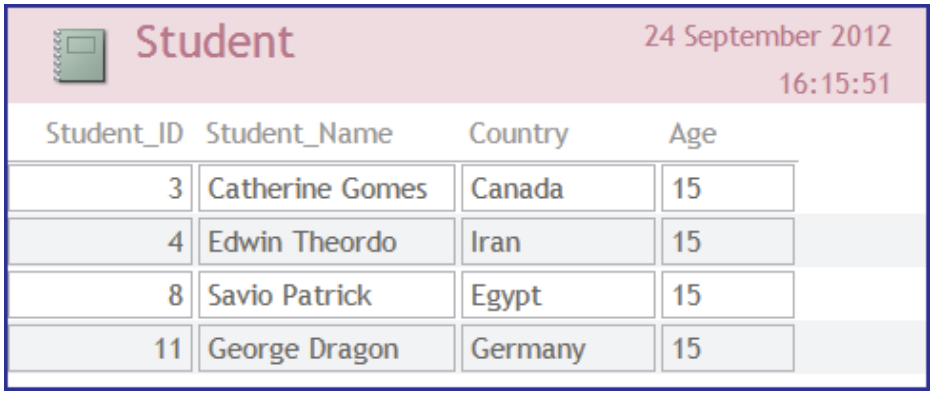


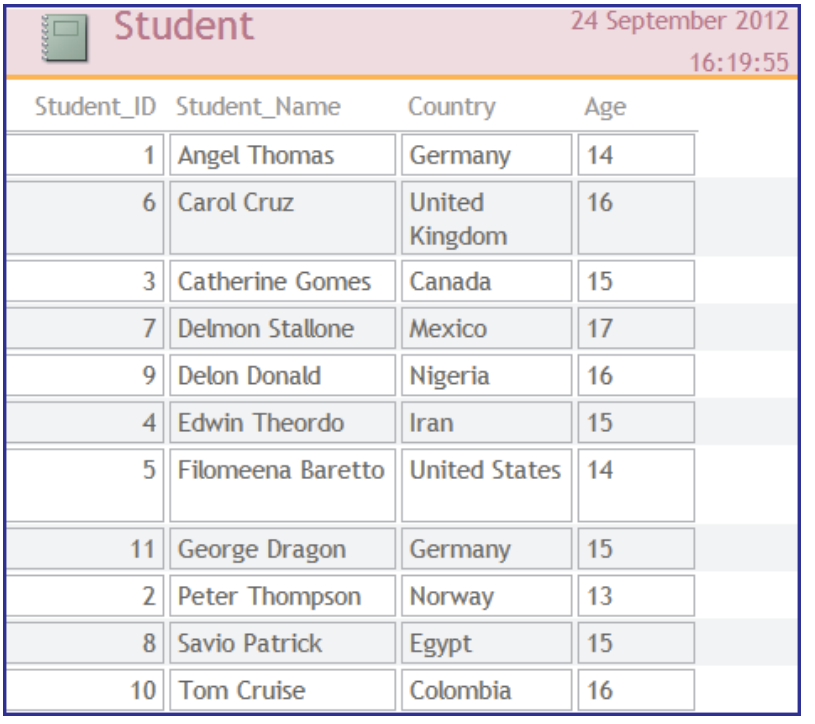
Figure 5.22: Filtered Records

Similarly, users can sort the records in a report. For example, users may want to view the student names in ascending order.

Concepts

To apply sort for the `Student_Name` column, perform the following steps:

1. Select the `Student _ Name` column in the Student report.
2. Select the **Home** tab in the **Sort & Filter** group, click **Ascending**. The `Student _ Name` column will be displayed in ascending order. Figure 5.23 displays the `Student_Name` field values in ascending order.



Student

24 September 2012
16:19:55

Student_ID	Student_Name	Country	Age
1	Angel Thomas	Germany	14
6	Carol Cruz	United Kingdom	16
3	Catherine Gomes	Canada	15
7	Delmon Stallone	Mexico	17
9	Delon Donald	Nigeria	16
4	Edwin Theordo	Iran	15
5	Filomeena Baretto	United States	14
11	George Dragon	Germany	15
2	Peter Thompson	Norway	13
8	Savio Patrick	Egypt	15
10	Tom Cruise	Colombia	16

Figure 5.23: Records in Ascending Order

To remove sort, click **Remove Sort**.

Concepts

5.9 Check Your Progress

1. A _____ is the best way to print data and format data and also summarize data in an organized manner.

(A)	Table	(C)	Form
(B)	Report	(D)	Query

2. Which of the following wizards will display the grouping levels to be applied to the reports?

(A)	Sort	(C)	Report
(B)	Navigation	(D)	Blank Form

3. Which of the following allows specifying the names of table fields, controls, reports, or properties of those fields or controls?

(A)	Identifiers	(C)	Functions
(B)	Operators	(D)	Constants

4. Tom is working on an online community Web site form. He wants to create and display a formatted list of the registered users who have deactivated their profile from last two months, along with total count. Which of the following features allows him to accomplish this task?

(A)	Charts	(C)	Tables
(B)	Reports	(D)	Forms

5. Which of the following can be added to a report to enable users to go to a specific location or file on the computer, it can also be used to get connected to the Internet and visit the specified Web site?

(A)	Pivot Table	(C)	Sort
(B)	Filter	(D)	Hyperlink

5.9.1 Answers

1.	C
2.	B
3.	A
4.	B
5.	D

Summary

- Reports are the summarized final output of the database.
- Reports allow users to present the tables and query the data in the available formats.
- Users can make use of the Blank Report tool when only a few records are to be selected to display on the report.
- In Access 2010, the report properties are used to customize the report and to add various functions in the report.
- An expression is similar to a formula used to generate a result.
- Date functions allow user to filter the date fields inserted in the report.
- A Hyperlink can be used to get connected to the Internet and visit the specified Web site.

Try it Yourself

1. Oxem Cars is an automobile agency which is headquartered in Hong Kong. The agency has designed new cars and launched them in the market. The agency's Web site has already displayed the models of the new cars on the home pages and has received orders for the newly launched cars. Now, they want to create a report on the number of customers who have placed orders for cars. Assume that you have to create a database named OxemCarsDB having following tables as shown in figure 5.24.

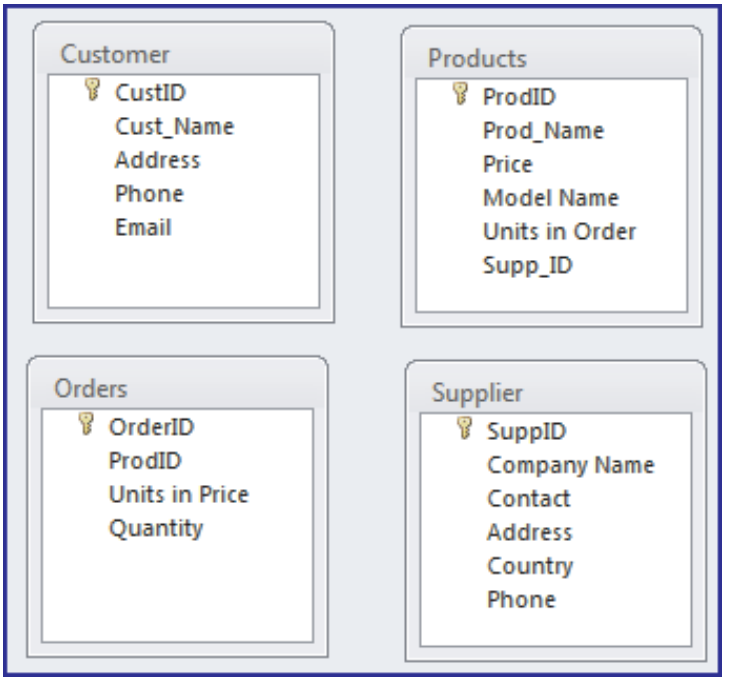


Figure 5.24: OxemCarsDB

Perform the following tasks:

a. Create a detailed report to display the list of customers who have placed the orders for the cars.

b. Create a report to display the list of cars that are sold by the OxemCars.

Concepts

Session - 5 (Workshop)

Working with Reports

In this workshop, you will learn to:

- Create reports in Access 2010
- Create a blank report, customize it, add themes and images, and apply sorting

5.1 Working with Reports

You will view and practice how to create reports in Access 2010. You will also view and practice how to create a blank report, customize it, add themes and images, and apply sorting.

➔ Creating Reports

➔ Creating a Blank Report, Customizing it, Adding Themes and Images and Applyijng Sorting

Note - Please refer to the respective lab deliverable for demonstrations and guided simulations.

Session - 6

Advanced Database Operations

Welcome to the Session, **Advanced Database Operations**.

This session explains about the working of macros in Access 2010. It also explains how a user can import and export different kind of data formats using Access 2010. Finally, it provides an overview of different security tools to secure a database and publish it on the Web using Access 2010.

In this Session, you will learn to:

- Explain macros
- Describe the process of creating and running a macro
- Explain the process of importing and exporting data in Access 2010
- Explain the features and tools to secure a database
- Describe publishing a database on the Web

6.1 Introduction

Consider a scenario where a departmental store has to perform certain tasks frequently on a database. These tasks may include: updating the stock table, running queries, and printing the sales report. A database manager has to perform each of these tasks individually, from time to time, on the database which results in repetition of the tasks.

Access 2010 provides a tool named 'macros' that can automate the tasks performed repeatedly on the database. This reduces the burden of performing them individually, thereby increasing the functionality of the application.

6.2 Macros

A macro is a tool that allows users to group a series of actions for automating tasks performed on a database. Macros are embedded to provide functionality within various database objects, such as forms, controls, and reports. For example, a form containing a command button can be associated with a macro to perform several actions, such as query execution and printing of a report. Thus, by recording both these actions within a macro, and then running the macro anytime later, increases productivity and saves time.

Figure 6.1 shows an example of execution of actions performed by a macro.

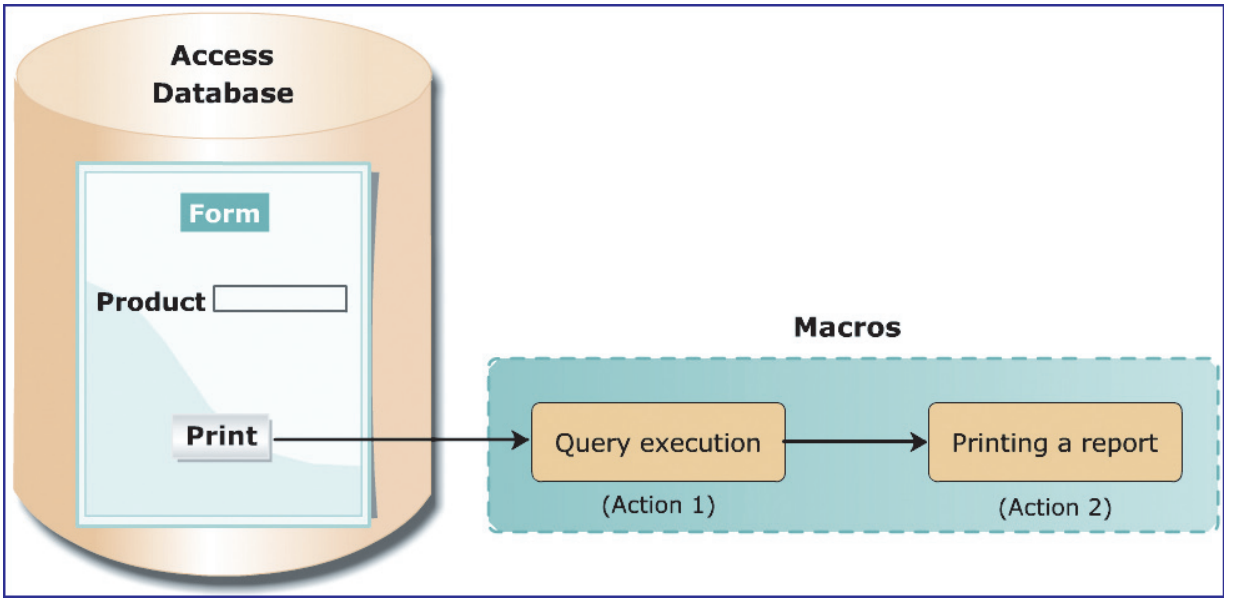


Figure 6.1: Example of a Macro Execution

All the actions captured by a macro are written in a language called Microsoft Visual Basic for Applications (VBA). This means that macros are programs, but it is not necessary for users to know VBA to build macros.

Concepts

6.2.1 Components of a Macro

A macro consists of macro actions, conditions, and arguments. The macro actions are ready to use VBA commands, while conditions are the constructs that control the actions. They form the building blocks of a macro. The brief description of these components is as follows:

- **Actions** – Actions specify operations that are performed by a macro, when the condition is met or a certain event occurs. For example, actions can be operations, such as, opening a report, executing a query, displaying a message, or finding a record.

- **Conditions** – Conditions are criteria that must be met before a macro performs the actions. Conditions are formed using expressions that are evaluated to true or false.

- **Arguments** – The information provided for an action is referred to as arguments. Arguments can contain one or more values. An action may or may not contain arguments. If an action needs a value, then a box for the corresponding argument is displayed.

Access 2010 provides a new **Macro Designer** window that helps to create complex macros with reduced errors. It provides a user interface with two panes namely, **Macro** window and the **Action Catalog** pane, separated by a split bar.

Figure 6.2 shows the **Macro Designer** interface with **Macro** window and **Action Catalog** pane. It is displayed when a new macro is created.

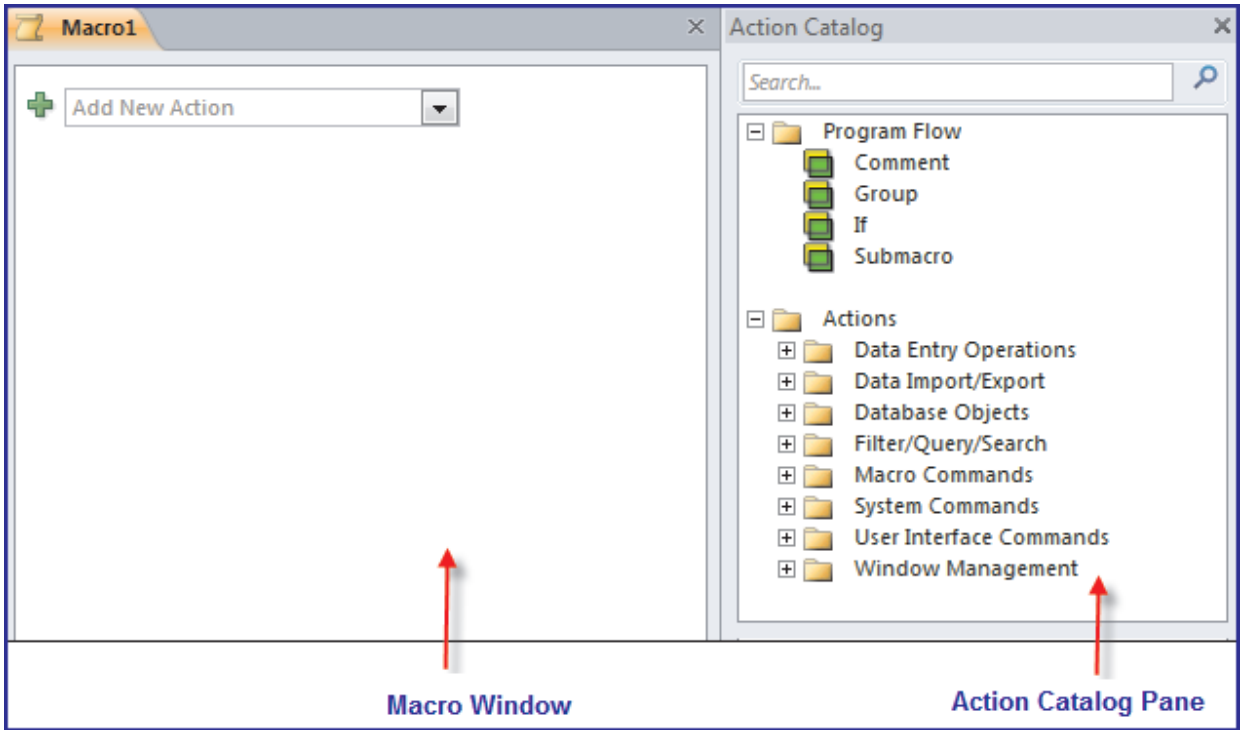


Figure 6.2: Macro Designer Window

Concepts

As shown in figure 6.2, the **Macro Designer** window contains a drop-down list which allows the user to add the specific action to a macro.

The **Action Catalog** pane contains two nodes namely, Program Flow and Actions. The Program Flow node allows the user to create conditions. Similarly, the Actions node allows the user to add actions to a newly created macro. Both the nodes are displayed with a **+** sign which indicates that they are expandable. Thus, to expand any of the node, click the **+** sign in the **Action Catalog** pane.

Figure 6.3 shows the expandable nodes in the **Action Catalog** pane.

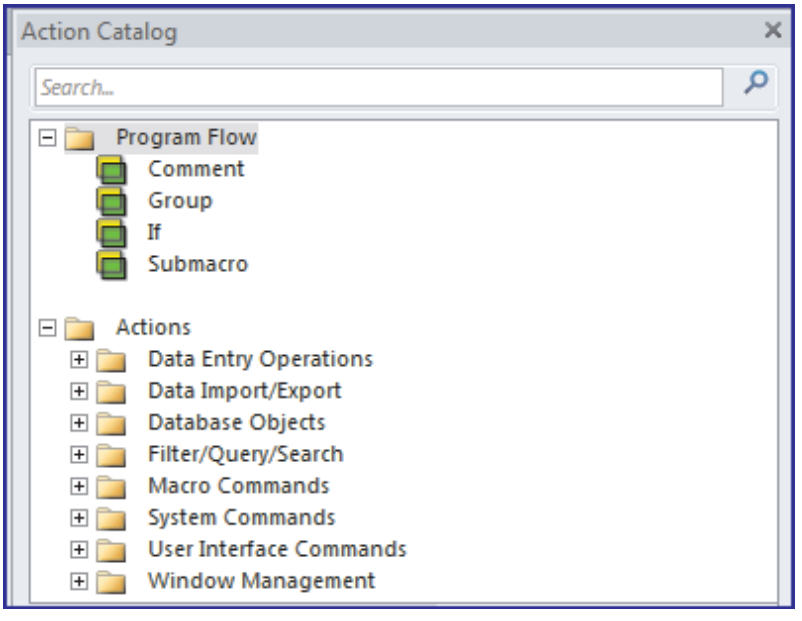


Figure 6.3: Expandable Nodes in Action Catalog Pane

As shown in figure 6.3, the **Actions** node comprises actions that are organized in different categories. To access an action, expand its node, which will display all the corresponding actions in that category.

Concepts

Figure 6.4 shows the various actions available within the **Data Entry Operations** category.

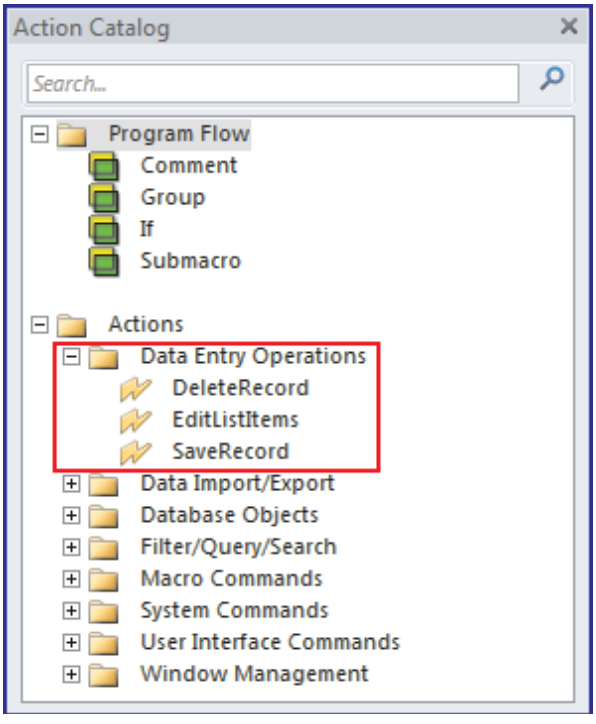


Figure 6.4: Data Entry Operations Action

Additionally, **Action Catalog** pane also display a node named **In this Database**. This node lists all macros created in the database.

6.2.2 Actions in a Macro

Access 2010 provides a wide range of actions. These actions can be added to a macro by selecting an action from the **Action** drop-down list in the **Macro** window. Alternatively, a user can also double-click or drag-and-drop an action from the **Action Catalog** pane onto the **Macro** window.

Some of the commonly used actions for creating a macro are as follows:

- The **OpenQuery**, **OpenReport**, **OpenTable**, **OpenForm**, and **OpenModule** actions allow the user to open database objects.
- The **FindNextRecord**, **FindRecord**, **RefreshRecord**, and **ApplyFilter** actions allow the user to filter and search records.
- The **MaximizeWindow**, **MinimizeWindow**, **RestoreWindow**, and **CloseWindow** actions allow the user to modify Access 2010 window.
- The **AddMenu**, **BrowseTo**, **MessageBox**, and **setMenuItem** actions control the interface displayed on the screen.

Concepts

- The **Beep**, **DisplayHourglassPointer**, **QuitAccess**, and **CloseDatabase** actions allow the user to make changes to the database system.

- The **GoToRecord**, **GoToControl**, and **SetProperty** actions allow the user to navigate between the records, controls, and set property of a form or report.

- The **DeleteRecord**, **SaveRecord**, and **EditListItems** actions allow the user to make changes to the data of a table.

6.2.3 Creating a Macro

Macros can be of two types that are described as follows:

- **Standalone Macro** – It is created and stored as a separate object in the database and can be bound to different objects in the database. These macros are visible in the Navigation Pane.

- **Embedded Macro** – It is created within the event of an object, such as form or report and belongs only to that object.

The steps to create a standalone macro are as follows:

1. Click the **Create** tab.
2. In the **Macros & Code** section, click **Macro**.

Figure 6.5 shows **Macro** button in the **Create** tab.

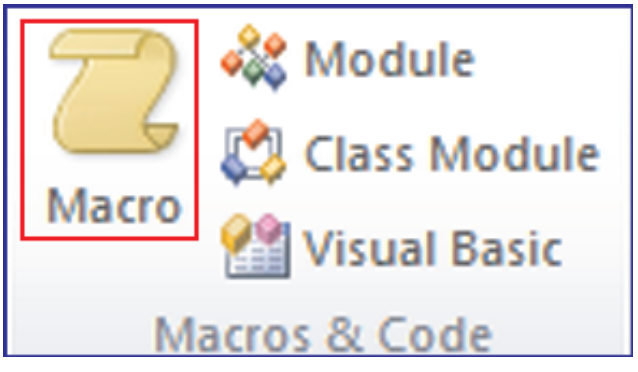


Figure 6.5: Macro Button

3. On clicking **Macro**, a **Macro Designer** window is opened. It contains a **Macro** window and the **Action Catalog** pane.
4. To create a macro using **Macro** window, click the arrow to display the actions drop-down list.
5. From the drop-down list, select the **MessageBox** action.

Concepts

Figure 6.6 shows the selection of **MessageBox** action in the **Macro** window.

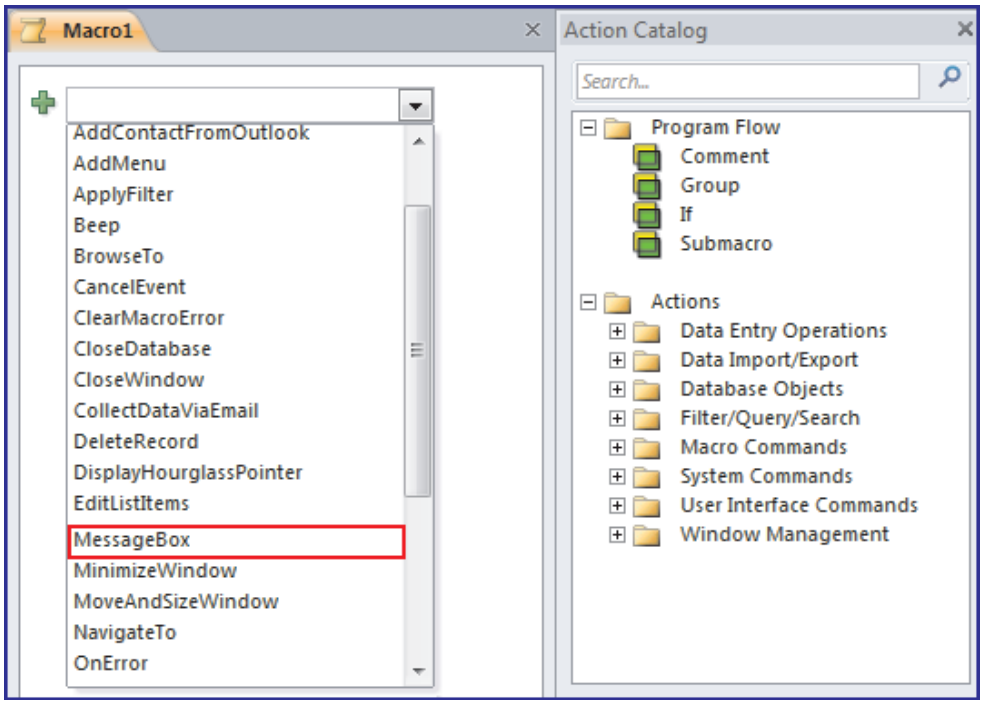


Figure 6.6: MessageBox Action

The **Macro** window is displayed with the **MessageBox** action. It contains arguments, such as **Message**, **Beep**, **Type**, and **Title**. The user needs to input values for the arguments. Some arguments are provided with default values which indicate that they are optional arguments.

Figure 6.7 shows the **MessageBox** action associated with the arguments.

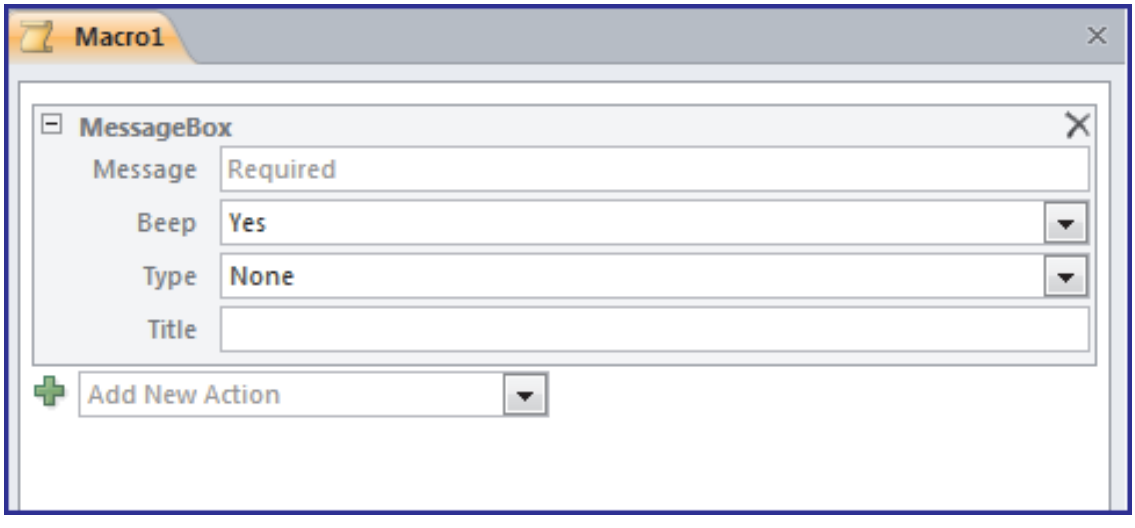


Figure 6.7: MessageBox Action with Arguments

As shown in figure 6.7, the **Message** argument defines the text that will be displayed in the message box. The **Beep** argument determines the sound that is heard on displaying the message box.

The **Type** argument defines the icon that includes: None, Critical, Warning, and Information displayed in the message box. Lastly, the **Title** argument accepts the text that will appear as title of the message box.

6. Type `Welcome to Macro` in the **Message** box.
7. Select **Information** from the **Type** drop-down list.
8. Type `Simple Macro` in the **Title** box.

Figure 6.8 shows the **MessageBox** action with the assigned user inputs.

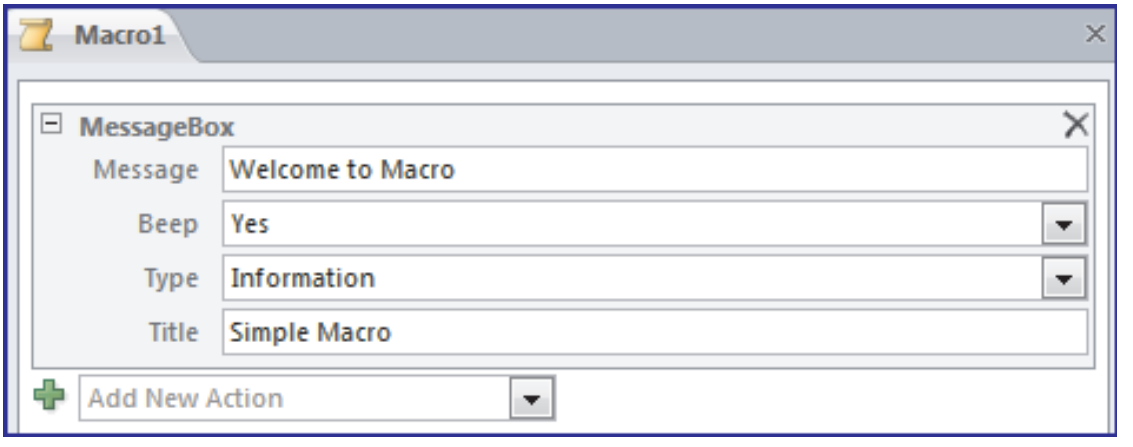


Figure 6.8: MessageBox Action with User Inputs

9. Click the button to close the macro window. A dialog box is displayed asking 'Do you want to save changes to the design of macro 'Macro1'?'.
10. Click **Yes** to display the **Save As** dialog box.
11. Type `MessageMacro` in the **Macro Name** box and click **OK** to save the macro.

`MessageMacro` is displayed in the Navigation Pane under **Macros** as shown in figure 6.9.

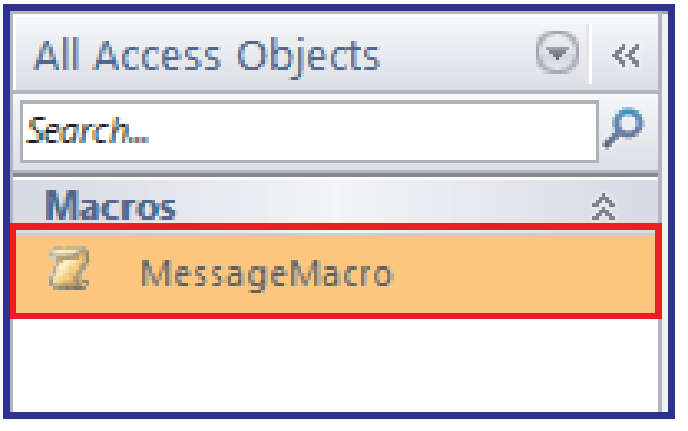


Figure 6.9: Navigation Pane - MessageMacro

Concepts

6.2.4 Running a Macro

A macro can be run in many different ways. It can be run directly, in a macro group, from another macro, form, or report. A macro can be run directly, if it is opened in the **Design View**.

The steps to run a macro in the **Design View** are as follows:

1. Right-click `MessageMacro` in the Navigation Pane.
2. Click **Design View** from the context menu.
3. Click **Run** in the **Tools** section of the **Design** tab.

Figure 6.10 shows the **Design** tab with **Run** button.

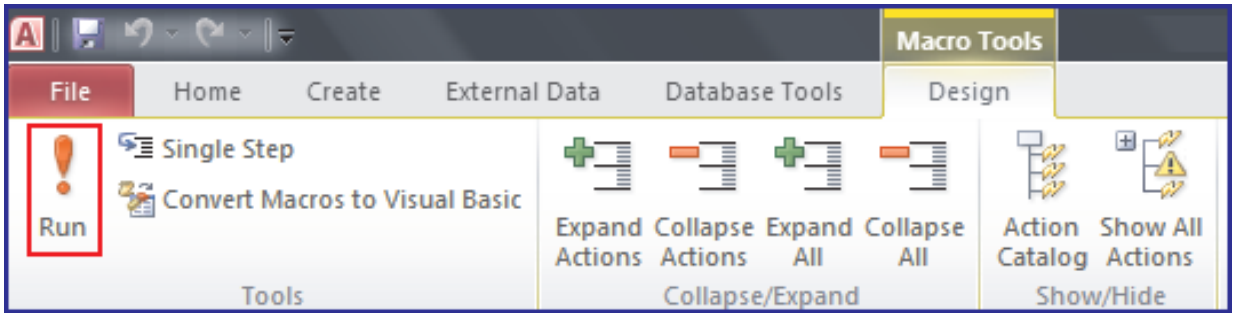


Figure 6.10: Design Tab

Figure 6.11 shows the execution of `MessageMacro` displaying a message box.

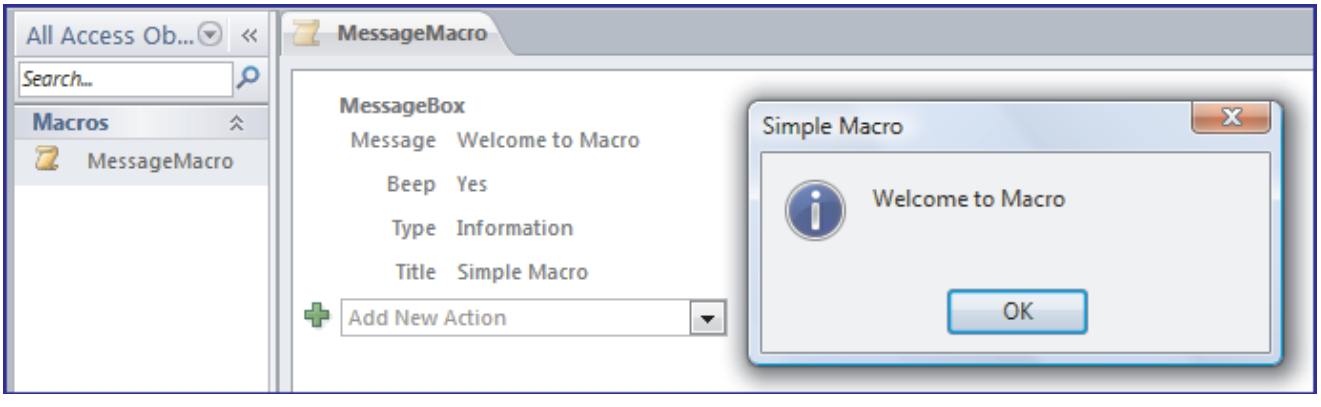


Figure 6.11: Execution of MessageMacro

Similarly, the steps to run a macro, when it is not opened in the **Design View** are as follows:

1. On the **Database Tools** tab, in the **Macro** section, click **Run Macro**.
2. Select the appropriate macro from the **Macro Name** drop-down list and click **OK**.

The macros can be bounded to an event of a control on the form or report. When the event occurs, it triggers the macro.

Concepts

6.2.5 Data Macros

A data macro is a new feature introduced in Access 2010. Data macros are used to apply the logic and functionality to the data stored within the tables of a database. They are automatically executed (fired) whenever any add, update, or delete operation is performed on the table. They are also fired for the objects, such as forms, reports, or queries bound to a table that has a data macro attached to it.

Some of the scenarios for using data macros are as follows:

- ➔ To check the value of a field before adding a record to the table.
- ➔ To maintain the history of changes made to a record.
- ➔ To generate an e-mail whenever the value of a field changes.
- ➔ To verify that the data stored in a table is accurate.

There are two kinds of data macros namely, event macros and standalone macros. The event macros run whenever any action is performed on the table. The standalone macros are named macros that run whenever they are invoked by their name.

Table 6.1 lists some of the events supported by data macros.

Event	Description
BeforeChange	Creates logic that runs and validates the data before a record is saved to the table
BeforeDelete	Creates logic that runs and validates the data before a record is deleted from the table
AfterInsert, AfterUpdate, AfterDelete	Creates logic that runs after the data is added, updated, or deleted from the table

Table 6.1: Events Supported by Data Macros

To add the data macro, open the table and in the **Table** tab, click the **Before Change/Before Delete** in the **Before Events** group.

6.3 Importing and Exporting Data

Access 2010 has the ability to interchange the data from many programs, such as Word 2010 and Excel 2010. It provides various ways through which data can be moved in and out of the database. The feature that is used to support data exchange is **Import and Export**.

Following are some of the uses of import and export feature supported by Access 2010:

- Data created in other programs can be combined with the data residing in the database.
- Data can be transferred between the programs.
- Data can be collected and stored for longer duration.
- Data can be exported to programs such as Excel 2010 for analysis.

6.3.1 Exporting Data

Exporting data means to transfer the data from one database to another database or program. Access 2010 allows exporting a table, form, report, module, macro, or query from one database to another. For example, a departmental store, using both Excel 2010 and Access 2010 applications, may need to export the data from Access 2010 to Excel 2010 for analysis.

Exporting data saves the time required for copying the entire data. It also allows sharing of data among the workgroup on different applications.

Figure 6.12 shows exporting of data to other programs.

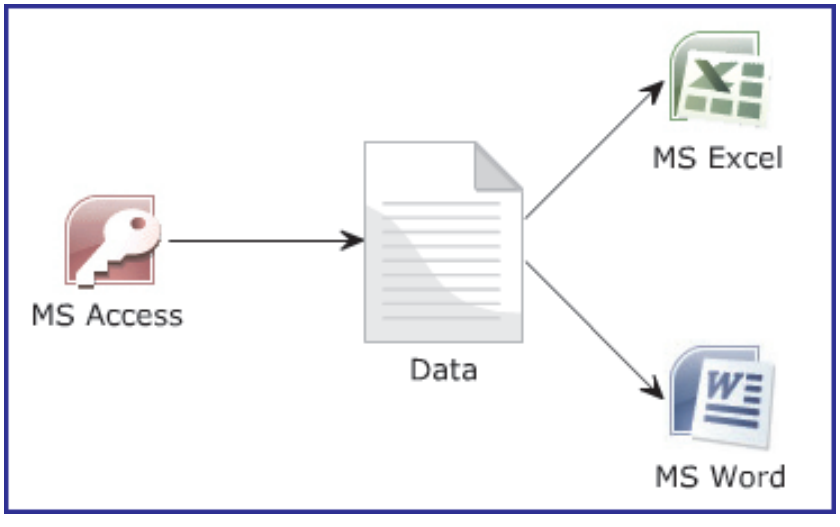


Figure 6.12: Exporting Data

Apart from this, Access 2010 supports exporting of data to other formats, such as Portable Document Format (PDF) file and XML Paper Specification (XPS) file. These files preserve the format and can be shared easily with other users. They can also be viewed online, printed, posted, or distributed through mails.

Consider a situation where the ToyHeaven company that was described in an earlier session wants to share its product details with the marketing people. The data is used to compare and discuss about the sales growth of their products in the recent years.

Concepts

To accomplish this task, the data needs to be exported from the `Product` table to Excel 2010 worksheet.

The steps to export the data from Access 2010 database to an Excel 2010 worksheet are as follows:

1. Open the `ToyHeaven` database.
2. In the Navigation Pane, select the `Product` table to be exported.
3. On the **External Data** tab, click **Excel** in the **Export** section.

Figure 6.13 shows the selection of Excel 2010 to export the data.

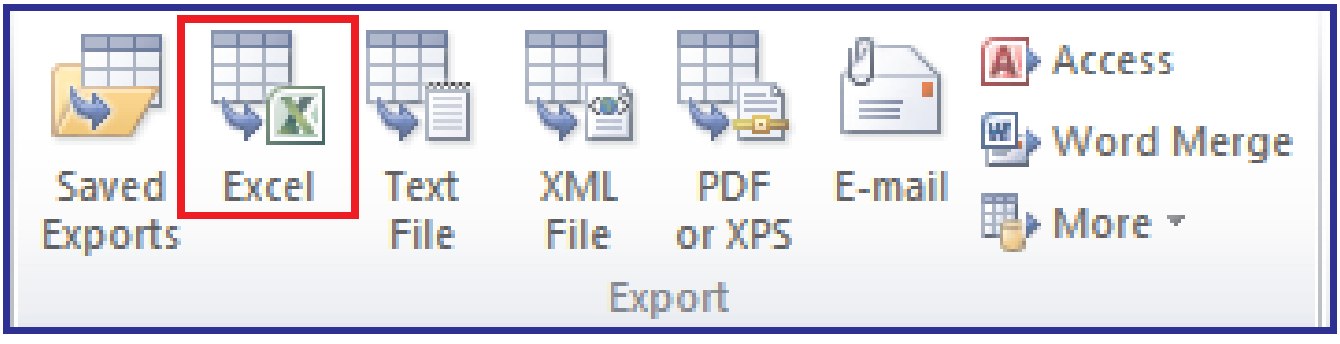


Figure 6.13: Selecting Excel for Exporting Data

Access 2010 opens the **Export – Excel Spreadsheet** wizard.

4. In the wizard, type an appropriate destination file name in the **File Name** box. Also, select the appropriate format from the **File Format** drop-down list. The default file format is Excel Workbook (*.xlsx).
5. Click **OK** to export the table data.
6. Select the **Save export steps** check box.

Concepts

Figure 6.14 shows the **Save Export Steps** in the wizard.

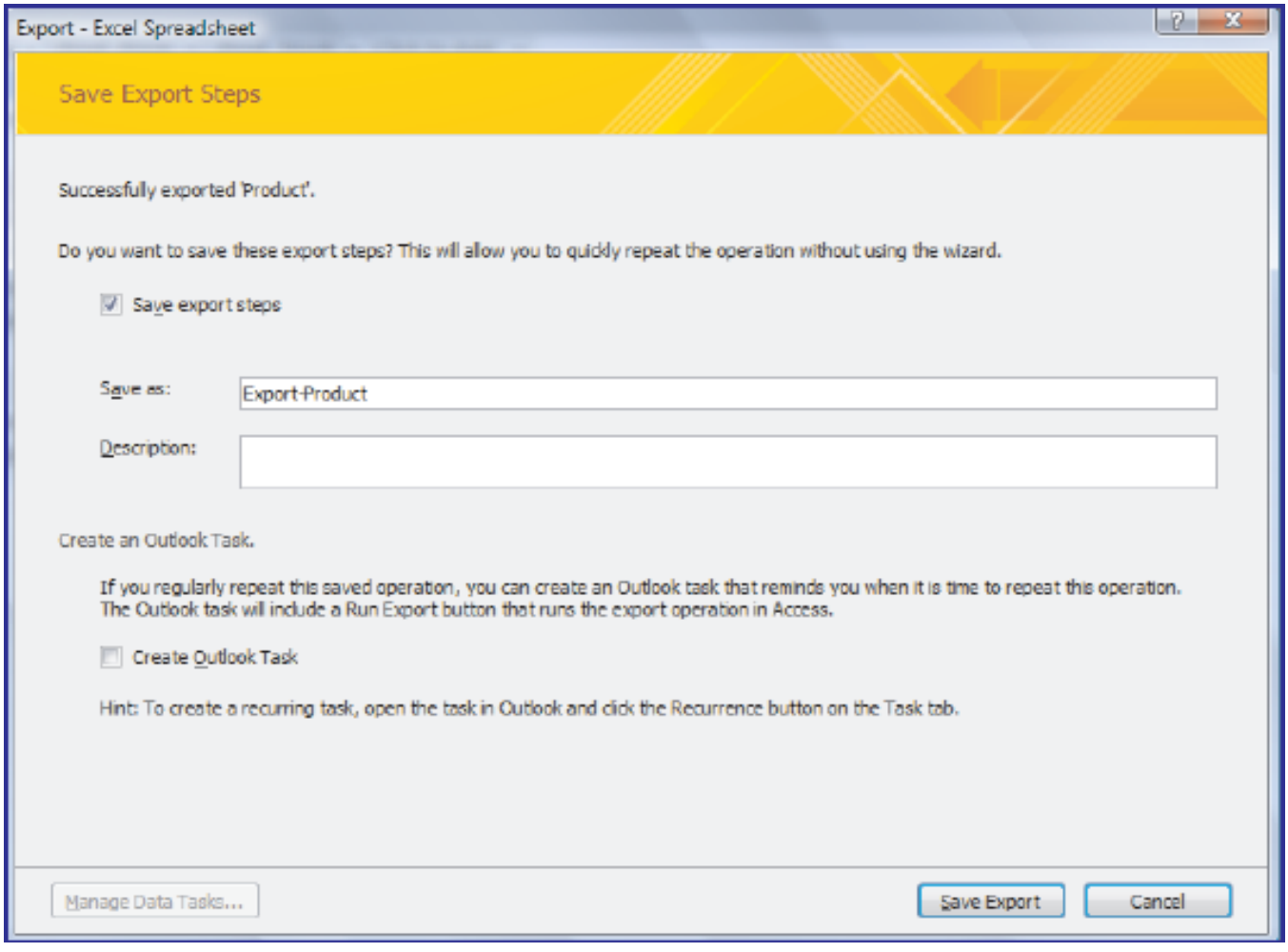


Figure 6.14: Save Export Steps

7. Click **Save Export** to save steps on the **Ribbon**. This allows users to perform the same export operation as and when required.

Figure 6.15 shows the exported data in the Excel 2010 worksheet.

	A	B	C	D	E
1	Prod_ID	Prod_Name	Prod_Category	Prod_Price	Prod_Stock
2	P0001	Hotwheels	Car	400	20
3	P0002	Blocks	Construction	200	100
4	P0003	Barbie	Doll	500	200
5	P0004	Silly Putty	Creative	200	250
6	P0005	Lego Mindstorms	Educational	500	100
7	P0006	Speak and Spell	Educational	200	150
8	P0007	Capsela	Construction	250	100
9	P0008	Construx	Construction	300	150
10	P0009	Dinky Cars	Car	100	200
11	P0010	Fulla	Doll	250	100
12	P0011	G.I.Joe	Action Figure	100	200
13	P0012	Jumping Jack	Action Figure	120	150

Figure 6.15: Exported Data in Excel 2010

Note - To run the steps again, click **Save Exports** button in the **Export** section.

6.3.2 Importing Data

Importing data means to collect data from another database or program. The data can be directly copied to the required database, but importing and linking provides better flexibility.

In Access 2010, data can be imported from Word 2010 or Excel 2010 applications. When data is imported, a copy of the imported data is created at the required location. Thus, the source data is not affected in the process of importing data.

Figure 6.16 shows importing of data to Access 2010 database from other programs.

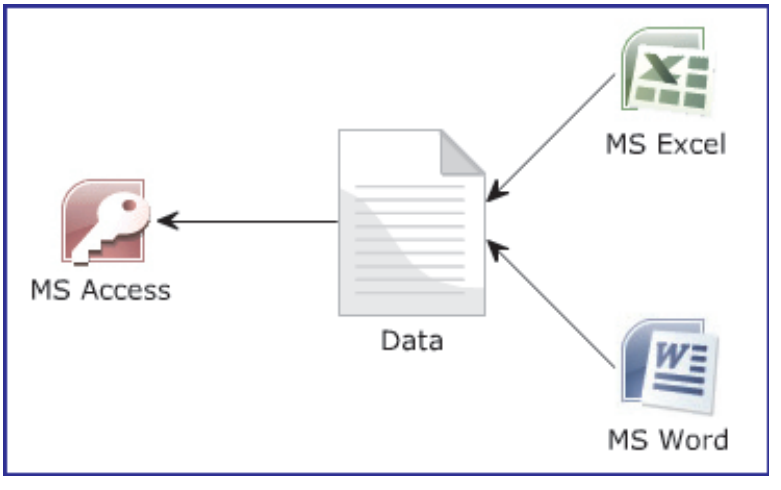


Figure 6.16: Importing Data

Concepts

In Access 2010, data can be imported from other databases or programs, such as a table, query, report, form, or module. It supports importing of multiple objects in a single operation. It is also possible to import tables with relations.

The steps to import the data from Excel 2010 to Access 2010 database are as follows:

1. Open the `ToyHeaven` database.
2. On the **External Data** tab, click **Excel** in the **Import & Link** section.

Figure 6.17 shows the selection of Excel program to export the data.

Figure 6.17: Selecting Excel for Importing Data

Access 2010 opens the **Get External Data – Excel Spreadsheet** wizard.

3. Click **Browse** to select the excel file from the local system into the **File Name** box. The selected file acts as the source.
4. Click **OK**.
5. Select the worksheet from which data needs to be imported and then, click **Next**.
6. Select **First Row Contains Column Headings** check box. This indicates whether to treat the first row as heading or data.
7. Click **Next** to display the **Field Options** containing **Field Name**, **Data Type**, **Indexed**, and a check box asking **Do not import field (Skip)**.
8. Select each column and change its data type.
9. To specify whether Access 2010 should add a primary key or not, click the appropriate button.
10. Click **Next**.
11. Type `Product _ Details` in the **Import to Table** box.
12. Click **Finish**.
13. Select **Save import steps** check box.
14. Click **Save Import** to save steps on the **Ribbon**. This allows users to perform the same import operation as and when required.

Concepts

Figure 6.18 shows the data imported in the `Product_Details` table.

Product_Details

ID	Prod_ID	Prod_Name	Prod_Categ	Prod_Price	Prod_Stock
1	P0001	Hotwheels	Car	400	20
2	P0002	Blocks	Construction	200	100
3	P0003	Barbie	Doll	500	200
4	P0004	Silly Putty	Creative	200	250
5	P0005	Lego Mindstorı	Educational	500	100
6	P0006	Speak and Spel	Educational	200	150
7	P0007	Capsela	Construction	250	100
8	P0008	Construx	Construction	300	150
9	P0009	Dinky Cars	Car	100	200
10	P0010	Fulla	Doll	250	100
11	P0011	G.I.Joe	Action Figure	100	200
12	P0012	Jumping Jack	Action Figure	120	150
*	(New)				

Figure 6.18: Imported Data in Access 2010

Note - A data source can be linked by creating a linked table using **Get External Data** wizard. By linking, changes made to the source data will be reflected in the linked table.

6.4 Security

The data in a database is prone to theft or misuse. It can be modified by an unauthorized person, which is a threat to the confidentiality of the data. Access 2010 provides a security model that applies security to a database. The security model provides encryption tool that prevents unauthorized access to the data in a database.

In Access 2010, encryption is done by setting a password for the database. Access 2010 uses a strong algorithm to encrypt the data, which makes it unreadable for the unauthorized users. The encrypted data is accessible only if the correct password is provided.

Figure 6.19 shows the encryption of data to secure the database from unauthorized access.

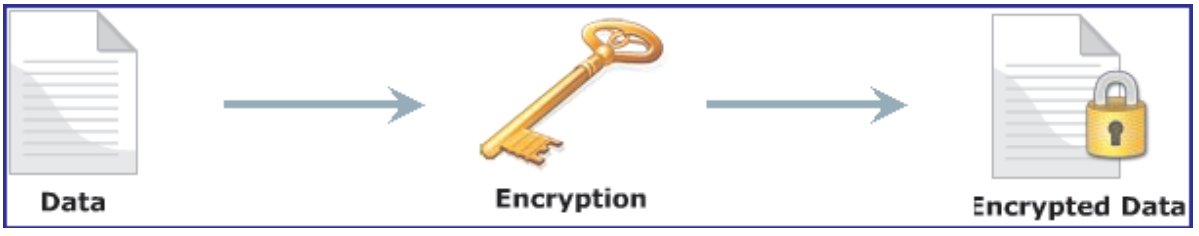


Figure 6.19: Encryption of Data

Concepts

6.4.1 Encryption

The encryption tool uses a database password to secure a database. It encrypts the password and makes the data unreadable to other tools. In order to work with the database, a user needs to enter a password.

The steps to use a password to protect Access 2010 database are as follows:

1. Click **File** tab.
2. Click **Open**. The **Open** dialog box is displayed.
3. Browse to select the database to be encrypted.
4. Click the **Open** down arrow and click **Open Exclusive**.

   Figure 6.20 shows the selection of **Open Exclusive** in the **Open** dialog box.

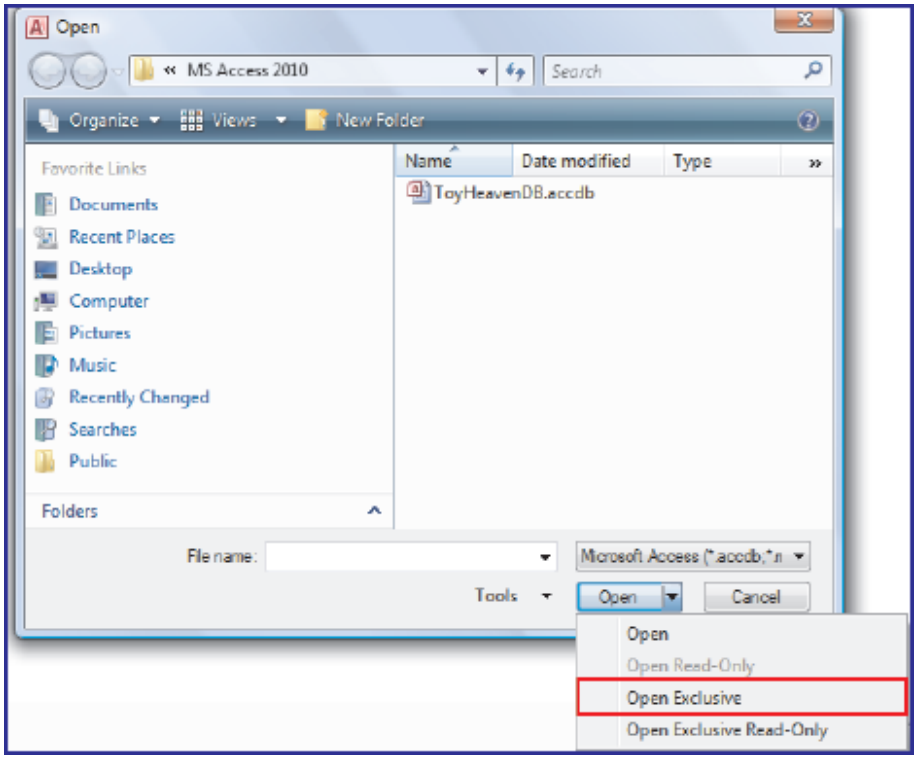


Figure 6.20: Open Dialog Box

5. On the **File** tab under **Info** option, click **Encrypt with Password** to display **Set Database Password** dialog box.

Figure 6.21 shows the **Set Database Password** dialog box.

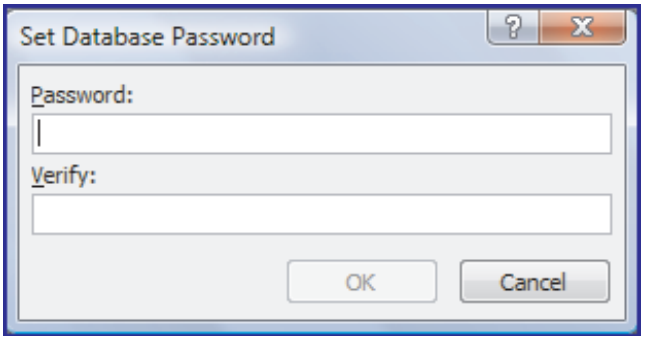


Figure 6.21: Set Database Password Dialog Box

6. Type a password in the **Password** box.
7. Type the same password in the **Verify** box.
8. Click **OK** to save password for the database.

6.4.2 Decryption

Decryption is the process of decoding the encrypted password information.

The steps to decrypt a password for Access 2010 database are as follows:

1. Open the database that has password protection. This displays the **Password Required** dialog box.

   Figure 6.22 shows the **Password Required** dialog box.

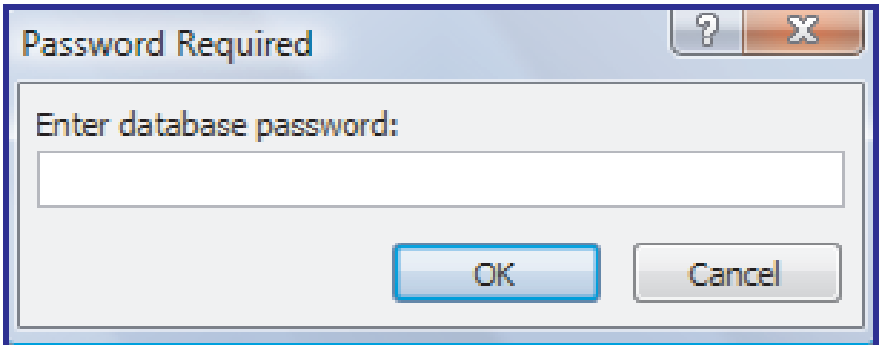


Figure 6.22: Password Required Dialog Box

2. Type the password in the **Enter database password** box.
3. Click **OK** to open the database.

Concepts

6.4.3 Trust Center

Whenever an Access 2010 database is opened for the first time, a **Message Bar** tool displays the security warning. The security warning is regarding enabling the macros and ActiveX controls present in the database.

Figure 6.23 shows the **Message Bar** tool with a security warning on opening an Access 2010 database.

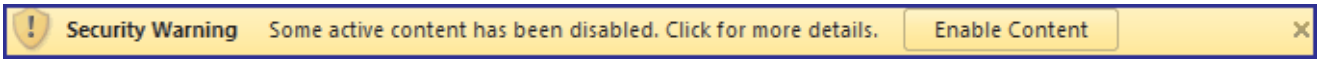


Figure 6.23: Message Bar Tool

The **Message Bar** tool indicates that the database is not trusted. The non-trusted databases are those that have several security risks from different Access 2010 components. These components are as follows:

- Queries that insert, update, and delete data from the tables
- Complex macros designed to corrupt the data
- Unsafe expressions
- VBA code

Access 2010 provides a feature named Trust Center that helps to secure the data in the database. It performs a set of security checks whenever the database is opened. For example, if a database file with extension `.accdb` or `.accde` is opened, then Access 2010 submits its location to the Trust Center. The Trust Center checks whether the location is trusted or not. If location is trusted, then it runs the database with all the functionalities. Trust Center is also used to create or change the trusted locations for the databases in Access 2010.

6.4.4 Creating a Trusted Location in Trust Center

The database opened in a trusted location ensures that all components are trusted. Also, it suppresses the security warning displayed for trusted databases.

The steps to create a trusted location in Access 2010 are as follows:

1. Click the **File** tab and click **Options**. The **Access Options** dialog box is displayed.
2. Under **Microsoft Access Trust Center** heading, click **Trust Center Settings**.

Figure 6.24 shows the **Access Options** dialog box with **Trust Center Settings**.

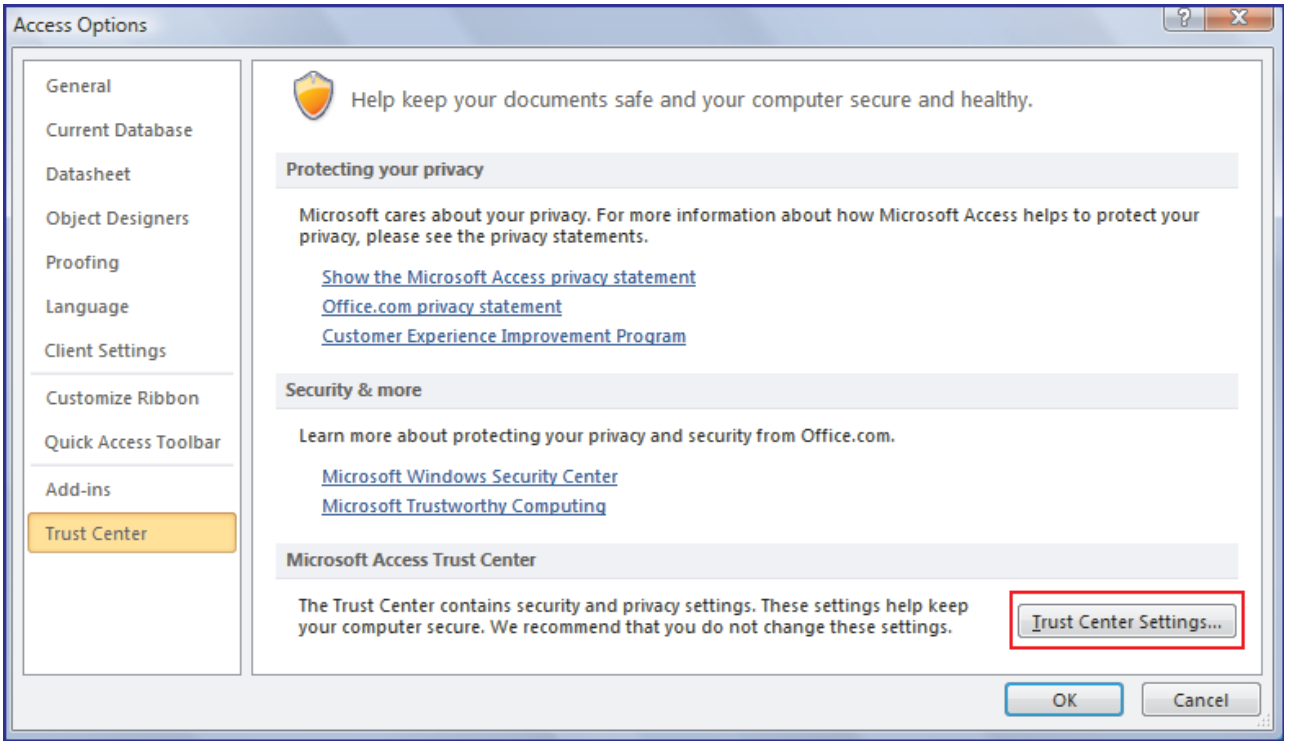


Figure 6.24: Access Options Dialog Box

Access 2010 displays the **Trust Center** Window; click **Trusted Locations** in the left pane.

3. To create a new trusted location, click **Add new location**. The **Microsoft Office Trusted Location** dialog box is displayed.
4. Click **Browse** and select a folder to be added a trusted location in the Trust Center.
5. Click **OK** to close the **Trust Center** Window.
6. Click **OK** to close the **Access Options** dialog box.

Alternatively, the `.accdb` file can be copied or moved to the trusted location using **Windows Explorer**. Access 2010 database files that are placed in a trusted location ensure that all VBA codes, macros, and expressions are executed on opening that database.

6.5 Publishing a Web Database

Access 2010 supports importing and linking of data to large databases. Large databases provide high level of security and are managed by database administrators. This can however become a drawback for the user, as they always need to depend on administrators for performing even simple tasks. To overcome this drawback and make it easier for users and IT departments to work with Access 2010 database, Microsoft introduced tools that allow users to publish a database on the Web. This means a Web database is created and stored on the server for users.

Concepts

Microsoft provides a platform named SharePoint that is used for publishing Web databases created in Access 2010. The database published on SharePoint platform can be shared between the users. For example, a database designer can access data from SharePoint site for various database objects, such as forms, or reports. Similarly, database users can access data from SharePoint site to view and analyze it.

Access 2010 database allows sharing, managing, and updating data from SharePoint site.

6.5.1 Creating an Access 2010 Web Database

Publishing an Access Web database to SharePoint site is a multistep process which includes wizards and signing into the Web site for completing the step list.

Consider the scenario of **ToyHeaven**, where the company decides to create a Web database displaying `Product` details. The Web site will be published further on SharePoint site for its users.

The steps to create an Access 2010 Web database are as follows:

1. Start **Microsoft Access 2010**. This displays **Microsoft office Backstage** view.
2. In the **Available Templates**, click **Blank Web database**.
3. Type `ToyHeavenWebDatabase` in the **File Name** box and click **Create**. This creates a Web database with a table named `Table1`. By default, table is opened in the **Datasheet** view.
4. Rename the table as `Product`.

Table 6.2 shows the records of the `Product` table.

Prod_ID	Prod_Name	Prod_Category	Prod_Price	Prod_Stock
P0001	Hotwheels	Car	400	20
P0002	Blocks	Construction	200	100
P0003	Barbie	Doll	500	200
P0004	Silly Putty	Creative	200	250
P0005	Lego Mindstorms	Educational	500	100
P0006	Speak and Spell	Educational	200	150
P0007	Capsela	Construction	250	100
P0008	Construx	Construction	300	150
P0009	Dinky Cars	Car	100	200
P0010	Fulla	Doll	250	100
P0011	G.I. Joe	Action Figure	100	200
P0012	Jumping Jack	Action Figure	120	150

Table 6.2: Product Table Records

To publish the `ToyHeavenWebDatabase` to SharePoint server 2010, the steps are as follows:

1. On the **File** tab, click **Save & Publish**.
2. On the right pane under **Publish**, click **Publish to Access Services**. This displays **Access Services Overview** pane. The **Access Services Overview** pane provides options namely, **Check Web compatibility** and **Publish to Access Services**.

Figure 6.25 shows the **Publish to Access Services** window.

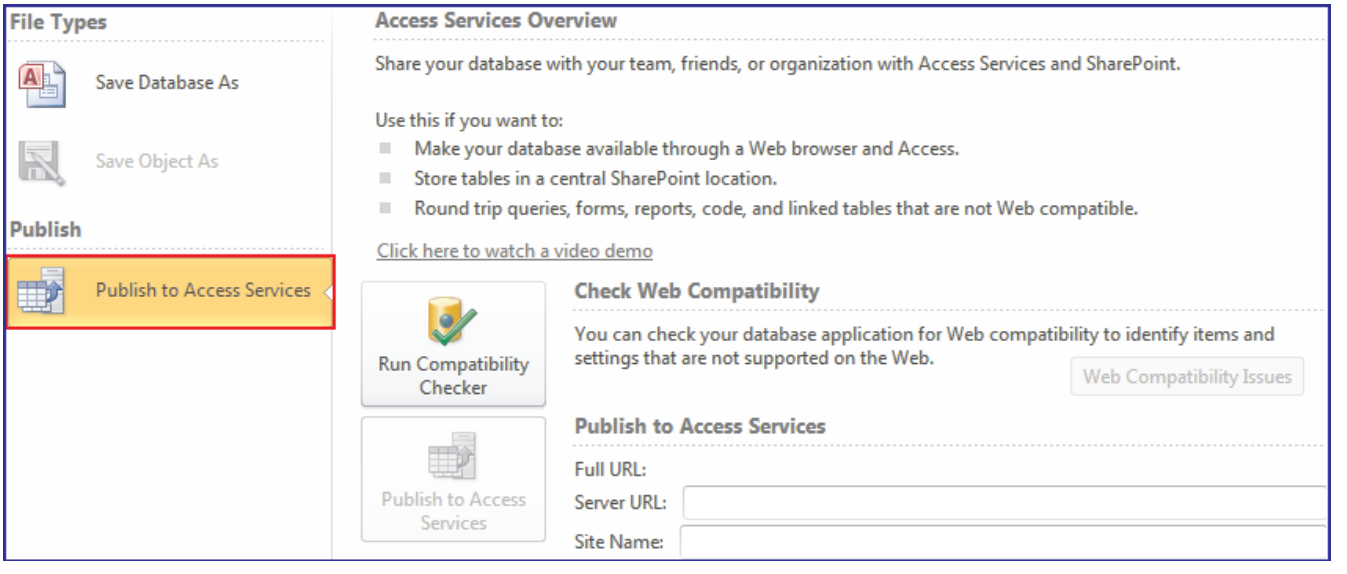


Figure 6.25: Publish to Access Services

Concepts

3. Click **Check Web compatibility** to check whether the database items are compatible with the Access Services. If any error occurs, it notifies those errors by displaying the **Web Compatibility Issues** tab.
4. Type the URL, `http://aptechaccess2010.sharepoint.com` in the **Server URL** box. This is the domain name of the SharePoint site.
5. Type `ToyHeavenSite` in the **Site Name** box. This is the name by which the database is published on SharePoint site.
6. Click **Publish to Access Services**.
7. SharePoint prompts for the user credentials. Type the username and password to access the site on the SharePoint server.
8. Click **OK**. Access 2010 publishes the database on the specified URL and displays a '`Published Succeeded Message`' to the user.
9. Open Web browser and type `http://aptechaccess2010.sharepoint.com/ToyHeavenSite` in the **Address** bar. It opens the Web database in the browser.

Figure 6.26 shows the Web database displayed in the browser.

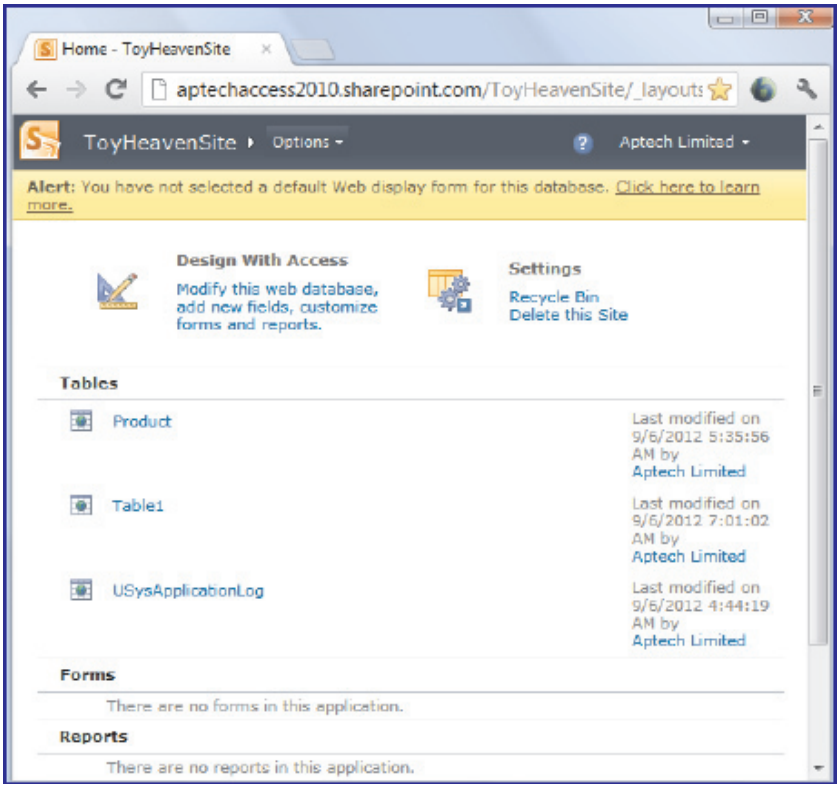


Figure 6.26: Web Database Displayed in Browser

10. Click **Product** and select **Modify Product**. This opens the database in Access 2010. Any changes made in the `Product` table are synchronized with the data stored on the server.

Note - To publish Access 2010 Web database on SharePoint Server, the users need to register at `http://www.microsoft.com/en-us/office365/sharepoint-online.aspx` Web site. To register for SharePoint Service, click **FREE TRIAL** on the main page. This displays a free 30 days subscription page. Click **FREE TRIAL** under **Small business (Plan P1)** which prompts for creating a trial account. The users need to fill a form to create username and password that are used to access the SharePoint site. It also creates SharePoint Web site address on which the database must be published. The information regarding username, password, and site address is notified through an e-mail message to users.

6.6 Check Your Progress

1. Match the terms with their corresponding descriptions.

Term		Description	
(a)	Encryption	1.	Publishes Web databases created in Access 2010
(b)	Decryption	2.	Decodes the encrypted password information
(c)	Macro	3.	Encrypts the password and makes the data unreadable
(d)	SharePoint	4.	Allows users to group series of actions

(A)	a-3, b-2, c-4, d-1	(C)	a-4, b-3, c-2, d-1
(B)	a-2, b-3, c-4, d-1	(D)	a-1, b-2, c-3, d-4

2. Which of the following options can be bound to a control through its event?

(A)	Standalone Macro	(C)	Embedded Macro
(B)	Expression Macro	(D)	Control Macro

3. Which of the following tool checks whether the location is trusted and runs the database with all the functionalities?

(A)	Encryption	(C)	Decryption
(B)	Database Security	(D)	Trust Center

4. _____ displays node lists of all the macros created in a database in the action catalog pane.

(A)	Program Flow	(C)	Statements
(B)	In this Database	(D)	Actions

Concepts

5. _____ means to transfer the data from one database to another database or program.

(A)	Importing	(C)	Exporting
(B)	Copying	(D)	Transferring

6.6.1 Answers

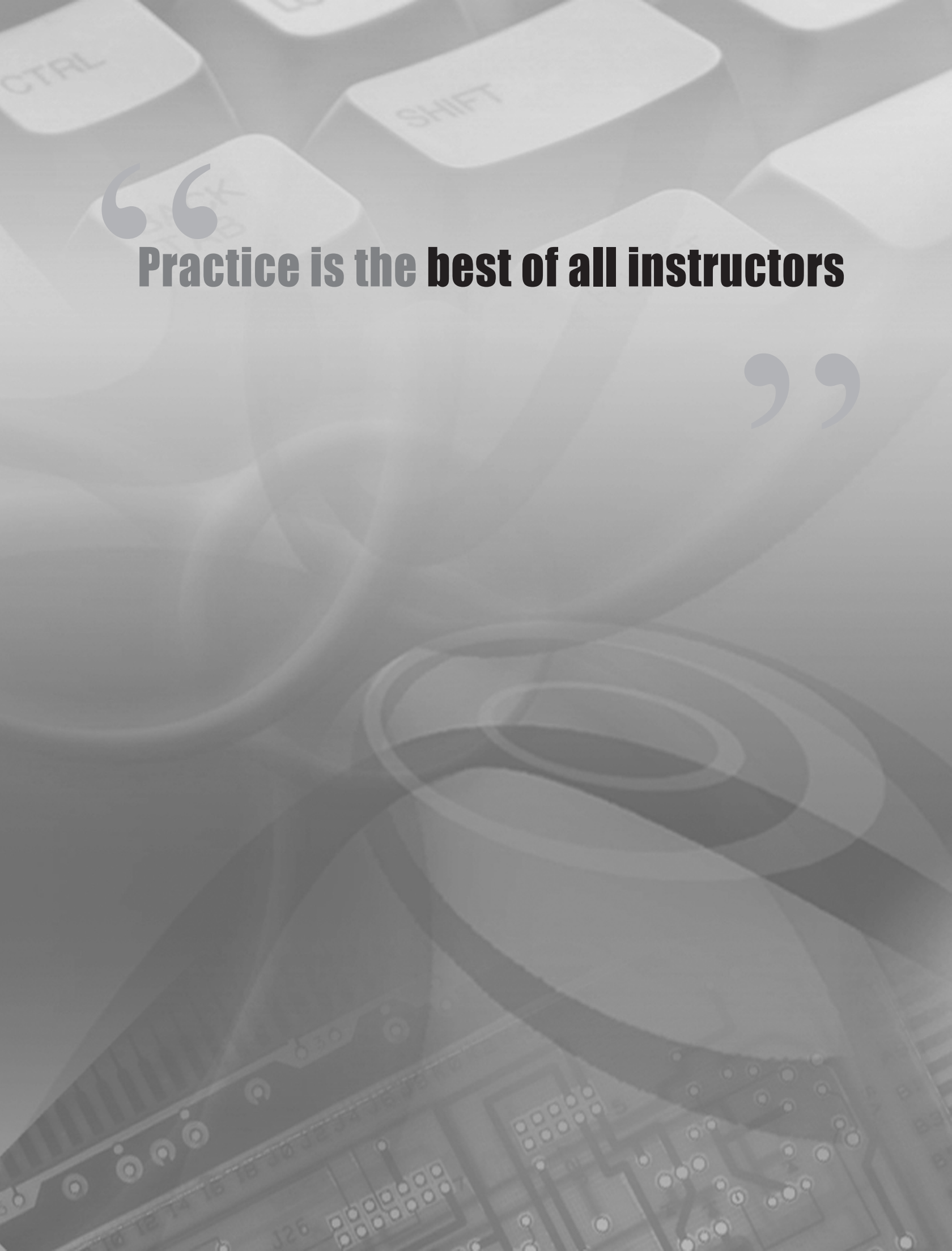
1.	A
2.	A
3.	D
4.	B
5.	C

Summary

- A macro is a tool that allows users to group series of actions related to a specific database task.
- A macro automates one or more actions and consists of macro actions, conditions, and arguments.
- Macros are of two types namely, standalone and embedded.
- Access 2010 provides a new Macro Designer Window that helps to create complex macros with reduced errors.
- Exporting data means to transfer the data from one database to another database or program.
- Importing data means to collect data from another database or program.
- Access 2010 provides a security model that supports encryption tool to prevent unauthorized access to the data in a database.
- Microsoft provides a platform named SharePoint for publishing Web databases using Access 2010.

Try it Yourself

1. Design a form and create a macro that allows a user to search for a record on the form. If the matched record is found, it should display all the other details in the form controls. Otherwise, display an error message to the user.
2. Export the table created in the `ToyHeaven` database to the format which can be opened in the acrobat reader program. Similarly, import it back to Access 2010 database.


Practice is the best of all instructors

Session - 6 (Workshop)

Advanced Database Operations

In this workshop, you will learn to:

- ➔ Create a macro in Microsoft Access 2010 and use Visual Basic for Applications
- ➔ Create a data macro

6.1 Working with Macros

You will view and practice how to use macros and VBA in Access 2010.

➔ Creating a macro

➔ Write VBA code on events

➔ Create a data macro to fire after the table data is modified

Note - Please refer to the respective lab deliverable for demonstrations and guided simulations.

Appendix

Specifications for MS Access 2010

The following specifications apply to databases:

Attribute	Value
Maximum size for an Access 2010 database (.accdb), including all database objects and data	2 GigaBytes (GB) minus the space needed for system objects
Maximum number of objects in a database	32,768
Maximum number of modules	1,000
Maximum characters in an object name	64
Maximum characters in a password	14
Maximum number of concurrent users	255

The following specifications apply to tables:

Attribute	Value
Maximum number of characters in a table name	64
Maximum characters in a field name	64
Maximum fields in a table	255
Maximum table size	2 GB minus the space needed for the system objects
Maximum number of characters in a Text field	255
Number of characters in a validation rule including punctuations and operators	2,048
Maximum number of characters in a field or table description	255
Maximum number of characters in a field property setting	255

The following specifications apply to queries:

Attribute	Value
Maximum number of enforced relationships	32 per table, minus the number of indexes that are on the table for fields or combinations of fields that are not involved in relationships
Maximum number of tables in a query	32 (Can be lower if the query includes multivalued lookup fields)

Attribute	Value
Maximum number of joins in a query	16 (Can be lower if the query includes multivalued lookup fields)
Maximum number of fields in a recordset	255
Sort limit	255 characters in one or more fields
Number of AND operators in a WHERE or HAVING clause	99 (Can be lower if the query includes multivalued lookup fields)
Maximum number of characters in an SQL statement	Approximately 64,000 (Can be lower if the query includes multivalued lookup fields)

The following specifications apply to forms and reports:

Attribute	Value
Number of characters in a label	2,048
Number of characters in a text box	65,535
Form or report width	22,75 inches
Section height	22.75 inches
Number of levels of nested forms or reports	7
Number of fields or expressions that you can sort or group on in a report	10
Number of headers and footers in a report	1 report header or footer 1 page header/footer; 10 group headers/footers
Number of printed pages in a report	65,536

“You must do the things
you think you cannot do”