

Microsoft Access 2013

Lab Practicals

Elbeshir Elgalal September 2015

**Recommended Books:**

1) Microsoft Office Access 2013 inside Out Jeff Conrad and John L. Viescas Microsoft Press, 2013.

2) Microsoft Office Access 2010 Inside Out; Jeff Conrad and John L. Viescas

Microsoft Press, 2010.

**Contents**

**Page**

**Practical 1:** Access 2013 New Features 3

1.1 Starting and Quitting Microsoft Access 2013 4

1.2 Getting Help 6

1.3 Exercises using a Sample Database 6

**Practical 2:** Building an Access 2013 Application 17

2.1 Introduction 18

2.2 Implementing the Company Database Schema 19

2.2.1 Creating a New database 23

2.2.2 Creating Tables 25

**Practical 3:** Adding Data to Tables and Setting up Relationships 33

3.1 Adding data to Department and Employee Tables 34

3.2 Setting up relationships 35

3.3 Lookup Columns and Drop down List of values 40

**Practical 4:** Creating Forms 47

4.1 Introduction 48

4.2 Creating a Single table Form using Quick Create Command 48

4.3 Creating a Single table Form Using the Form Wizard 53

4.4 Modifying a Form's structure 56

4.5 Creating a Multi-table Form 65

4.6 Using Forms 68

**Practical 5:** Creating Queries 79

5.1 Introduction 80

5.2 Using QBE Tool 81

5.3 Write a Query Using SQL 86

5.4 Queries for the Company Database 87

5.5 Queries for ClydeRental Database 88

**Practical 6:** Creating Reports 91

6.1 Introduction 92

6.2 Grouping, Sorting and Totalling Information 95

6.3 Using Report Wizard with a Single table 98

6.4 Using Report Wizard with Many tables 102

**Appendixes** 105

Appendix2.1 Fields Data Types and Constraints 107

Appendix3.1 Sample Data 111

**Practical 1**

**Access 2013 Features**

Before you start MS Access open your home drive folder (H: drive) and create a folder. Call the folder 'Database Practicals'. Inside the folder create 6 subfolders. One subfolder for each practical (Practical\_1, Practical\_2, etc.)

**1.1 Starting and Quitting Microsoft Access 2013**

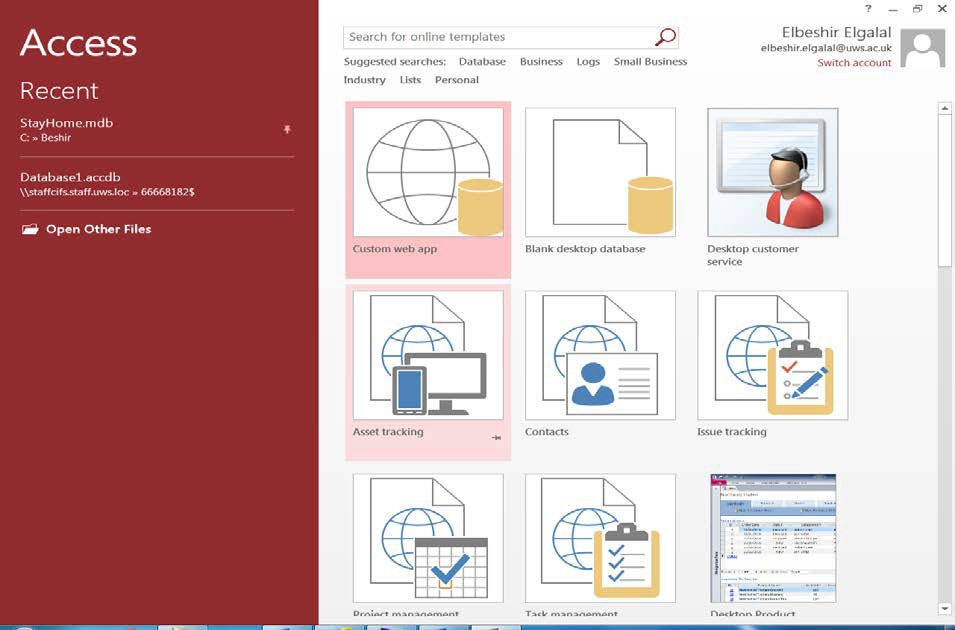
**To start Microsoft Access 2013**

Click on Windows **Start** button; select **Programs**, then **Microsoft Office** and then select

**Microsoft Office Access 2013** from the drop down menu.

**Start**  Programs  Microsoft office  Microsoft Office Access 2013.

The first time you open Office Access 2013 in the labs you will see the screen in Figure 1.1.



**Figure 1.1 Microsoft Office Backstage View**

The first pane contains a list of the most recently opened database names and the second pane contains a list of templates that be used to create the named application database. You can use the first button in the first row to create a new database (see Practical 2).

Click **Open Other Files** and the following window will open.

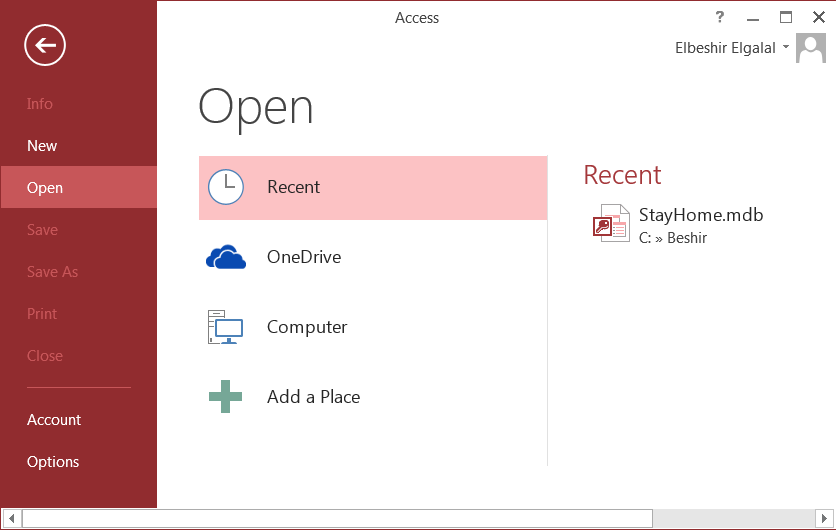


Figure 1.2

In the first pane on left-hand side there are a number buttons but only few of them are available

(dimmed). The buttons have the following functionality:

♦ **Info –** give information about an opened database

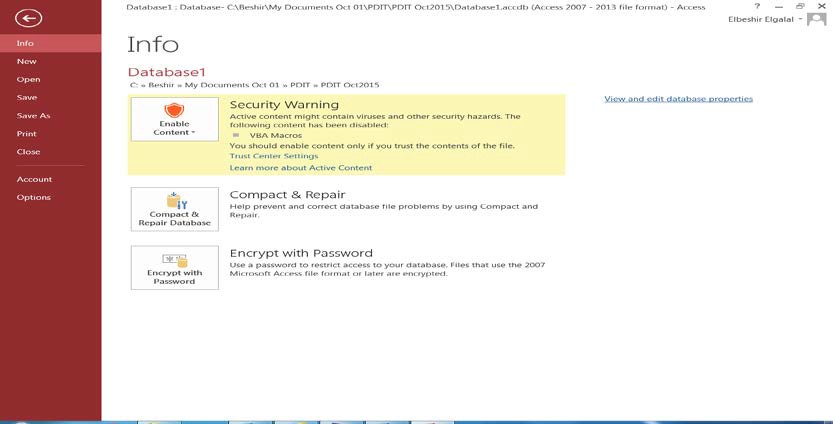


Figure 1.3

♦ **New –** for creating a **New database** application. Click **New** button to open the list of buttons on the second pane in Figure 1.1.

♦ **Open –** Opens the window in Figure 1.2 and allow user to open any of the previously opened databases appearing in the list or click '**Computer'** button and navigate to the desired folder storing the required database to be opened.

♦ **Save** – save design changes to a database object that is currently open and has the focus in the Navigation pane.

♦ **Save As** – save a copy of the current open object that has the focus or the object that have the focus in the Navigation pane.

♦ **Print –** opens the following windows with three print options to print the required object.

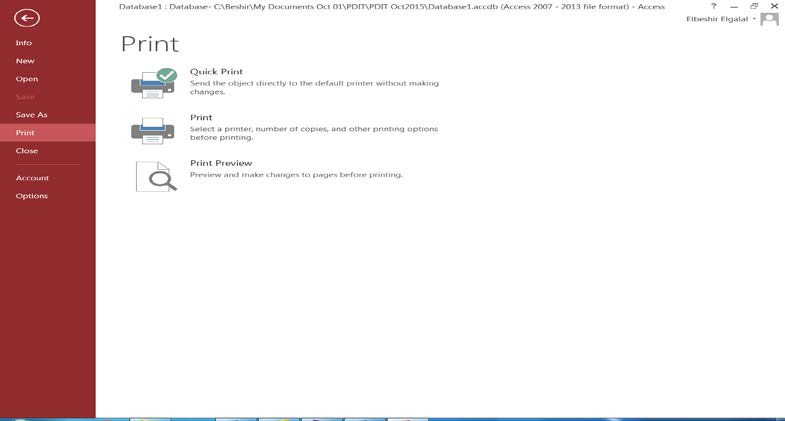


Figure 1.4

♦ **Close** – close the currently open database and return to the database main window.

♦ **Account** – show the current user account details.

♦ **Options** – Open the following window show the different options that can be set for the currently open database.

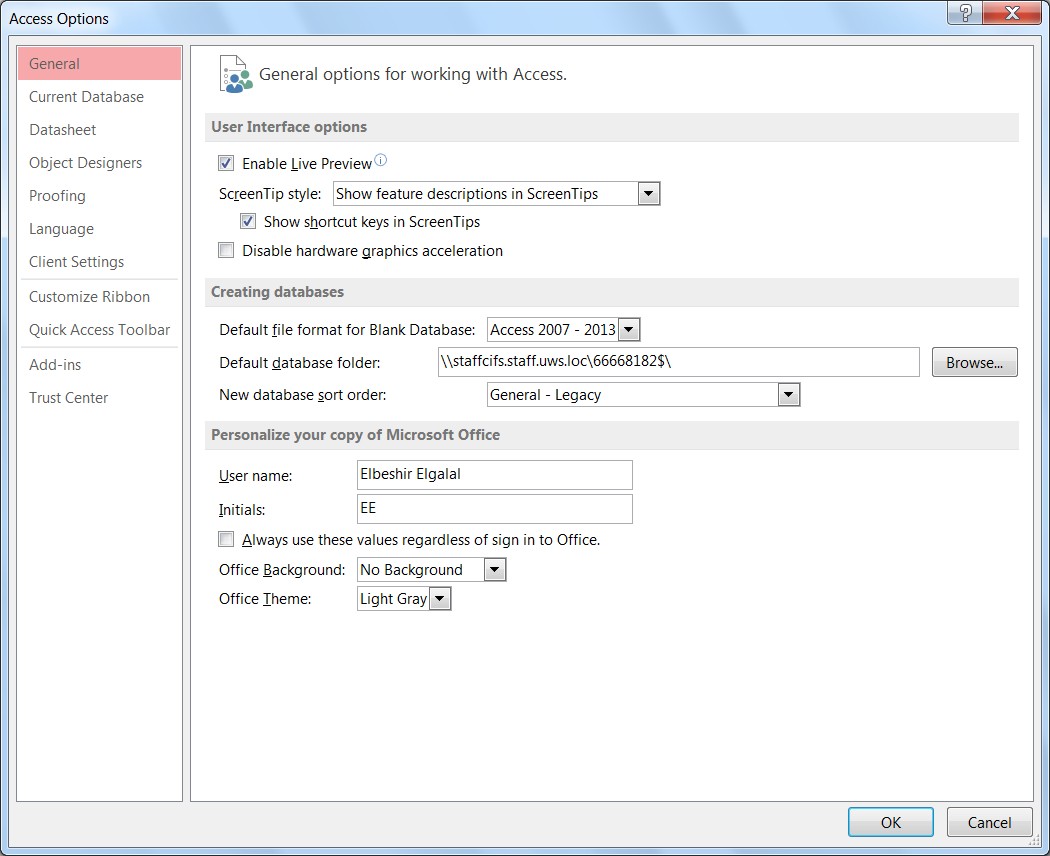


Figure 1.5

In second pane under **Open**

♦ **Recent -** displays the names of the recently opened databases;

♦ **OneDrive –** allows you to access any stored files on the Cloud;

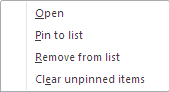
♦ **Computer –** displays a list of links to most recently opened folders and a **Browse** button to navigate to any drive of folders on your PC;

♦ **Add a Place –** allows you to add locations to make it easier to save Office documents on the cloud.

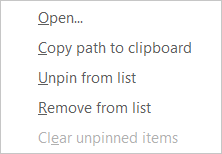
**Additional Options to Manage the List of Recent Database or folders**

A **pushpin button** is displayed to the right of each database file name or folder path. Click this button if you want to pin that specific database file of folder path to the list of displayed items (folders details appear when you click **Computer** button).

Right-click any recent database and Access displays a drop-down menu with four options.



OR



In first case the item is pinned to the list of items but in the latter it is not.

**1.2 Getting Help**

To access the **Help** system, click the question mark button  on the top-right corner of the

Ribbon menu bar.

If you are not connected to the internet you will see only the content of the Help system which is stored on your system when Microsoft Access 2013 is installed. More Help information is made available from Microsoft Online Office system if you are connected to the internet.

**1.3 Exercises Using a Sample Database**

Copy the sample database ‘StayHome.mdb’ from **Moodle** to your **H: drive** folder and open it (see figure 1.6) using Microsoft Access 2013. The database is stored in Module Materials section in the folder:

Module Materials  Databases  StayHome.mdb

**To copy the database file:** right-click the file and select 'Save Link As ....'. Navigate to the folder you want to save the file and save it.

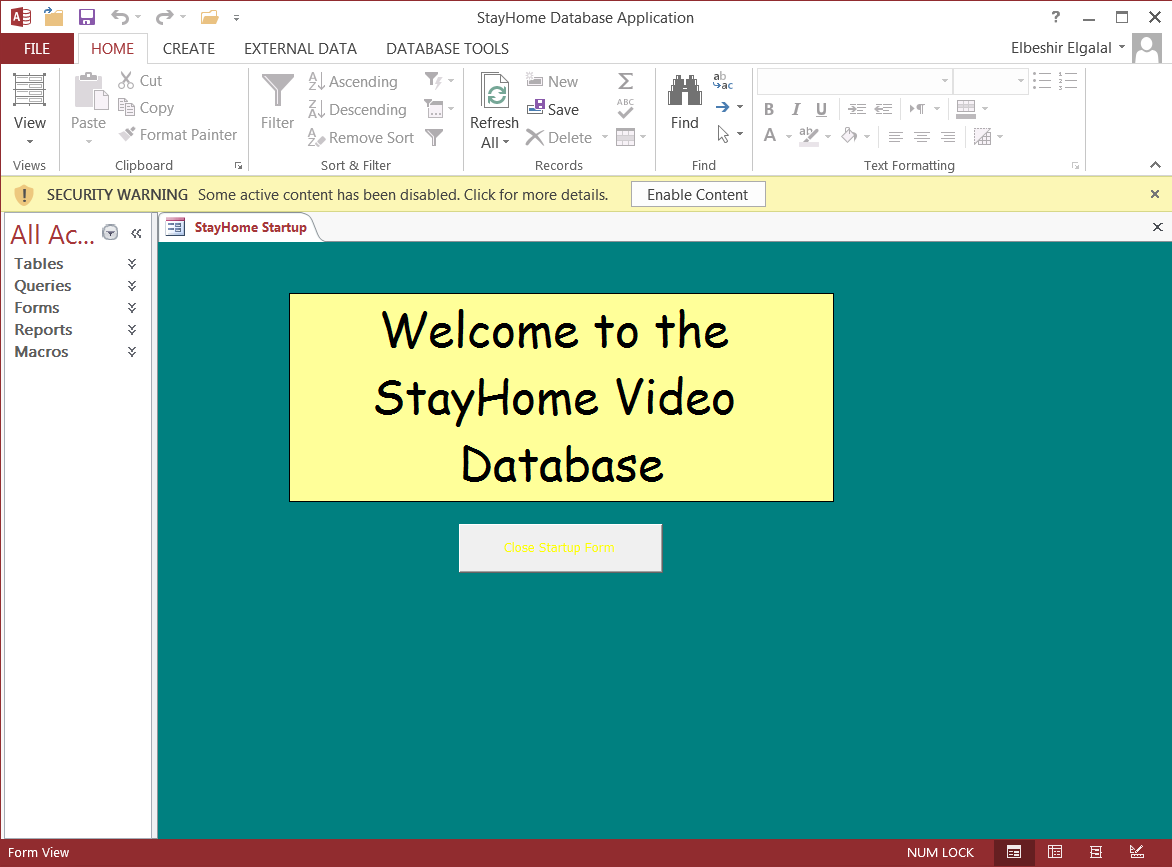


Figure 1.6

The database window consists of a number of user interface elements. The top-left corner of the screen contains Access 2013 icon button  (see figure1.6) followed by a few buttons on the

**Quick Access Toolbar**. The toolbar hold the frequently used commands within Access 2013. You can customize the toolbar by adding or removing command buttons. Beneath the Quick Access Toolbar is the **Ribbon** which consists of a series of tabs that contain many commands, options and drop-down list boxes. The Ribbon replaces menus and toolbars from previous versions of Access. Most of the commands required for developing or using Access 2013 databases are available on the Ribbon and as such you will interact heavily with the Ribbon when developing or using Access 2013 databases.

Beneath the Ribbon is the **Message Bar** with a security warning message. This message appears when you open the database for the first time. Click the message for more information

on security. Consult Help for more information on Access 2013 security model. Type: ‘*Security in*

*Access 2013’* in Help search box. Click **Enable Content** to get rid of the message.

On the left side of the screen (under the Ribbon) is the **Navigation Pane**. The Navigation pane replaces the database window from previous MS Access versions. The various database objects (i.e. tables, forms, queries, reports, etc.) are displayed in the Navigation pane. To minimize the Navigation pane click the **Shutter Bar** Open/Close Button (the double-arrow button <<) at the top right corner of the pane.

To the right of the Navigation pane is the **object display** window. In this window database objects are displayed (a start-up form is displayed in Figure 1.6). Beneath the navigation pane and object display window is the **Status bar**. The status bar displays text messages and different command button relative to the current open object.

A list of objects appears in the Navigation Pane under the *menu bar* with the heading **All Access Objects**. Click the menu bar or the downward arrow head at the end the menu bar to open the Navigation Pane menu. The menu list (see Figure 1.7) show a number of titles grouped

under two headings: **Navigate to Category** and **Filter By Group.**

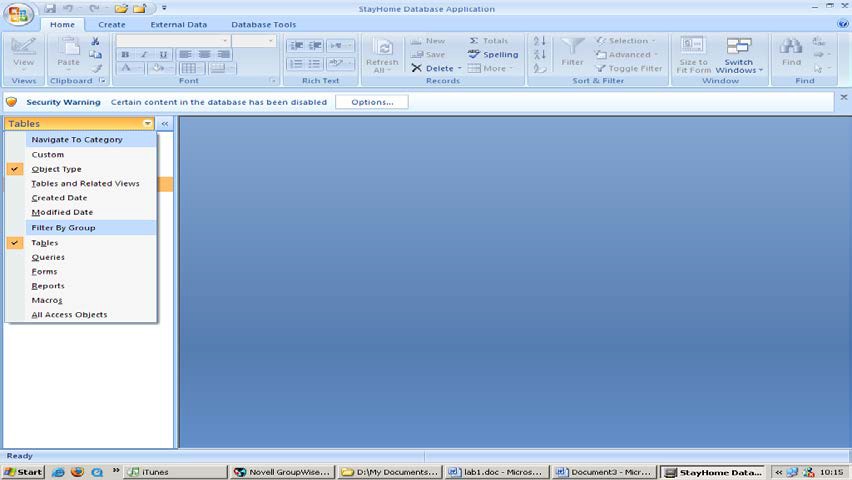


Figure 1.7

**Exercise 1.1**

You can display the database objects in different format in the Navigation pane.

♦ Select ‘Object Type’ under Navigate to Category and ‘Tables’ under Filter by Group and observe what happens in the Navigation Pane.

♦ While ‘Object Type’ is still selected, select ‘Queries’, ‘Forms’ etc. and see what happens in the Navigation Pane.

When **All Access Objects** under Filter By Group is selected all the database objects are shown in the Navigation Pane grouped under the titles: Tables, Queries, Forms, Reports and Macros. Each of these groups can be collapsed or expanded by clicking the header of each object or the double arrows to the right of the text (two arrow heads pointing upward or downward).

**Exercise 1.2**

♦ Select ‘Tables and Related Views’. All tables in the database will be displayed under

**Filter By Group**. Select each table in turn and see what happens in the Navigation Pane.

When ‘Tables and Related Views’ category is selected and a table name is selected from the list under Filter By Group, the navigation Pane displays the table name followed by ‘:’ and the word ‘Table’, followed by objects that uses this table i.e. forms, queries, reports, etc.

When ‘**Unrelated Objects’** is selected Access displays all objects that does not use any of the tables in the database.

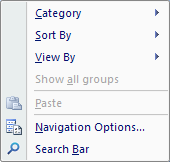
♦ Select ‘All tables’ and see what happens in the Navigation Pane.

**Exercise 1.3**

♦ Select ‘Object Type’ and ‘All Access Objects’ in the Navigation Pane menu and experiment with opening and closing the different objects that are displayed in the navigation Pane.

**Using Drop down menu**

Right-click the menu bar at the top of the Navigation Pane and the following drop-down menu will open.



**Exercise 1.4**

♦ **Category** - press and select each category in turn see what happens to the object shortcuts in the Navigation pane.

♦ **Sort by** - sort the object shortcuts in ascending and descending order by: Name, Type, etc.

♦ **View by** – view object shortcuts by: Details, Icon and List.

♦ **Search Bar** – tsearch for a database object shortcut. Select **Search Bar** and type the shortcut of the object name. Note that Access will return all available shortcuts titles that contain the string of characters you typed.

♦ **Navigation Options** – click the link and see available options in the window that opens

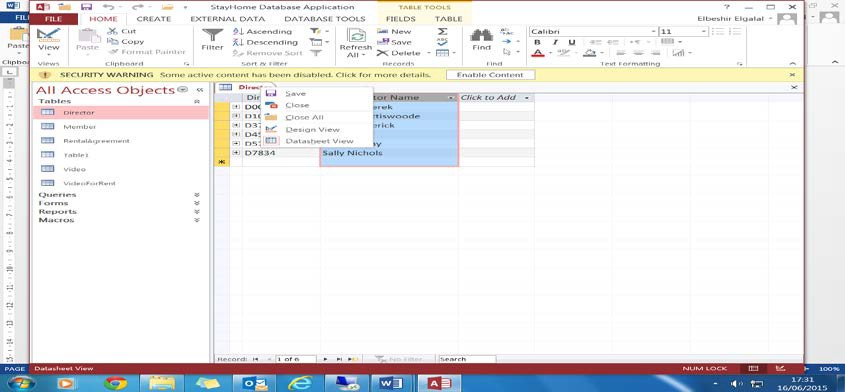
**Opening and Closing Objects**

Objects are opened in the objects' windows on the right-hand side of the Navigation pane. An object can be opened by double-clicking the object shortcut on the Navigation Pane or from the drop-down menu (right-click the shortcut and click open). Objects can be opened in different views. Tables and Queries can be opened in Datasheet or Design view. Forms can be opened in Form, Layout and Design view. Reports can be opened in Print, Layout and Design view. When an object is opened in whichever view the applicable commands on the Ribbon (under Home tab) becomes active and can be used with the open object.

To close an object click the **‘x’** at the top right-hand of the object display window or place the cursor on the object name on the top part of the opened object display window (right-click and select **Close** from the drop-down menu). The drop-down menu shows all the possible views in which the current object can be opened, as well as, **Save**, **Close** and **Close All** if more than one object is open (see Figure 1.8).

Close Object

**Figure 1.8**



**Exercise 1.5**

♦ Click **All Access Objects** menu bar and select ‘Object Type’ under Navigate To Category and ‘All Access Objects’ under 'Filter by Group' to see object shortcuts of all available groups (i.e. tables, forms, queries, etc.).

♦ Open one or more objects by double clicking the object shortcut. The object will open in the standard view (Tables in Datasheet view; Forms in Form view, etc.). Open in a different view using the **View** button on the Ribbon or right-click the name of the opened object in the top part of the object display window and use the option in the drop-down menu.

♦ Minimize, maximize or close the database window using the window commands at the top right corner.

**Exercise 1.6**

♦ Click ‘File’ on the Ribbon and then click **Options**. Access Options window will open (see Figure 1.9). The options apply to the whole database. The selected ones are the default. Click each title in the left-hand side pane to see the relevant options.



Figure 1.9

**The Quick Access Toolbar**

The Quick Access Toolbar is by default displayed above the Ribbon in the left-hand corner. It can also be displayed below the Ribbon (Right-click the toolbar and chose the required option from the drop-down menu).

The Quick Access Toolbar displays the user’s most frequently used command buttons. This allows the user to gain quick access to the more common commands of Access 2013. Only four command buttons (Open, Save, Undo and Redo) are included in the toolbar by default. The toolbar can be customized by including additional command buttons or removing exiting ones.

Click the small arrow at the right end of the Quick Access Toolbar and the **Customize Quick Access Toolbar** menu will be displayed as show in Figure 1.6. You can add or remove a command to/from the Quick Access Toolbar by clicking the command on the 'Customize Quick Access Toolbar' menu. To add more commands than those available on the Customize Quick Access Toolbar menu click the downward arrow head under 'Choose commands from' and click

‘ALL Commands’.

**Exercise 1.7**

♦ Add the commands 'Open' and ‘Close Database’ and any other commands of your choice to the Quick Access Toolbar and order them (from left to right) so that ‘Open‘ command is first, followed by ‘Save’, followed by ‘Close Database’, etc.

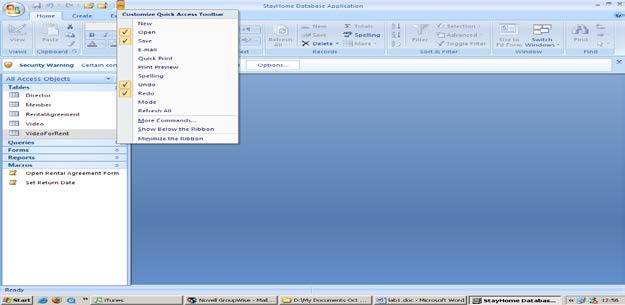


Figure 1.10

**The Ribbon Feature**

One of the biggest changes to the new user interface in Access 2007- 2013 is the Ribbon. The

Ribbon (Figure 1.11) replaces menu bars and toolbars in previous MS Access versions.

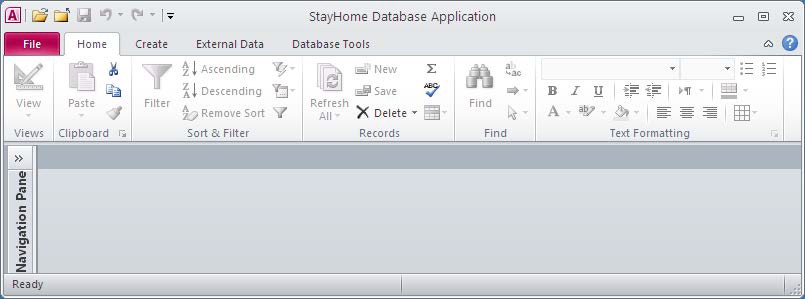


Figure 1.11

The Ribbon is context-rich environment displaying all the programs functions and commands, with large icons for key functions and smaller icons for less-used functions. The Ribbon consists of four main tabs: **Home**, **Create**, **External Data**, and **Database Tools**. The four main tabs are visible at all times when you are working with Access 2013 because they contain the most common tools you need when working with any database object. Other tabs, called *contextual tabs*, appear and disappear to the right of the 'Database Tools' tab when working with a specific database objects and in various views.

Each tab has commands that are further organized into groups. The name of each group is listed at the bottom, and each group has various commands logically grouped by subject matter. To make thing easy to find each command is labelled. If you rest the mouse pointer on a specific command, Access displays a *ScreenTip* that contains the name of the command and a short description that explains what you can do with the command. If a command includes a small downward arrow head click the arrow head to display options available for the command.

**The Ribbon Tabs**

**Home Tab**

The Home tab has the following groups:

♦ **Views -** Most objects in Access can be viewed in two or more ways. When an object is open it has the focus, you can use the **View** command to switch to another view.

♦ **Clipboard -** Use the commands in this group to move data to and from Clipboard (an area in memory). Click the downward arrow to see the item in the Clipboard.

♦ **Sort & Filter** - Use commands in this group to sort and filter data.

♦ **Records** - use commands in this group to work with records including deleting and saving changes.

♦ **Find** - use commands in this group to search and replace data, go to a specific record or select one or all records.

♦ **Window –** use commands in this group to resize windows or select one of several windows that you have open. Note that Access displays this group only when you have set your database to display Overlapping Windows rather than Tabbed Documents.

♦ **Text Formatting –** use command in this group to change how Access displays text.

You can also design fields in your database to contain data formatted in Rich Text.

**Create Tab**

♦ The tab contains commands that allow you to create new database objects. The tab contains groups with commands to create: Templates, Tables, Queries, Forms, Reports, and Macros &Code.

♦ You can use macros or modules to automate your applications (Search Help for

Macros and Modules)

**External Data tab**

The tab provides commands to import from or link to data in external sources or export data to external sources. The tab has the following groups:

♦ **Import & Link** - commands in this group let you link to data or import data or objects from other Access databases, Microsoft Excel spreadsheets, Windows SharePoint Services (see chap 8 in reference 1, see Help on the subject), and other sources.

♦ **Export -** commands in this group let you export objects to another Access database or export data to Excel, Word, Windows SharePoint Services, and other applications.

**Database Tools Tab**

The database Tools tab will show slightly different icons when using Access 2013 database (.accdb) and when using earlier versions of Access (i.e. 2000, 2002, and 2003) databases (.mdb). The database Tool tab includes the following groups:

♦ **Tools** – Compact or Repair Database

♦ **Macro** - commands in this group let you open Visual Basic editor, run a macro or convert a macro.

♦ **Relationships** - use the Relationships command to view and edit your table relationships.

♦ **Analyse** - use the commands in this group to print a report about a database object or run one of the two analysis wizards.

♦ **Move Data** – wizards in this group allow you to move some or all tables to SQL Server or move all tables to another Access database and create links to the moved tables in the current database.

♦ **Administer** – This group is displayed on the Database Tools tab only when you open an Access database file created in Access 2000, 2002, or 2003 (.mdb). The Switchboard Manager command starts the Switchboard Manager to assist you with building a switchboard form for navigating through your application.

**Collapse and Restore Ribbon**

Three options:

♦ Double click on any of the tabs. All groups disappear from the screen but the tabs are still available. To restore the Ribbon to it full size, simply click on any tab.

♦ Press **Ctrl+F1** to restore Ribbon to full size.

♦ Put the mouse pointer anywhere on the Ribbon click the mouse right-button and select **Minimize the Ribbon** option**.** To restore right-click any tab to open the drop-down menu of the minimized ribbon and select Minimize the Ribbon option.

**Practical 2**

**Building an Access 2013 Application**

**2.1 Introduction**

In this practical you will learn to create a relational database application using Microsoft Access

2013 Database Management System (MS DBMS). The application is part of a Company database and consists of 6 tables, namely: Department, Employee, Project, Dependent, worksOn and Qualifications. Figure 2.1 shows the conceptual schema of the database application which is followed by the Relational schema that is derived from it.

Controls

0..\*

Supervise

Supervisee

Supervisor

0..1

0..\*

Employee

nin {PK}

name fName lName

Employs

1..\* 1..1

Manages

1..1

Department

deptNumber {PK}

deptName {AK}

deptPhone {AK}

Project

projNumber {PK} projName {AK} budget

startDate finishDate

WorksOn

0..\* 1..\*

hours

address

town street postcode

qualifications[1..\*]

sex birthDate salary

1..1 0..1

startDate

Has

Dependent

firstName {PPK}

sex

/age

1..1

0..\*

birthDate

relationship

Figure 2.1 Company Database Conceptual Schema (An ER diagram)

**The Company Relational Database Schema**

**Department** {deptNumber, deptName, deptPhone, mgrNin, mgrStartDate} Primary key: deptNumber

Foreign key: mgrNin references Employee (nin) Alternate keys: deptName, deptPhone

**Employee** {nin, fName, lName, town, street, postCode, sex, birthDate, salary, superNIN, deptNumber}

Primary key: nin

Foreign key: superNin references Employee (nin)

Foreign key: deptNumber references Department (deptNumber)

**Project** {projNumber, projName, budget, startDate, finishDate, deptNumber} Primary key: projNumber

Foreign key: deptNumber references Department (deptNumber) Alternate key: projName

**Dependent** {empNin, fName, birthDate, sex, relationship} Primary key: empNin, fName

Foreign key: empNin references Employee (nin)

**WorksOn** { nin, projNumber, hours} Primary key: nin, projNumber

Foreign key: nin references Employee (nin)

Foreign key: projNumber references Project (projNumber)

**EmpQualifications** {nin, qualification} Primary key: nin, qualification

Foreign key: nin references Employee (nin)

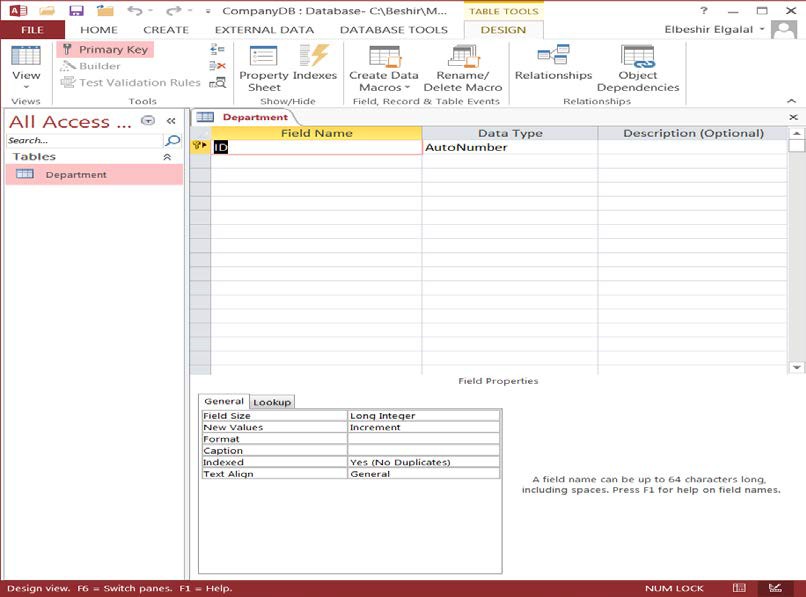
**2.2 Implementing the Company Database Schema**

To implement the above schema you must create a table for each relation. To create a table we must describe to MS Access the structure of each table, in particular, the column name, data type and size and the constraints that should hold on each column. The constraints are concerned with maintaining the consistency of the data in the table. MS Access provides a template that allows the user to define the table’s fields when defining the table in **design view.** The template consists of two sections as shown in Figure 2.2. The upper section of the template allows the user to specify the field name, data type and a description. The description field is optional. The field is used to type any general comments about the field. The comments are not stored in the table

The bottom section of the window specifies the field properties (constraint) that has to be enforced by the database engine. When the user clicks on the name or the data type of the field the properties of the field appear in the bottom section of the window. Note that the field properties that appear differ from field to another depending on the field’s Data Type.

The next two sections briefly describe the most used *data types* and field properties. For more details consult the **Access Help** and the recommended books.

Figure 2.2 Defining a new table in design view



The key symbol shows which field (or fields) is the primary key for the table.

The Field Name identifies the data stored in a field. A field name can contain up to **64 characters**, including spaces. The Data Type tells MS Access what kind of data goes in the field, such as text, numbers, dates or currency.

**Data Types**

A brief description of the most commonly used data types is given below:

♦ The **Text** data type is used for words (such as names), for combinations of words and numbers (such as addresses), and for numbers that are not used in mathematical calculations (such as telephone numbers). Up to 255 characters.

♦ The **AutoNumber** data type stores sequential integer numbers automatically generated by MS Access. You cannot enter or change the data in this type of field at any time. Access increments the values in an AutoNumber field whenever you add a new record to a table.

♦ The **Date/Time** data type is used for dates and times.

♦ The **Number** data type is used for numeric data that can be used in **mathematical** calculations, and also for fields that are related to other Number fields or to AutoNumber fields.

♦ The **Currency** data type is used for currency values such as pounds, dollars, or yen. You do not have to manually enter a currency symbol, by default Access applies the currency symbol (£, $, ¥, and so on) specified in your Windows regional settings.

♦ The **Yes/No** data type is used to store ‘yes’ or ‘no’ ‘true’ or ‘false’ values (Boolean data).

Access stores the numeric value zero (0) for ‘false’ and -1 for ‘true’. When you click a field that is set to this data type, Access displays either a check box or a drop-down list, depending on how you format the field. If you format the field to show a list, you can select either **Yes** or **No**, **True** or **False**, or **On** or **Off** from the list, again depending on the format applied to the field. You cannot enter values in the list or change the values in the list directly from a form or table.

♦ The **Memo** data type is used to store long textual descriptions up to a maximum size of

64,000 characters. Memo fields cannot be indexed.

♦ The **OLE Object** data type is an object (such as an Excel spreadsheet, a Word document, graphics, sounds, or other binary data) linked or embedded in an Access table.

♦ The **hyperlink** data type stores text or combinations of text and numbers stored as text and used as a hyperlink address.

**Field Properties**

Note that the list of properties depends on the Data Type. For example, the Data/Time data type has no field size property, whereas a Text data type does. This is because the space required to store a Date/Time field is always the same, whereas text fields can vary in length. Many properties have sensible default values, so in general we need only to change a few properties for each field. We will concentrate on the following properties:-

♦ **Field Size –** specify a text field size. Text field can be from 0 to 255 with a default length

255. This can be rather wasteful e.g. National Insurance Number field, for example, is always only eight characters long. Using excessively long fields not only wastes space, but can lead to slower processing.

♦ **Format** - this specifies how the field is to be displayed. Dates, for example, can be shown as 1/1/2001, or 1-Jan-2001, or Saturday, January 1st, 2001 and so on.

♦ **Input Mask** - specify restrictions on data format. For example, a National Insurance Number always starts with two capital letters followed by six digits. We can enforce this requirement by specifying an input mask.

♦ **Caption** - in **Datasheet View**, the column names are the field names. These might not be as easy to understand as we would like: not everyone would know that NIN is short for National Insurance Number. In certain cases it is a good idea therefore to display a short yet more self explanatory piece of text rather than the field name. This is achieved by defining the **Caption** property for the field.

♦ **Default Value** - Specifies a value that is automatically entered in a field when a new record is created. For example, if the majority of employees work in department D01 we could specify D01 as the default value for the **deptNumber** field of **Employee**. When data in entered and the user does not type a value for the field 'deptNumber', 'D01' is automatically entered in the field. In this way the user need to type only the values other than 'D01' and if the majority of employees work for department 'D01' then this reduces typing.

♦ **Validation Rule and Validation Text -** a validation Rule allows you to limit the values that can be put in a field. We can use these, for example, to ensure that sex can only be male or female, that a salary must be a positive number, and that all employees were born after

1/1/1900. The validation Text is displayed in an alert window on the screen when an attempt is made to insert a value that violates validation rule.

♦ **Required**. This is set to **yes** if we require that the field cannot be left blank**.**

♦ **Indexed**. This property is automatically set **Yes (No Duplicates)** for the Primary key field and should be set **Yes (No Duplicates)** to all **Alternate** keys (e.g. **deptName** as no two departments can have the same name). For fields which are not unique the 'indexed' property is set to '**No'** by default.

For more details on Data Types and Field properties of a field consult **Help**. Alternatively place the cursor in:

• The data type column and then **press F1** to get information about the different data types;

• The property row of a data type and then **press F1** to get information about the property.

**2.2.1 Creating a New Database**

To implement the Company database schema we will first create a new blank database. To create a blank database using MS Access 2013 do the following:

Start MS Access 2013 and the window in Figure 2.3 below will open. Click the second icon

in the first row and an additional window will appear as in Figure 2.3a.

Click here to start a new blank database

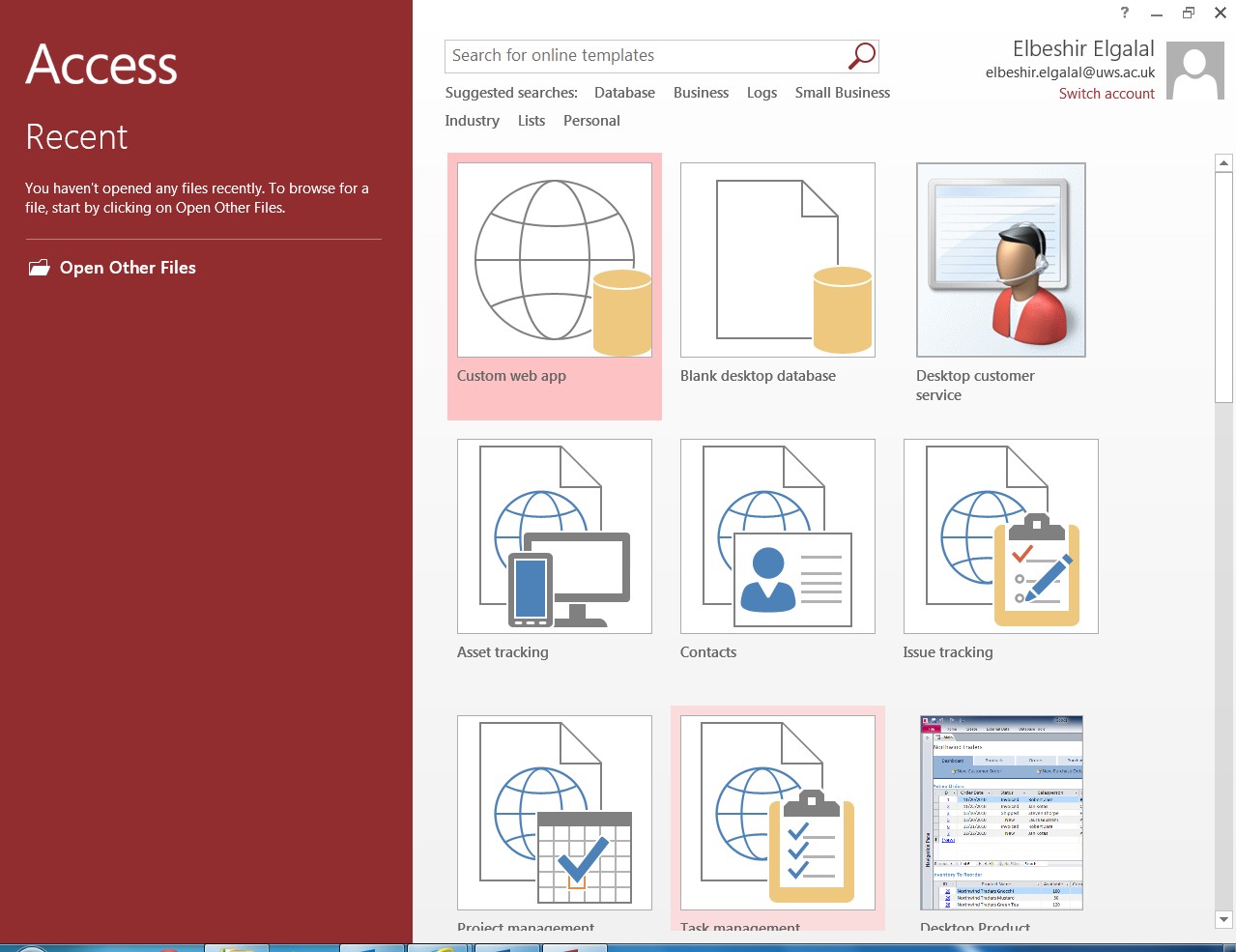


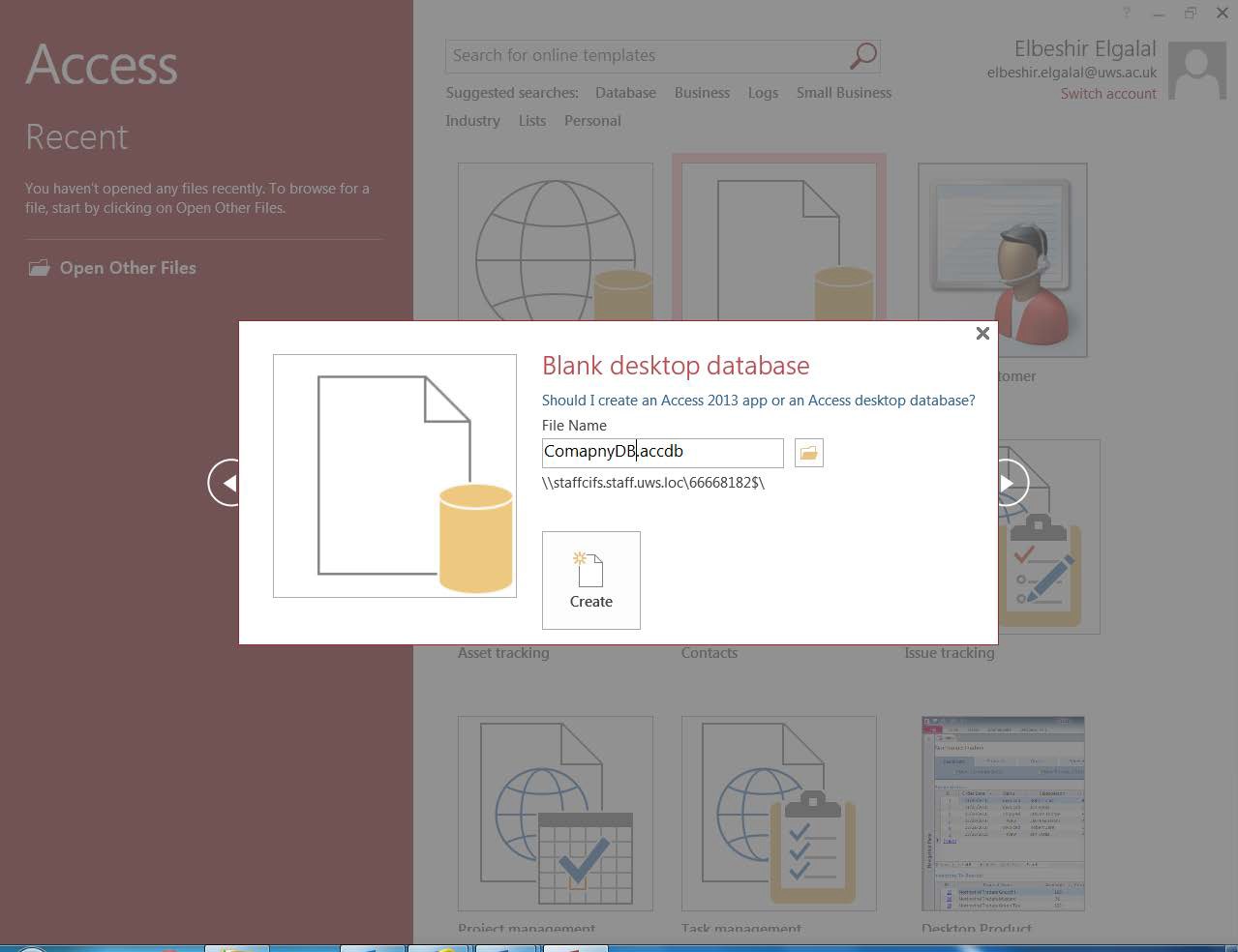
Figure 2.3 Creating a New Blank Database

**1.** Type the database file name in the box

**2.** Click the button and navigate to folder you wish to store the database file

3. Finally click here to create the database

Figure 2.3a Creating a new Blank Database



**Exercise 2. 1**

♦ Create the Company database application using the blank database option

Click Blank Database icon in Figure 2.3a, specify the file name (call the file **CompanyDB.accdb**) and click the button on the right-hand side of the File Name box, as shown in Figure 2.3a, navigate to the folder where you want to store the database file and click OK button.

When you click the **Create** button Access create the new database and displays the Navigation pane of the new database and opens a new blank table in Datasheet view, as shown in Figure

2.4.

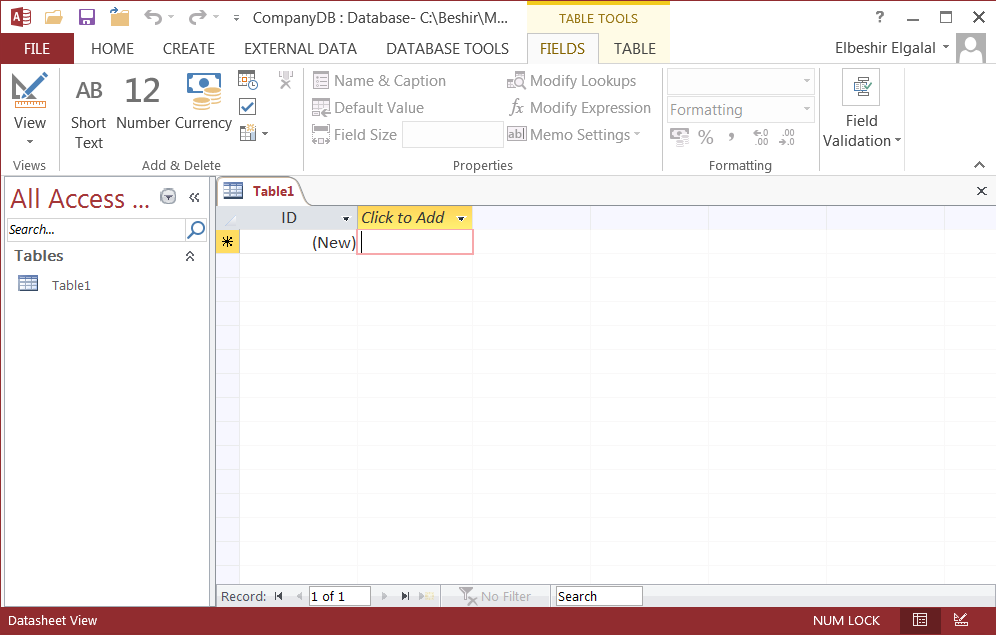


Figure 2.4 Table opened in datasheet view

If you click the **'Create'** button before navigating to the desired folder the database will be created in the default folder (i.e. My Documents).

**2.2.2 Creating tables**

A table can be viewed in two ways:

♦ Datasheet view – you can see the data in the table in a way that resembles a spreadsheet;

♦ Design view – allow you to define the structure and properties of the table.

When you create a new blank database Access, by default, creates a new table in Datasheet view called Table1 with an ID field already defined. This is what you see in Figure 2.4. You can add sample data to the table and let Access work out the data types for the data. **This is NOT a recommended practice**. It is good practice to first create the table structure and then populate it with data.

**2.2.2.1 Creating the Department Table**

**Department** {deptNumber, deptName, deptPhone, mgrNin, mgrStartDate} Primary key: deptNumber

Foreign key: mgrNin references Employee (nin) Alternate keys: deptName, deptPhone

The above table schema is from the Company Relational Database Schema on **page 15**.

**Exercise 2.2**

If you the window in Figure 2.4 is still open right click on the tab for Table1 in the Navigation Pane and select **Design View** from drop down menu that appears or select the tab for Table1 and click the icon above View in the Ribbon or click the downward arrow above the word 'views' on the Ribbon and select **Design View** from the drop down menu. In all cases a dialogue box will open and invite you to give a name to the table. Type **Department** and then click **OK**. The window in Figure 2.5 will appear.

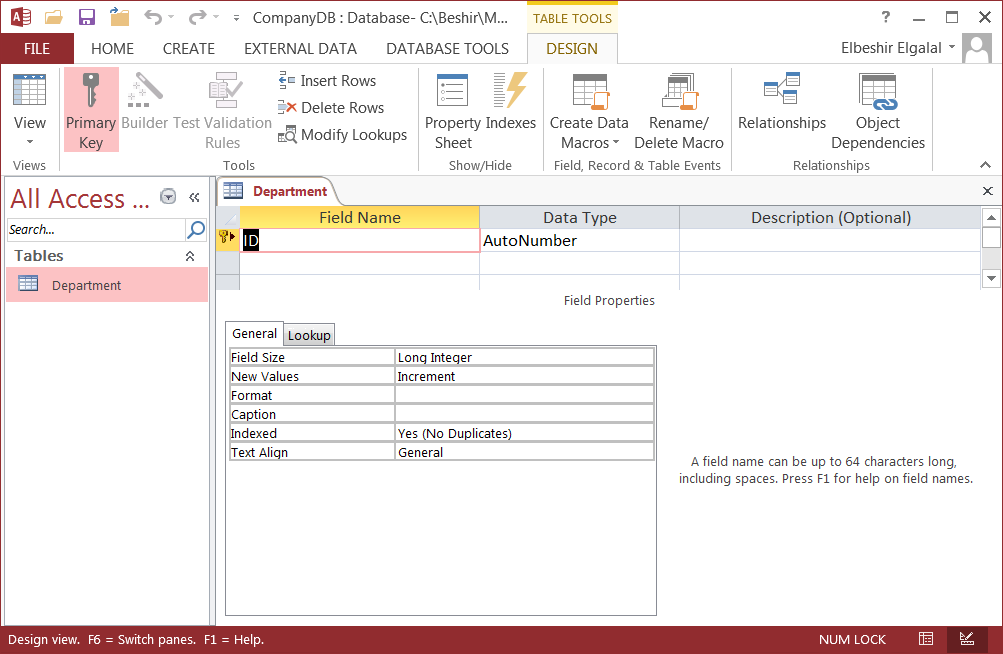


Figure 2.5 Table design view

You must define each field in the Department table. Each field has a **Field Name,** a **Data Type**, and optionally, a **Description**. These are shown in the top section of table design view. Every

field has properties. When you click on any of the field names in the top section, the properties of the selected field appear in the bottom section.

The **key** symbol shows which field (or fields) is the primary key for the table.

The **Field Name** identifies the data stored in a field. A field name can contain up to 64 characters, including spaces. The **Data Type** (field domain) tells Access what kind of data goes in the field, such as text, numbers, dates or currency.

When you create a table, Access assumes that the first field is called **ID**, with Data Type **AutoNumber**, and that this field will be the primary key for the table (key icon appears on the left side of the field name). However, our first field is called **deptNumber,** so overwrite **ID** with **deptNumber**. The Data Type for **deptNumber** is **Short** **Text**, so click on **AutoNumber**, click the **Short** **Text** from the drop down list which appears.

A brief description of the most commonly used data types is given in pages 20 and 21.

Before you start adding the remaining fields and setting their properties you should know more about Input Mask property. **InputMask** property makes data entry easier and controls the values users can enter. The table in Figure 2.6 show some of the characters used in an inputMask and their meaning. For more details on Input Mask property search Help for **Input Mask property or**

place the cursor in **Input Mask property row** and click help.

|  |  |
| --- | --- |
| 0 | Digit (0 to 9, entry required, plus [+] and minus [–] signs not allowed). |
| 9 | Digit or space (entry not required, plus and minus signs not allowed). |
| # | Digit or space (entry not required; spaces are displayed as blanks while in Edit mode, but blanks are removed when data is saved; plus and minus signs allowed). |
| L | Letter (A to Z, entry required). |
| ? | Letter (A to Z, entry optional). |
| A | Letter or digit (entry required). |
| a | Letter or digit (entry optional). |
| & | Any character or a space (entry required). |
| C | Any character or a space (entry optional). |
| > | Causes all characters to be converted to uppercase. |

Figure 2.6

**Creating the Remaining Fields in the Department Table**

To add or delete fields the table must be opened in **design** view. For each remaining field in the

**Department** table create a row in the design window.

Using the details given in the Department table in **Appendix2.1** type each remaining field name in the **Field Name** column, choose the field data type from the **Data Type** column, and set the field's properties (see Figure 2.7)

Note that the list of properties depends on the Data Type. For example, the **startDate** field (which is a **Date/Time** field) has no field size property, whereas **deptNumber** (which is a text field) does. This is because the space required to store a **Date/Time** field is always the same, whereas text fields can vary in length.

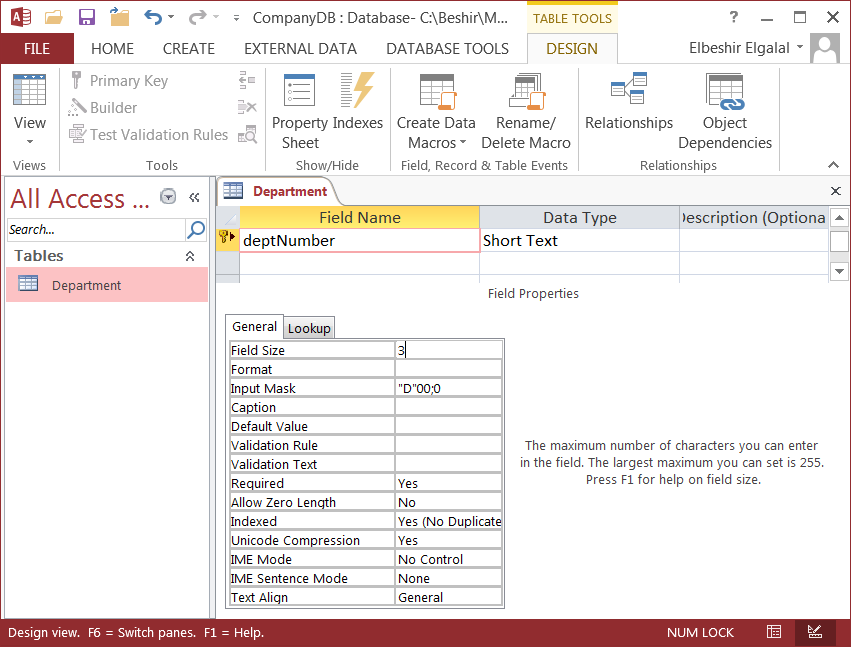


Figure 2.7

**Note:**

♦ the **indexed** property of the **deptNumber** field (primary key) is automatically set when the

**key symbol** appears on the left-hand side of the field name;

♦ Input mask should be **"D"00;0.** In this mask, the “D” means that capital D is automatically inserted into the field, **00** means that this must be followed by exactly two digits. The characters **;0** ensures that the **letter D** is stored in the table (rather than just appearing when you type a value into the field).

♦ In the mgrNin field, **required** is set to **yes**. This ensures that a department must have a manager (see structural constraints on the Company Database ER diagram).

♦ In **deptName** and **deptPhone,** index is set to **Yes (No Duplicates).** This ensures that no two departments have the same name or phone number. Both fields are Alternate keys (index property set to: **Yes (No Duplicates**)).

♦ There is a validation rule for **startDate**. This is in two parts, the rule itself, which says that the date must be later than 1/1/1900, and the validation text which will be displayed if an attempt is made to put a date in the field which is before this date.

After you finish setting all the fields of Department table, close the table and confirm that you wish to save the changes. The Navigation Pane shows you (see Figure 2.8) that you now have a

single object in the database: a table called **Department**.

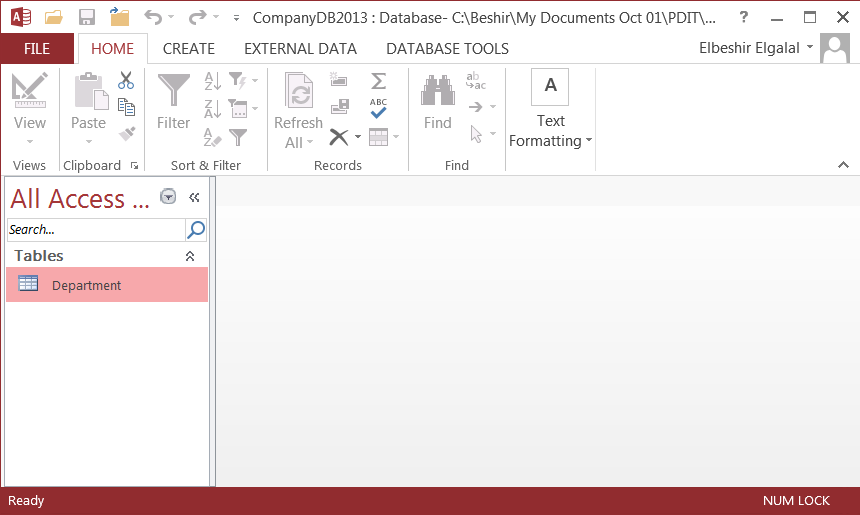


Figure 2.8

**2.2.2.2 Creating the Employee table**

Employee {nin, fName, lName, town, street, postCode, sex, birthDate, salary, superNIN, deptNumber}

Primary key: nin

Foreign key: superNin references Employee (nin)

Foreign key: deptNumber references Department (deptNumber)

**Exercise 2.3**

Click the tab **Create** and then click **Table Design** icon on the ribbon to create a new blank table in design view. Define the field names, data types and set the field properties as shown in the Employee table in Appendix2.1.

Note that the first field is the primary key of the table. To set the field as a primary you need to add the **key** icon on the left hand side of the field. Click the field name to select it and then click the **key** icon on the ribbon. Observe once you have added the key icon the field property **Indexed** will automatically be changed to **Yes(No Duplicates)** to enforce integrity constraint that values in this field must all be unique (no duplicate values).

Close the **Employee** table and confirm that you wish to save the changes. To close the table either click the **'x'** on the top right corner of the table window or place the cursor on the table name tab on table window, right-click and select **Close** from the drop down menu. The drop down menu has other facilities such as: opening the table in Design view if it is open in Datasheet view or vice versa, close all opened tables, etc.

**2.2.2.3 Create Remaining tables**

**Exercise 2.4**

For each of the remaining tables in the Company database schema click the tab **Create** and then click **Table Design** icon on the ribbon to create a new blank table in design view. Define the field names, data types and set the field properties as given in the corresponding tables in Appendix2.1. For each table you must set the primary key.

Note that the primary key for the tables: Dependent, WorksOn and EmpQualifications consist of

**two** fields. To set a **multi-field primary key** do the following:

♦ Select the first field by placing the cursor in the column on the left hand side of the Field

Name (the cursor shape will change into an arrow) and left-click.

♦ Press the shift key on the keyboard and hold and then select the second field;

♦ Click the **key** icon on the ribbon.

A key icon will appear next to each field of the primary key (see Figure 2.9).

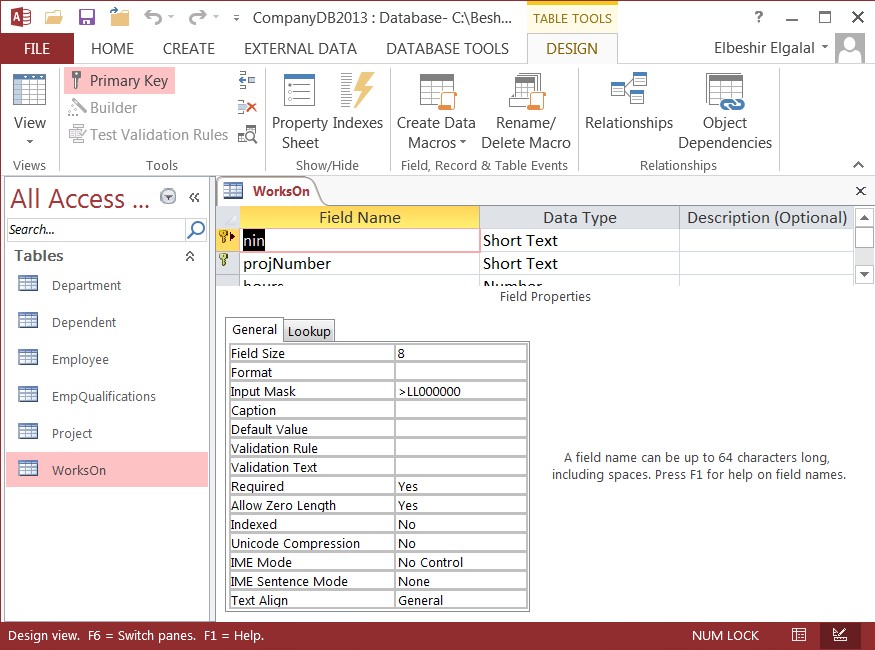


Figure 2.9

After creating all tables the navigation Pane will look as in Figure 2.10. To open any of the tables click the table icon in the navigation pane, right-click and then select Design View from the drop down menu to open the table in design view or Open from the drop down menu to open the table in datasheet view.

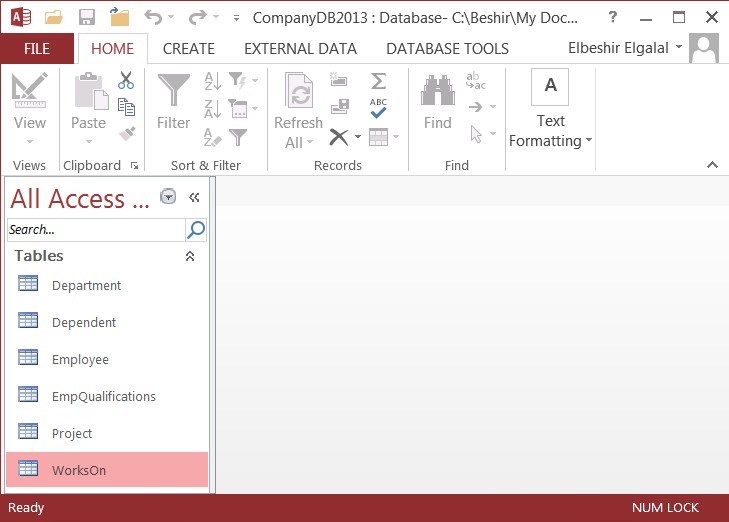


Figure 2.10

**Practical 3**

**Adding Data to Tables and Setting up Relationships**

**3.1 Adding Data to Department and Employee tables**

**Exercise 3.1**

**(i) Open the Department table in Datasheet View** and enter the data in Figure 3.1.

Click in the first column (deptNumber) and type the value D01. Press the keyboard tab key to get to the next field. When you get to the end of the row another tab will take you to the next row.

**Table 1** (**Department**)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **deptNumber** | **deptName** | **deptPhone** | **mgrNin** | **mgrStartDate** |
| D01 | Finance | 0131 6597794 | AR218148 | 03/03/1985 |
| D02 | Personnel | 0131 6629977 | CW954608 | 07/09/1995 |
| D03 | Engineering | 0141 3712958 | FL238319 | 29/10/1980 |
| D04 | Education | 0141 3346759 | GS866278 | 14/05/1998 |
| D05 | IT | 0141 8483875 | LY170573 | 19/06/2000 |

Figure 3.1

**(ii) Check if the constraints created so far** on the Department table schema are enforced by trying to enter a record with:

♦ Department Number: D01

♦ No Department Number

♦ Department Number: X33

♦ Manager NIN: FGT00000

♦ Department Name: Finance

♦ Phone Number: 0131 6597794

♦ Manager Start Date: 1/1/1890

♦ Manager NIN: AR218148

In all cases except the last you will get an error. **Why?** And why you don't get an error in the last case?

**(iii) Add data to Employee table:** enter all the data in Table 2 in Appendix3.1 in the

Employee table.

**(iv) Check if the constraints** created so far on the Employee table are enforced by trying to enter a record with any of the following values:

♦ Salary: £40000

♦ Sex: T

♦ birthDate: 25/10/1895

♦ Employee nin : AR218148

♦ A blank nin value

**Exercise 3.2**

**(i)** Enter a data record in the Employee table with following values:

nin: AB222222 superNin: AF111111 deptNumber: D99

Insert any suitable values in the remaining fields.

The record will be created even though you will enter wrong values for the foreign keys. There no employee with National Insurance Number: **AF111111** in the Employee table and no department with department number: **D99.** This is allowed because we have not set the relationships and enforced **Referential Integrity** constraints **yet**. See

Lecture 1 for the definition of Referential Integrity.

**3.2 Setting up relationships**

Before setting the relationships **print** out (if you did not do that yet) the ER diagram in Figure

2.1 and the relational schema that follows it in **Practical 2** so you can refer to them easily. From your ER diagram you can see that there are **six** relationships: **Fou**r of them (controls, Employs, Has, and Supervises) has cardinality ratio one-to-many (1:\*), one (Manages) with cardinality ratio one-to-one (1:1) and one (WorksOn) with many-to-many (\*:\*) cardinality ratio. Cardinality ratio can be worked out from the relationship structural constraints (See **Lecture Notes**, **Section2** for information on structural constraints and cardinality ratio).

**3.2.1 Setting one-to-many 'Employs' Relationship**

The relationship 'Employs' is implemented by storing the primary key value of the department which employs a certain employee as a foreign key in **deptNumber** field in Employee table. Foreign key values must refer to existing primary key values.

To set the 'Employs' relationship do the following:

**(a)** Click **Database Tools** tab and then **Relationships** button on the Ribbon. This will produce the window in Figure 3.2. If 'Show Tables ' window does not open, because some tables are already added to the window, click the 'Show Tables' button on the Ribbon.

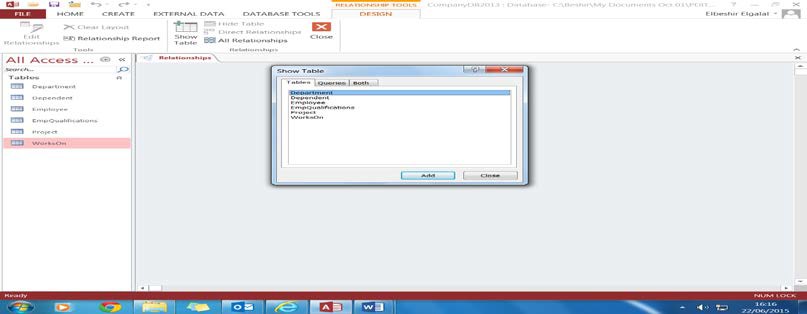


Figure 3.2

**(b)** Click **Department**, then **Add**, then **Employee**, then **Add**, then **Close** 'Show Tables' window. Relationships window will change as shown in Figure 3.3.

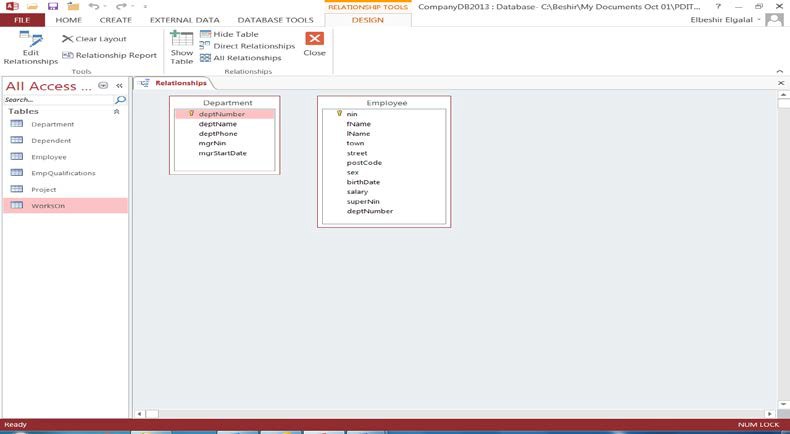


Figure 3.3

**(c)** Left-click and hold **deptNumber** field in **Department** table;

**(d)** Drag the field over to **deptNumber** field in **Employee** table and release the mouse button. The dialogue box in Figure 3.4 is displayed.

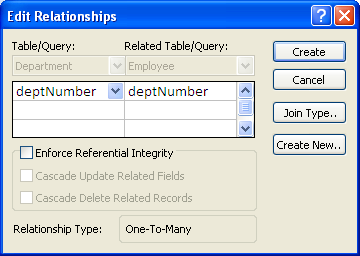


Figure 3.4

**(e)** Tick the box for **Enforce Referential** Integrity and click **Create**. When you enforce referential integrity you are telling MS Access to check that the values inserted in the foreign key in the Employee table do exist in the Primary key field in Department table. Because the record entered in **Exercise 3.2** contains wrong values an error message will be displayed stating that referential integrity is being violated. The value D99 violates referential integrity, as there is no such department number value in deptNumber field (primary key) of the Department table. Also the value AF111111 in *superNin* field will violates referential integrity when you try to set the recursive

relationship 'Supervises' (see exercise 3.11) as there is no **Employee** with this National

Insurance Number.

Close the error message window and click **Cancel** tab in the **Edit Relationships** window. Correct the wrong values or delete the record and repeat steps **(a)** to **(e)** above. The Relationships window will look like Figure 3.5.

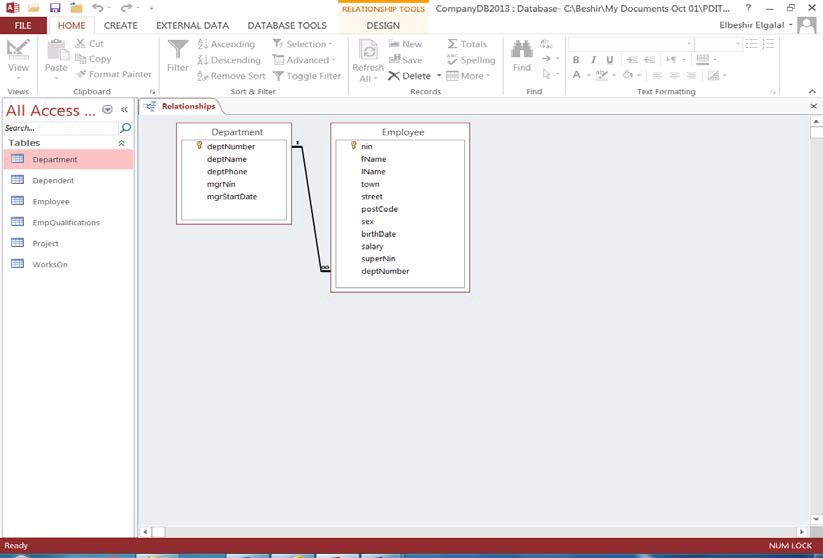


Figure 3.5

Note how the ‘many’ end of the relationship is identified with the ‘infinity’ symbol ∞**.** Close the

**Relationships** window confirming that the layout is to be saved.

Once you tick Enforce Referential Integrity box on the **Edit Relationships** dialogue box two tick boxes will be highlighted as shown Figure 3.6.

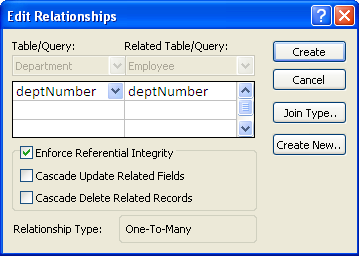


Figure 3.6

♦ **Cascade Update Related Fields** – this means if the Primary key value (deptNumber in Department table) is changed then change corresponding values in the foreign key field in Employee table and in any other table. It is good practice to tick this box.

♦ **Cascade Delete Related Records** - this means if a **record** in Department table is deleted then delete all related records in Employee table and any other tables. For example if department D06 record in the Department table is deleted; all records in Employee table with a foreign key equal **D06** will be deleted. This option is used in

certain cases.

**Exercise 3.4**

♦ Open the Relationship window, click the **Employs** relationship line and then click **Edit**

**Relationships** button on the Ribbon;

♦ Tick the remaining boxes (i.e. Cascade Update ..., and Cascade Delete ....)

♦ Close the **Relationships** window and save changes;

♦ Add a **new** department record in the Department table ( e.g. D06);

♦ Add two records in the Employee table and assign D06 as foreign key;

♦ Edit the new department number and change it to D11. Open Employee table and observe that the deptNumber field values D06 are also changed to D11;

♦ Delete the row of the newly added department (i.e. D06) in Department table. Open

Employee table and observe the effect.

**3.2.2 Setting one-to-one 'Manages' relationship**

The relationship **Manages** state that every department is managed by one Employee only. The field property '**Required**' is set to **Yes** (see Department table in Appendix2.1). This means the field value cannot be NULL.

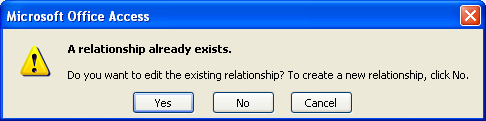
To set the 'Manages' relationship do the following:

**(a)** Click **Database Tools** tab and then **Relationships** button on the Ribbon. This will produce the window in Figure 3.5.

**(b)** The required tables are already in the window and you don't need to add any more tables. Left-click and hold **nin** field in Employee table;

**(c)** Drag the field over to **mgrNin** field in Department table and release the mouse button.

The following message will appear. Click **No** and the dialogue box in Figure 3.7 appears.



The relationship type box displays: **one-to-many** but we require **one-to-one**. You need to set **mgrNIn** field (in Department table) **'indexed'** property to **Yes (No Duplicates)** so that the relationship type appears as one-to-one. This enforces the fact that the values in this column

must be unique (an alternate key field). An employee may manage only one department, therefore an employee National Insurance Number can appear **only once** in the mgrNin

column.

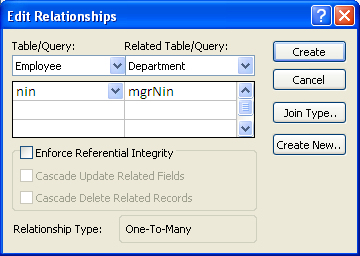


Figure 3.7

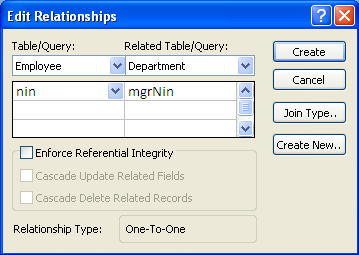
**Exercise 3.5**

♦ Click the **Cancel** button on the **Edit Relationships** dialogue box in Figure 3.7 and close the **Edit Relationships** window;

♦ Open **Department** table in **design** view;

♦ Select **mgrNin** field and set the property 'Indexed' to **Yes(No Duplicates).** Close the table and save the changes;

♦ Repeat steps **(a)** to **(c)** in **Section 3.2.2** and the following dialogue box will appear.



♦ Tick the box next to **Enforce Referential Integrity** and the box next to **Cascade Update**

**Related Fields** and press **Create**. This will produce the window in Figure 3.8.

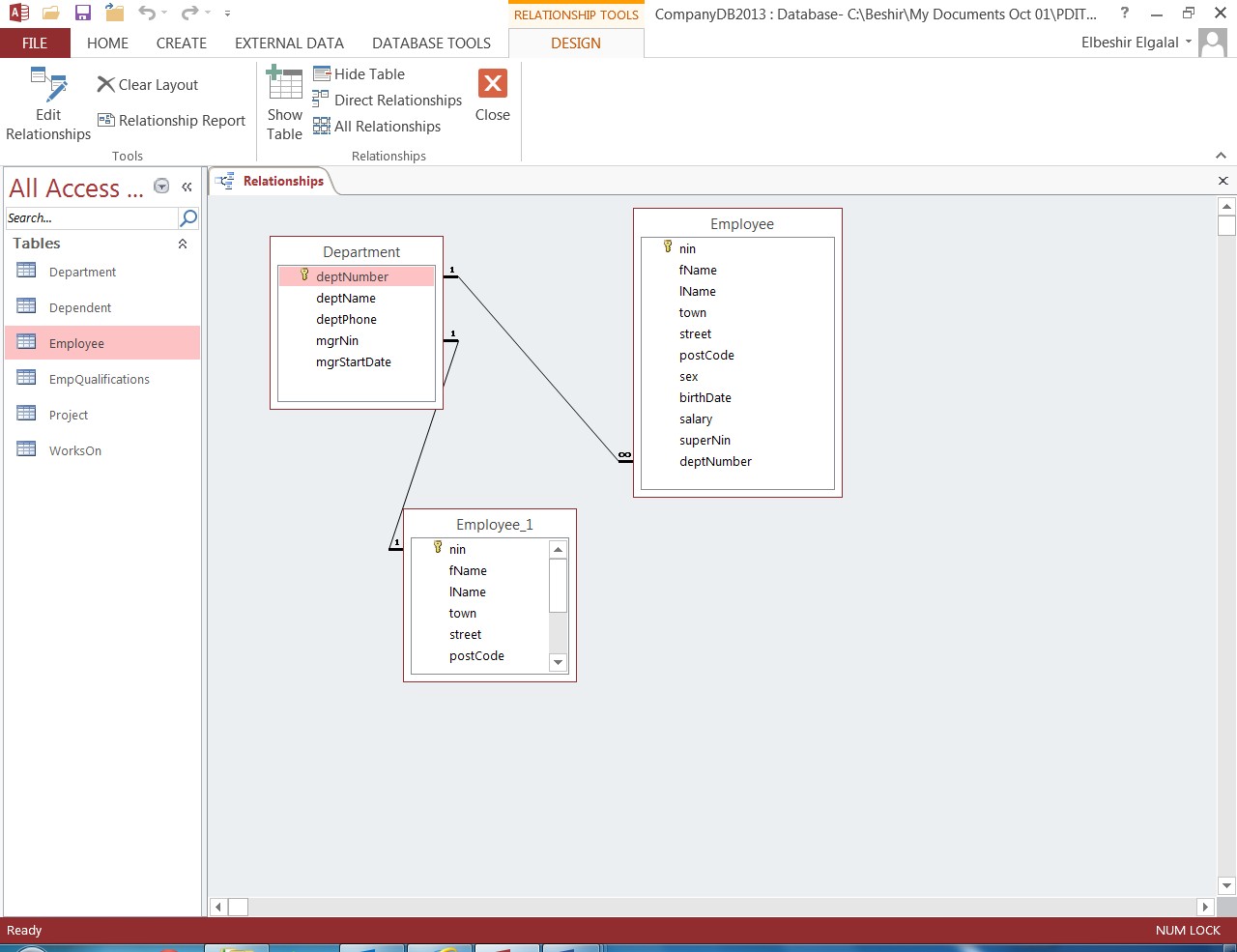


Figure 3.8

**3.3 Lookup Columns and Drop Down list of values**

To avoid entering wrong values for **foreign keys** or repeatedly typing the same value you would like to select foreign key values from a valid list of primary key values. Similarly you should be able to select a repeatedly entered value from a drop down list of values.

You do this by changing a field in the table to a Lookup column (or field) by setting up the **lookup** properties of the field. A lookup column (or field) is a field in a table whose value is retrieved from **another table** or from a **value list**. Converting a foreign key field (e.g. deptNumber in Employee table) to a lookup column will enable you to display the list of the corresponding primary keys from the parent table (i.e. Department table). Similarly you can convert a field to lookup column that retrieve its values from a drop down **value list** which you specify. These can be done by either setting the **lookup properties** of a field or using the **Lookup Wizard**

Displaying a list of the primary keys only will not be very helpful to the user. The list should contain more fields form parent table (primary key table) **so that the user can make a more informed choice**. Remember only the value of the primary key will be stored in the foreign key field even though the list shows more than one field.

**3.3.1 Setting up the Lookup Properties of a Field**

You can set up the field to get its values from another table (**combo** box) or they can be from a list of values you supply (**list** of values).

**Exercise 3.6**

To convert **deptNumber** field in **Employee** table to a **combo box** lookup column do the following:

**(a)** Open **Employee** table in design view;

**(b)** Click on **deptNumber** field and then the **Lookup tab** in the properties section;

**(c) Set** the **properties** as follows (see Figure 3.10):

♦ **Display Control** – set to **Combo Box.** A combo box displays a list of possible values that appears when you click it.

♦ **Row Source Type** – set to **Table/Query,** as the values for the deptNumber field will be taken from the values that appear in a table or query, rather than from a fixed predefined list

♦ **Row Source** – set to **Department**, because that’s where the values will come from.

♦ **Bound Column –** set to **1**, as the values of **deptNumber** are to be found in the first column of the **Department** table.

♦ **Column Count** – set to **3**, as only 3 columns of department will be displayed when **deptNumber** is to be selected. Property can be set to 4 or 5 4 fields or all fields of Department table to be displayed.

♦ **Column Heads –** set to **Yes**, to ensure that the list of possible values includes the names of the columns.

♦ **Column Widths –** set to 3cm;3cm;3cm. Each column width is set to 3cm.

♦ **List Width** – set to **9cm** or **15cm** if **'column count'** field is set to 5. Width large enough to accommodate chosen fields.

♦ **Limit to List** – set to **Yes:** we don’t want to type in a value rather than choose a value from the list.

♦ **Allow Value List Edits -** set to **No** as you don't want to change the values of the primary keys.

♦ **Show Only Row Source values -** set to **Yes**. Only the primary key row values should be shown

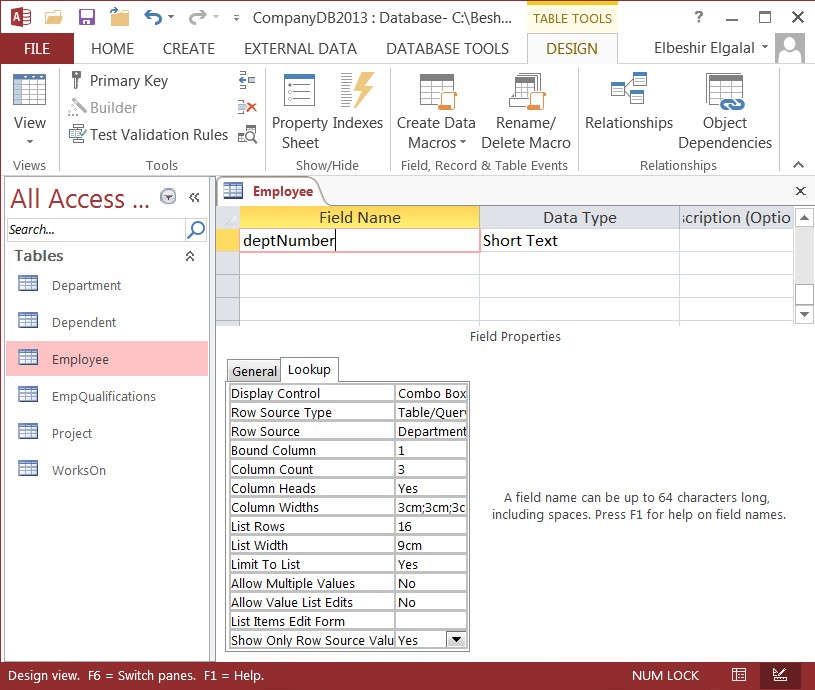


Figure 3.10

**(d)** Close Employee table and save changes. Open the table in datasheet view and click any of the values or an empty cell in deptNumber column and then click the drop-down arrow on the right-hand side of the cell. A list of all the departments records (3 fields only; the property **Column Count** = 3 in Figure 3.10) will be displayed.

Remember when you select a record only the **primary** key field value of a **Department** will be stored in the foreign key field of **Employee.**

**Exercise 3.7**

To convert the **town** field in **Employee** table to **list of values** lookup column do the following:

**(a)** Open Employee table in design view;

**(b)** Click **'town'** field and then **Lookup tab** in the properties section;

**(c)** Set the **properties** as follows (see Figure 3.11):

♦ **Display Control** should be **List Box.** A list box displays a list of possible values that appears when you click drop-down arrow.

♦ **Row Source Type** - should be **Value List** as the list values are not coming from any table or Query.

♦ **Row Source -** type the values to appear in list separated by semicolons.

♦ **Bound Column** – list is not bound to any column of a table. The number can be either

1 or 0. The value 1 tells Access to display the current values in the column. The value 0 tells Access to clear the current values in the column.

♦ **Column Heads -** set to **No**. Access uses the name of the column on which the list is build.

♦ **Column Widths –** leave blank

♦ **Allow Value List Edits –** set to **Yes.** This enable user to add, delete or edit the items to be displayed in the list box.

♦ **List Items Edit Form –** leave blank.

♦ **Show only Row Source Values** – set to **Yes**.



Figure 3.11

**Exercise 3.8**

♦ Repeat the steps in Exercise 3.7 for the 'sex' field. Remember the **Row Source** values should be: **M** and **F** only.

**Exercise 3.9**

♦ Using the steps in Exercise 3.6 convert all remaining foreign key fields, namely:

♦ **superNin** in Employee, **mgrNin** in Department, **empNin** in Dependent, **deptNumber** in Project, **nin** in WorksOn, **projeNumber** in WorksOn, and **nin** in EmpQualifications to **combo** boxes lookup columns

**Exercise 3.10**

♦ Add one or two records in the **Project** and **Dependent** tables making use of the lookup columns.

♦ Clear the **Project** and **Dependent** tables.

♦ **Populate** all the empty tables with the data given in **Appendix3.1**.

**Exercise 3.11: Set the remaining relationships**

♦ First click on **Database Tools** tab and then **Relationships** button on the **Ribbon**. Click on the button **Show Table** add the remaining tables to the Relationships window and then use the steps given in **section 3.2.1** to set the relationships. Enforce Referential

integrity and also tick the box for **Cascade Update Related Fields**.

**Note:**

♦ All remaining relationships are one-to-many relationships except WorksOn which is many-to-many relationship. A many-to-many relationship is implemented as two one-to- may relationship.

♦ WorksOn relationship is implemented as one-to-many between Employee and WorksOn tables and one-to-many relationship between Project and WorksOn tables.

♦ The recursive relationship Supervise is between Employee (supervisor role) and Employee (supervisee role) is one-to-many relationship. Because only one table is involved in the relationship you need to add another copy (Employee\_1) of Employee table (if it does not already exist in the Relationships window). Once you have two copies of the table Employee drag **nin** field in Employee\_1 and drop it on superNin field in Employee.

♦ A copy of Employee table was created when you implemented the one-to-one

'Manages' relationship

♦ After setting all relationships the **Relationships** window should look like Figure 3.12.

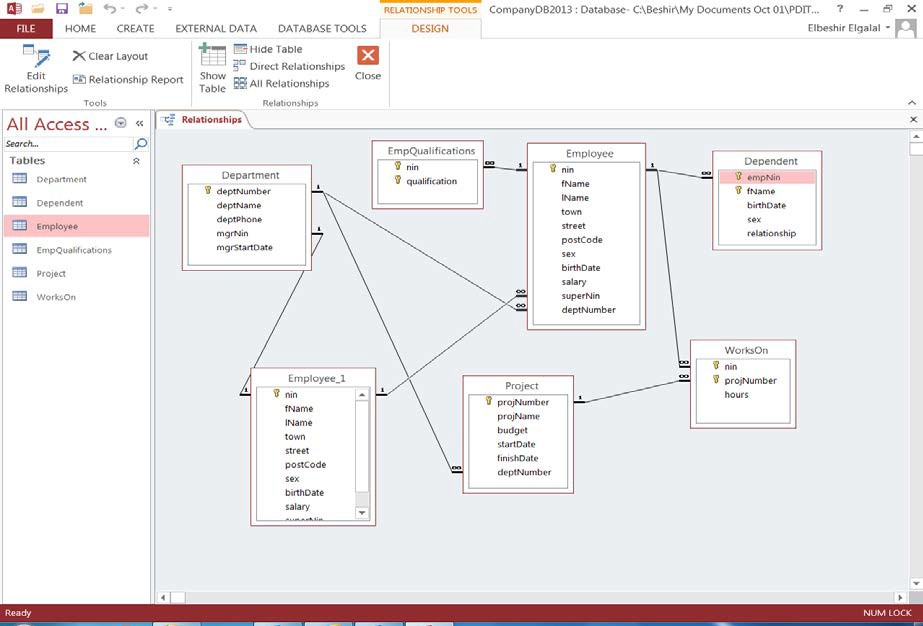


Figure 3.12

Two ways you can delete a relationship line:

1. Right-click the relationship line and select **'Delete'.**

2. Select the relationship line (left-click on the line) and press the **Delete** key on the keyboard or the **Delete button** in the Records group in the Home tab on the Ribbon.

Two ways to edit a relationship:

1. Right-click relationship line and select 'Edit Relationship'

2. Select relationship line (left-click) press the **Edit Relationships** button on the Ribbon.

**Exercise 3.12**

♦ Save the final Relationships window and print it.

**Practical 4: Creating Forms**

**4.1 Introduction**

In this practical you will learn to create Forms. Using forms you can create a graphical user interface to your database application. Forms can be used to read, add, delete or update data stored in the underlying tables linked to the forms. You can create forms based on a single table, more than one table or on a query. Access 2013 provides different tools to create forms, namely:

**(a)** Create a form from scratch using Design tools. This option is long and complex;

**(b)** Create a form using Quick create commands;

**(c)** Create a form using the Form Wizard.

In this practical you will learn how to create a form using option (b) and (c). This practical will cover the creation of:

♦ **Single table, single record forms** - the form allows the user to enter information for one record

♦ **Single table Split forms** – a spilt form gives two views of the data in the underlying table at the same time – Form view and a Datasheet view. The two views are connected to the same data source and are synchronized with each other at all times. Selecting a field in one part of the form selects the same field in the other part of the form. You can add, edit, or delete data from either part (provided the record source is updateable and you have not configured the form to prevent these actions). Working with split forms gives you the benefits of both types of forms in a single form. For example, you can use the datasheet portion of the form to quickly locate a record, and then use the form portion to view or edit the record.

♦ **Multiple Items (Continuous) Forms -** A multiple item form lets you show information from more than one record at a time. The form resembles a datasheet format as the data is arranged in rows and columns and multiple records are displayed a time. However, because it is a form, there are more customization options than with a datasheet. You can add features such as graphical elements, buttons, and other controls.

♦ **Multi-table forms** – Form built on more than one table.

♦ **Graphical User Interface** – us the forms to create a simple graphical user interface to the database

**4.2 Creating a Single table Form using Quick Create Commands**

To create a single form for the Employee table using Quick create commands, proceed as follows:

**(a)** Close all open objects in Access (i.e. tables, forms, etc.), leaving just the list of the tables in the Navigation Pane;

**(b)** Select the Employee table by clicking on it once;

**(c)** Click the **Create** tab on the Ribbon;

**(d)** Click the **Form** button in the Forms group and the window in Figure 4.1 is displayed.

Control Layout Indicator

Label control Text box control

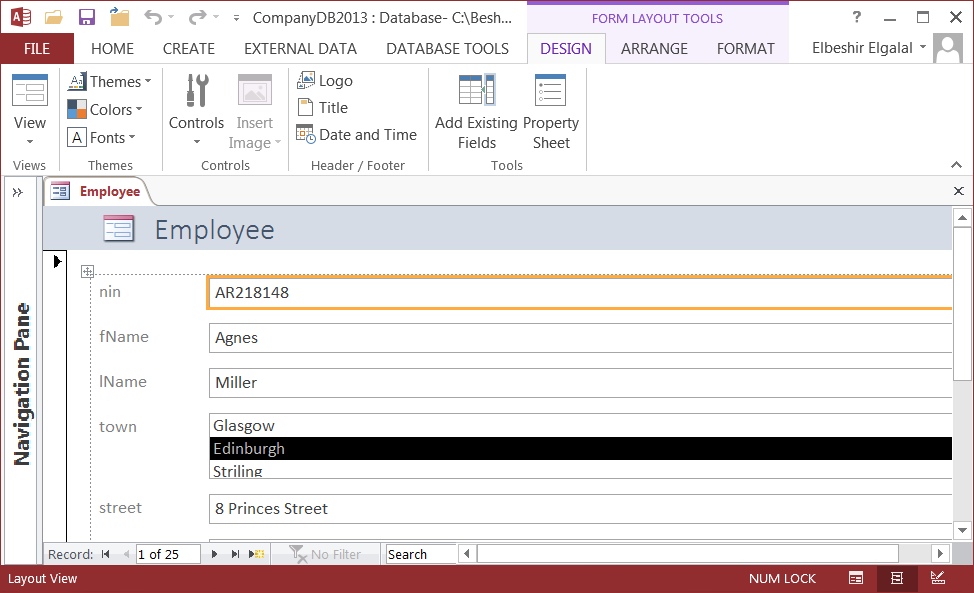


Figure 4.1

Access immediately creates a new single form based on the Employee table, including a control for every field in the table, and displays it in **Layout** view as shown in Figure 4.1. Each control consists of two parts a Label (with a Caption) and a text field which is **bound** with the corresponding field in the underlying table. Access also creates a bitmap picture logo and a label for the form's title in the **Header** section.

Switch to **Design** view for this form by clicking the **arrow** below the **View** button in the **Views group** on the **Home tab** and click **Design View.** A form has header and footer and you can also add a form page header and form page footer. Page headers and footer are used when you create a multi-page form. Design View is used when you want to customize a form (change the structure or design of the form e.g. add new controls, changing the controls position, size, colour, font, etc). You can only see the form structure but no data. To see the data you must switch to **Form** view or **Layout** view.

You can use **Layout** view to view the data in the underlying table but you cannot manipulate the data (you cannot add, delete or update data in the table). You can also modify the form structure in this view. Unlike **Design** view, **Layout** view enables you to work with the various control elements and form sections using live data. If, for example, you need to resize the a

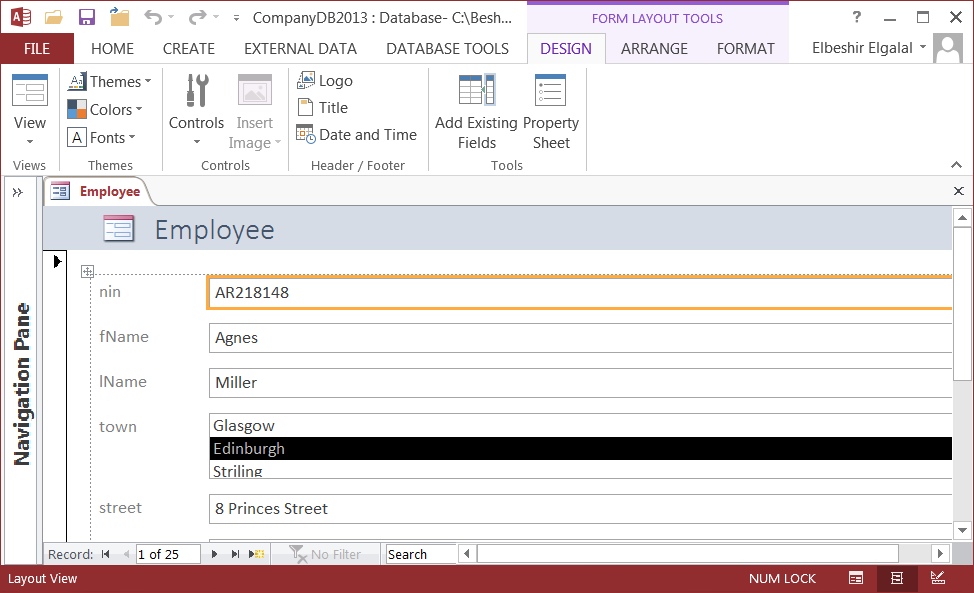
text box to fit the available data, you don't have continually to switch back and forth between **Form** and **Design** view to see if the size change works effectively – you actually see the data in the text box while resizing the control. In Layout view you have limited facilities for customizing the form.

To view, change, insert, or delete data via a form, you can use the **Form** view. To summarize a form can be viewed in three ways: **Form**, **Layout** and **Design** view.

**To Move from Record to Record on a Form**

Open form in Form or Layout view. Use the navigation buttons in lower-left corner (see

Figure 4.2) of the forms window.



First Record

Previous Record

Record Number box

New Record

Last Record

Next record

Figure 4.2

Exercise 4.1

♦ Open Employee form in Design view and then open the form's properties sheet.

♦ To open a Form property sheet click on the form top left corner (Form's button, see Figure 4.3) and then click the Property Sheet button on the Ribbon or just double click the Form's button (Form top left corner) and the Form Property sheet will open.

♦ Observe that form's Default View property is set to Single Form

Form button

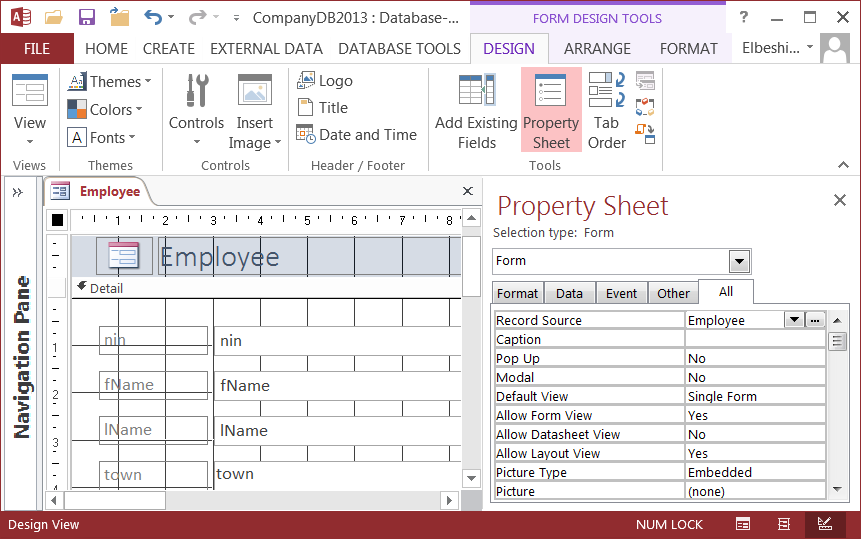


Figure 4.3

♦ You can also display the Form’s Property sheet as follows: click the Form’s button (see Figure 4.3)  click Design tab on the Ribbon  click Property button in the Tools group.

**Exercise 4.2: Create a split form**

**(a)** Close and save the previous Employee form (Call the form Employee also);

**(b)** Select **Project** table in the navigation Pane by clicking on it once;

**(c)** Click the **Create** tab on the Ribbon;

**(d)** Click the **More Forms button** in the **Forms** group and click **Split Form** from the drop- down menu. The window in Figure 4.4 is displayed.

The top of the Form window displays the fields from the Project table in Single Form view, and the bottom of the form displays all records from the Project table in Datasheet view. Use the form navigation arrows (See Figure 4.2) to move through the records and you see that that the two views are synchronized with each other. Changing any value in the lower part of the form is reflected in upper part of the form.

♦ Switch to Design view and open the form's properties sheet;

♦ Check the Form Default View and Split Form Orientation properties.

♦ Close the form and don't save it when prompted.

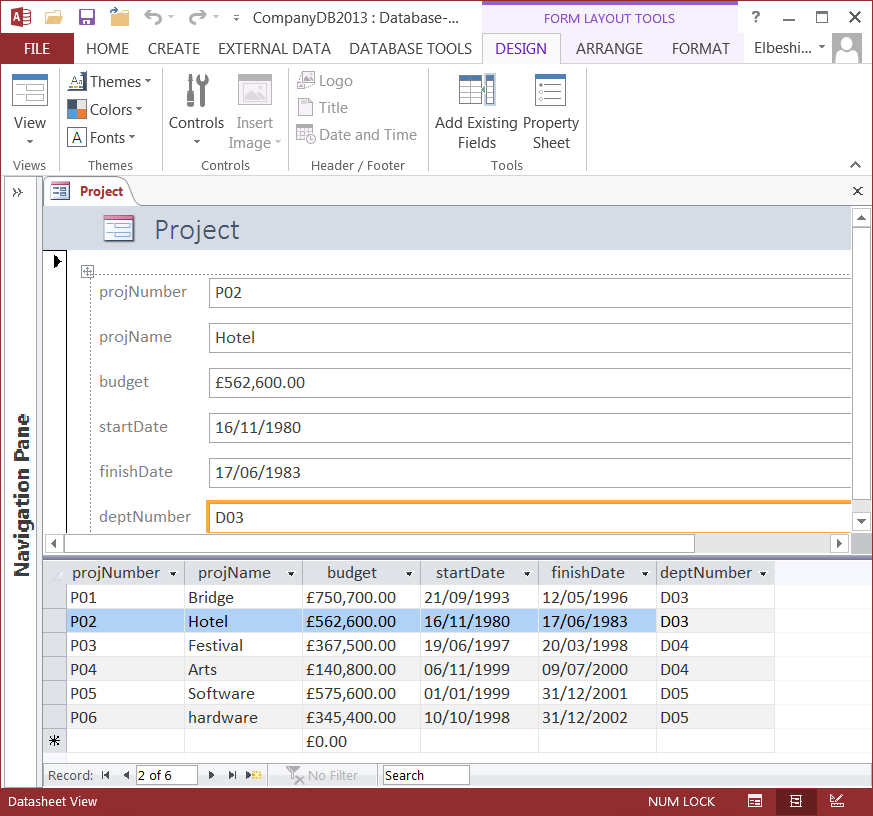


Figure 4.4

**Exercise 4.3 Create a Multiple items form**

♦ Select the Employee table by clicking on it;

♦ Click the **Create** tab on the Ribbon  click the **More Forms** button in the **Forms** group

 click **Multiple Items** from the drop-down menu. Access immediately creates a new **continuous** form based on all the fields in the Employee table and displays it in Layout view.

♦ Switch to Design view and check the form properties (Default View is set to continuous

Forms);

♦ Close the form and don't save when prompted.

**4.2.1 Automatic Subdatasheet Creation**

If Access finds a table that has a one-to-many or a one-to-one relationship with the table or query that you used to create the form, Access adds a Subdatasheet to the form that is based on the related table or query.

**Exercise 4.4**

♦ Create a simple form based on the Project table.

There is a one-to-many relationship between the **Project** table and **WorksOn** table, the Subdatasheet displays all the records in the WorksOn table that pertain to the current **Project** record (See Figure 4.5). If you decide that you do not want the Subdatasheet on the form, you can delete the Subdatasheet by switching to **Design** view, select the datasheet, and then press DELETE.

If there is more than **one** one-to-many or one-to-one relationship to the table that you used to create the form (e.g. Employee), Access **does not** add any Subdatasheet to the form.

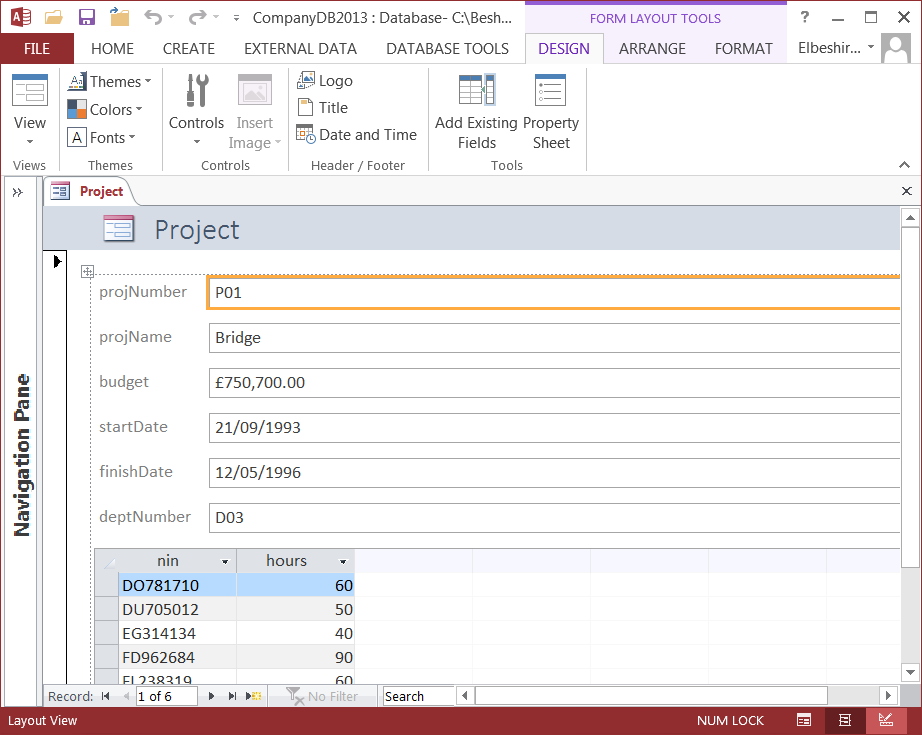


Figure 4.5

**4.3 Creating a Single table Form Using the Form Wizard**

The ‘quick create form’ commands are easy to use, but you have no flexibility on how Access initially creates the form. The **Form Wizard** is another tool you can use to quickly create forms in your database. Select **Employee** table in the Navigation pane, click the **Create** tab, and then click the **Form Wizard** button. Access opens the first page of the Form Wizard as shown in Figure 4.6.

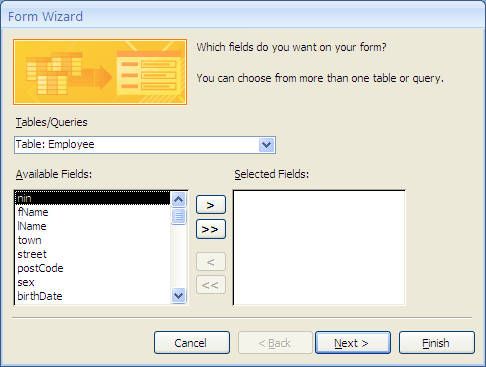


Figure 4.6

You can:

♦ Select any field from **Available Fields** list and click the single right arrow (>) button to copy the field to the **Selected Fields** list;

♦ Click the double right arrow (>>) button to copy all available fields to the **Selected**

**Fields** list;

If you copy a field in error, you can:

♦ Select the field in the Selected Fields list and click the single left arrow (<) button to remove the field from the list;

♦ Remove all the selected fields and start all over by clicking the double left arrow (<<)

button.

For this example:

♦ Click the double right arrow button to use all the fields in the **Employee** table in the new form, and then click **Next** button.

♦ Select **Columnar** layout (try other layout also), and then click **Next**;

♦ **Form Wizard** will open the final page and asks for a title for your form. Type the name of the table (i.e. Employee in this example) and click **Finish**. The wizard places this title in the **Caption** property of the form and also save the form with this name (If you already have a form called Employee, Access appends a number to the end of the name to create a unique name). When you press the Finish button the form in Figure

4.7 is displayed.

♦ If you want to edit the form before saving it select **'Modify the form's design'** before you click **Finish.**

In Figure 4.7 the 'town' and 'sex' fields are shown as drop down **list** of values. They can be changed to Combo boxes so that they appear like **'deptNumber'** field by modifying the form

(see Exercise 4.7) or deleting the form and creating it again on a modified Employee table.

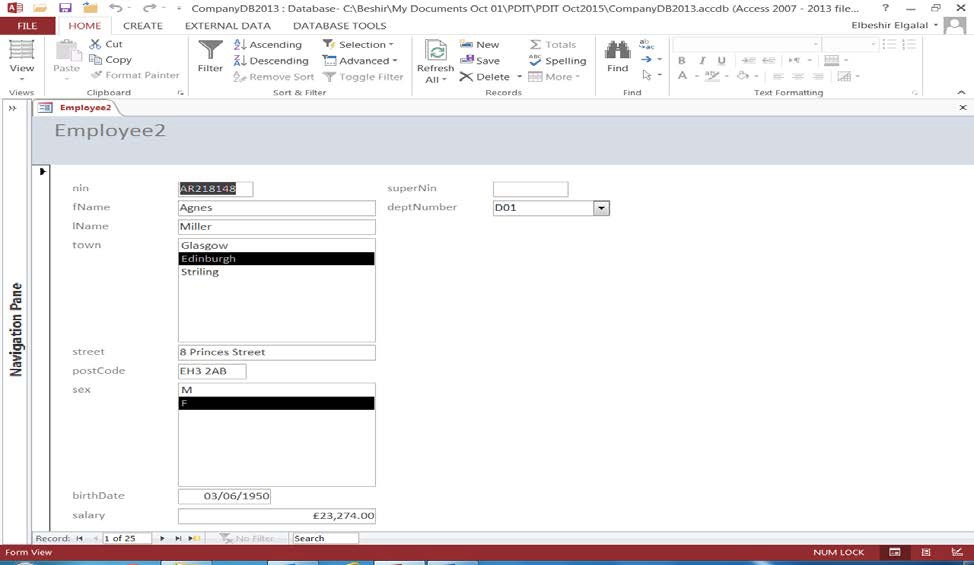


Figure 4.7

Note the Form Wizard does not create a Subdatasheet.

**Exercise 4.5**

Modify the Employee table:

♦ Open the table in design view;

♦ Click the 'town' field and change the 'Display Control' property in the **Lookup** tab to

Combo box (see Exercise 3.7 in Practical 3);

♦ Click 'sex' field and change the 'Display Control' property in the Lookup tab to Combo box also.

♦ Save the table and then create the Employee form again on the newly edited table.

**Exercise 4.5a**

♦ Using the form Wizard creates a form for each of the remaining tables.

♦ The Employee table is involved in more than one relationship. Experiment with removing all relationships except one and create a form on Employee using **Form** button option (see section 4.2.1) in the Forms group.

**4.4 Modifying a Form's structure**

In this section you will practice how to change the structure of the form. You will learn how to move, resize, add or delete and change properties of the form controls. Open **Employee** form in Design view.

Access divides a form into **five** sections in design view, namely:

♦ **Form header** – prints at the top of every page. Currently it contains the form title.

♦ **Page header** – prints at the top of every page. Only Form header is added by default.

To see the form Page header you need to add it (see exercise 4.6).

♦ **Detail** – contains the fields from the table. When you are viewing data, the **Detail** section is repeated for each record. When you print the form, the detail section shows as many records as will fit on a page.

♦ **Page footer** – prints at the bottom of every page. Currently blank but you can add a text box to print the page numbers. The page footer will only appear, if it contains any information, when printed or in print preview.

♦ **Form footer** – prints at the bottom of every page (currently blank). When you are viewing data the footer appears once at the bottom of the window, if it contains any information.

**Exercise 4.6**

♦ Open Employee form in Design view and check if all the sections are displayed. If the **page header/footer** is not displayed right-click the Form Header bar and select **Page Header/Footer** from the drop down menu.

♦ Print preview of the form. Click the **File** button (top left corner of Access window)**,** select

**Print** and then **Print Preview**.

Before modifying the form makes sure that the **Grid** lines are visible by turning the Grid on. To turn the Grid on right-click the Detail section and select the Grid Lines button from the pop-up menu. To make individual adjustment (moving, resizing, colouring, etc.) to the labels and text boxes controls, you need to **ungroup** the controls by turning **off** the **Control layout**. The controls' grouping is created by Access when creating the form.

**Note:**

♦ **Control layout indicator** appears on forms created using the **Form** command only (see Forms group under CREATE tab) and can be removed when the form is in **Layout view** or **Design view** (see Exercise4a below). Once removed the form can be modified in **Layout** view or **Design** view. In **Layout** view controls' **size** and **move** handles are not visible.

♦ **Control layout indicator** does **not** appear on the form if the form is created using

**Form Wizard.** Forms created using Form Wizard can be modified in **Design** view

**Exercise 4.6a**

♦ **Turn off the Control Layout.**

In **Layout view** click on any control (text box or label part) and the symbol will appear in the top left corner of the **Detail** section of the form (see Figure 4.1). The symbol is called **Control layout indicator.** Click on the symbol and all the controls will be selected and you can move them in any direction as a group of objects. To turn off the Control layout do the following:

♦ Open form in design view;

♦ Select any control on the form.

♦ Click **Arrange** tab on the **Ribbon** and click **Select Layout** in **Rows & Columns**

group. All controls will be selected.

♦ Right-click on any control and select **Layout** from pop-up menu and then select **Remove Layout** from pop-up submenu**.** The controls will be ungrouped and their **size** and **move** handles will be shown as in Figure 4.8.

The big square box in the top left corner of each label and each text box are the **move** handles (you can use them to move the box or the label). The small square boxes in the middle of the horizontal line and the vertical line of the labels and text boxes are **size** handles. The size handles are displayed at the top, bottom, left and right and middle sides of the control when it is selected.

Move handle Left/right size handle Up/down size handles

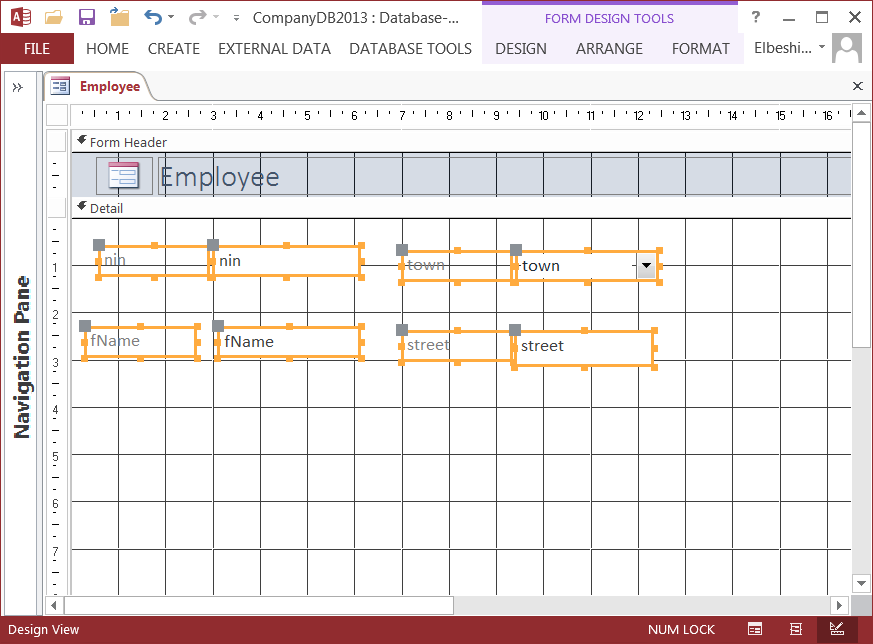


Figure 4.8

Now you have removed the controls layout you can make adjustments to the individual controls.

**Exercise 4.7**

♦ Using the Employee form you created in Exercise 4.5 and change the fields 'town' and

'sex' to **List** boxes.

**To change the fields do the following:**

♦ Open Employee Form in design view;

♦ Right-click 'town' or 'sex' field, select **'Change To'** from the drop-down menu and choose

**List box.** Follow the same process if you want to change the field back to Combo box.

**Note:**

Changing the two fields to List boxes on the **form** does not change the field property in the table. The fields **Display Control** property in the **Lookup** section will still be **Combo Box** and not **List Box**. Similarly if the fields were changed to Combo boxes from List boxes.

To avoid changing the fields every time you create a form on Employee table you need to edit the Employee table and change the 'town' and 'sex' fields' **Display Control** property to the required type before creating a form on the table.

Each item on a form including the form itself as whole is considered by **Access** as an **object** and has a set of properties that can be changed. When you open a form in design view the Ribbon changes as shown in Figure 4.9 on page 59. To see the form’s properties make sure that the **Form button** is clicked (see Figure 4.9) and then click Property Sheet button on the Ribbon.

To see the properties of any object (e.g. textbox, label, header, footer, etc.) on the form you must first select object by clicking on it and then click the **Property Sheet** to display its properties. To modify an object (including the form itself) you can change the object's properties. You can also change the size and position of an object manually.

To get more information about any property, place the cursor in the property field and press F1 on the keyboard.

Form button

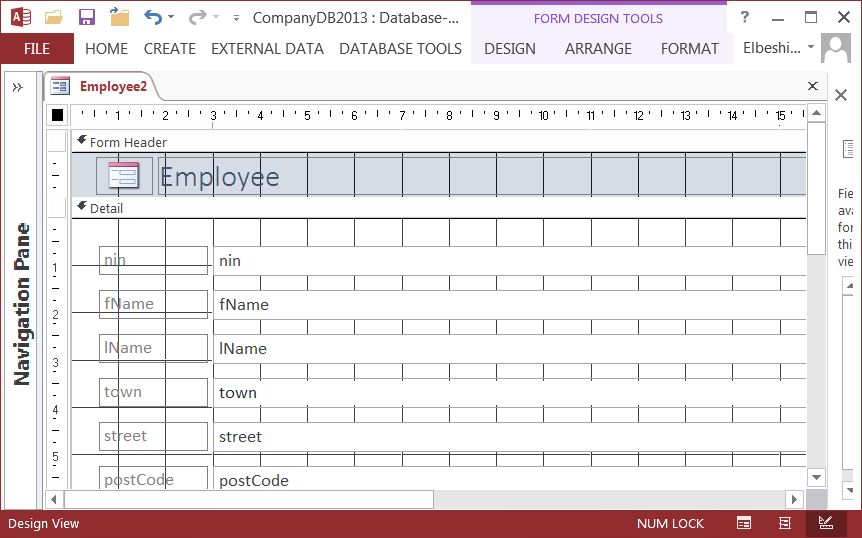


Figure 4.9

**Select and Resize Controls**

*Controls* are the objects on your form that display data from a field or words in a title. In

Design view, you can select, resize, move, and manipulate the controls on a form.

**To select a control**

♦ Click the control (the textbox or the label).

♦ When you select a control, Access displays **'size'** and '**move'** handles around the control.

You can also select more than one control at a time. This is a convenient method for moving or aligning a group of controls.

**To select more than one control**

♦ Position the pointer beside (not on) one of the controls you want to select, left-click and hold, and then drag diagonally through all the controls you want to select. While you drag, Access draws a rectangle around the controls. When you release the mouse button, all the controls in or touching the rectangle are selected. (You can also select a group of controls by holding down the SHIFT key while you click each control in turn)

**To resize a textbox or a label**

♦ Drag the handles on the top and bottom to resize the textbox or label vertically.

♦ Drag the handles on the left and right sides to resize the textbox or label horizontally.

♦ Drag the handles in the corners to resize the textbox or label both vertically and horizontally.

♦ The size of the text box or label should be appropriate to the maximum length of the values to be displayed in the field.

**Move Controls**

All **Text box** controls on your form have attached labels. You can move a control and its attached label together, or you can move them separately.

**To move a textbox and attached label controls together**

♦ Click anywhere in the **Text** box control. The pointer (cursor) changes into a **cross**.

The Text box is highlighted as well as the **‘move’** handles of both the Label and the

Text control. Left-click and hold.

♦ Drag the control to any part of the form. Note that **Access** moves the Text box and its attached Label together.

♦ When the control is positioned where you want it, release the mouse button.

**Move a textbox or attached label control separately**

When you move a Text box or a Label not using **‘move’** handles both parts of the control move together. To move each part separately:

♦ Select the control (Textbox or Label part). When you select the Label part the label only is selected.

♦ Position the pointer over the Label or the Textbox **'move'** handle. When the pointer is over the handle, it changes to a cross. Left-click and hold.

♦ Drag the control to the required position;

♦ When the Label or Textbox is where you want it, release the mouse button.

**Add a Label and Edit a Label Text**

A *label* is a type of control you can place on your form to add information. A freestanding label is not bound to a field (like a text box) or attached to a control (like a text box's label). You can use the controls in the **Controls** group on the Design tab to place new controls on your form.

**To draw a freestanding label**

♦ Click the **Label** control in the **Controls** group in the **Design** tab.

♦ Move the cursor where you want to place the label on the form and left-click and drag.

♦ Type text in your label. As you type, **Access** sizes the label to fit your text. You can also resize the label control first and then type the text.

**To edit text in a label**

♦ Select the label by clicking it.

♦ Click the label again. Access places the cursor in the text of the label. Now you can edit the text. You can also now change the font type, size, characteristics (e.g. bold, italic), colour, etc.

Similarly to enter text in a text box: click the control to select and then click it again to insert the cursor in the box.

**To delete a Control**

♦ Click the text box to select both the label and the text box control.

♦ Note that to delete a label without deleting the text box, select the label.

♦ Press the **Delete** key on the keyboard or click the **Home** tab on the Ribbon and then click the **Delete** command in the **Records** group.

**To change the visual style of a control**

♦ Select the control and use the various **visual properties** of the control (back style, back colour, etc.)

**To lengthen a section (e.g. Header, footer, detail, etc) of a Form**

♦ Position the cursor on the lower edge of the section title. The cursor will change into a cross.

♦ Left-click, hold and drag the border up and down to lengthen or shorten the section size.

A form can be viewed in three ways: you use the form in **Form View**, minor changes to layout can be performed in **Layout View**, and more complex changes are made in **Design View.** Layout view allows you to edit the layout of the form and see the table data at the same time. You cannot see the data in **Design** view. Create a new form for the **Employee** table using **Form** button in **Forms groups.**

Open the new **Employee** form in design view (right click on its name, and select **Design**

**View)** and the layout shown in Figure 4.14 is displayed.

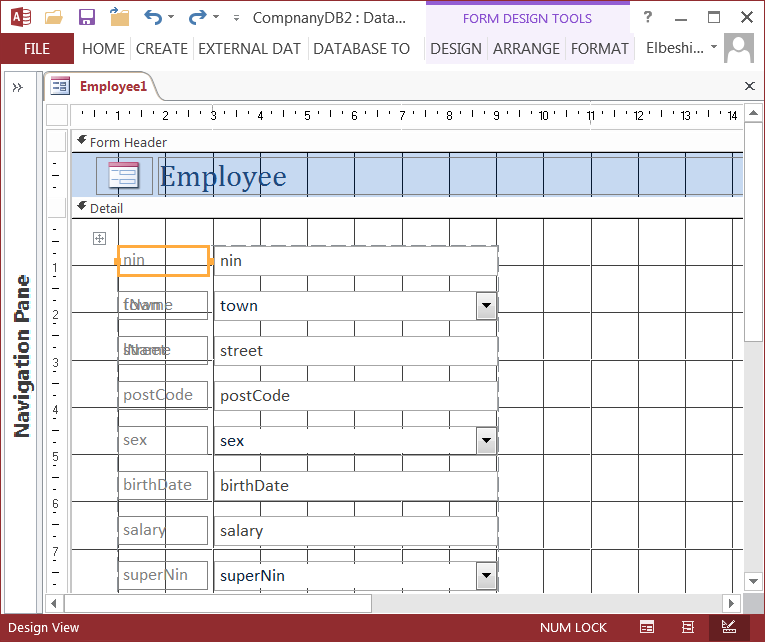


Figure 4.14

The information on a form is contained in ‘controls’. In Figure 4.14 there are label controls, text controls that take data from the Employee table, and Combo boxes (controls which allow you to select a value from a list, such as the Combo box for town, sex and deptNumber)

**Exercise 4.10**

♦ Edit the form in Figure 4.14 so that it looks like Figure 4.15

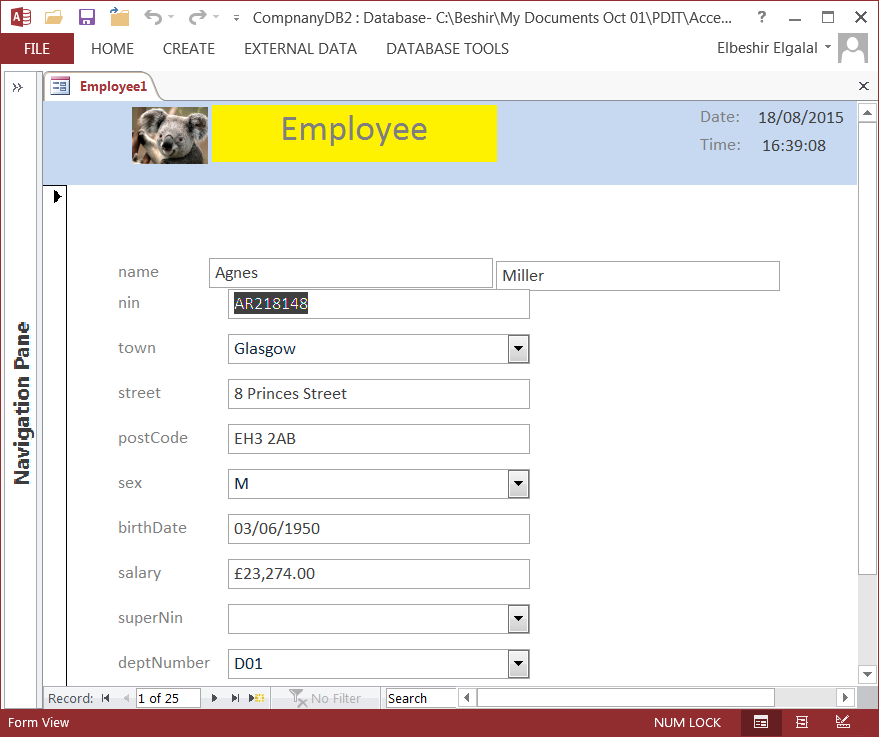


Figure 4.15

Changes to be made:

1) Change title

♦ Click the title label;

♦ Change text to Employees;

♦ Move title control to the middle of the Form Header;

♦ Click **Design** tab and then click **Property Sheet** button in **Tools** group;

♦ The properties sheet will be displayed;

♦ Change the following properties:

♦ Font size 22

♦ Text Align center

♦ Font Weight bold

♦ Back Color yellow or any other colour you prefer (click dotted button to display the colours palette)

♦ Change the label size.

2) Add a Logo on the left-hand side of Form Header

♦ Click the **Logo** button in **Header/Footer** group in **Design** tab;

♦ Choose a picture of your choice.

3) Add Date

♦ Click Text Box button in **Controls** group in under **Design** tab;

♦ Move mouse pointer to the top right-hand corner of the **Form Header** and then left- click. A text box and it label control will be displayed on the form. Select the Label control double click control to position cursor inside the control and write **Date:**

♦ Position mouse pointer inside the text box control (click text box control twice) and write **=Date().**

♦ Date() is a function that displays the current date.

♦ Select text box controls and set the following properties:

♦ Back Style - Transparent; Border Style – Transparent

4) Add Time

♦ Repeat same steps for Date except type: **=Time()** inside the text box control.

♦ Time() is a function that displays the current time.

5) Reduce size of more than one Control (texts or labels)

♦ **Control Layout indicator ** is still on

♦ Position mouse pointer just above first **label** (i.e. nin) until the pointer changes to a downward arrow . Left-click and all the label controls will be selected.

♦ Left-click on any right-side or left-side size handles (see Figure 4.8; cursor shape should change to ) and drag to the left/right. Release mouse button when desired size is reached.

♦ **Repeat** above instructions for the text box controls.

6) Turn off **Layout Control** (see exercise 4.6a) of the controls: **fName** and **lName**

When you create a from **Access** stack (group) all the controls so that they can be moved together using the Layout control (see Figure 4.1). To change size, position or move to individual labels and text box controls you need to turn the **Layout Control** off from the specific control(s). For example:

♦ To remove layout control from fName and lName controls do the following:

♦ Click fName label, press and hold the **Shift key** on the keyboard and then click fName text box control, lName label control and lName text box control.

♦ Click **Arrange** tab under **Form Design Tools** tab and then click the **Remove Layout** button in the **Table** group. The selected controls will be ungrouped and can be adjusted individually.

♦ Click any of the grouped controls (e.g. nin) and the control layout indicator will appear. Click and hold the **layout indicator** and then drag the stacked controls downward.

♦ Edit fName and lName controls and position them as shown in Figure 4.15.

7) You can also insert a picture as the form's background using the 'Picture' property.

**Exercise 4.11**

♦ Edit the structure of two other forms to suit your own taste (make use of the controls and form properties)

**4.5 Creating a Multi-table Form**

Multi-table form can be created between tables which have one-to-one or one-to-many relationship.

♦ Click **Create** then **Form Wizard.**

♦ Tell the wizard which tables or queries the form is to be based on. Click the downward arrow under **Tables/Queries** and click on **Table: Department.** Select the fields to appear on the form. It is easier to select them in the appropriate order at this point rather than reorder them, on the form itself. Select deptNumber, deptName, deptPhone, mgrNin, and startDate as in Figure 4.16

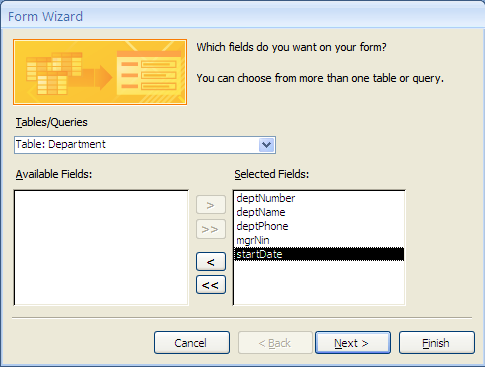


Figure 4.16

♦ Click the downward arrow under **Tables/Queries** and click **Table: Employee.** Select the fields to appear on the form, namely: nin, fName, lName, sex, birthDate, salary and superNin as in Figure 4.17. You don’t need the department number from

Employee as it would be repeated for every employee.

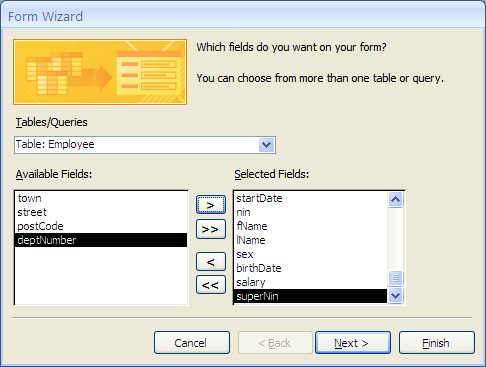


Figure 4.17

♦ Click **Next** and Figure 4.18 is displayed. Click **Next** and a form with a subform is displayed, with **Department** as the form, and **Employee** as the subform.

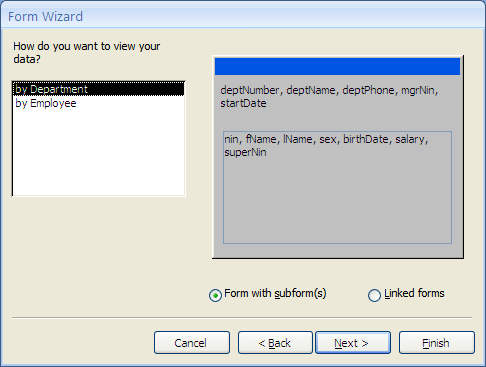


Figure 4.18

♦ Choose **Tabular** layout and click **Next** in the window that opens. Change the form name to DepartEmployee (or any name of your choice that reflect that the form is a multi-table form) and click Finish. The form in Figure 4.19 will be displayed.

♦ If you want to modify the form you should chose the second option (Modify the form's design) before clicking Finish.

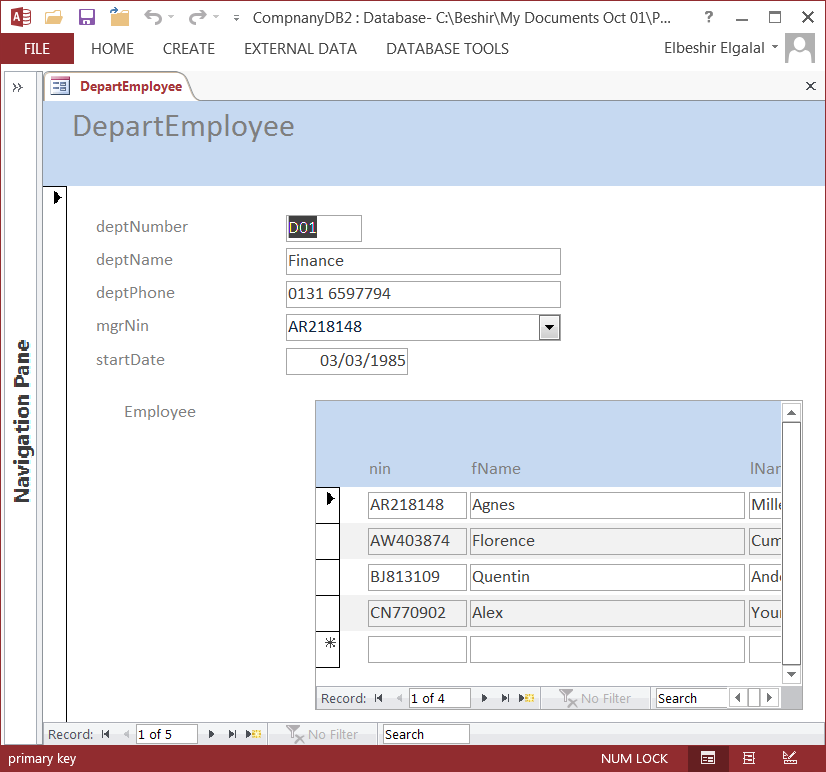


Figure 4.19

**Exercise 4.12**

♦ Create the following **multi-table forms** and edit their layout to suit your taste:

♦ A multi-table form based on the tables: Department and Project.

♦ A multi-table form based on the tables: Employee and Dependent.

♦ A multi-table form based on the tables: Department, Employee, and WorksOn.

**4.6 Using Forms**

We can use a form to add, delete and update records in a table. Open the Employee form in Form view as shown in Figure 4.20. Hold the cursor over some of the buttons (e.g. Last record, First record, etc.). After a short pause a ‘**Tool Tip’** appears telling you what the button does. If you can’t remember or don’t know what a button does, that’s a quick way of finding out. There are 25 records in the **Employee** table. Click on the **Next record**, **Last record, previous record** and **First record** buttons to scan through the records.

**Exercise 4.13**

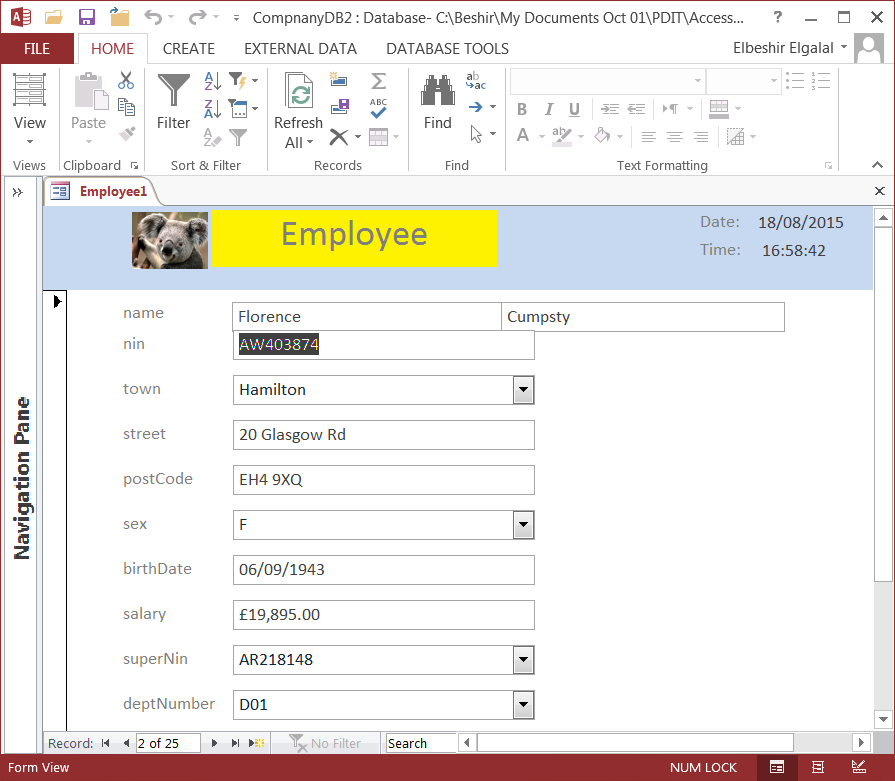
♦ Insert the following record in the table using the form. You should make use of the

Combo boxes when entering data in the fields: town, sex, superNin and deptNumber.

First name: **Peter** Last name: **Murphy** nin; **YX123456** Town: **Paisley** Street: **10 New street,** Post code: **PA1 2BE** sex: **M** birthDate: **25/08/1954** superNin: **CW954608** Salary: **£20000** deptnumber: **D02**

Close

Save Record selector



First Record

Previous Record

Record Number box

New Record

Last Record

Next record

Drop down boxes

Figure 4.20

Press **Save** to save the record.

If at any time, you wish to abandon your attempt to add a new record, press **Esc** on the keyboard.

**Exercise 4.14**

Using the Employee form try to input records with any of the following values (one at a time)

and see what happens:

♦ An existing National Insurance Number;

♦ Gender other than M or F;

♦ A superNin of non-existent employee (e.g.IU999999);

♦ A negative annual salary;

♦ A department number that does not exist.

**Exercise 4.15**

♦ Open the Project form and add a new record;

♦ Delete the record.

To delete a record you must select it first. Click the **Record selector** (see Figure 4.20). The **Selector** pane will go black. Press the keyboard **Delete** key (or the Delete button in the Records group in the **Home** tab of the Ribbon). Access will ask you to confirm that the record is to be deleted. Click **Yes** to confirm.

It is time to create your own User Interface **Start-up** form which you can use to open any object you have created (i.e. tables, forms, queries, reports, macros, etc.). Two **options**:

1) Create one blank form and put on it all the required controls to open the database objects (see Figure 4.21);

2) Create a number of forms linked in hierarchical structure.

Main Form

Form1 Form2 Form3

.................

Form**n**

Form3.1 Form3.2

etc.

Any form can have controls invoking other forms or controls invoking database objects or both.

The first option is suitable if we have few objects that require controls that can fit in one form only. Option2 is more flexible. Use **option2** to create your user interface form. The option is explained in more details in the following pages.

The form in Figure 4.21 uses **option1** and shows only some of the object of the database. Even for a small database like the CompanyDB database it is not possible to put all required controls on one form in an easy usable format.

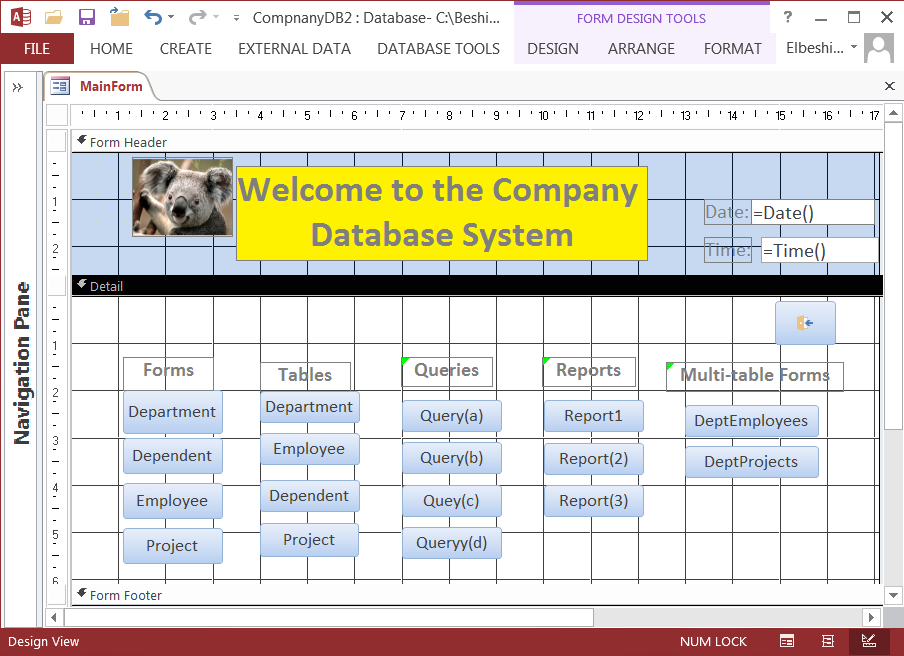


Figure 4.21

You can put a background picture of your choice to the form as shown in Figure 4.21a by setting the picture property of the form. The button with arrow on top right-hand side corner of the form is the Exit button. When the button is clicked the form is closed.

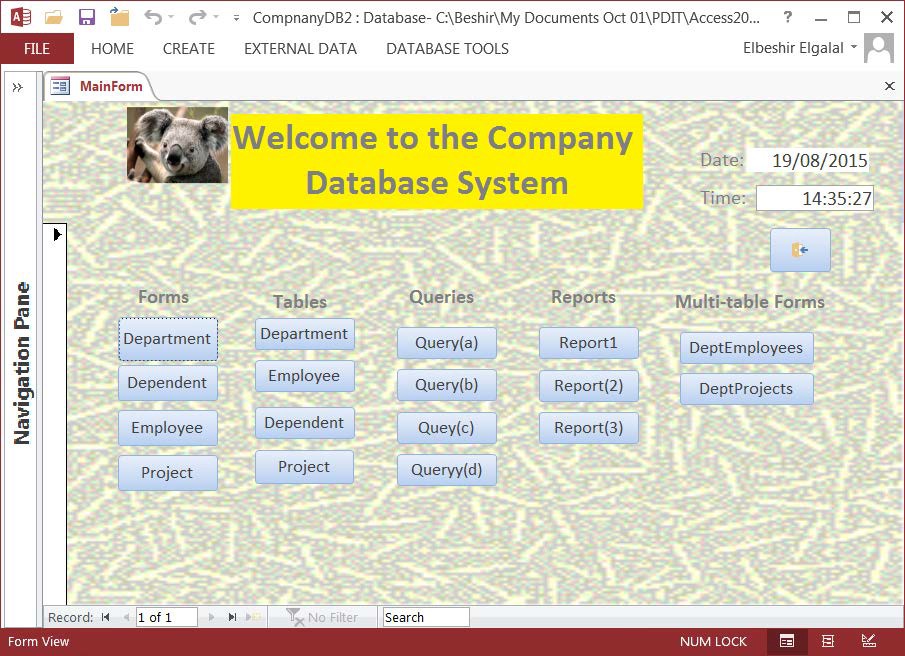


Figure 4.21a

**Option 2**

In this option we start with a main form (see Figure 4.22) and make the controls placed on it

open other forms on which we place controls that open the database objects or other forms.

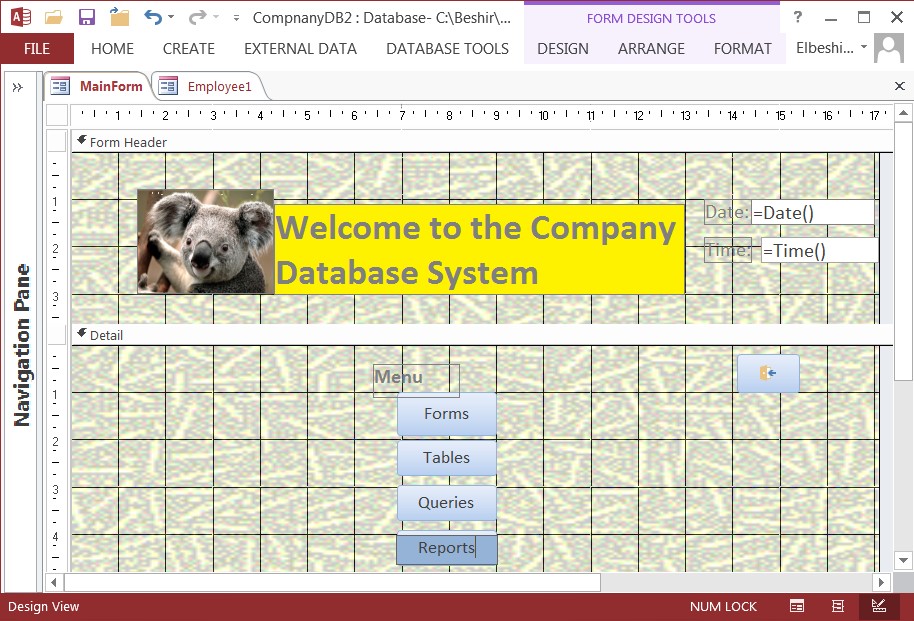


Figure 4.22

When you press the 'Forms' button the form in Figure 4.23 will open and the main form will

close.

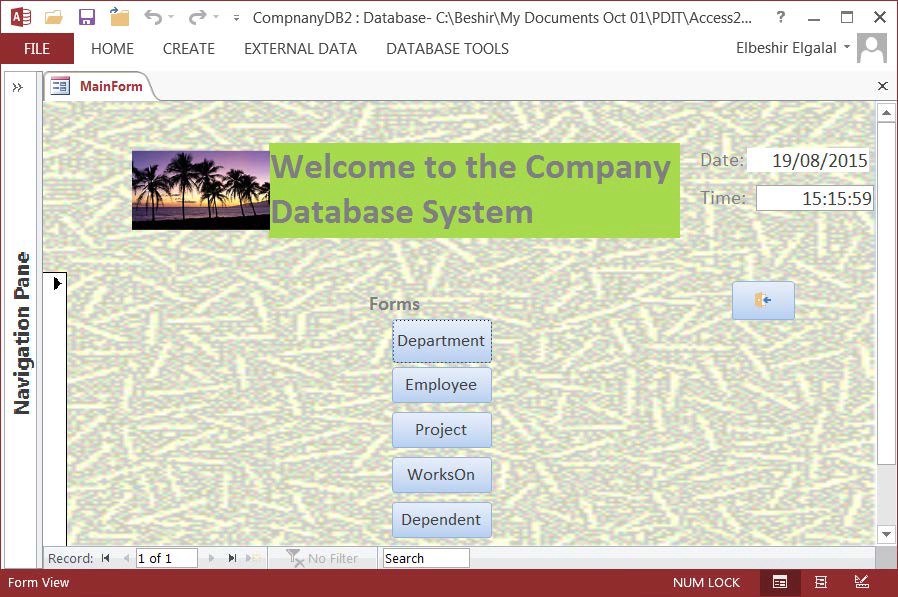


Figure 4.23

Similarly when you press the 'Tables' button a form with a set of controls will open allowing you to open the table objects in the database. Each of the other buttons opens a form. To avoid a clutter of windows (many forms are opened on the screen) only one form should be open at any one time. Therefore when you open a **child** form you should close the **parent** form and when you close the child form you should open the parent form.

Opening and closing forms or opening other database object (e.g. queries, tables, etc.) requires that a **Macro** be associated with the button so that when the button is clicked the instructions (actions) in the macro are executed. What is a macro??

A macro is a tool that allows you to automate tasks and add functionality to your forms, reports, and controls.

**Steps to create a Macro and attached to a button:**

♦ Create a macro with all the commands you want to be executed and save it in the database;

♦ Open form in Design view, select the button you want to attach the macro to, and display its properties;

♦ Click '**Event'** tab on Properties menu;

♦ Update **'on Click'** property: click the downward arrow and select the macro you want to associate with the button. This means whenever the button is clicked the all the commands in the macro will be executed.

**How to create a Macro in Access 2013**

♦ Click **Create tab** on the Ribbon.

♦ Click **Macro** button in the **Macros & Code** group and the window in Figure 4.24 will open.

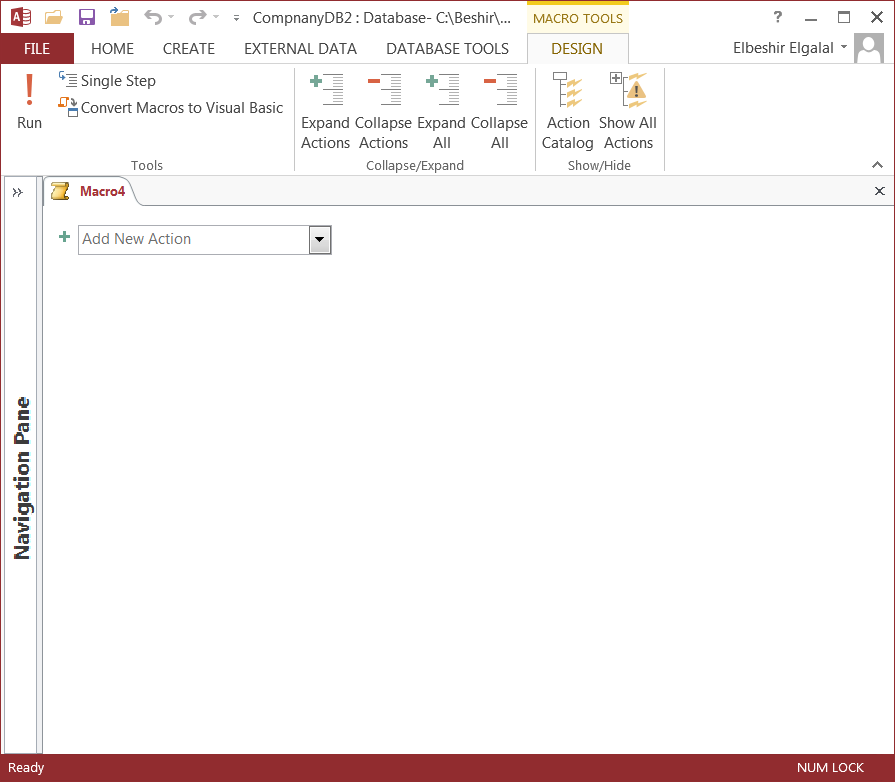


Figure 4.24

♦ Click **'Action Catalog'** button under **Design tab** in **Show/Hide** section a window

(within the current window) with three groups will open as shown in Figure 4.25.

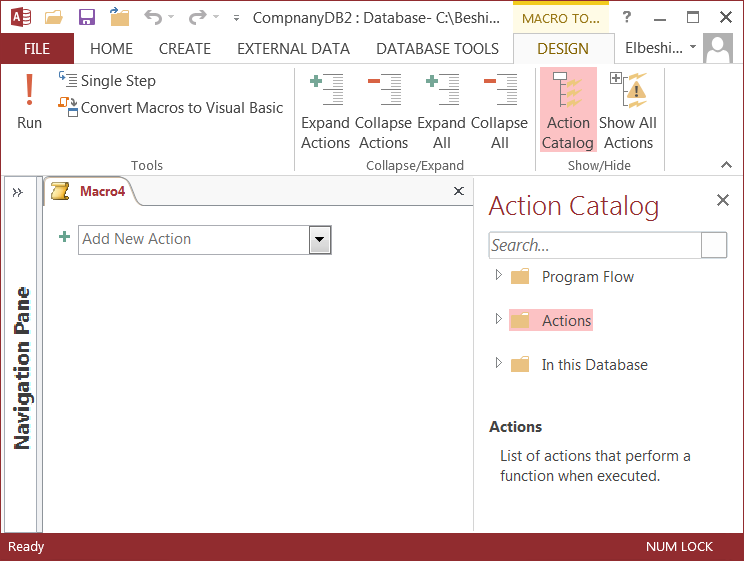


Figure 4.25

♦ Place the cursor over each link or click the link to see text explaining what the section consists of.

♦ **Program Flow** - section list advanced features to use when creating a complex macro.

♦ **Actions** –section consists of many actions grouped in many groups;

♦ **In this Database** – list object and the associated macros in the current database.

♦ **Two ways to add actions to the macros:**

1. Expand the groups, find and double click the action you want to perform;

2. Click the downward arrow next to the box on left-hand side pane and select the action you want to perform (i.e. OpenForm, OpenTable, OpenReport, etc.)

In both cases if the action selected is 'OpenForm' the macro window will change to the one in Figure 4.26.

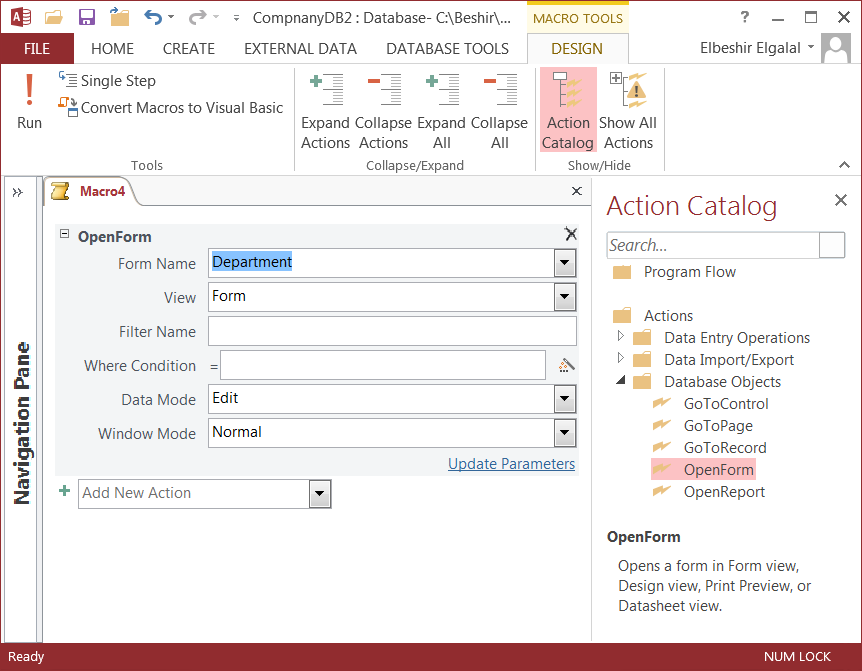


Figure 4.26

♦ Choose the form name, the view in which you want to open the form (default is Form view) but form can be opened in **Design** view, **Print Preview** or **Data Sheet** view.

♦ Choose **Edit** mode for Data mode. Move the cursor over any of the boxes to find more information. Ignore **Filter Name** and **Where Condition** boxes.

♦ If you want the macro to execute more than one action choose the next action and fill required boxes. Repeat until you finish all required actions.

♦ To close any open window (i.e. form, table, query or report window) click the downward arrow and select **CloseWindow** action. See Figure 4.27.

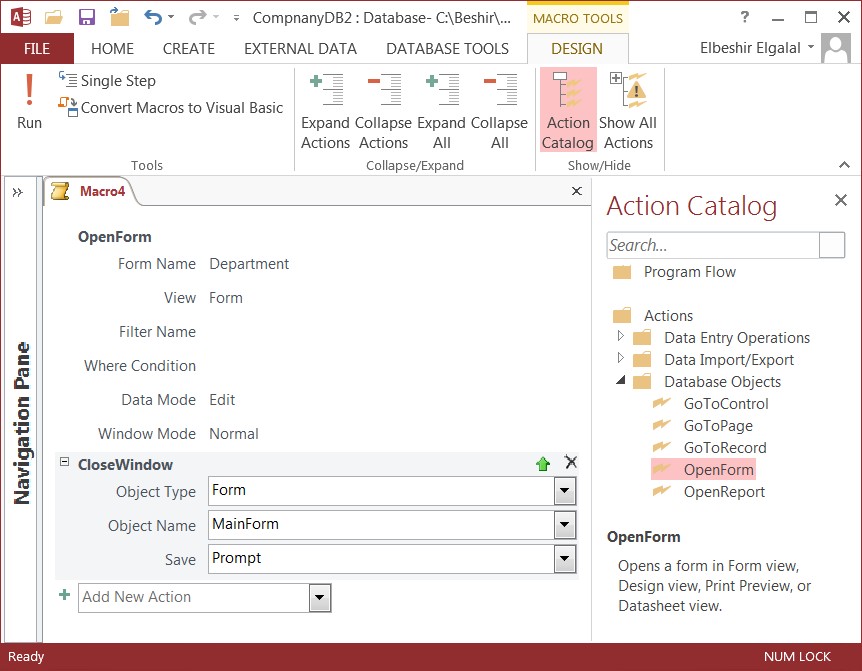


Figure 4.27

♦ Click **x** at the right-hand side of the Macro pane (not Action Catalog pane) and in the window that opens type a name or just accept the suggested name and click **OK** button.

Consult **Access Help** for more information on Macros.

**Exercise 4.16**

♦ Create a Start-up form from scratch using Design Tools and following Option 2. Save it and call it MainForm (or StartupForm.

♦ The form should open automatically when you open the **Company** database.

To make the main form open automatically do the following:

♦ Click File (top left corner of Access window) and then click **Options** from the drop- down menu;

♦ In the window that opens click **Current Database** button on the left-hand side pane;

♦ Click the downward arrow head in the box next to **Display Form:** and select the name of your **main** form.

♦ You can change the title and the image to be displayed in at the top of Access main window using the **Application Title** and **Application Icon** boxes.

**Note:**

♦ The main form that opens automatically should have the necessary buttons to open all objects in the database and close the form.

♦ Each button (on the main form) when pressed should close the parent form (main form) and open another form that have a button for each object you want to open and a button to close the form and open the parent form;

♦ Use your own imagination and liking (i.e. choice of colours, layout, etc.) when creating the forms. Your design does not need to follow the given examples;

♦ Make use of the design tools and **Access Help**;

♦ To start click **Create** tab on the Ribbon, then click **Form Design** button in the **Forms**

group and the window in figure 4.28 is displayed.

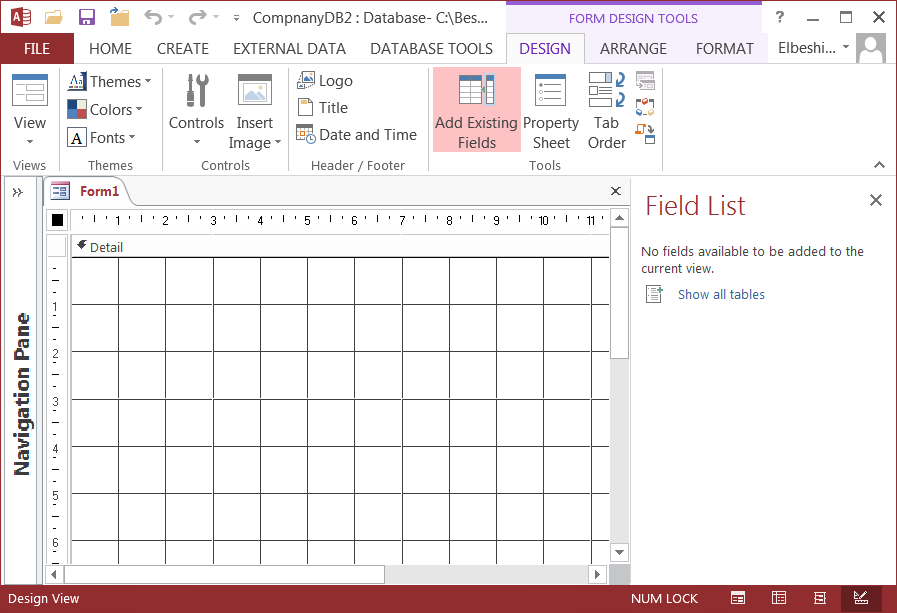


Figure 4.28

**Practical 5: Creating Queries**

**5.1 Introduction**

In this practical you will learn how to create simple and complex queries. To perform the exercises in this practical you should have created all the tables of the Company database, populated the tables with data, created the relationships between the tables and enforced referential integrity. In the last set of exercises you will use another database application.

Access 2013 provides two ways to begin creating a new query:

♦ Using Query Wizard;

♦ Using Query Design.

You will use the **Query Design** option in this practical. Queries can be created using Query-By- Example (**QBE**) tool or writing the query using Structured Query Language (**SQL**) syntax. To create a query:

♦ Start the database system (i.e. Company database);

♦ Click **Create** tab on the Ribbon and then click **Query Design** button in the **Queries**

group. The Query window together with **Show Table** window is displayed as shown in

Figure 5.1.

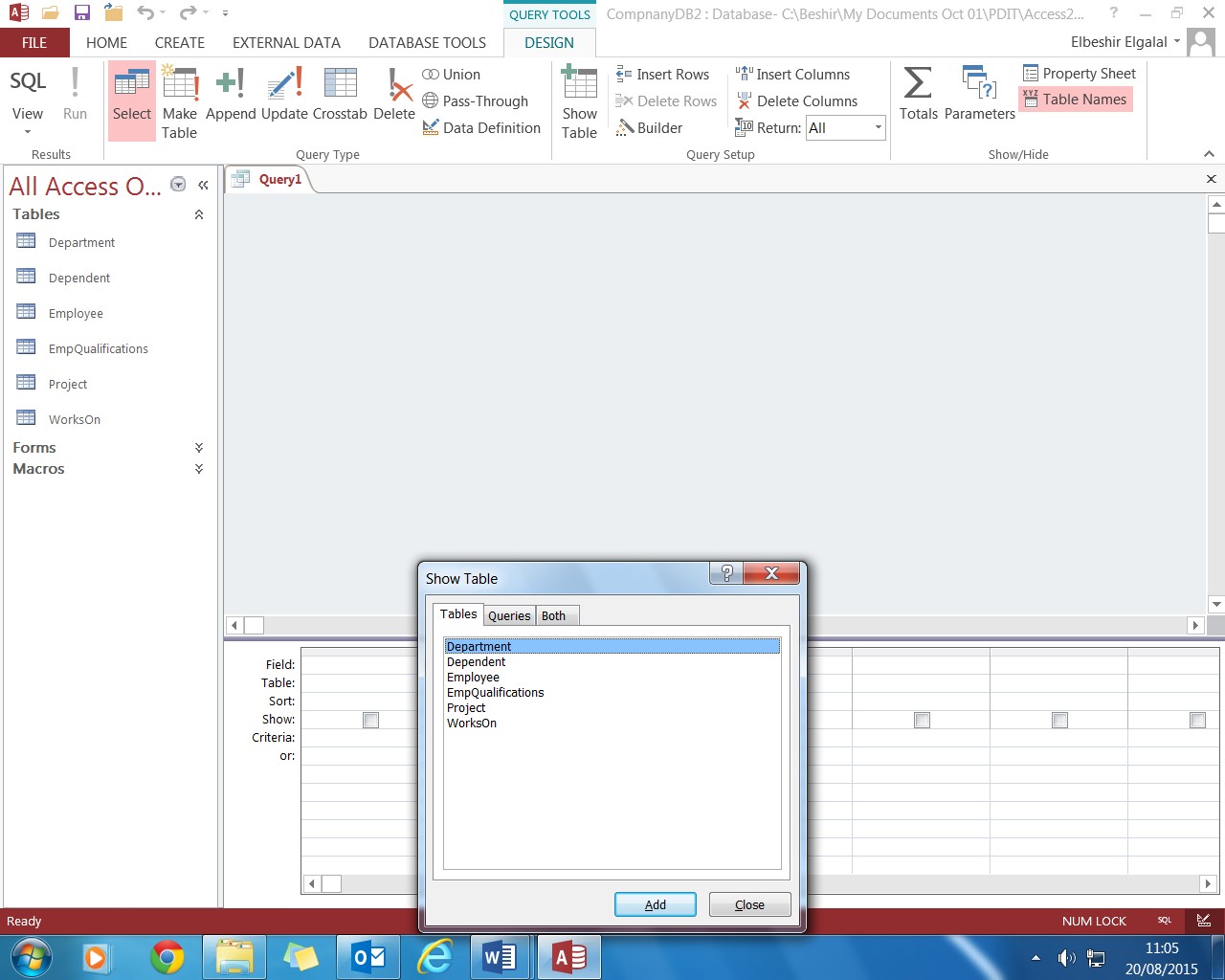


Figure 5.1

At this point you must decide how you are going to create the query:

♦ Option1: use QBE tool.

♦ Option2: write the query using SQL.

From Figure 5.1 you can see that option1 is the default and that the default query type is **Select**

query (highlighted in orange on the Ribbon)

**5.2 Using QBE Tool**

If **Show Table** window is still open click **Employee**, then click **Add** button and then click **Close**

button. The Query window in design view will change as shown in Figure 5.2.

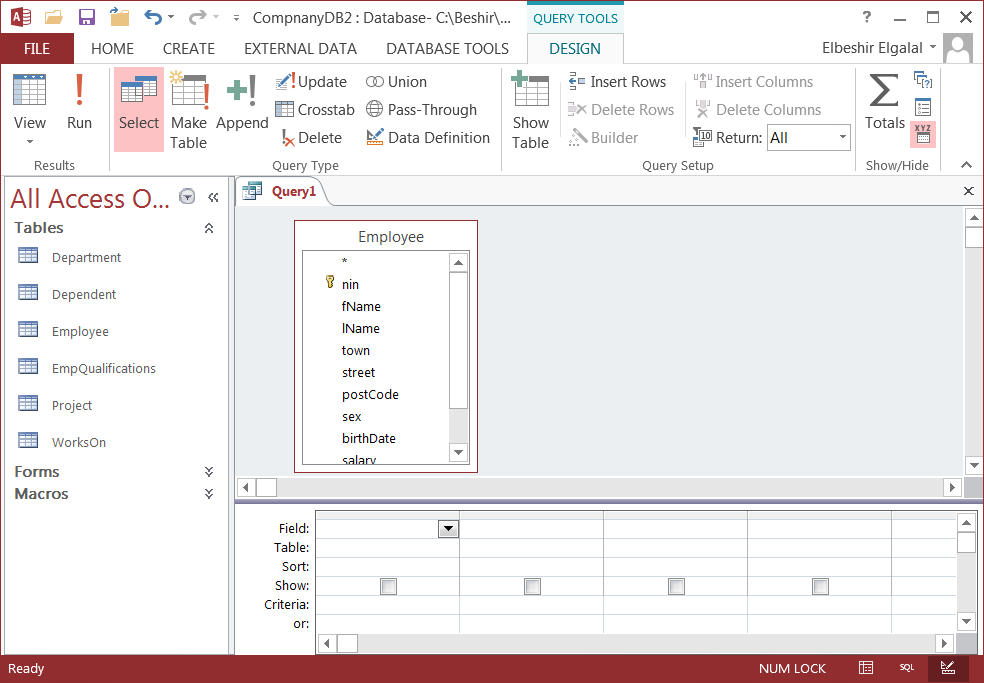


Figure 5.2

**The Query Window**

The Query window in design view has two main parts. In the upper part you add the field list for the tables or queries you chose for a specific query. The lower part of the window is the design grid, in which you do all the design work. Each column in the grid represents one field in the query. A field can be a simple field from one of the tables or a calculated field based on several fields in the tables.

♦ You use the first row of the design grid (**Field:)** to select fields – the fields you want to see in the resulting **records set** when the query is executed. You can select the fields in a number of ways:

1. Click in the downward arrow head at the end of the field and select the field name from the drop down list. If the arrow head is not visible click in the field first.

2. Click and hold the field name in the table field list in the upper part of the Query window. Drag the field to the desired column.

3. Double click the field name in the table field list in the upper part of the Query window. The field will be placed in the next available column.

4. At the top of each table fields' list in the upper part of the Query window (and also next to the first entry in the Field drop-down list in the design grid) is an asterisk (\*) symbol. This symbol is shorthand for selecting "all fields in the table or query" with one entry on the **Field** line (table name followed by a dot (.) and an asterisk (\*)).

♦ Note you can add individual fields (from the same table) to the grid in addition to the asterisk in order to define criteria for those fields, but you should clear the **Show** check box for the individual fields so that they don't appear twice in the records set.

♦ The second row (**Table:**) shows you the name of the table from which you selected a field. If you don't see this row, you can display it by clicking **Table and Names** in the **Show/Hide** group on the **Design** tab below **Query Tools.**

♦ In the Sort row (**Sort:**), you can specify whether **Access** should sort the selected or calculated field in ascending or descending order.

♦ In the Show row (**Show:**) you can use the check boxes to indicate the fields that will be shown in the query result. Sometimes you want to add a field to the grid to allow you to select the required records (the field must satisfy a certain criteria) but you don't want to display the filed in the result. By default every field added to the grid will be displayed in the result unless you clear the **Show** check beneath the field to exclude it from the result.

♦ If you want to use aggregate functions (Group by, Count, Sum, etc.) in your query click **Totals** button in **Show/Hide** group in the **Design** tab under **Query Tools.** A new row will be added to the design grid. If you then click the downward arrow in any field a drop down list of aggregate functions will be displayed.

♦ Finally, you can use the Criteria row (**Criteria:)** and the row(s) labelled **Or:** to enter the criteria you want to use in the query. If more than one criteria are specified in the Criteria row they are ANDed (e.g. criteria on field A AND with criteria in field B AND ..., etc.), while criteria specified in a field in the '**or'** row are ORed with the criteria for the same field in the

**'Criteria:'** row.

**Exercise 5.1**

♦ List the first and last names, town and salary of all female employees who are employed by department number three.

Do the following:

♦ Click **Create** tab and then **Query Design** button in **Queries** group;

♦ Click **Employee** in **Show Table** window and then click **Add** button. Click **Close** button to close **Show Table** window;

♦ Double click the fields: fName, lName, town, sex, salary and deptNumber in the table fields list;

♦ In the **Criteria** row type F in the 'sex' field and D03 in deptNumber field (this is translated to sex = 'F' AND deptNumber = 'D03' in the query statement)

♦ The Query window will look like Figure 5.3.

♦ Click **Run** to run the query or click **View** button in the **Results** group in **Design** tab under **Query Tools** and click **Datasheet view** to execute the query Or simply click the **View** button). Query result is shown in Figure 5.4. The result is shown in Datasheet View format.

♦ Close and save the query giving the query a suitable name (e.g. query5\_1)

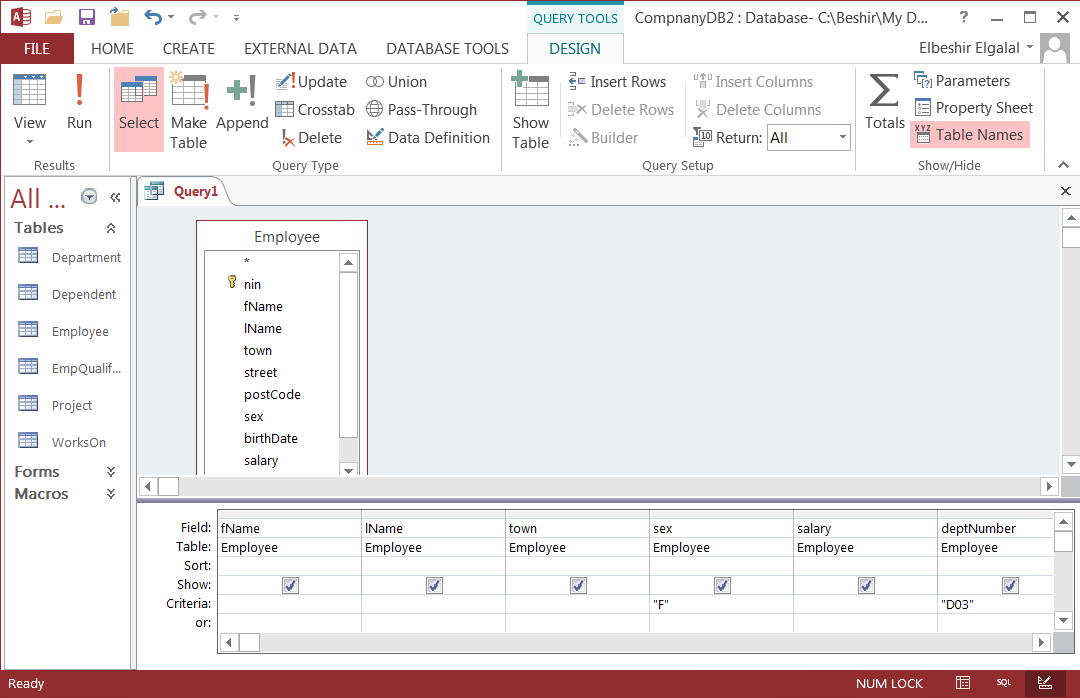


Figure 5.3 Query design

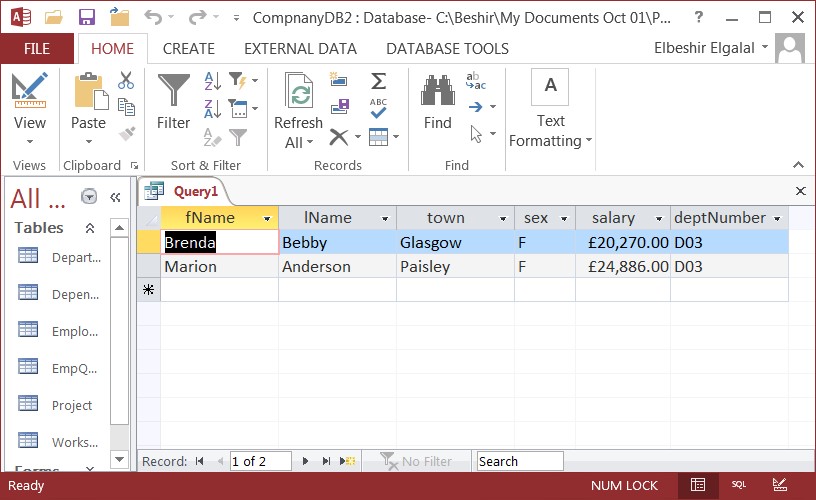


Figure 5.4 Query result

To view the query in design view click the **View** button in the **Home** tab. Click the downward arrow under the **View** button to see the different views in which you can display a query.

**Exercise 5.2**

Retrieving data from more than one table.

♦ List nin, first name, last name, date of birth and salary of all employees who work in the

IT department.

Two tables are required for this query: Employee and Department tables. Do the following:

(a) Click **Create** tab and then click **Query Design** in **Queries** group; (b) Select Employee in Show Table window and click **Add** button;

(c) Select Department in Show Table window and click Add button. Click **Close** button to close Show Table window and the Query window will look like Figure 5.5.

(d) Double click the required attributes from Employee and department field lists in the upper part of the Query window.

(e) In the **Criteria** row type IT in deptName field

(f) Click **Run** button to execute the query. (g) To edit the query open it in design view.

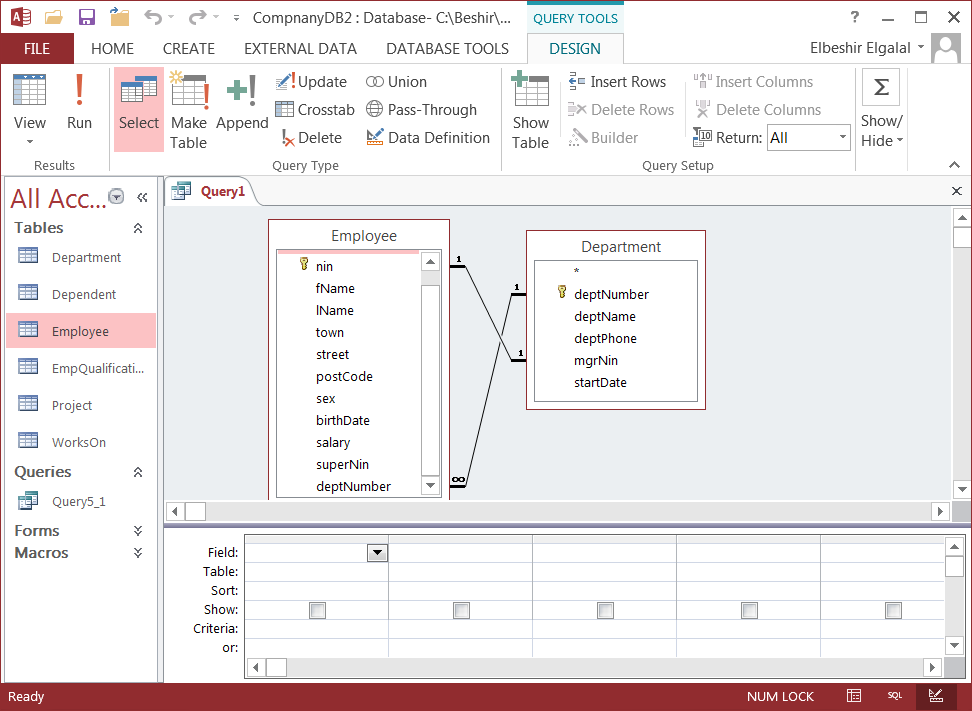


Figure 5.5 Two tables query

The query will produce **wrong** result. The data of the employee who manages the IT department and not the records of the employees who work in the IT department are displayed. This because there are two relationships between the Employee and Department (see Figure

5.5). One represents Employs relationship (Department Employs many Employees) and the second represents Manages relationship (an Employee may manage one department).

To get the right answer delete 1:1 relationship line in the Query window (click on 1:1 relationship line and press Delete button on keyboard). Click **Run** button and query will produce the right answer.

**Exercise 5.3**

♦ For every Employee who manages a department list the employee first name, last name, salary, department name and start date when he/she started managing the department.

1) Repeat steps (a) to (d) of exercise 5.2 and then run the query.

Although the query will produce the correct result the query is not efficient (view the query in

SQL view). The tables are unnecessarily joined on department number.

2) Remove 1:\* relationship link in the Query window and run the query again. View the query in

SQL view and it should be as shown below.

SELECT Employee.nin,Employee .fName, Employee.lName, Employee.birthDate, Employee.salary, Department.deptName

FROM Employee INNER JOIN Department ON Employee.nin = Department.mgrNin;

**Note:**

♦ In SQL view **Access** qualifies every field name with the field's table name. This is how **Access** produces SQL statements. In standard SQL fields names are qualified by the field's table name **only** if the field name is the **same** in more than **one table** that are used in the query.

♦ Access also uses INNER JOIN option. The same query can be written using **Equijoin** (using equality operator in WHERE clause). Using Equijoin and no field qualification (see lecture notes) the query in (2) above will be:

SELECT nin, fName, lName, birthDate, salary, deptName

FROM Employee, Department

WHERE nin = mgrNin;

**5.3 Write a Query Using SQL**

♦ Open the database system;

♦ Click **Create** tab on the Ribbon and then click **Query Design** button in the **Queries** group.

The Query window together with **Show Table** window is displayed as shown in Figure 5.1.

♦ Click **Close** button on **Show Table** window to close the window and click **SQL** button in

**Results** group on the Ribbon. The Query window will change as shown Figure 5.6.

♦ Write SQL statement, run it, close and save it.

♦ To edit the query open it in design view.

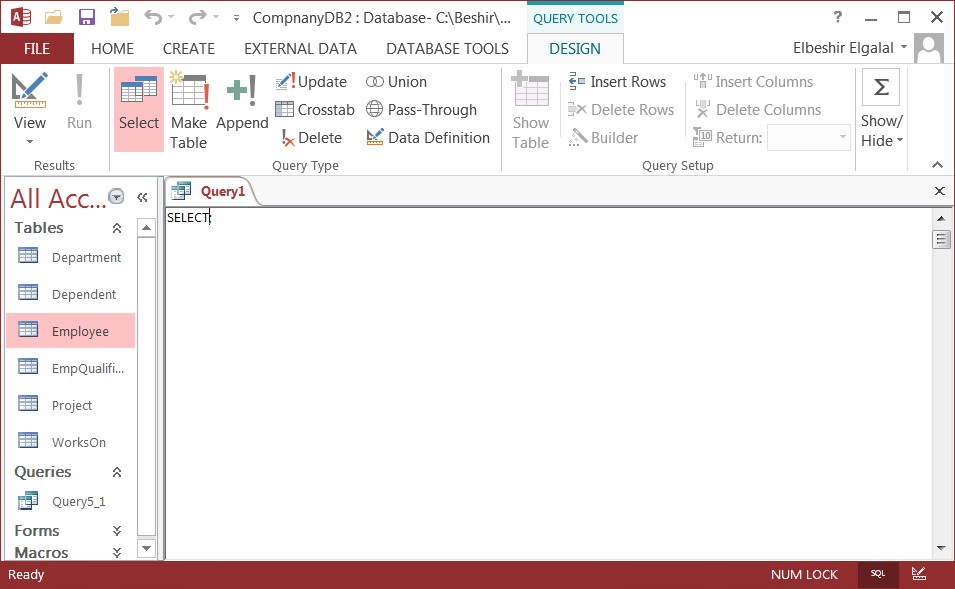


Figure 5.6

In the following exercises you **should:**

♦ Make use of the schema of the database you are using;

♦ Work out form the schema how you will solve the query and then write the query in SQL

or use QBE tools to do it.

It is advisable that you should try to solve the queries using SQL and **not** QBE. Make use of a table **alias** when using more than one table in a query.

**5.4 Queries for the Company Database**

Using the **Company** database application create and run the following queries: (a) List name, sex, date of birth and salary of all employees.

(b) List name, sex, date of birth and salary of all male employees born after 1950.

(c) List name, sex, date of birth, town, salary and department number of all employees who live in Glasgow

(d) List all details of each department.

(e) List name, town, post code, date of birth, and department number of all employees whose salary is more than 16000 but less than 25000.

(f) List national insurance number and qualification of all employees with BSc or MSc qualifications.

(g) List first name, date of birth and relationship of all female dependent born before 1980.

(h) List first and last names, town, date of birth and salary of all employees who work for the

Engineering department.

(i) List project name, budget and start and finish dates of projects controlled by IT or

Engineering department.

(j) List department name, phone number and manager start date of the department managed by Anges Miller.

(k) List (under appropriate headers) the minimum, maximum and average salaries of all employees.

(l) For each department list department number and the number of employees (under an appropriate header) who are employed by the department.

(m) For each department that controls more than one project list the department number and number of projects (under an appropriate header) it controls.

(n) For each project list the project name and the total number of hours spent on that project.

(o) For each employee with more than one qualification list the employee national insurance number, last name and the number of qualifications he/she has.

(p) List each qualification and the number of employees (under an appropriate header) who has the qualification.

(q) List the average salary of all female employees.

(r) List the names of employees who have a dependent with the same first name as themselves.

**Use Subqueries when creating the following queries**

(s) List first and last names, sex and department number of employees whose salary is greater than the average salary of all employees.

(t) List the names and department number of all employees who don't work on any project. (u) List the names and sex of all employees who don' have any dependent.

(v) List project name, budget, and start and finish dates of all projects with a budget greater than the average budget.

(w) List the names of all department managers who have no dependents.

(x) List national insurance number, first and last names, and salary of employees who supervise other employees.

(y) List the names, sex and date of birth of all employees who are supervised by Max

Beckham.

**5.5 Queries for ClydeRental Database**

Copy the database file ClydeRental.mdb from Blackboard to your H: drive. The file is stored in: SM2 Fundamentals of Database Systems  Course Materials  SQL & QBE  Databases

 ClydeRental.mdb

Do the following to copy the file to your H: drive:

♦ Right-click the file and select 'Save target As ....'

♦ Navigate to the H: drive and save the file.

Using ClydeRental database system create and run the following queries: (a) List details of all properties.

(b) For each branch located in Glasgow or Paisley list the branch number, phone and fax numbers, and city.

(c) List name and salary of all female staff.

(d) List property number, type and rooms for all properties in Glasgow with rent less than 600 pound.

(e) List properties in Glasgow with rent between 350 and 450 pounds. (f) List the first and last names of all staff and their qualifications.

(g) List the first and last names of all owners and the property number and city of their properties.

(h) List details of all properties offered by a branch located in Aberdeen.

(i) For each owner who owns a property offered by a branch located in Glasgow, list the owner first and last names, and the property number type and rent.

(j) List name, sex, date of birth, and salary of each branch manager and the date he/she started managing the branch.

(k) List, under an appropriate header, the sum of all staff salaries.

(l) List, under an appropriate header, the number of staff in each city.

(m) List, under an appropriate header, the average rent of each type of property in each city. (n) List, under an appropriate header, the average salary of all staff working in branch B4.

(o) For each branch list the branch number and the average salary (under an appropriate header) of all staff.

(p) For each branch list the branch number and the number of staff (under an appropriate header) who works in the branch.

(q) For each branch which employs more than 2 members of staff, list the branch number and the number of staff (under an appropriate header).

(r) For each staff list the staff number and name and his/her age in days, weeks, and years

(rounded to the nearest integer).

(s) For each staff list the staff number and name and his/her monthly salary (rounded to the nearest integer).

**Practical 6: Creating Reports**

**6.1 Introduction**

In this practical you will learn to create Reports. Forms allow you to view single records or a small group of related records displayed on the screen in an attractive way, but forms aren't the best way to print and summarise large data in your database. Reports can be based on **tables** and **queries.**

Reports are the best way to make a printed copy of information taken from the database and can be used to summarise and group data. Reports can be created using three methods:

♦ Design view;

♦ Report command;

♦ Report wizard.

**Design view** method is a long and tedious method where the user has to design and create the report by hand from scratch. **Report command** is an easy and quick way to create quality reports. Similar to the quick create form commands, the Report command is a **one-step process** – you're not presented with any options or dialog boxes; Access simply creates a generic report with one click. Reports created using the Report command are displayed in **Layout** view. Once created they can be edited in **Layout** view or in **Design** view.

**Exercise 6.2**

♦ Create a report based on **Project** table using the **Report** command. Do the following:

♦ Click **Project** in the **Tables** section of the **Navigation** pane;

♦ Click **Create** tab on the Ribbon and then **Report** button in the **Reports group** and the report in Figure 6.1 is displayed.

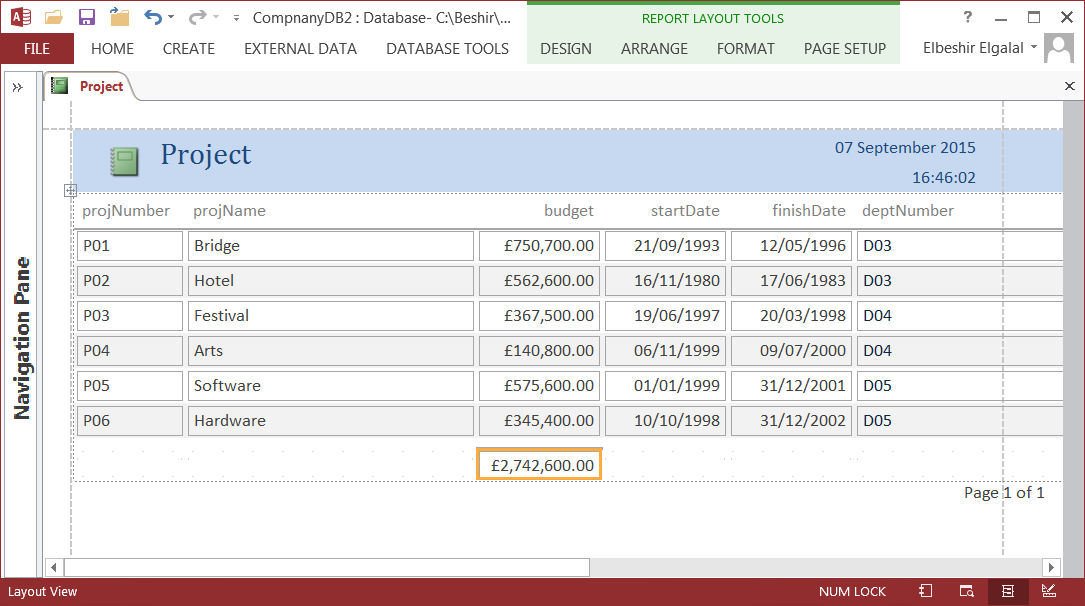


Figure 6.1 Report for the Project table in layout view

Click the downward arrow head on the **Views** tab on the Ribbon. Four views for reports are displayed, namely:

♦ **Report View** shows you the report on the screen

♦ **Print Preview** lets you see what the report will look like prior to printing.

♦ **Layout View** allows you to make basic changes to the appearance of the report.

♦ **Design View** allows more complex customization features (adding fields, functions, buttons, etc.

Initially the report is in **Layout** view. The dotted lines show the page margins. The layout can be edited. You can change the name of the columns, the title, the size of the columns, etc. To remove the budget field total you need to view the report in Design view and remove the textbox with the **Sum** function from the **Report Footer.**

In **Report** view the page margin lines will disappear. In **Report** and **Layout** views you can use the filtering capabilities of **Access** to drill down to subsets of the data. For example to see all the projects controlled by department number 03, place the cursor on the first value D03 in deptNumber column, right-click and select **Equals "D03"** from the drop down menu (see Figure

6.2). Only the records with D03 will be shown (See Figure 6.3). To remove the filter click **Toggle Filter** button in the **Sort & Filter** group on the **Home** tab. Access will again display all the records.

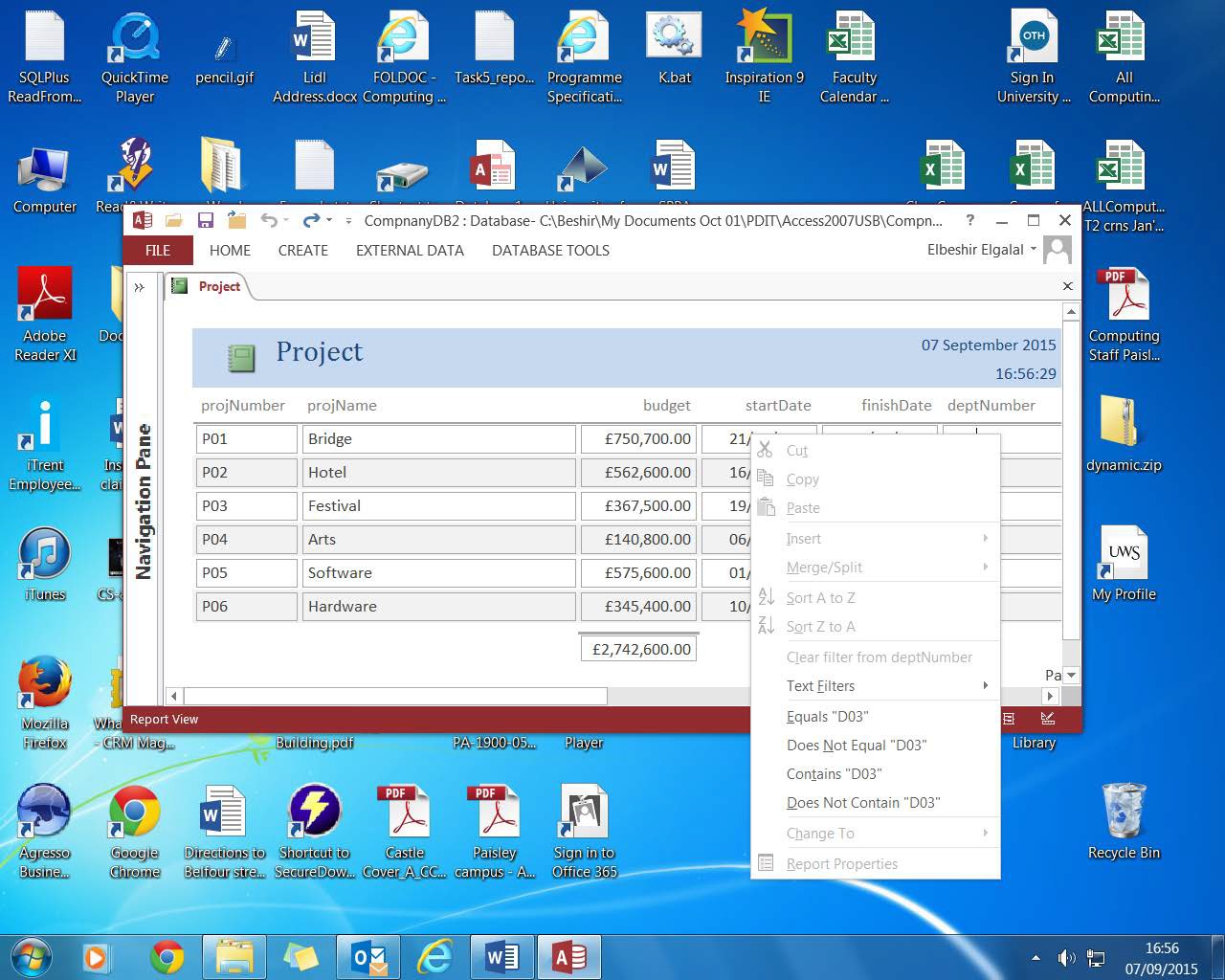


Figure 6.2

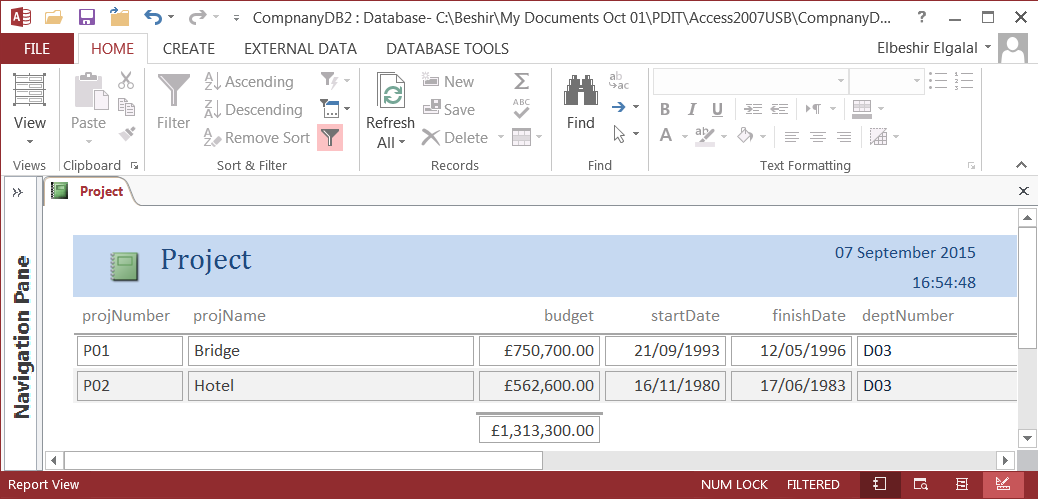


Figure 6.3 Projects controlled by department 03

**Exercise 6.2**

♦ Create a report for each of the tables: Employee, WorksOn, Dependent and

EmpQualifications tables using the **Report** command.

♦ Open **Employee** report in Layout view;

• Edit the report so that all the report columns are displayed in one page (use

**Landscape** layout).

**6.2 Grouping, Sorting and Totalling Information**

A key way in which reports differ from forms is that on reports you can group information for display using **Grouping & Sort** commandsin **Grouping & Totals** in the **Design** tab under **Report Layout Tool** (appears when you open a report in layout view).

**Exercise 6.3**

**a)** Open **project** report in layout view;

**b)** Click **Group & Sort** button to open the **Group, Sort and Total** pane beneath the report grid;

**c)** Click **Add a Group** button and choose **deptNumber** from the drop down list (if list does not

automatically appears press the downward arrow head and select deptNumber);

**d)** Click **Add a Sort** button and choose **ProjName** from dropdown list and the report in Figure

6.4 is produced.

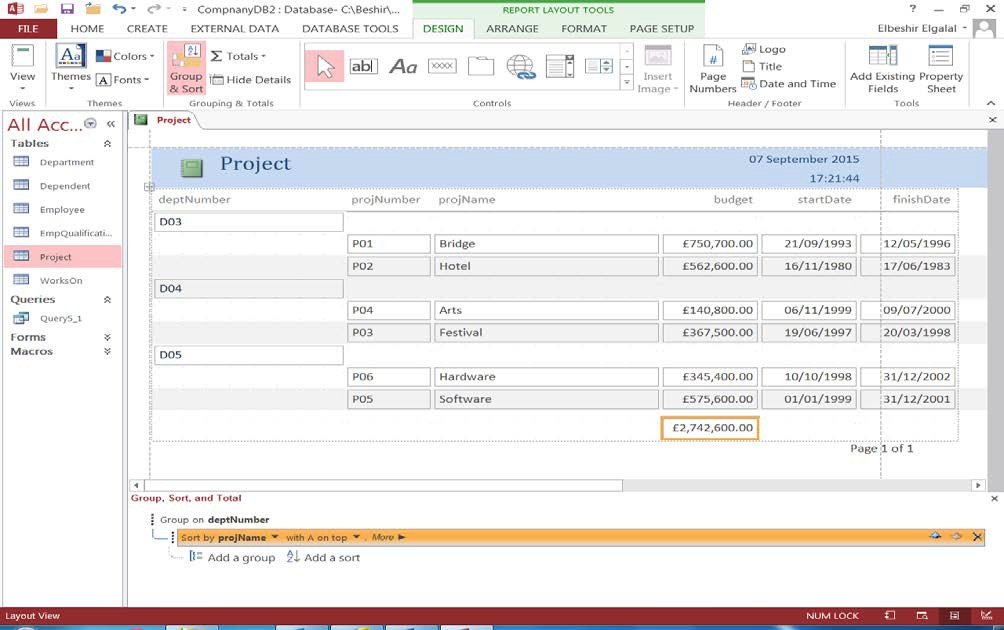


Figure 6.4 Project report

In Figure 6.4 you also see the 'budget' field total value. This is due to the **Sum** function which is inserted in the report footer. Open the report in **Design** view, and look at the footer area. The

report page numbers are produced by the expression in the report **Page** footer (see Figure 6.5).

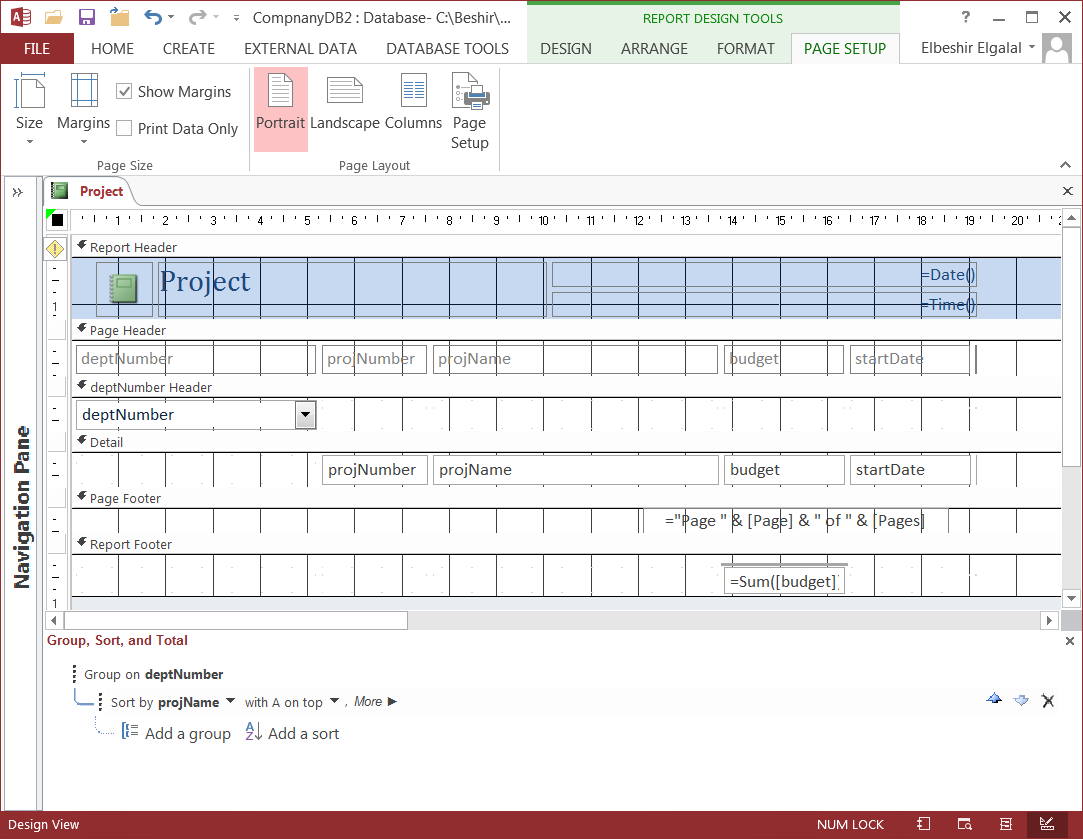


Figure 6.5 Project report in Design view



Figure 6.6 Edited report

**Exercise 6.4**

♦ Edit the report in Figure 6.4 so that it looks like Figure 6.6

Do the following to edit the report in Figure 6.4:

♦ **Turn off** the control layout so that you can move the controls individually :

• Open report in Design view;

• Click on any control, select **Arrange** tab on Ribbon and click **Select Layout** in

**Rows & Columns** group. All controls will be selected.

• Right-click any control and select **Layout** from the drop-down menu. Then select

**Remove Layout** from drop-down submenu.

♦ Open the report in **Layout** view;

♦ Click **Group & Sort** button in the **Grouping & Totals** group in the **Design** tab if it is not selected.

♦ Click on **Group on deptNumber** in the **Group, Sort and Total** pane beneath the report grid and then click **More** button. Click the downward arrow next to **without a footer section** and select **with a footer section** to add the deptNumber footer.

♦ Open the report in Design view and rearrange the controls moving the **budget** control header and text box to the end of the line.

♦ Expand the deptNumber Footer, and add a textbox with its label. Type: **Department budget** in the label control and **=Sum(budget)** in the text control. Add a **line** under the budget text control.

♦ Add a label to the Report Footer and type in it: **Total budget.** Add a **line** under the budget text control

♦ Arrange the new added controls so that you report look like the one in Figure 6.6.

Make use of the report properties to add additional feature to the report (background colour, a different logo, etc.). Make use of the **Grid** when rearranging the controls layout.

**Exercise 6.5**

a) Create a report based on WorksOn table. Group by projectNumber and count the number of employees working on each project, as well as, the sum of the total number of hours the employees worked on the project.

b) Create a report on EmpQualifications table. Group by qualification and count the number of employees for each qualification.

**6.3 Using Report Wizard with a Single table**

You can use the Report wizard to create simple reports based on a **single** table or reports based on **multiple related** tables or reports based on a query. Creating a report using the wizard allows you to choose the attributes that you want to appear in the report and as such is more flexible than the **Repor**t command.

**Exercise 6.6**

♦ Use the Wizard to create a report based on the Employee table. The report should display some of the Employee attributes.

Do the following:

♦ Selec**t Employee** table in navigation pane;

♦ **Click Create** tab and then **Report Wizard** in **Reports group** and Figure 6.7 is displayed;

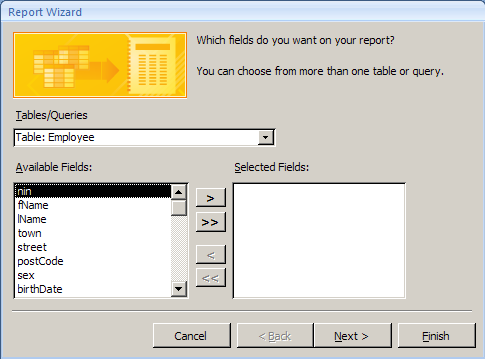


Figure 6.7 The Report wizard: selecting the fields to appear on the report

The dialogue box in Figure 6.7 allows you to select which fields are to appear on the report. You can select all of them by clicking >> or use the > button to select individual fields. Similarly you can use the < and << buttons to remove fields from the list of selected fields.

♦ Select required fields: fName, lName, sex, birthdate, salary, and deptNumber. And then click **Next** button. The dialogue box in Figure 6.8 will appear;

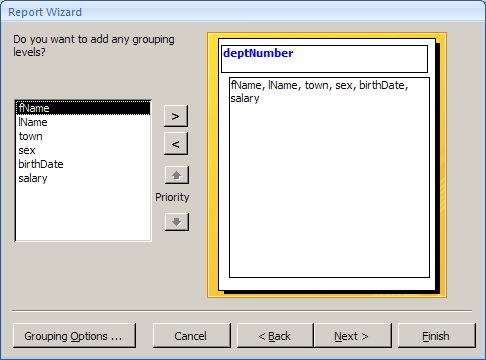


Figure 6.8 The Report Wizard: grouping data

♦ Click **Next** button and the dialogue box in Figure 6.9 will appear.

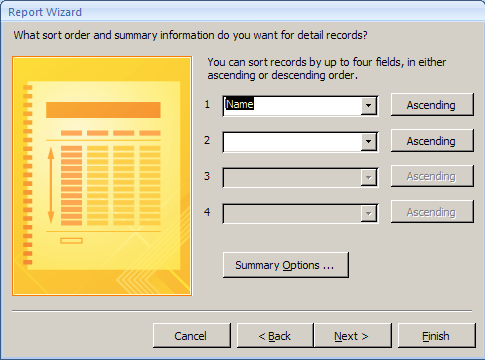


Figure 6.9 The report Wizard: Ordering result

♦ Order records in ascending order by **lName** and click **Next** button**.**

In the next screen (Figure 6.10), you specify the report layout and orientation and click next. (Experiment with the various options).

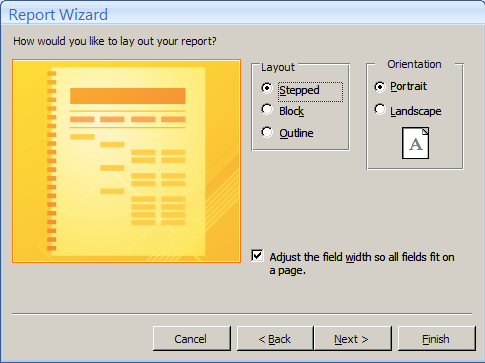


Figure 6.10 The Report Wizard: layout and style

The final dialogue box is shown in Figure 6.11. The text you select in **What title…** is used both as the title on the head of the report and the name of the report as it appears in the navigation pane.

♦ Choose a meaningful title and click **Finish** button and the report in Figure 6.12 will be produced. You can choose 'Modify the report's design' to display the report in design view, and edit the report to your own taste (add date and time, add a logo, change font and colour of heading, etc.). You should make use of Report and controls' properties.

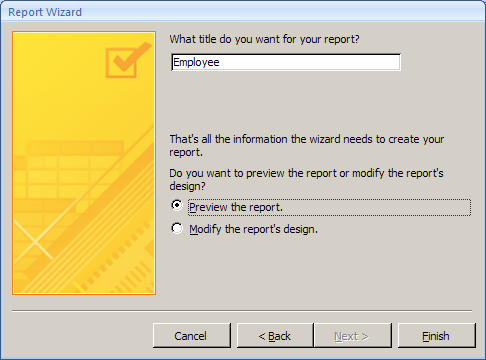


Figure 6.11

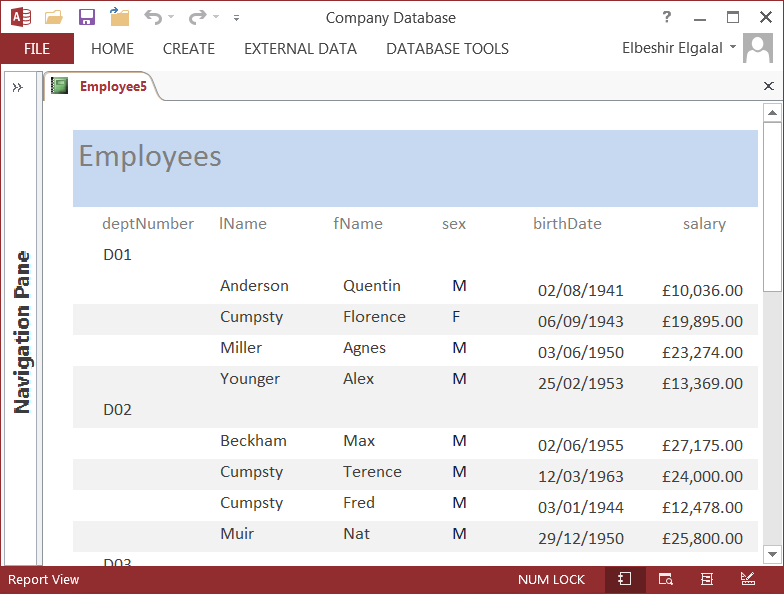


Figure 6.12 Employees' report

**Exercise 6.7**

♦ Using Report Wizard create a report based on **Dependent** table and a report based on **Project** table. Edit the reports: add date and time; page numbers, a logo, etc. Make use of the Grid when rearranging the controls layout.

**6.4 Using Report Wizard with Many tables**

When using more than one table, there must be a link (a relationship) between the tables otherwise you will get a meaningless report.

**Exercise 6.8**

♦ Create a report that shows each department name and phone number, the first and last names, date birth and salary of each employee who works in the department.

Do the following:

♦ Click **Create** tab and then **Report Wizard** in **Reports** group.

♦ Click downward arrow head and select Department table and select required fields;

♦ Click downward arrow head and select Employee table and select required fields;

♦ Click **Next** button and follow the same steps as using one table.

♦ The report should look like Figure 6.13.

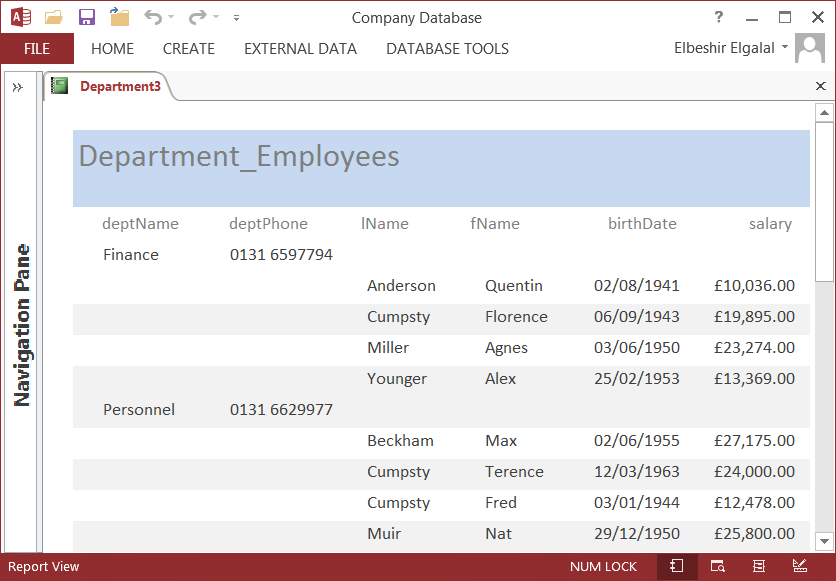


Figure 6.13

**Exercise 6.9**

♦ Create a report based on the following query:

SELECT projName budget, fName, lName, hours

FROM Project p, Employee e, WorksOn w

WHERE (p.projNumber = w.projNumber) AND (e.nin = w.nin);

Steps:

♦ Run and save the above query;

♦ Using the Wizard create a report based on the query.

**APPENDIXES**

**Appendix2.1: Fields' data types and constraints**

**Department table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **Field**  **Size** | **Format** | **Required** | **Input Mask** | **Indexed** | **Validation Rule** | **Validation Short**  **Text** |
| deptNumber | Short Text | 3 |  |  | "D"00;0 |  |  |  |
| deptName | Short Text | 30 |  | Yes |  | Yes(No  Duplicates) |  |  |
| deptphone | Short Text | 20 |  | Yes |  | Yes(No  Duplicates) |  |  |
| mgrNin | Short Text | 8 |  | Yes | >LL000000 |  |  |  |
| mgrStartDate | Date/Time |  | Short  date |  |  |  | >#01/01/1900# | Must be in or after  1900 |

**Required** property set to yes for the fields: department name, phone number and manager id because each department must have a name, a phone number and manager.

**Employee table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **FieldSize** | **Format** | **Required** | **Input Mask** | **Validation Rule** | **Validation Text** |
| nin | Short Text | 8 |  |  | >LL000000 |  |  |
| fName | Short Text | 30 |  | Yes |  |  |  |
| lName | Short Text | 30 |  | Yes |  |  |  |
| town | Short Text | 30 |  | Yes |  |  |  |
| street | Short Text | 40 |  | Yes |  |  |  |
| postcode | Short Text | 7 |  | Yes | >LL0 0LL |  |  |
| sex | Short Text | 1 |  |  |  | ="M" or "F" | Must be M or F |
| birthDate | Date/Time |  | Short Date |  |  | >#01/01/1900# | Must be in or after 1900 |
| salary | Currency |  |  |  |  | Between 10000 and  30000 | Must be between 10000 and 30000 |
| superNin | Short Text | 8 |  |  | >LL000000 |  |  |
| deptNumber | Short Text | 3 |  | Yes | "D"00;0 |  |  |

**Required** property for fName, lName, town, street and postcode fields is set to **Yes**. We don't want records data about employees if we don't know their names and address. Department number field's **Required** property is also set to **Yes** because in the conceptual schema each employee works for one department.

**Project table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **Field**  **Size** | **Format** | **Required** | **Input Mask** | **Indexed** | **Validation Rule** | **Validation Short**  **Text** |
| projNumber | Short Text | 3 |  |  | "P"00;0 |  |  |  |
| projName | Short Text | 30 |  | Yes |  | Yes(No Duplicates) |  |  |
| budget | Currency |  |  |  |  |  |  |  |
| startDate | Date/Time |  | Short  date |  |  |  | >#01/01/1900# | Must be in or after  1900 |
| finishDate | Date/Time |  | Short  date |  |  |  | >#01/01/1900# | Must be in or after  1900 |
| deptNumber | Short Text | 3 |  | Yes |  |  |  |  |

**Required** property is set to **Yes** for project name field because each project must have a name. Required property is also set to **Yes**

for department number field because in the conceptual schema every project is controlled by one department.

**Dependent table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **Field**  **Size** | **Format** | **Required** | **Input Mask** | **Indexed** | **Validation Rule** | **Validation Short**  **Text** |
| empNin | Short Text | 8 |  | Yes | >LL000000 |  |  |  |
| fName | Short Text | 20 |  | Yes |  |  |  |  |
| birthDate | Date/Time |  | Short  date |  |  |  | >#01/01/1900# | Must be in or after  1900 |
| sex | Short Text | 1 |  |  |  |  | ="M" or "F" | Must be M or F |
| relationship | Short Text | 30 |  |  |  |  |  |  |

empNin and fName fields cannot have NULL values because they part of the primary key. That is why **Required** property is set to

**Yes** for both fields.

**WorksOn table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **Field**  **Size** | **Format** | **Required** | **Input Mask** | **Indexed** | **Validation Rule** | **Validation Short**  **Text** |
| nin | Short Text | 8 |  | Yes | >LL000000 |  |  |  |
| projNumber | Short Text | 3 |  | Yes | "P"00;0 |  |  |  |
| hours | Number | integer |  |  |  |  |  |  |

**nin** and **projNumber** fields cannot have **NULL** values because they part of the **primary key**.

**EmpQualifications table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **Field**  **Size** | **Format** | **Required** | **Input Mask** | **Indexed** | **Validation Rule** | **Validation Short**  **Text** |
| nin | Short Text | 8 |  | Yes | >LL000000 |  |  |  |
| qualification | Short Text | 30 |  | Yes |  |  |  |  |

**nin** and **qualification** fields cannot have **NULL** values because they part of the **primary key**.

**Steps**

**Appendix3.1: Sample Data**

a) Enter some of the sample data, given in the next pages, manually to check that the constraints (e.g. mask, primary key, foreign key, etc.) are enforced;

b) Remove the data from all tables except **Department table;**

c) Import all the data given in the next pages into the database tables.

**Note:**

Before you import the data for **Employee** table you need to edit the Mask for ‘postCode’ field. Read the values in ‘postCode’ field in Table 2 and you will notice that some post codes have two digits after the first letter. The current mask for ‘postCode’ field will generate an error. See the table on page 27 in Practical 2 and change the mask for ‘postCode’ field so that you don’t get an error when you import the **Employee’s** table data.

To import the data, given in the tables in the next pages, in the database tables do the following:

1. Copy the data of a table including the columns headings;

2. Open Excel 2013 and put the cursor in the first cell on the left hand corner;

• Right-click and click the first icon under **Paste Options** in the drop down menu;

• Increase the size of Excel cells if the column names do not appear in one line the values are displayed as # characters;

3. Store the spreadsheet on your disk;

4. Start Microsoft Access 2013 and click **EXTERNAL DATA** tab on the Ribbon;

5. Click **Excel** icon in the **Import & Link** group;

6. The window in Figure 1 below will open;

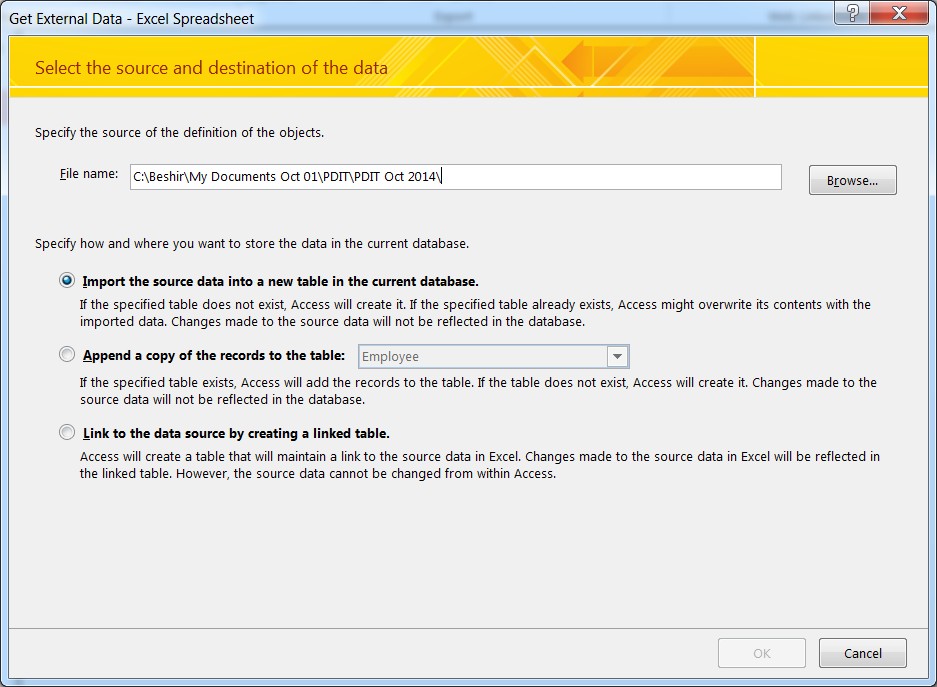


Figure 1

7. Click **Browse** button and navigate to the folder where you stored the **Excel** file, select it and click **Open** button;

8. Click the button next to: **Append a copy of the records to the table;**

9. Select the table you want to store the data and click **OK;**

10. Make sure that the data in all fields is correct and click **Next**;

11. Click **Finish** and then **Close** in the windows that open;

12. Check if all the data has been copied to the table;

13. Go to 1 and repeat the process for the other tables.

**Table 2 (Employee)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **nin** | **fName** | **lName** | **town** | **street** | **postcode** | **sex** | **birthDate** | **salary** | **superNin** | **deptNumber** |
| AR218148 | Agnes | Miller | Edinburgh | 8 Princes Street | EH3 2AB | F | 03/06/1950 | £23,274.00 |  | D01 |
| CW954608 | Max | Beckham | Edinburgh | 2 Bankhead Str. | EH3 1XY | M | 02/06/1955 | £27,175.00 |  | D02 |
| FL238319 | Alex | Beeby | Glasgow | 4 Glen Road | G33 7HL | M | 30/06/1968 | £26,390.00 |  | D03 |
| GS866278 | Sally | Cumpsty | Paisley | 14 Canal Street | PA1 2XY | F | 04/01/1945 | £24,000.00 |  | D04 |
| LY170573 | Cedric | Younger | Glasgow | 16 Woodland Rd | G12 3HN | M | 29/10/1963 | £26,821.00 |  | D05 |
| AW403874 | Florence | Cumpsty | Edinburgh | 20 Glasgow Rd | EH4 9XQ | F | 06/09/1943 | £19,895.00 | AR218148 | D01 |
| BJ813109 | Quentin | Anderson | Edinburgh | 22 London Road | EH3 6XQ | M | 02/08/1941 | £10,036.00 | AR218148 | D01 |
| CN770902 | Alex | Younger | Edinburgh | 15 Castle Street | EH6 7LN | M | 25/02/1953 | £13,369.00 | AR218148 | D01 |
| CQ078763 | Nat | Muir | Edinburgh | 26 Hamilton Road | EH7 6HH | M | 29/12/1950 | £25,800.00 | CW954608 | D02 |
| DN612612 | Terence | Cumpsty | Glasgow | 17 Duke Street | G41 3BN | M | 12/03/1963 | £24,000.00 | CW954608 | D02 |
| DN996833 | Fred | Cumpsty | Glasgow | 3 Manor Road | G75 8PJ | M | 03/01/1944 | £12,479.00 | CW954608 | D02 |
| DO781710 | John | Anderson | Glasgow | 18 Dale Street | G11 5QX | M | 01/07/1946 | £19,972.00 | EG314134 | D03 |
| DU705012 | Brenda | Beeby | Glasgow | 5 Novar Drive | G12 9AX | F | 26/05/1967 | £20,270.00 | EG314134 | D03 |
| EG314134 | Marion | Anderson | Paisley | 88 Gordon Street | PA1 3AB | F | 14/06/1959 | £24,886.00 | FL238319 | D03 |
| FD962684 | Larry | Robertson | Stirling | 10 King Street | FK8 1BD | M | 20/04/1955 | £22,880.00 | FL238319 | D03 |
| FP932137 | George | Birtwhistle | Glasgow | 32 Queen Street | G11 3HN | M | 21/11/1962 | £14,549.00 | EG314134 | D03 |
| FQ600778 | Deborah | Cramond | Edinburgh | 18 High Street | EH8 9NM | F | 09/11/1960 | £18,609.00 | GS866278 | D04 |
| HH189883 | Harry | Younger | Glasgow | 66 Main Drive | G52 4NZ | M | 17/01/1947 | £16,089.00 | GS866278 | D04 |
| HN120169 | Quentin | Quinn | Glasgow | 3 Castle Road | G44 4BN | M | 12/02/1946 | £23,990.00 | GS866278 | D04 |
| HP650969 | Florence | Muir | Stirling | 6 Milton Road | FK7 8EX | F | 31/05/1954 | £14,081.00 | GS866278 | D04 |
| JB162844 | Max | Birtwhistle | Hamilton | 14 Oliver Street | ML1 0AY | M | 06/02/1940 | £16,490.00 | LY170573 | D05 |
| KH471537 | Irene | Musgrove | Glasgow | 12 London Road | G16 8QZ | F | 02/05/1946 | £16,709.00 | LY170573 | D05 |
| KN289846 | Rita | Cramond | Hamilton | 16 Clyde street | ML3 1BN | F | 15/02/1958 | £19,885.00 | LY170573 | D05 |
| LL679812 | Florence | Cano | Glasgow | 14 New Street | G18 7XZ | F | 17/03/1955 | £23,231.00 | LY170573 | D05 |
| NK805161 | Polly | Birtwhistle | Paisley | 15 Mull Avenue | PA3 2XY | F | 13/11/1960 | £18,251.00 | LY170573 | D05 |

**Table 3 (Dependent)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **empNin** | **fName** | **birthdate** | **sex** | **relationship** |
| AR218148 | Emily | 28/01/1980 | F | daughter |
| CN770902 | John | 21/10/1979 | M | son |
| CW954608 | Katie | 04/10/1980 | F | daughter |
| DN612612 | Terence | 01/05/1988 | M | son |
| DO781710 | Harriet | 28/06/1942 | F | mother |
| DU705012 | Richard | 15/02/1976 | M | brother |
| DU705012 | Tammy | 24/10/1995 | F | daughter |
| FD962684 | Polly | 13/01/1960 | F | sister |
| FD962684 | Alex | 21/10/1982 | M | son |
| FD962684 | Jane | 06/11/1932 | F | mother |
| GS866278 | Deborah | 27/06/1950 | F | wife |
| KH471537 | Larry | 09/10/1969 | M | son |
| KN289846 | Karrie | 22/01/1940 | F | sister |
| HH189883 | Peter | 23/06/1968 | M | son |
| HH189883 | Chris | 04/06/1970 | F | daughter |
| HN120169 | Frank | 29/06/1950 | M | brother |
| HP650969 | Brenda | 27/02/32 | F | mother |
| LL679812 | Fred | 28/11/1977 | M | son |
| LY170573 | Irene | 21/11/1988 | F | daughter |
| LY170573 | Nancy | 11/12/1991 | F | daughter |
| LY170573 | Roger | 27/07/1994 | M | son |
| LY170573 | Tom | 03/10/1967 | M | son |
| NK8051619 | Oliver | 12/05/1985 | M | son |

**Table 4 (Project)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **projNumber** | **projName** | **budget** | **startdate** | **finishDate** | **deptNumber** |
| P01 | Bridge | £750,700.00 | 21/09/1993 | 12/05/1996 | D03 |
| P02 | Hotel | £562,600.00 | 16/11/1980 | 17/06/1983 | D03 |
| P03 | Festival | £367,500.00 | 19/06/1997 | 20/03/1998 | D04 |
| P04 | Arts | £140,800.00 | 06/11/1999 | 09/07/2000 | D04 |
| P05 | Software | £575,600.00 | 01/01/1999 | 31/12/2001 | D05 |
| P06 | hardware | £345,400.00 | 10/10/1998 | 31/12/2002 | D05 |

**Table 5 (WorksOn)**

|  |  |  |
| --- | --- | --- |
| **nin** | **projNumber** | **hours** |
| DO781710 | P01 | 60 |
| DO781710 | P02 | 80 |
| DU705012 | P01 | 50 |
| DU705012 | P02 | 90 |
| EG314134 | P01 | 40 |
| EG314134 | P02 | 70 |
| FD962684 | P01 | 90 |
| FD962684 | P02 | 70 |
| FL238319 | P01 | 60 |
| FL238319 | P02 | 30 |
| FP932137 | P01 | 80 |
| FP932137 | P02 | 70 |
| FQ600778 | P03 | 70 |
| FQ600778 | P04 | 90 |
| GS866278 | P03 | 50 |
| GS866278 | P04 | 70 |
| HH189883 | P03 | 100 |

**Table 5(WorksOn) continued**

|  |  |  |
| --- | --- | --- |
| **nin** | **projNumber** | **hours** |
| HH189883 | P04 | 60 |
| HN120169 | P03 | 70 |
| HN120169 | P04 | 85 |
| JB162844 | P05 | 66 |
| JB162844 | P06 | 95 |
| KH471537 | P05 | 80 |
| KH471537 | P06 | 55 |
| KN289846 | P05 | 88 |
| KN289846 | P06 | 76 |
| LL679812 | P05 | 55 |
| LL679812 | P06 | 65 |
| LY170573 | P05 | 67 |
| LY170573 | P06 | 76 |
| NK805161 | P05 | 65 |
| NK805161 | P06 | 85 |

**Table 6(EmpQualifications)**

|  |  |
| --- | --- |
| **nin** | **qualification** |
| AR218148 | BSc |
| AR218148 | MSc |
| AW403874 | HND |
| BJ813109 | BSc |
| CN770902 | BSc |
| CQ078763 | BA |
| CW954608 | BA |
| CW954608 | Mphil |
| DN612612 | BA |
| DN612612 | HNC |
| DN996833 | BSc |
| DN996833 | HND |
| DO781710 | BSc |
| DU705012 | BSc |
| DU705012 | HND |
| EG314134 | BSc |
| FD962684 | BSc |
| FD962684 | MSc |
| FL238319 | BSc |
| FL238319 | MSc |
| FP932137 | BSc |
| FQ600778 | BSc |
| FQ600778 | MSc |
| GS866278 | BSc |
| GS866278 | PhD |
| HH189883 | BSc |

**Table 6 (EmpQualifications) Continued**

|  |  |
| --- | --- |
| **nin** | **qualification** |
| HH189883 | HND |
| HN120169 | BSc |
| HP650969 | BSc |
| HP650969 | MSc |
| JB162844 | BSc |
| JB162844 | HND |
| KH471537 | BSc |
| KN289846 | BSc |
| KN289846 | HND |
| LL679812 | BSc |
| LL679812 | MSc |
| LY170573 | BSc |
| LY170573 | MSc |
| LY170573 | PhD |
| NK805161 | BSc |
| NK805161 | HND |

**Note:** Sample data for the **Department** table in given in **Practical 3.**