# NEAR EAST UNIVERSITY

## Faculty of Engineering

## Department of Computer Engineering

# VIDEO STORE MANAGEMENT PROGRAMMING SYSTEM

**Graduation Project**
**COM-400**

**Student:** Müge Kütük (20032372)

**Supervisor:** Assoc. Prof. Rahib Abiyev

**Nicosia-2008**

# ACKNOWLEDGMENT

*"First of all I would like to thank my supervisor Assoc. Prof. Rahib Abiyev for his way of treatment with me, he never rejected me when I had any question and also we contacted otherside of the university.*

*I don't forget other instructors, Mr. Elbrus Imanov was also helped me whenever I had any problem or he gave me any ideas. I thank them very much with my all for their advices.*

*Secondly I would like to thank my family for their molar tooth of graduated me from here. I know that because of their selfless I was here and I never want to make them disappointed. So we trust each other and I will be graduated.*

*Thirdly I want to thanks my friends Halime Yılmaz, Emre Doğan, S.Serhan Eldemir, Cüneyt Şeker their helped me in my project and gave me ideas and parted her time with me.*

*Finally I would like to thank the many faculty and staff who have enriched my experience at Near East University, specially vice president Prof. Dr. Fakhreddin Mamedov..."*

# ABSTRACT

Computer science has developed tremendously over the last decades.It is possible to state this in terms of both hardware and software. Programming is always providing the scientists a continuous systematic development in their studies and research. In this project it's been constructed a special program related to Video Store Management Programming. The video store management should not be regarded as an isolated and unrelated field from the other industries but it is within this framework that the history of video store development should be examined .New concepts in video store design have been developed more recently in an effort to meet the changing preferences and new characteristics.

The video store management program consists of many departments like give-return video, search video, all customer, all videos, search sale, rental, supplier, quaries, help and about The program that been given in this thesis, resumes that the briefly in a quick time in order to have quick and economic services. On the other hand, the store development is suitable for researches and students in computer science ;the development of video store management program system is designed to help compute professionals who want to learn about exciting field and to serve as a basic reference.

The aim of this project is to create and to develop a project in a scientific method to introduce the gab between scientific theoretical life and work normal life.

In this project, it's been constructed a video store management program for the availability of information is incrementally important an all over the word, how to make a cays process in order to have a quick research, data process, analysis process.

Finally, all file enclosed full details about the project.

# TABLE OF CONTENTS

## CHAPTER TWO

## CHAPTER THREE

# LIST OF FIGURES

# CHAPTER 1

## 1.BASIC CONCEPT OF DELPHI

### 1.1.Introduction to Delphi

Although I am not the most experienced or knowledgeable person on the forums I thought it was time to write a good introductory article for Delphi

### 1.2.What is Delphi?

Delphi is a Rapid Application Development (RAD) environment. It allows you to drag and drop components on to a blank canvas to create a program. Delphi will also allow you to use write console based DOS like programs.

Delphi is based around the Pascal language but is more developed object orientated derivative. Unlike Visual Basic, Delphi uses punctuation in its basic syntax to make the program easily readable and to help the compiler sort the code. Although Delphi code is not case sensitive there is a generally accepted way of writing Delphi code. The main reason for this is so that any programmer can read your code and easily understand what you are doing, because they write their code like you write yours.

For the purposes of this series I will be using Delphi 6. Delphi 6 provides all the tools you need to develop, test and deploy Windows applications, including a large number of so-called reusable components.
Borland Delphi, provides a cross platform solution when used with Borland Kylix - Borland's RAD tool for the Linux platform.

### 1.2.1.Delphi Compliers

There are two types complier for Delphi

- **Turbo Delphi** : Free industrial strength Delphi RAD (Rapid Application Development) environment and compiler for Windows. It comes with 200+ components and its own Visual Component Framework.

- **Turbo Delphi for .NET**: Free industrial strength Delphi application development environment and compiler for the Microsoft .NET platform.

### 1.2.2. What kind of programming can you do with Delphi?

The simple answer is "more or less anything". Because the code is compiled, it runs quickly, and is therefore suitable for writing more or less any program that you would consider a candidate for the Windows operating system.

You probably won't be using it to write embedded systems for washing machines, toasters or fuel injection systems, but for more or less anything else, it can be used (and the chances are that probably someone somewhere has!)

Some projects to which Delphi is suited:

- Simple, single user database applications
- Intermediate multi-user database applications
- Large scale multi-tier, multi-user database applications
- Internet applications
- Graphics Applications
- Multimedia Applications
- Image processing/Image recognition
- Data analysis
- System tools
- Communications tools using the Internet, Telephone or LAN
- Web based applications

This is not intended to be an exhaustive list, more an indication of the depth and breadth of Delphi's applicability. Because it is possible to access any and all of the Windows API, and because if all else fails, Delphi will allow you to drop a few lines of assembler code directly into your ordinary Pascal instructions, it is possible to do more or less anything. Delphi can also be used to write Dynamically Linked Libraries (DLLs) and can call out to DLLs written in other programming languages without difficulty.

Because Delphi is based on the concept of self contained Components (elements of code that can be dropped directly on to a form in your application, and exist in object form, performing their function until they are no longer required), it is possible to build applications very rapidly. Because Delphi has been available for quite some time, the number of pre-written components has been increasing to the point that now there is a component to do more or less anything you can imagine. The job of the programmer has become one of gluing together appropriate components with code that operates them as required.

### 1.2.3.History Of Delphi

Delphi was one of the first of what came to be known as "RAD" tools, for Rapid Application Development, when released in 1995 for the 16-bit Windows 3.1 . Delphi 2, released a year later, supported 32-bit Windows environments, and a C++ variant, C++ Builder , followed a few years after.

The chief architect behind Delphi, and its predecessor Turbo Pascal , was Anders Hejlsberg until he was headhunted in 1996 by Microsoft , where he worked on Visual J++ and subsequently became the chief designer of C Sharp programming language |C# and a key participant in the creation of the Microsoft .NET Framework.

In 2001 a Linux version known as Kylix programming tool| Kylix became available. However, due to low quality and subsequent lack of interest, Kylix was abandoned after version 3.

Support for Linux and Windows cross platform development (through Kylix and the CLX component library) was added in 2002 with the release of Delphi 6.

Delphi 8, released December 2003, was a .NET -only release that allowed developers to compile Delphi Object Pascal code into .NET Microsoft Intermediate Language|MSIL . It was also significant in that it changed its IDE for the first time, from the multiple-floating-window-on-desktop style IDE to a look and feel similar to Microsoft's Visual Studio.NET.

Although Borland fulfilled one of the biggest requests from developers (.NET support), it was criticized both for making it available too late, when a lot of former Delphi developers had already moved to C#, and for focusing so much on backward compatibility that it was not very easy to write new code in Delphi. Delphi 8 also lacked significant high-level features of the c sharp|C# language, as well as many of the more appealing features of Microsoft's Visual Studio IDE. (There were also concerns about the future of Delphi Win32 development. Because Delphi 8 did not support Win32, Delphi 7.1 was included in the Delphi 8 package.)

The next version, Delphi 2005 (Delphi 9), included the Win32 and .NET development in a single IDE, reiterating Borland's commitment to Win32 developers. Delphi 2005 includes design-time manipulation of live data from a database. It also includes an improved IDE and added a "for ... in" statement (like C#'s foreach ) to the language. However, it was criticized by some for its bugs; both Delphi 8 and Delphi 2005 had stability problems when shipped, which were only partially resolved in service packs.

In late 2005 , Delphi 2006 was released and federated development of C# and Delphi.NET, Delphi Win32 and C++ into a single IDE. It was much more stable than Delphi 8 or Delphi 2005 when shipped, and improved even more after the service packs and several hotfixes.

On February 8 , 2006 , Borland announced that it was looking for a buyer for its IDE and database line of products, which include Delphi, to concentrate on its Application Lifecycle Management|ALM line. The news met with voluble optimism from the remaining Delphi users.

On September 6 , 2006, The Developer Tools Group (the working name of the not yet spun off company) of Borland Software Corporation released single language versions

of Borland Developer Studio, bringing back the popular "Turbo" moniker. The Turbo product set includes Turbo Delphi for Win32, Turbo Delphi for .NET, Turbo C++, and Turbo C#. Each version is available in two editions: "Explorer"—a free downloadable version—and "Professional"—a relatively cheap (US$399) version which opens access to thousands of third-party components. Unlike earlier "Personal" editions of Delphi, new "Explorer" editions can be used for commercial development.

On November 14 , 2006, Borland announced the cancellation of the sale of its Development tools; instead of that it would spin them off into an independent company named "CodeGear"

### 1.2.4.Advantages&Disadvantages Delphi

**==Advantages==**

Delphi exhibits the following advantages:

- Rapid Application Development (RAD)
- Based on a well-designed language - high-level and strongly typed, with low-level escapes for experts
- A large community on Usenet and the World Wide Web (e.g. news://newsgroups.borland.com and Borland's web access to Delphi)
- Can compile to a single executable, simplifying distribution and reducing DLL versioning issues
- Many VCL and third-party components (usually available with full source code) and tools (documentation, debug tools, etc.)
- Quick optimizing compiler and ability to use assembler code
- Multiple platform native code from the same source code
- High level of source compatibility between versions
- Cross Kylix - a third-party toolkit which allows you to compile native Kylix/Linux applications from inside the Windows Delphi IDE, hence easily enabling dual-platform development and deployment

- Cross FBC - a sister project to CrossKylix, which enables you to cross-compile your Windows Delphi applications to multi-platform targets - supported by the Free Pascal compiler - without ever leaving the Delphi IDE
- Class helpers to bridge functionality available natively in the Delphi RTL, but not available in a new platform supported by Delphi
- The language's object orientation features only class- and interface-based Polymorphism in object-oriented programming|polymorphism

**Disadvantages**

- *Limited cross-platform capability for Delphi itself. Compatibles provide* more architecture/OS combinations
- *Access to platform and third party libraries require header files to be* translated to Pascal. This creates delays and introduces the possibilities of errors in translation.
- There are fewer published books on Delphi than on other popular programming languages such as C++ and C#
- A reluctance to break any code has lead to some convoluted language design choices, and orthogonality and predictability have suffered

### 1.3. Delphi 6 Editions

There are 3 editions in Delphi 6 :

- **Delphi Personal** - makes learning to develop non-commercial Windows applications fast and fun. Delphi 6 Personal makes learning Windows development easy with drag-and-drop visual programming.
- **Delphi Professional** - adds the tools necessary to create applications with the latest Windows® ME/2000 look-and-feel. Dramatically enhance functionality with minimal code using the power and flexibility of SOAP and XML to easily integrate Web Services into client-side applications.

- **Delphi Enterprise** - includes additional tools, extensive options for Internet. Delphi 6 makes next-generation e-business development with Web Services a snap.

This Program will concentrate on the Enterprise edition..

### 1.3.1. Delphi 6 Archite

Delphi 6 Architect is designed for professional enterprise developers who need to adapt quickly to changing business rules and manage sophisticated applications that synchronize with multiple database schemas. Delphi 2006 Architect includes an advanced ECO III framework that allows developers to rapidly deploy scalable external facing Web applications with executable state diagrams, object-relational mapping, and transparent persistence.

Delphi 6 Architect includes all of the capabilities of the Enterprise edition, and includes the complete ECO III framework, including new support for ECO State Machines powered by State Chart visual diagrams, and simultaneous persistence to multiple and mixed database servers.

- State Chart Diagrams
- Executable ECO State Machines
- Multi- and Mixed- ECO database support

### 1.3.2.Installation Delphi 6

To install Delphi 6 Enterprise, run INSTALL.EXE (default location C:\Program Files\Borland Delphi ) and follow the installation instructions.

We are prompted to select a product to install, you only have one choice "Delphi 6":


**Figure 1.1** The Select Page For Start Installation

While the setup runs, you'll need to enter your serial number and the authorization key (the two you got from inside a Cd rom driver ).

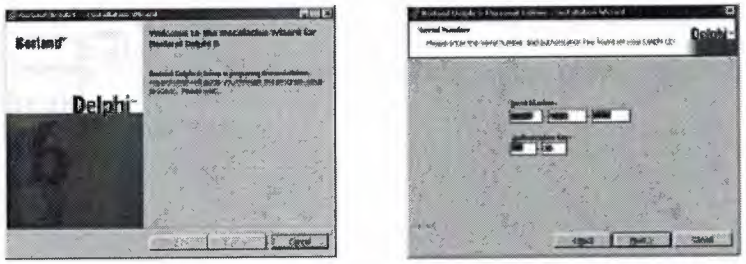



**Figure 1.2** Serial Number And Authorization Screen

Later, the License Agreement screen will popup:

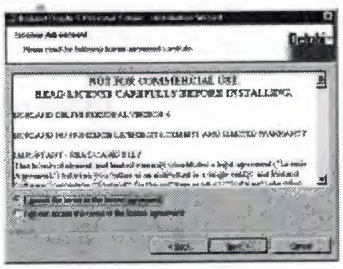



**Figure 1.3** License Agreement Screen

After that, you have to pick the Setup Type, choose Typical. This way Delphi 6 Enterprise will be installed with the most common options. The next screen prompts you to choose the Destination folder.

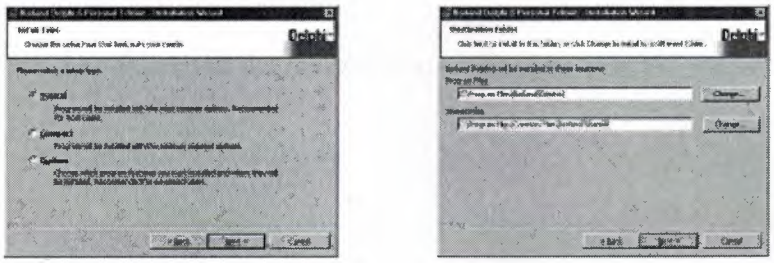


**Figure 1.4.**SetUp Type and Destination Folder Screen

At the end of the installation process, the set-up program will create a sub menu in the Programs section of the Start menu, leading to the main Delphi 6 Enterprise program plus some additional tools.


**1.5.**Start Menu Screen

**Figure 1.5.**Start Menu

For a faster access to Delphi, create a shortcut on the Windows Desktop.

## 1.4. A Tour Of The Environment

This chapter explains how to start Delphi and gives you a quick tour of the main parts and tools of the Integrated Development Environment(IDE)

### 1.4.1. Running Delphi For The First Time

You can start Delphi in a similar way to most other Windows applications:

- Choose Programs | Borland Delphi 6 | Delphi 6 from the Windows Start menu
- Choose Run from the Windows Start menu and type Delphi32
- Double-click Delphi32.exe in the $(DELPHI)\Bin folder. Where $(DELPHI) if a folder where Delphi was installed. The default is C:\Program Files\Borland\Delphi6.
- Double-click the Delphi icon on the Desktop (if you've created a shortcut)

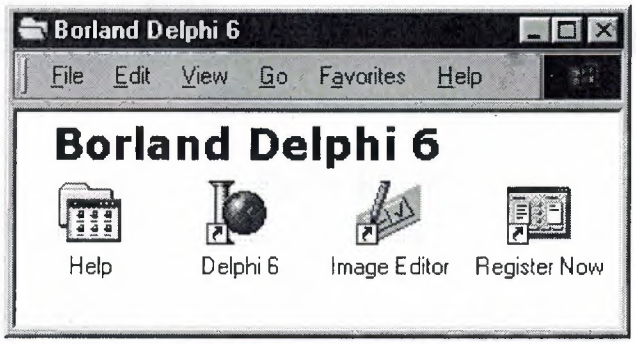



**Figure 1.6.**Borland Delphi 6 Folder

### 1.4.2. The Delphi IDE

As explained before, one of the ways to start Delphi is to choose Programs | Borland Delphi 6 | Delphi 6 from the Windows Start menu.

When Delphi starts (it could even take one full minute to start - depending on your hardware performance) you are presented with the IDE: the user interface where you can design, compile and debug your Delphi projects.

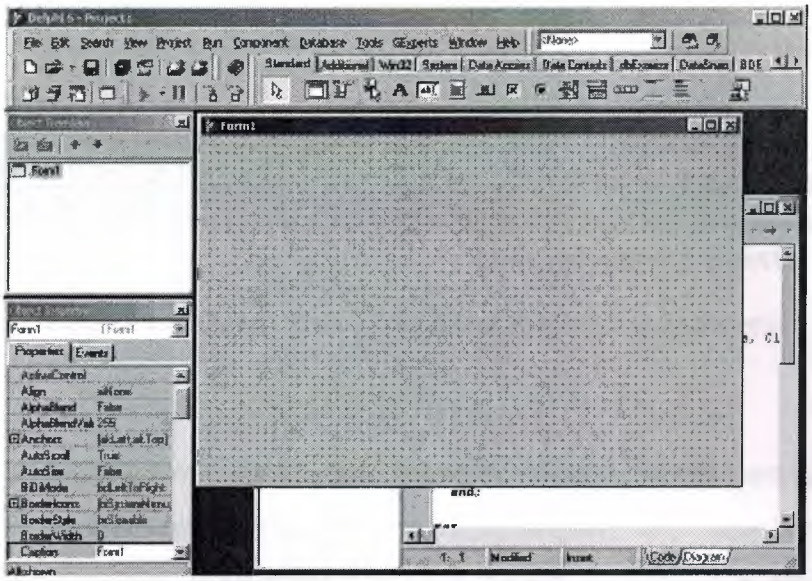


**Figure 1.7**.IDE

Like most other development tools (and unlike other Windows applications), Delphi IDE comprises a number of separate windows.

Some of the facilities that are included in the "Integrated Development Environment" (IDE) are listed below:

- A syntax sensitive program file editor
- A rapid optimising compiler
- Built in debugging /tracing facilities
- A visual interface developer
- Syntax sensitive help files
- Database creation and editing tools
- Image/Icon/Cursor creation / editing tools
- Version Control CASE tools

### 1.4.3. The Menus & Toolbar

The main window, positioned on the top of the screen, contains the main menu, toolbar and Component palette.

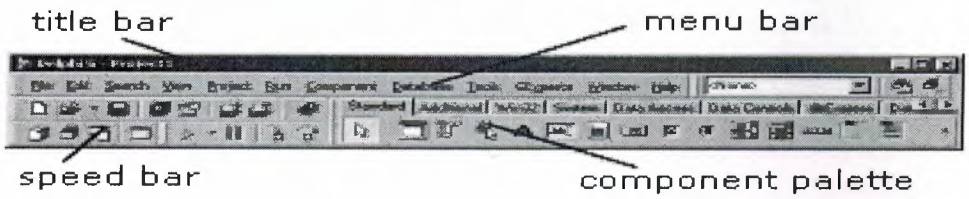



**Figure 1.8.**Menu ,Title , Speed Bar & Component Palette

The title bar of the main window contains the name of the current project (you'll see in some of the future chapters what exactly is a Delphi project). The menu bar includes a dozen drop-down menus - we'll explain many of the options in these menus later through this course. The toolbar provides a number of shortcuts to most frequently used operations and commands - such as running a project, or adding a new form to a project. To find out what particular button does, point your mouse "over" the button and wait for the tooltip. As you can see from the tooltip (for example, point to [Toggle Form/Unit]), many toolbuttons have keyboard shortcuts ([F12]).

The menus and toolbars are freely customizable. I suggest you to leave the default arrangement while working through the chapters of this course.

### 1.4.4. The Component Palette

You are probably familiar with the fact that any window in a standard Windows application contains a number of different (visible or not to the end user) objects, like: buttons, text boxes, radio buttons, check boxes etc. In Delphi programming terminology such objects are called controls (or components).Components are the building blocks of every Delphi application. To place a component on a window you drag it from the component palette. Each component has specific attributes that enable you to control your application at design and run time.

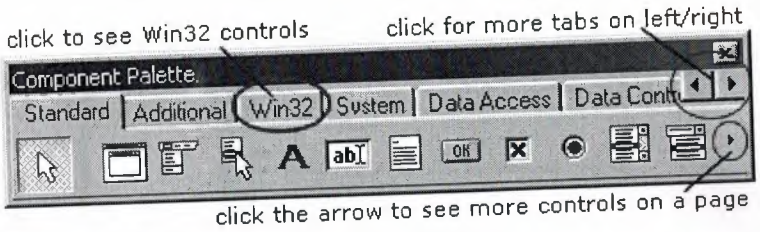



**Figure 1.9.**Component Palatte

Depending on the version of Delphi (assumed Delphi 6 Personal through this course), you start with more than 85 components at your disposal - you can even add more components later (those that you create or from a third party component vendor). The components on the Component Palette are grouped according to the function they perform. Each page tab in the Component palette displays a group of icons representing the components you can use to design your application interface. For example, the Standard and Additional pages include controls such as an edit box, a button or a scroll box.

To see all components on a particular page (for example on the Win32 page) you simply click the tab name on the top of the palette. If a component palette lists more components that can be displayed on a page an arrow will appear on a far right side of the page allowing you to click it to scroll right. If a component palette has more tabs (pages) that can be displayed, more tabs can be displayed by clicking on the arrow buttons on the right-hand side.

### 1.4.5. The Code Editor

Each time you start Delphi, a new project is created that consists of one *empty* window. A typical Delphi application, in most cases, will contain more than one window - those windows are referred to as forms.

In our case this form has a name, it is called Form1. This form can be renamed, resized and moved, it has a caption and the three standard minimize, maximize and close buttons. As you can see a Delphi form is a regular Windows window

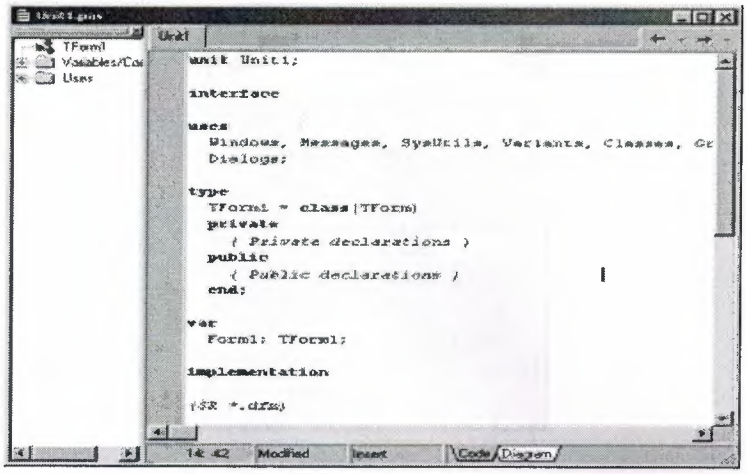



**Fig.1.10.**Code Editor Window

If the Form1 is the active window and you press [F12], the Code Editor window will be placed on top. As you design user interface of your application, Delphi automatically generates the underlying Object Pascal code. More lines will be added to this window as you add your own code that drives your application. This window displays code for the current form (Form1); the text is stored in a (so-called) unit - Unit1. You can open multiple files in the Code Editor. Each file opens on a new page of the Code editor, and each page is represented by a tab at the top of the window.

### 1.4.6. The Object Inspector

Each component and each form, has a set of properties – such as color, size, position, caption – that can be modified in the Delphi IDE or in your code, and a collection of events – such as a mouse click, keypress, or component activation – for which you can specify some additional behavior. The Object Inspector displays the properties and events (note the two tabs) for the selected component and allows you to change the property value or select the response to some event.

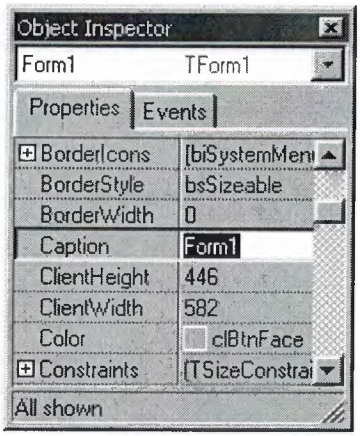



**Figure 1.11.**Object Inspector

For example, each form has a Caption (the text that appears on it's title bar). To change the caption of Form1 first activate the form by clicking on it. In the Object Inspector find the property Caption (in the left column), note that it has the 'Form1' value (in the right column). To change the caption of the form simply type the new text value, like 'My Form' (without the single quotes). When you press [Enter] the caption of the form will change to My Form.

Note that some properties can be changed more simply, the position of the form on the screen can be set by entering the value for the Left and Top properties - or the form can be simply dragged to the desired location.

### 1.4.7. The Object TreeView

Above the Object Inspector you should see the Object TreeView window. For the moment it's display is pretty simple. As you add components to the form, you'll see that it displays a component's parent-child relationships in a tree diagram. One of the great features of the Object TreeView is the ability to drag and drop components in order to change a component container without losing connections with other components.

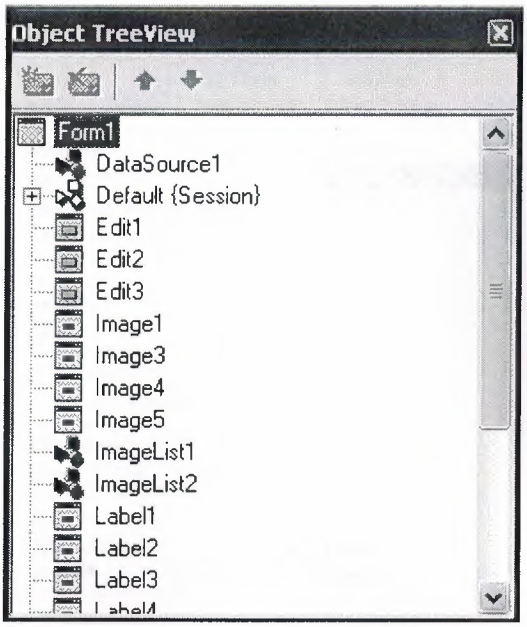



**Figure 1.12.**Object Tree View

The Object TreeView, Object Inspector and the Form Designer (the Form1 window) work cooperatively. If you have an object on a form (we have not placed any yet) and click it, its properties and events are displayed in the Object Inspector and the component becomes focussed in the Object TreeView.

### 1.4.8.Class Completion

Class Completion generates skeleton code for classes. Place the cursor anywhere within a class declaration; then press Ctrl+Shift+C, or right-click and select Complete Class at Cursor. Delphi automatically adds private **read** and **write** specifiers to the declarations for any properties that require them, then creates skeleton code for all the class's methods. You can also use Class Completion to fill in class declarations for methods you've already implemented.

To configure Class Completion, choose Tools|Environment Options and click the Explorer tab.

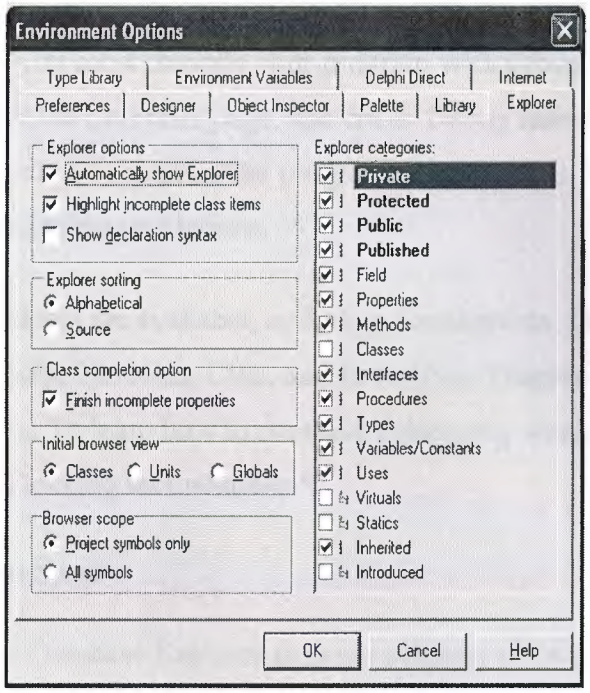
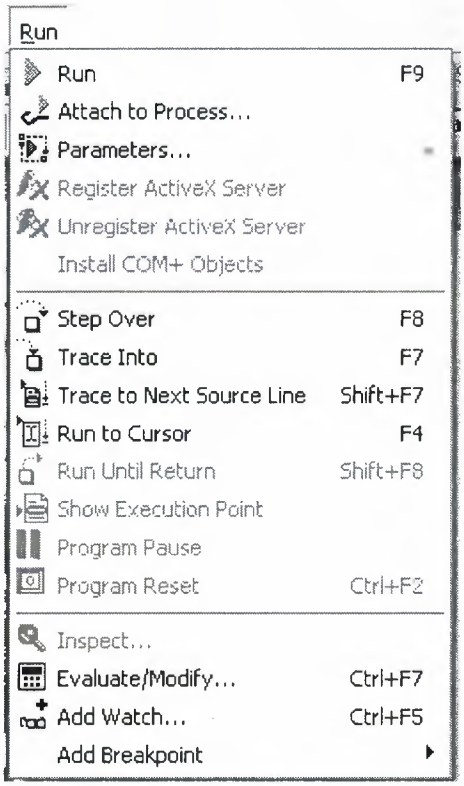



**Fig.1.13.**Class

### 1.4.9 Debugging applications

The IDE includes an integrated debugger that helps you locate and fix errors in your code. The debugger lets you control program execution, watch variables, and modify data values while your application is running. You can step through your code line by line, examining the state of the program at each breakpoint.




Choose any of the debugging commands from the Run menu.

Some commands are also available on the toolbar.

**Figure1.14.**Run

To use the debugger, you must compile your program with debug information. Choose Project|Options, select the Compiler page, and check Debug Information. Then you can begin a debugging session by running the program from the IDE. To set debugger options, choose Tools|Debugger Options.

Many debugging windows are available, including Breakpoints, Call Stack, Watches, Local Variables, Threads, Modules, CPU, and Event Log. Display them by choosing View|Debug Windows. To learn how to combine debugging windows for more convenient use, see "Docking tool windows".

### 1.4.10.Exploring databases

The SQL Explorer (or Database Explorer in some editions of Delphi) lets you work directly with a remote database server during application development. For example, you can create, delete, or restructure tables, and you can import constraints while you are developing a database application.

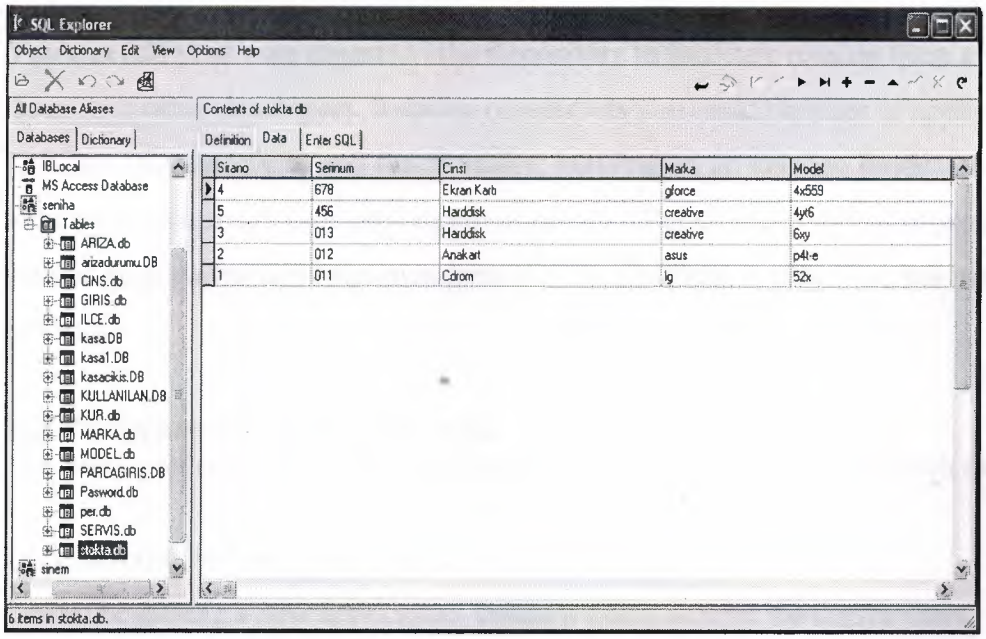



**Figure 1.15.**SQL Explorer

### 1.4.11.Templates and the Object Repository

The Object Repository contains forms, dialog boxes, data modules, wizards, DLLs, sample applications, and other items that can simplify development. Choose File|New to display the New Items dialog when you begin a project. Check the Repository to see if it contains an object that resembles one you want to create.

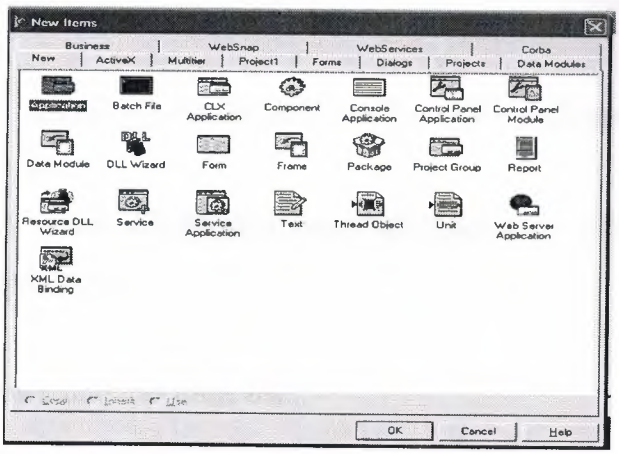



**Figure 1.16**.New Item

You can add your own objects to the Repository to facilitate reusing them and sharing them with other developers. Reusing objects lets you build families of applications with common user interfaces and functionality; building on an existing foundation also reduces development time and improves quality. The Object Repository provides a central location for tools that members of a development team can access over a network.

## 1.5.Programming With Delphi

The following section provide an overwiew of software development with Delphi.

### 1.5.1.Starting a New Application

Before beginning a new application, create a folder to hold the source files.

1. Create a folder called Seniha in the Projects directory off the main Delphi directory.
2. Open a new project.

Each application is represented by a project . When you start Delphi, it opens a blank project by default. If another project is already open, choose File|New Application to create a new project.

When you open a new project, Delphi automatically creates the following files.

- Project1.DPR : a source-code file associated with the project. This is called a project file.
- Unit1.PAS : a source-code file associated with the main project form. This is called a unit file.
- Unit1.DFM : a resource file that stores information about the main project form. This is called a form file.

Each form has its own unit and form files.

3. Choose File|Save All to save your files to disk. When the Save dialog appears, navigate to your Seniha folder and save each file using its default name.

Later on, you can save your work at any time by choosing File|Save All.
When you save your project, Delphi creates additional files in your project directory. You don't need to worry about them but don't delete them.
When you open a new project, Delphi displays the project's main form, named Form1 by default. You'll create the user interface and other parts of your application by placing components on this form.

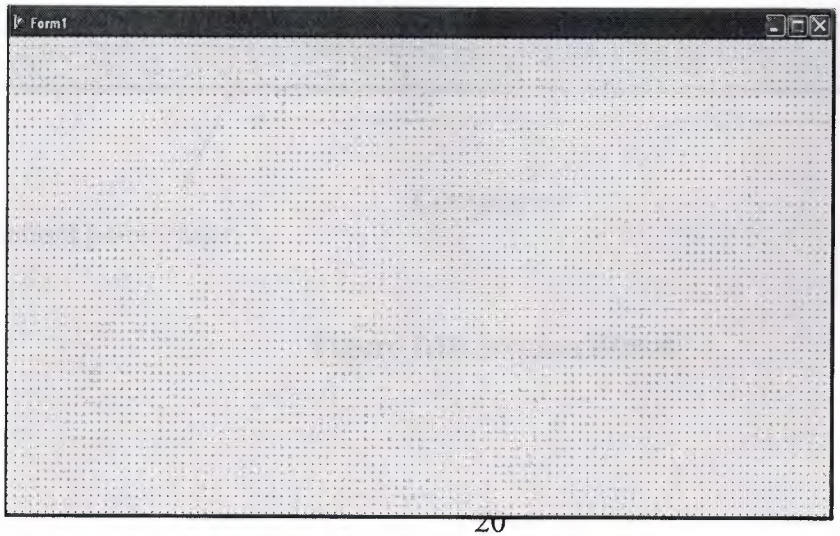



**Figure 1.17.**Form Screen

The default form has maximize , minimize buttons and a close button , and a control menu.Next to the form, you'll see the Object Inspector, which you can use to set property values for the form and components you place on it.

The drop-down list at the top of the Object Inspector shows the current selected object.when an object is sellected the Object Inspector show its properties.

### 1.5.1.1. Setting Property Values

When you use the Object Inspector to set properties, Delphi maintains your source code for you. The values you set in the Object Inspector are called *design-time* settings.

For Example ; Set the background color of Form1 to Aqua.

Find the form's Color property in the Object Inspector and click the drop-down list displayed to the right of the property. Choose clAqua from the list.

### 1.5.2. Adding objects to the form

The Component palette represents components by icons grouped onto tabbed pages. Add a component to a form by selecting the component on the palette, then clicking on the form where you want to place it. You can also double-click a component to place it in the middle of the form.

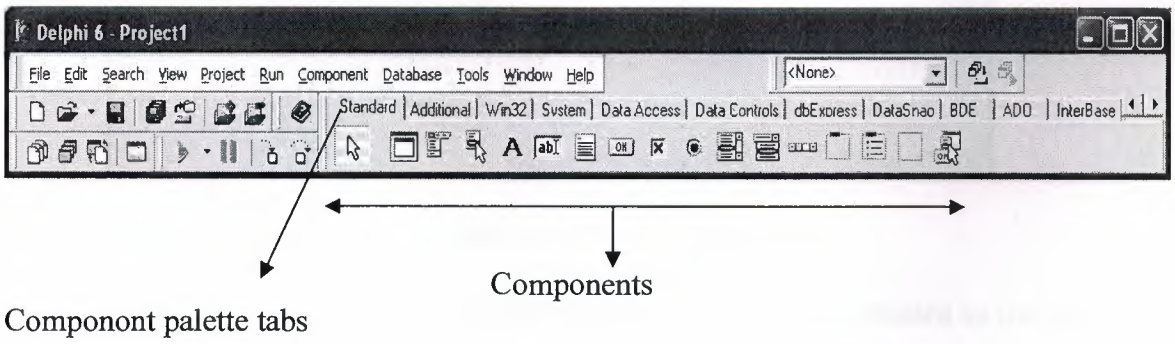



**Figure 1.18**.Standart Button

### 1.5.3.Add a Table and a StatusBar to the form:

Drop a Table component onto the form. Click the BDE tab on the Component palette. To find the *Table* component, point at an icon on the palette for a moment; Delphi displays a Help hint showing the name of the component.

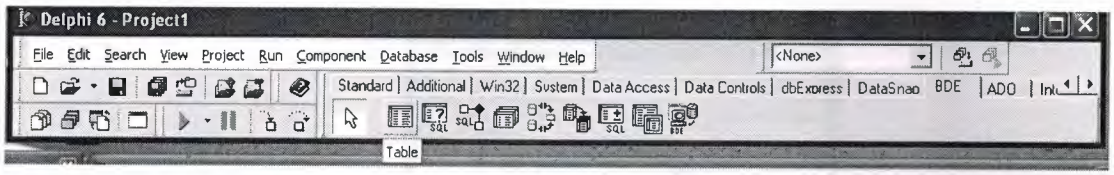



**Fig.1.19.**BDE Component palette

When you find the Table component, click it once to select it, then click on the form to place the component. The Table component is nonvisual, so it doesn't matter where you put it. Delphi names the object Table1 by default. (When you point to the component on the form, Delphi displays its name--Table1--and the type of object it is--TTable.)

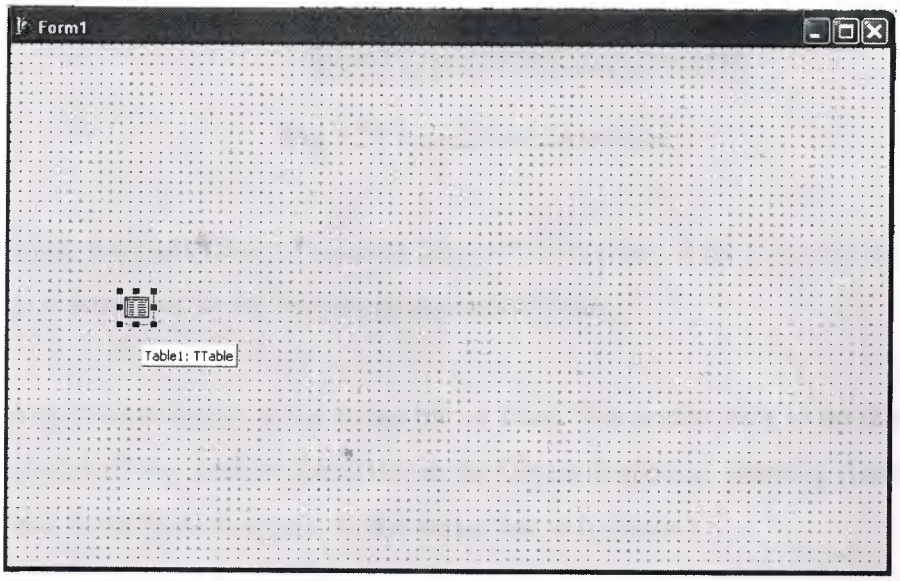



**Figure 1.20.**Table In The Form

Each Delphi component is a class; placing a component on a form creates an instance of that class. Once the component is on the form, Delphi generates the code necessary to construct an instance object when your application is running.

Set the DatabaseName property of Table1 to DBDEMOS. (DBDEMOS is an alias to the sample database that you're going to use.)

Select Table1 on the form, then choose the DatabaseName property in the Object Inspector. Select DBDEMOS from the drop-down list.

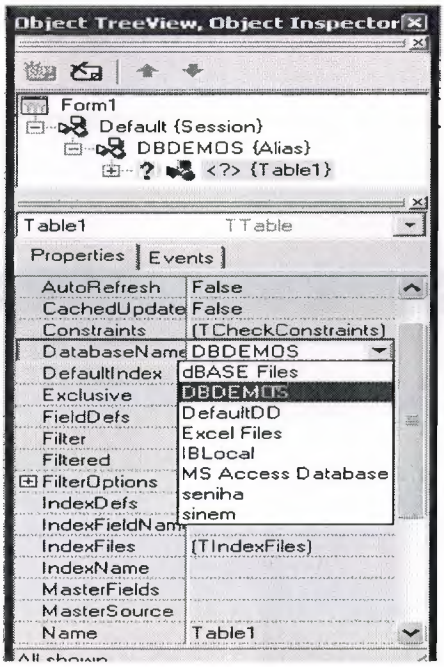



**Fig.1.21**.Select DatabaseName

Double-click the StatusBar component on the Win32 page of the Component palette. This adds a status bar to the bottom of the application.

Set the AutoHint property of the status bar to True. The easiest way to do this is to double-click on False next to AutoHint in the Object Inspector. (Setting AutoHint to True allows Help hints to appear in the status bar at runtime.)

### 1.5.4. Connecting to a Database

The next step is to add database controls and a DataSource to your form.

1. From the Data Access page of the Component palette, drop a DataSource component onto the form. The DataSource component is nonvisual, so it doesn't matter where you put it on the form. Set its DataSet property to Table1.

2. From the Data Controls page, choose the DBGrid component and drop it onto your form. Position it in the lower left corner of the form above the status bar, then expand it by dragging its upper right corner.

If necessary, you can enlarge the form by dragging its lower right corner. Your form should now resemble the following figure :

The Data Control page on Component palette holds components that let you view database tables.

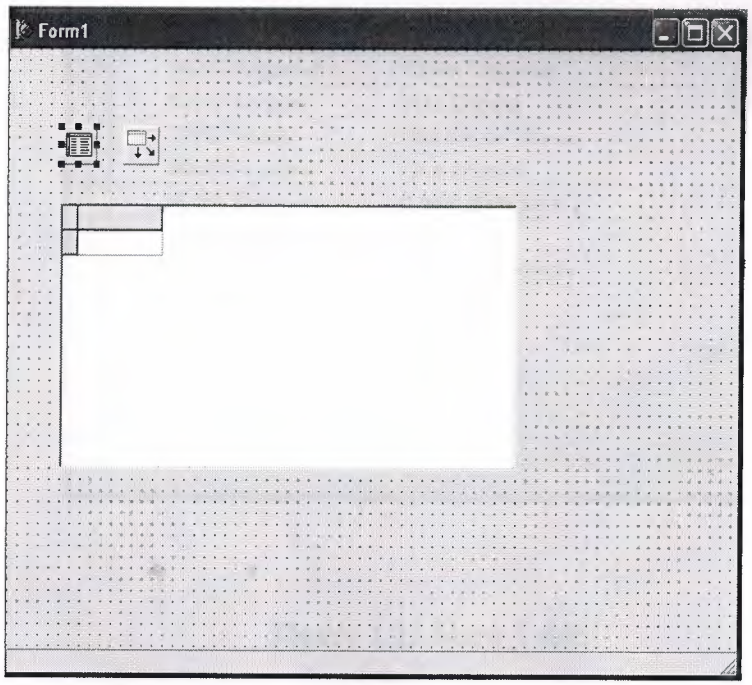



**Figure 1.22.**DBGrid In The Form

Set DBGrid properties to align the grid with the form. Double-click Anchors in the Object Inspector to display akLeft, akTop, akRight, and akBottom; set them all to True.

3. Set the DataSource property of DBGrid to DataSource1 (the default name of the DataSource component you just added to the form).

Now you can finish setting up the *Table1* object you placed on the form earlier.

4.Select the Table1 object on the form, then set its TableName property to BIOLIFE.DB. (Name is still Table1.) Next, set the Active property to True.

When you set Active to True, the grid fills with data from the BIOLIFE.DB database table. If the grid doesn't display data, make sure you've correctly set the properties of all the objects on the form, as explained in the instructions above. (Also verify that you copied the sample database files into your ...\Borland Shared\Data directory when you installed Delphi.)

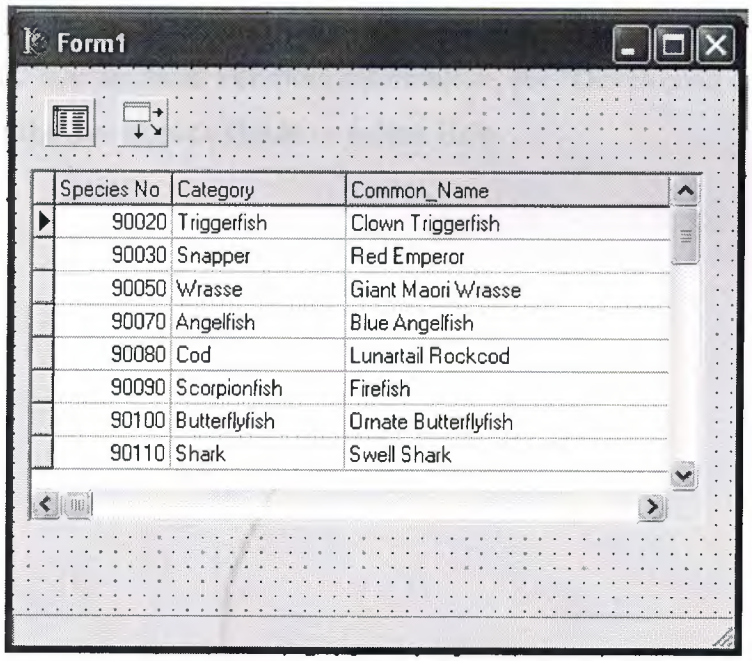
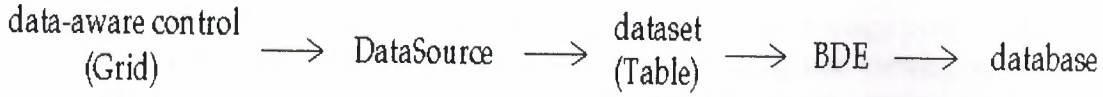



**Figure 1.23.**Show Table

The DBGrid control displays data at design time, while you are working in the IDE. This allows you to verify that you've connected to the database correctly. You cannot, however, edit the data at design time; to edit the data in the table, you'll have to run the application.

4. Press F9 to compile and run the project. (You can also run the project by clicking the Run button on the Debug toolbar, or by choosing Run from the Run menu.)
5. In connecting our application to a database, we've used three components and several levels of indirection. A data-aware control (in this case, a DBGrid) points to a DataSource object, which in turn points to a dataset object (in this

case, a Table). Finally, the dataset (Table1) points to an actual database table (BIOLIFE), which is accessed through the BDE alias DBDEMOS. (BDE aliases are configured through the BDE Administrator.)




This architecture may seem complicated at first, but in the long run it simplifies development and maintenance. For more information, see "Developing database applications" in the Developer's Guide or online Help.

# CHAPTER 2

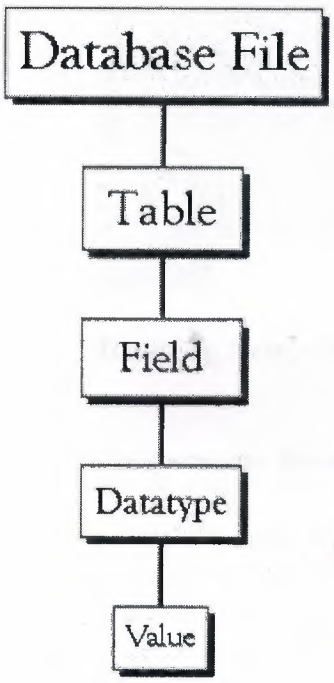
MICROSOFT ACCESS

## 1.6 MICROSOFT ACCESS DESCRIPTION

- Microsoft Access is a powerful program to create and manage your databases. It has many built in features to assist you in constructing and viewing your information. Access is much more involved and is a more genuine database application than other programs such as Microsoft Works.

  This tutorial will help you get started with Microsoft Access and may solve some of your problems, but it is a very good idea to use the Help Files that come with Microsoft Access, or go to Microsoft's web site located at http://microsoft.com/office/access/default.htm for further assistance.

  First of all you need to understand how Microsoft Access breaks down a database. Some keywords involved in this process are: *Database File, Table, Record, Field, Data-type*. Here is the Hierarchy that Microsoft Access uses in breaking down a database.




**Database File:** This is your main file that encompasses the entire database and that is saved to your hard-drive or floppy disk.
Example) StudentDatabase.mdb

**Table:**A table is a collection of data about a specific topic. There can be multiple tables in a database.
Example #1) Students
Example #2) Teachers

**Field:**Fields are the different categories within a Table. Tables usually contain multiple fields.
Example #1) Student LastName
Example #2) Student FirstName

**Datatypes:**Datatypes are the properties of each field. A field only has 1 datatype.
FieldName) Student LastName
Datatype) Text

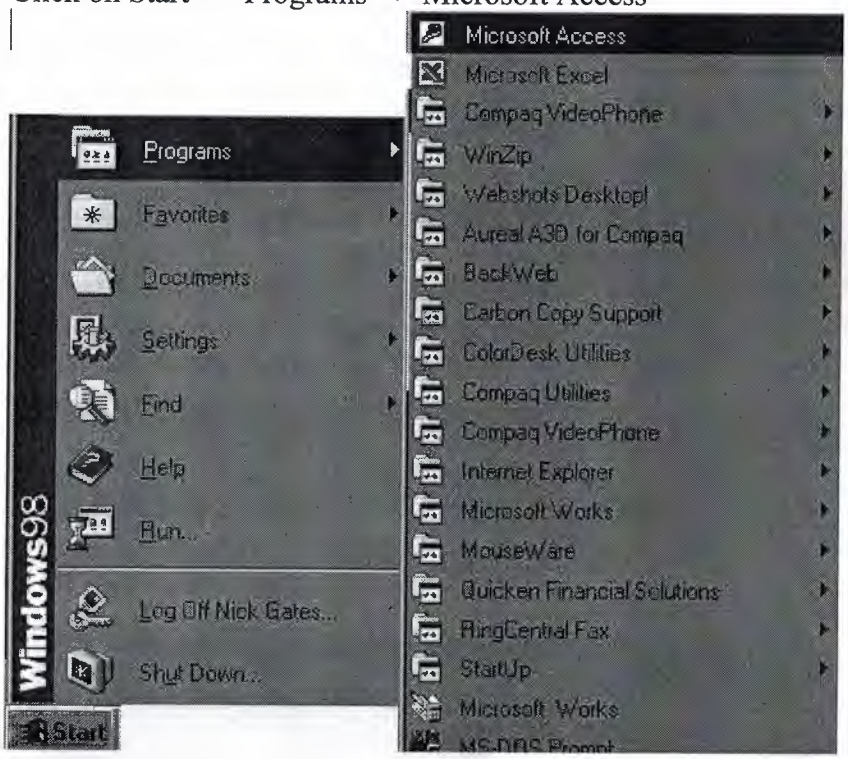
- 

  This tutorial will help you get started with Microsoft Access and may solve some of your problems, but it is a very good idea to use the Help Files that come with Microsoft Access (or any program you use for that matter), or go to Microsoft's web site located at http://microsoft.com/office/access/default.htm for further assistance.

### 1.6.1 Starting Microsoft Access

- Two Ways
  1. Double click on the Microsoft Access icon on the desktop.


  2. Click on Start --> Programs --> Microsoft Access




**Creating New, and Opening Existing Databases**

The above picture gives you the option to:

- Create a New Database from scratch
- Use the wizard to create a New Database
- Open an existing database

The white box gives you the most recent databases you have used. If you do not see the one you had created, choose the More Files option and hit OK. Otherwise choose the database you had previously used and click OK.

### 1.6.2 Create a database using the Database Wizard

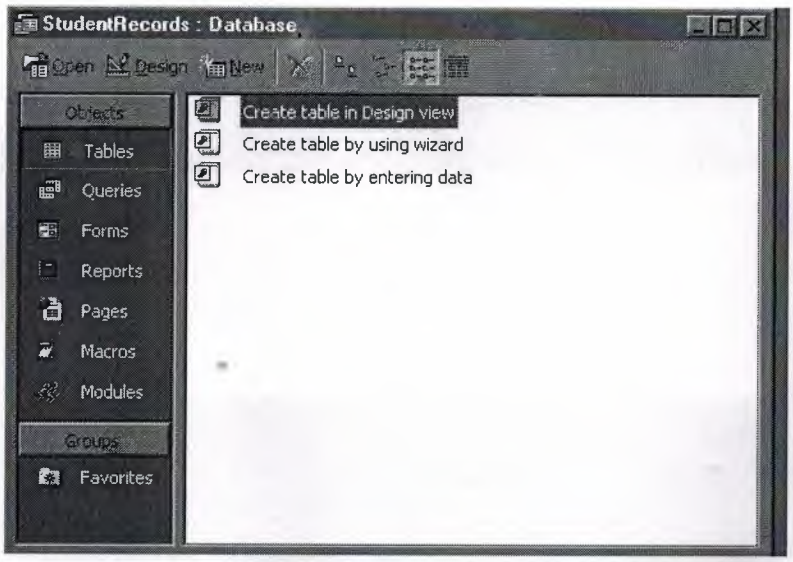
1. When Microsoft Access first starts up, a dialog box is automatically displayed with options to create a new database or open an existing one. If this dialog box is displayed, click **Access Database Wizards, pages, and projects** and then click **OK.**

   If you have already opened a database or closed the dialog box that displays when Microsoft Access starts up, click **New Database** on the toolbar.
2. On the **Databases** tab, double-click the icon for the kind of database you want to create.
3. Specify a name and location for the database.
4. Click **Create** to start defining your new database

#### 1.6.2.1 Create a database without using the Database Wizard

1. When Microsoft Access first starts up, a dialog box is automatically displayed with options to create a new database or open an existing one. If this dialog box is displayed, click **Blank Access Database**, and then click **OK**.

   If you have already opened a database or closed the dialog box that displays when Microsoft Access starts up, click **New Database** on the toolbar, and then double-click the **Blank Database** icon on the **General** tab.
2. Specify a name and location for the database and click **Create**. (Below is the screen that shows up following this step)



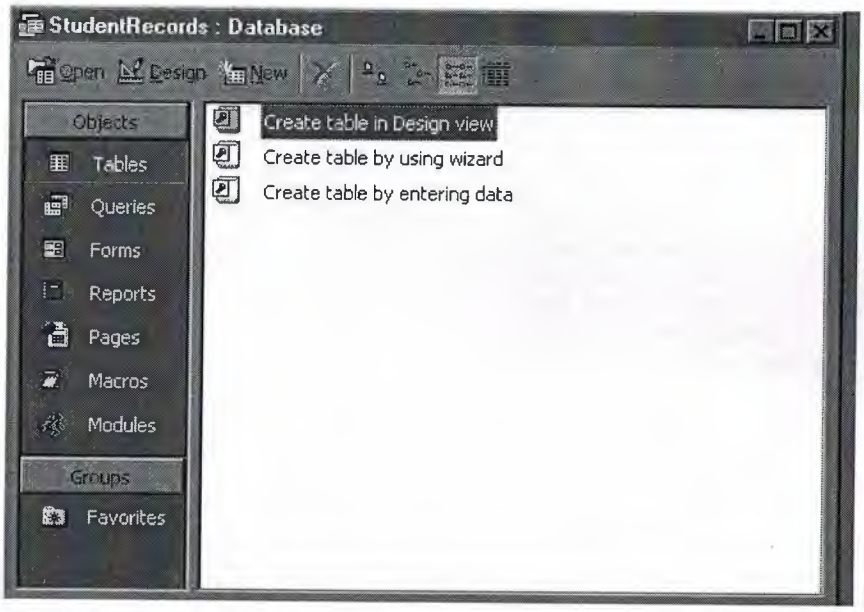

## 1.7 Tables

A table is a collection of data about a specific topic, such as students or contacts. Using a separate table for each topic means that you store that data only once, which makes your database more efficient, and reduces data-entry errors.

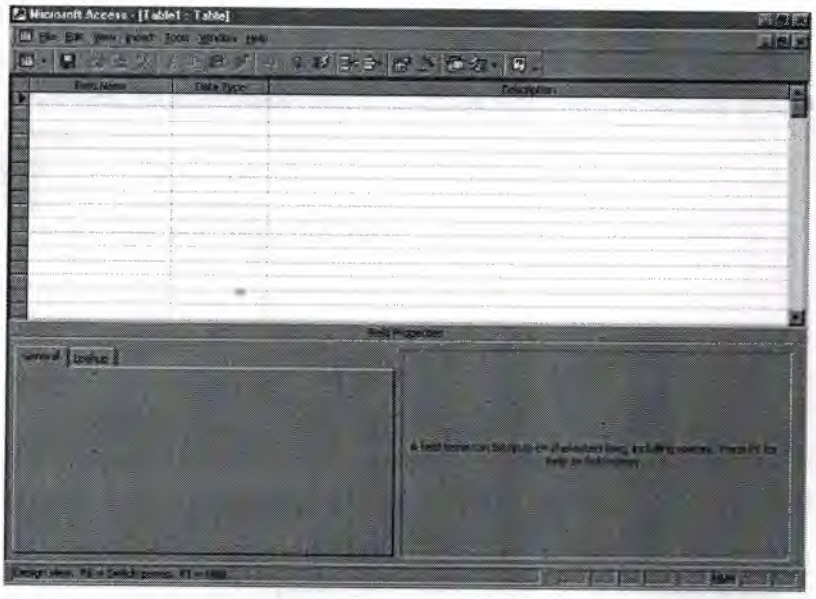
Tables organize data into columns (called **fields**) and rows (called **records**).

### 1.7.1 Create a Table from scratch in Design view

1. If you haven't already done so, switch to the Database Window You can press F11 to switch to the Database window from any other window.




2. Double-Click on **"Create table in Design view"**.
   *(DESIGN VIEW)*



3. Define each of the fields in your table.
   - Under the Field Name column, enter the categories of your table.
   - Under Data Type column, enter the type you want for you categories.

- The attribute of a variable or field that determines what kind of data it can hold. For example, in a Microsoft Access database, the Text and Memo field data types allow the field to store either text or numbers, but the Number data type will allow the field to store numbers only. Number data type fields store numerical data that will be used in mathematical calculations. Use the Currency data type to display or calculate currency values. Other data types are Date/Time, Yes/No, Auto Number, and OLE object (Picture).

- Under the Description column, enter the text that describes what you field is. (This field is optional).
- For our tutorial enter the following items:

| Field Name | Data Type | Description |
|---|---|---|
| Soc Sec # | Text | Social Security Number. Uniquely identifies a student |
| First Name | Text | Student's First Name |
| Last Name | Text | Student's Last Name |
| BirthDate | Date/Time | Student's Birthdate |
| Address | Text | Students Address |
| City | Text | City student resides in |
| State | Text | State student resides in |
| Zip | Text | Zip Code student resides in |
| Phone | Text | Student's home phone number |

### 1.7.2 Primary Key

- One or more fields (columns) whose value or values uniquely identify each record in a table. A primary key does not allow Null values and must always have a unique value. A primary key is used to relate a table to foreign keys in other tables.
- **NOTE:** You do not have to define a primary key, but it's usually a good idea. If you don't define a primary key, Microsoft Access asks you if you would like to create one when you save the table.
- For our tutorial, make the **Soc Sec #** field the primary key, meaning that *every* student has a social security number and no 2 are the same.
  - To do this, simply select the Soc Sec # field and select the primary key button

  - After you do this, Save the table

### 1.7.3 Switching Views

- To switch views form the datasheet (spreadsheet view) and the design view, simply click the button in the top-left hand corner of the Access program.

| Datasheet View | Design View |
|---|---|
|  Displays the view, which allows you to enter raw data into your database table. |  Displays the view, which allows you to enter fields, data-types, and descriptions into your database table. |

### 1.7.4 Entering Data

- Click on the Datasheet View and simply start "chugging" away by entering the data into each field. **NOTE:** Before starting a new record, the **Soc Sec #** field must have something in it, because it is the Primary Key. If you did not set a Primary Key then it is OK.

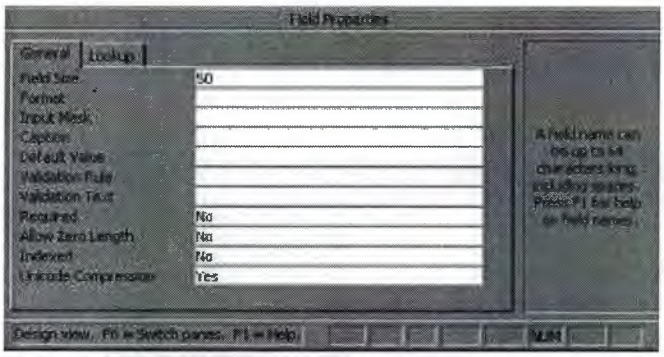

### 1.7.5 Manipulating Data

- **Adding a new row**
  - Simply drop down to a new line and enter the information
- **Updating a record**
  - Simply select the record and field you want to update, and change its data with what you want
- **Deleting a record**
  - Simply select the entire row and hit the Delete Key on the keyboard

### 1.7.6 Advanced Table Features w/Microsoft Access

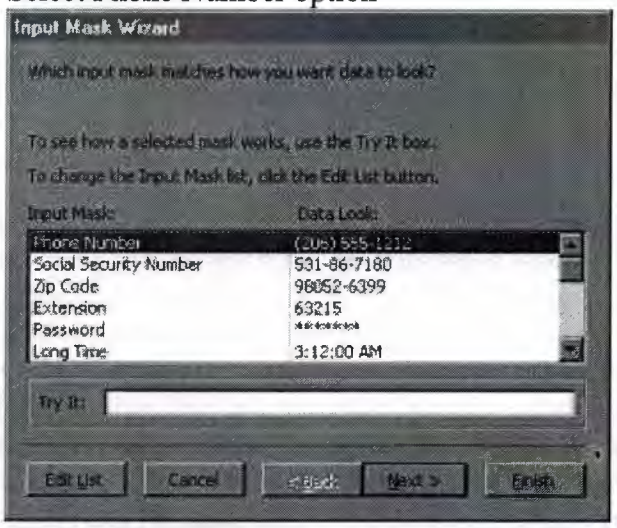
- **Assigning a field a specific set of characters**
  - Example) Making a Social Security Number only allows 9 characters.
    1. Switch to Design View
    2. Select the field you want to alter

3. At the bottom select the General Tab




4. Select **Field Size**
5. Enter the number of characters you want this field to have

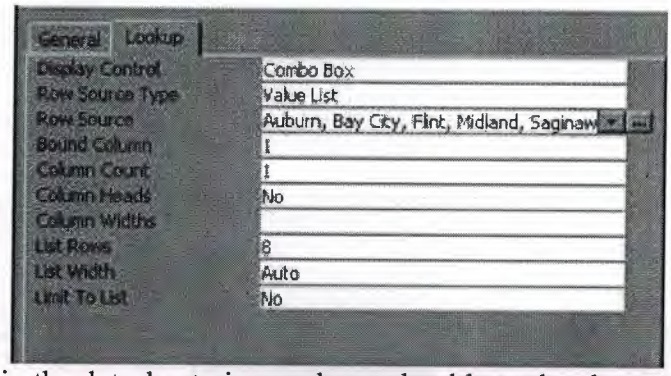
- **Formatting a field to look a specific way (HINT: You do not need to assign a field a specific set of characters if you do this)**
  - Example) Formatting Phone Number w/ Area Code (xxx) xxx-xxxx
    1. Switch to Design View
    2. Select the field you want to format
    3. At the bottom select the General Tab
    4. Select **Input Mask Box** and click on the **...** button at the right.
    5. Select Phone Number option



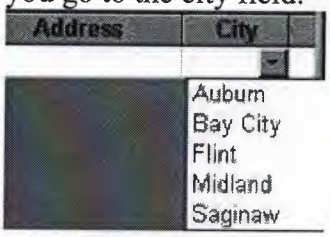

    6. Click on Next
    7. Leave *!(999) 000-0000 the way it is*. This is a default.
    8. Click Next
    9. Select which option you want it to look like
    10. Click Next
    11. Click Finish

- **Selecting a value from a dropdown box with a set of values that you assign to it. This saves you from typing it in each time**
  - Example)Choosing a city that is either Auburn, Bay City, Flint, Midland, or Saginaw
    1. Switch to Design View
    2. Select the field you want to alter (City)
    3. At the bottom select the Lookup Tab
    4. In the **Display Control** box, select **Combo Box**
    5. Under **Row Source Type**, select **Value List**

6. Under **Row Source**, enter the values how you want them displayed, separated by a comma. (*Auburn, Bay City, Flint, Midland, Saginaw*)
   - **NOTE:**This will not alphabetize them for you, so you will have to do that yourself. It should look something like this:




7. Select in the datasheet view and you should see the change when you go to the city field.




## 1.8 Relationships

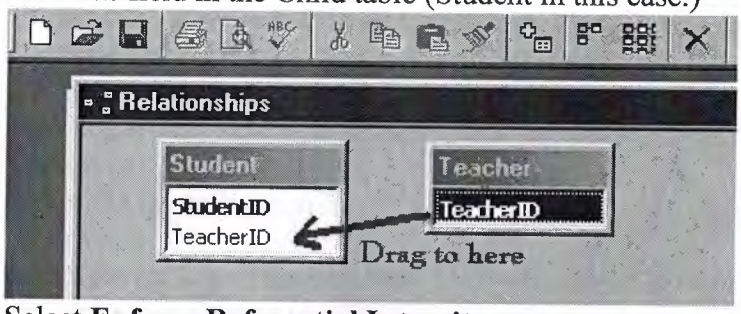
After you've set up multiple tables in your Microsoft Access database, you need a way of telling Access how to bring that information back together again. The first step in this process is to define relationships between your tables. After you've done that, you can create queries, forms, and reports to display information from several tables at once.

A relationship works by matching data in key fields - usually a field with the same name in both tables. In most cases, these matching fields are the primary key from one table, which provides a unique identifier for each record, and a foreign key in the other table. For example, teachers can be associated with the students they're responsible for by creating a relationship between the teacher's table and the student's table using the TeacherID fields.

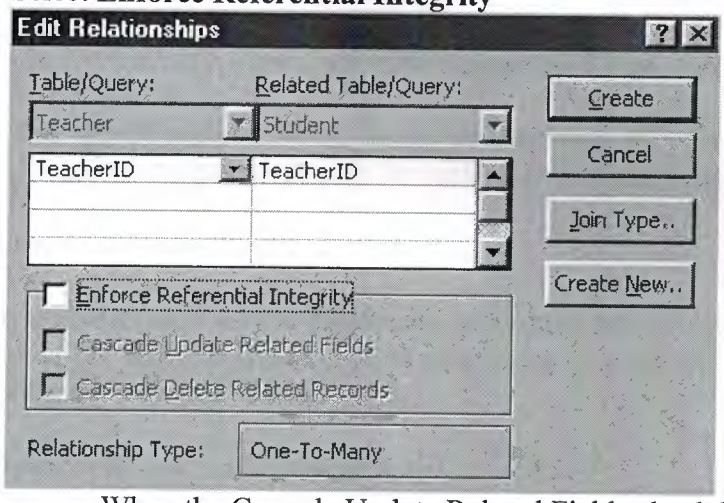
Having met the criteria above, follow these steps for creating relationships between tables.

1. In the database window view, at the top, click on Tools ---> Relationships
2. Select the Tables you want to link together, by clicking on them and selecting the Add Button

3. Drag the primary key of the Parent table (Teacher in this case), and drop it into the same field in the Child table (Student in this case.)




4. Select **Enforce Referential Integrity**




   - When the Cascade Update Related Fields check box is set, changing a primary key value in the primary table automatically updates the matching value in all related records.
   - When the Cascade Delete Related Records check box is set, deleting a record in the primary table deletes any related records in the related table
5. Click Create and Save the Relationship

## 1.9 Forms

A form is nothing more than a graphical representation of a table. You can add, update, delete records in your table by using a form. **NOTE:** Although a form can be named different from a table, they both still manipulate the same information and the same exact data. Hence, if you change a record in a form, it will be changed in the table also.

A form is very good to use when you have numerous fields in a table. This way you can see all the fields in one screen, whereas if you were in the table view (datasheet) you

would have to keep scrolling to get the field you desire.

### 1.9.1 Create a Form using the Wizard

It is a very good idea to create a form using the wizard, unless you are an advanced user and know what you are doing. Microsoft Access does a very good job of creating a form using the wizard. The following steps are needed to create a basic form:

1. Switch to the Database Window. You can do this by pressing F11 on the keyboard.
2. Click on the **Forms** button under **Objects** on the left side of screen
3. Double click on **Create Form Using Wizard**
4. On the next screen select the fields you want to view on your form. Most of the time you would select all of them.
5. Click Next
6. Select the layout you wish
7. Click Next
8. Select the style you desire...**HINT**: if you plan on printing your form, I suggest you use a light background to save on printer toner and ink
9. Click Next
10. Give you form a name, and select **Open the Form and enter information**
11. Select **Finish**
12. You should see your form. To adjust the design of your form, simply hit the design button (same as with the tables), and adjust your form accordingly

### 1.9.2 Reports

A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.

### 1.9.3 Create a Report using the Wizard

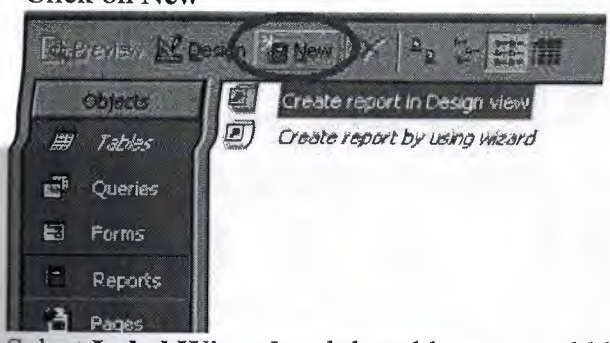
As with the Form, it is a very good idea to create a report using the wizard, unless you are an advanced user. Microsoft Access does a very good job using the wizard to create reports.

1. Switch to the Database Window. You can do this by pressing F11 on the keyboard.
2. Click on the **Reports** button under **Objects** on the left side of screen
3. Double click on **Create Report Using Wizard**
4. On the next screen select the fields you want to view on your form. Most of the time you would select all of them.
5. Click Next
6. Select if you would like to group your files. Keep repeating this step for as many groupings as you would like.
7. Click Next
8. Select the layout and the paper orientation you desire
9. Click Next
10. Select the style you desire...**HINT**: if you plan on printing your report, I suggest you use a light background to save on printer toner and ink
11. Click Next
12. Give you report a name, and select **Preview the Report**
13. Select **Finish**
14. You should see your report. To adjust the design of your report, simply hit the design button (same as with the tables), and adjust your report accordingly

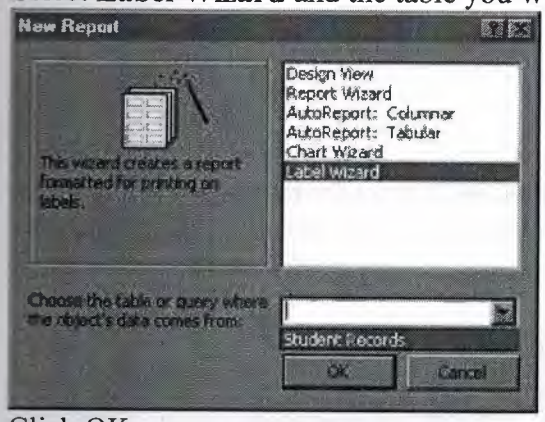
### 1.9.4 Creating Mail Merge Labels using a Wizard

Microsoft Access lets you create Mailing Labels for your database that you have. To do this do the following:

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




4. Select **Label Wizard** and the table you would like to get your information from.




5. Click OK
6. Select the layout of your labels
7. Click Next
8. Select the font size and color you want on each label
9. Click Next
10. Select how you want your label to look
11. Click Next
12. Select how you want your labels sorted
13. Give your label report a name and preview it

# CHAPTER 3

## 3.1 Picture Simulation Form

When the program stars there will be a simulation during ten seconds.


Figure 3.1

## 3.2 Login Screen Form

The user enters the username and password to login to program.if the information provided by the user is correct the main form will be opened else it will not be displayed.There is an option for the user to change login information.When the change password button is pressed a form will be displayed for the user to change the password.

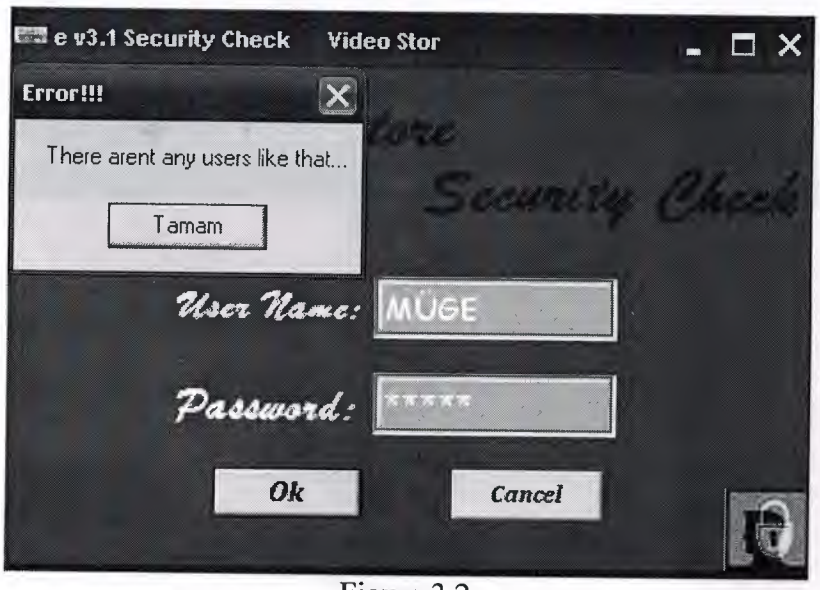



Figure 3.2


Figure 3.3


Figure 3.4

### 3.2 Main Menu Form

The aim of the main menu is to use the program easily, faster and use all the process screens or necessary program at the same time. Main menu of video store contains customers, sale, rental, videos, supplier, queries, help, about.

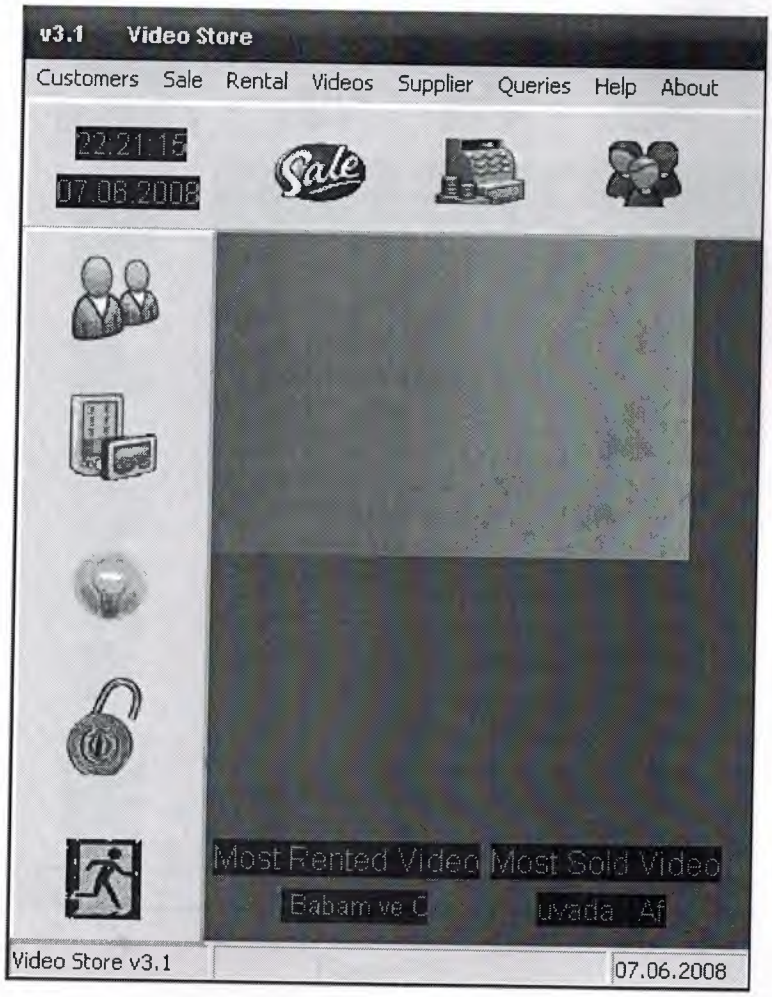



Figure 3.5

## 3.3 CUSTOMERS

This form contains informations all customers, customer operation and customer search.

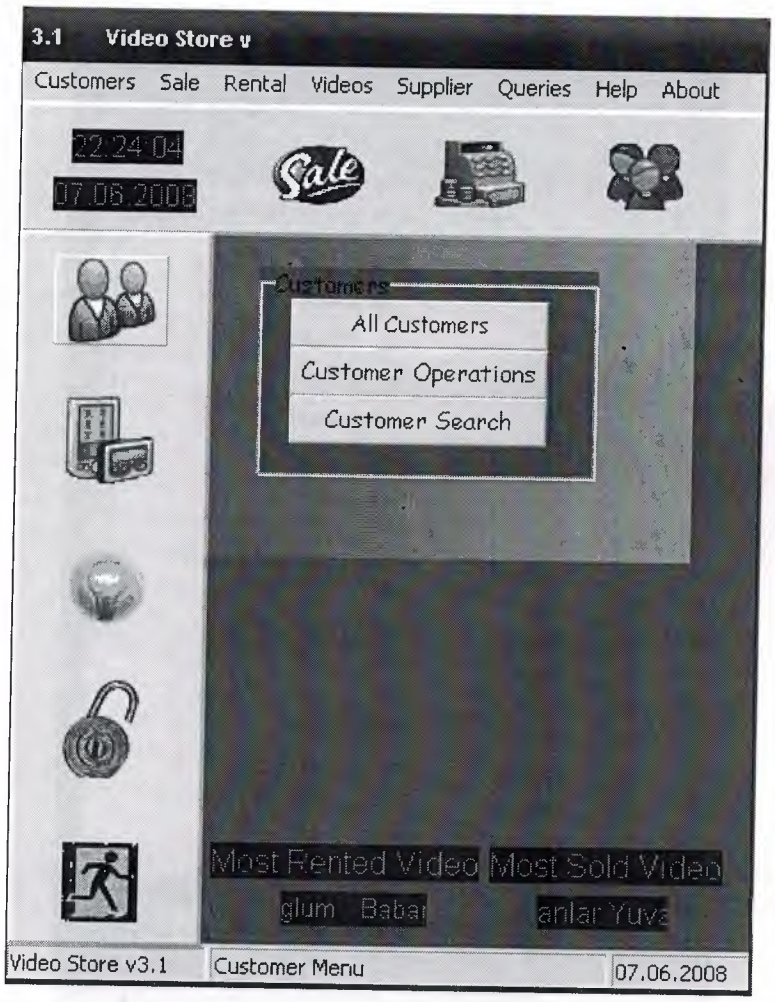



Figure 3.6

### 3.3.1 All Customers

This form contains the customers informations.It list of all customers with customer id, customer name, surname, telephone, address, city county in database.

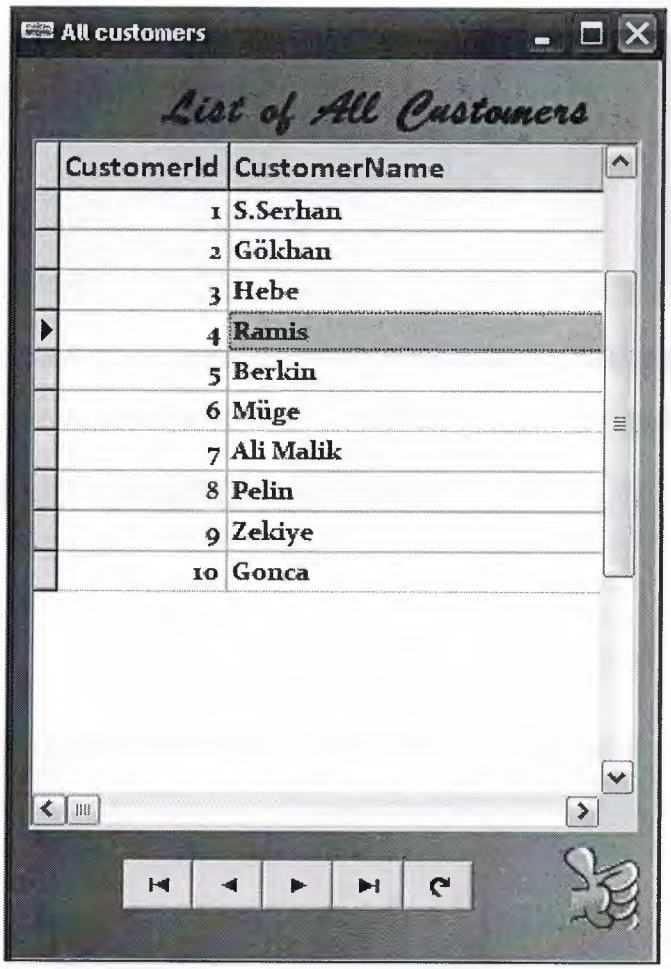



Figure 3.7

### 3.3.2 Customer Operation

This form shows personal informations of data. The customer can be searched from the database. The customer information will be entered by the user.


Figure 3.8

### 3.3.3 Searching Customer

The aim of the form is to use the searching customers. The form allows to reach the customer informations easily and faster.

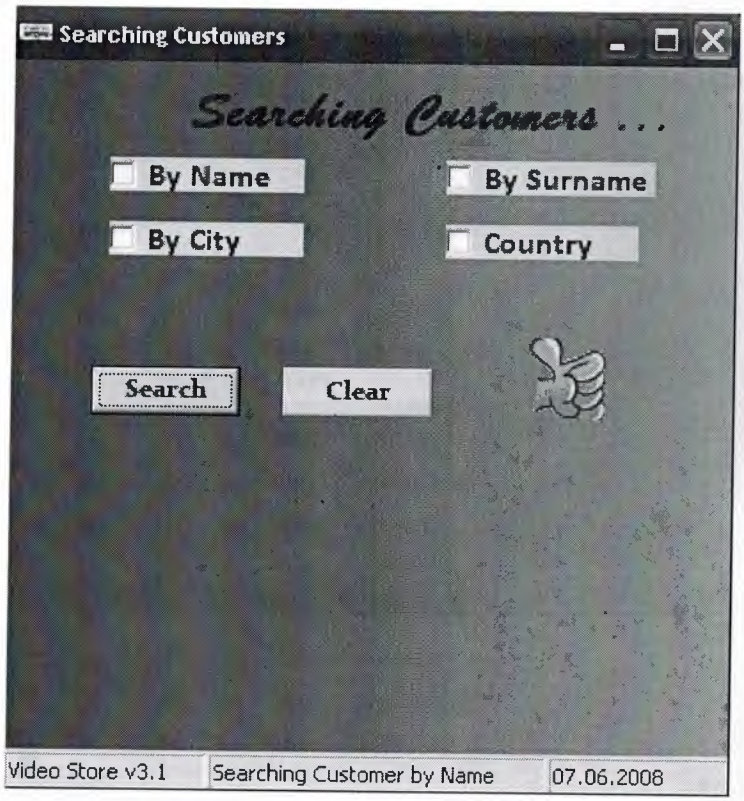



Figure 3.9

### 3.3.3.1 Searching Customer by Name

This form is searching customer by name. When user choose the customer name search option,the program will show the customer datas according to the their customer names from the data base.

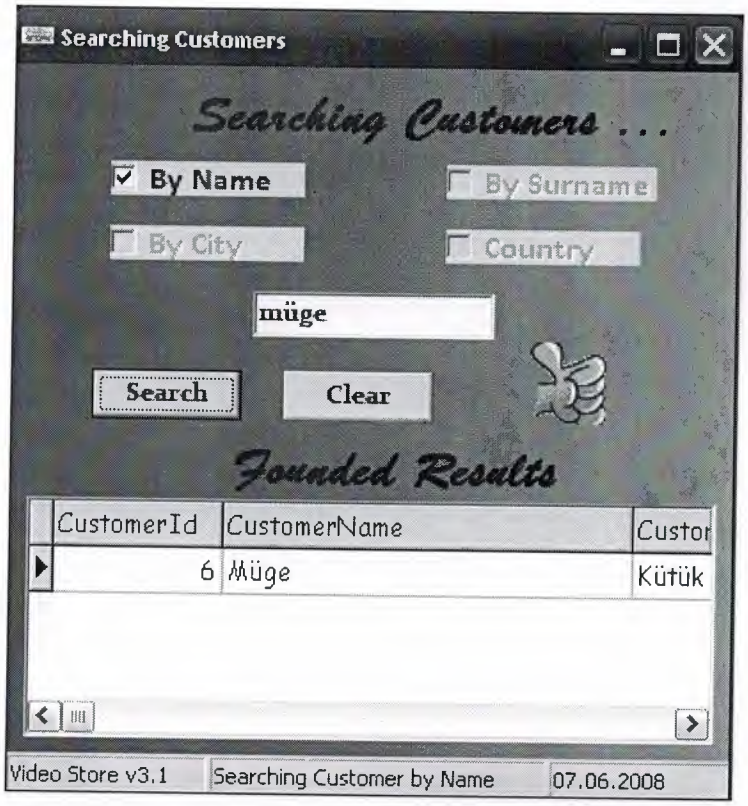



Figure 3.10

### 3.3.3.2 Searching Customer by Surname

In this form allows to search customer by surname. When user choose the customer surname search option,the program will show the customer datas according to the their customer surnames from the data base.

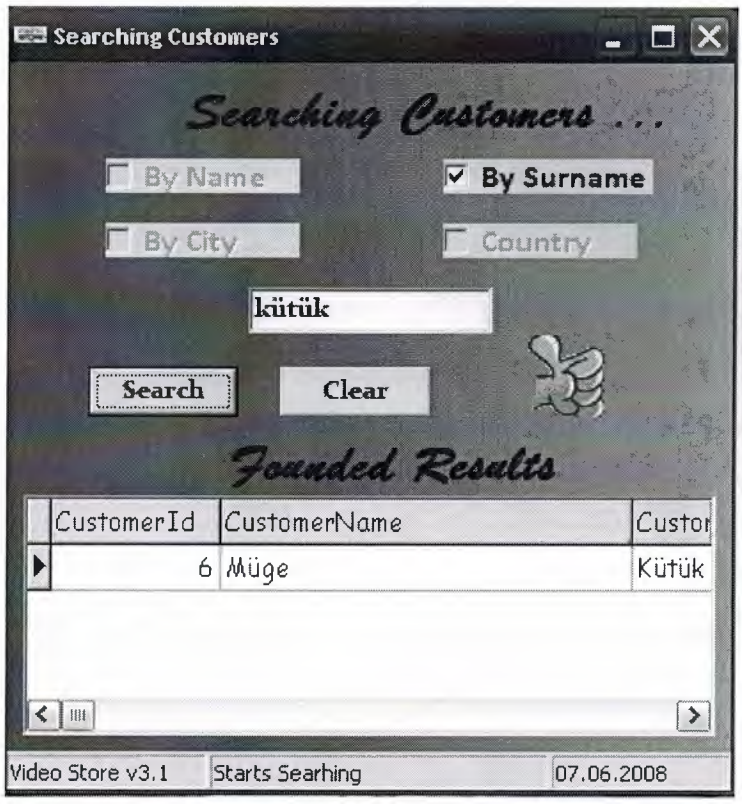



Figure 3.11

### 3.3.3.3 Searching Custemers by City

This form will show the customer datas according to the their citys from the data base.

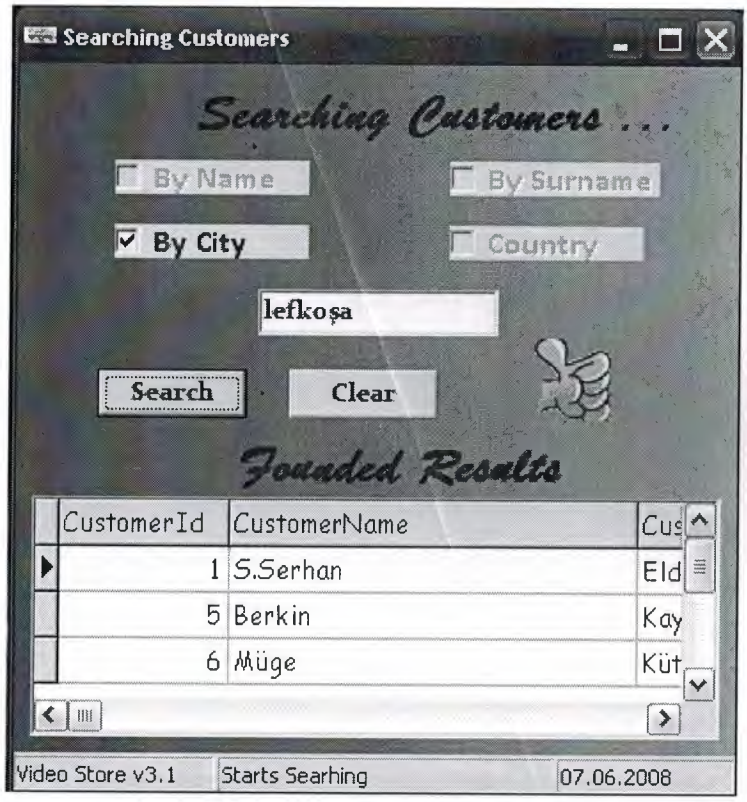



Figure 3.12

### 3.3.3.4 Searching Customer by Country

This form is searching customer by country.When user choose the country search option,the program will show the customer datas according to the their countries from the data base.

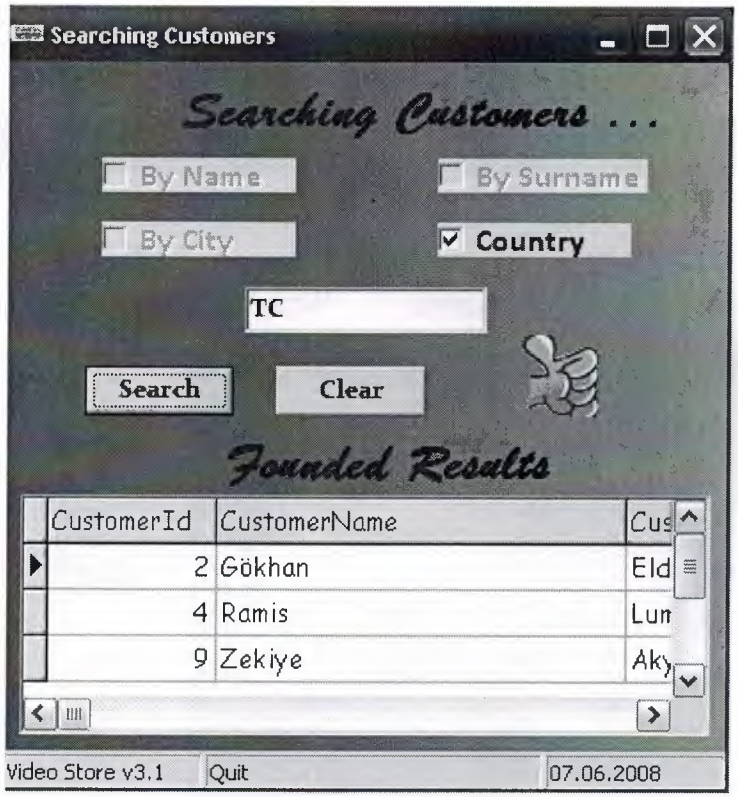



Figure 3.13

## 3.4 Videos

### 3.4.1 All Videos

This window contains the video informations. List of all videos with video id, video name, video description, video director, video type,video year, unit price, video stock and video picture in database.

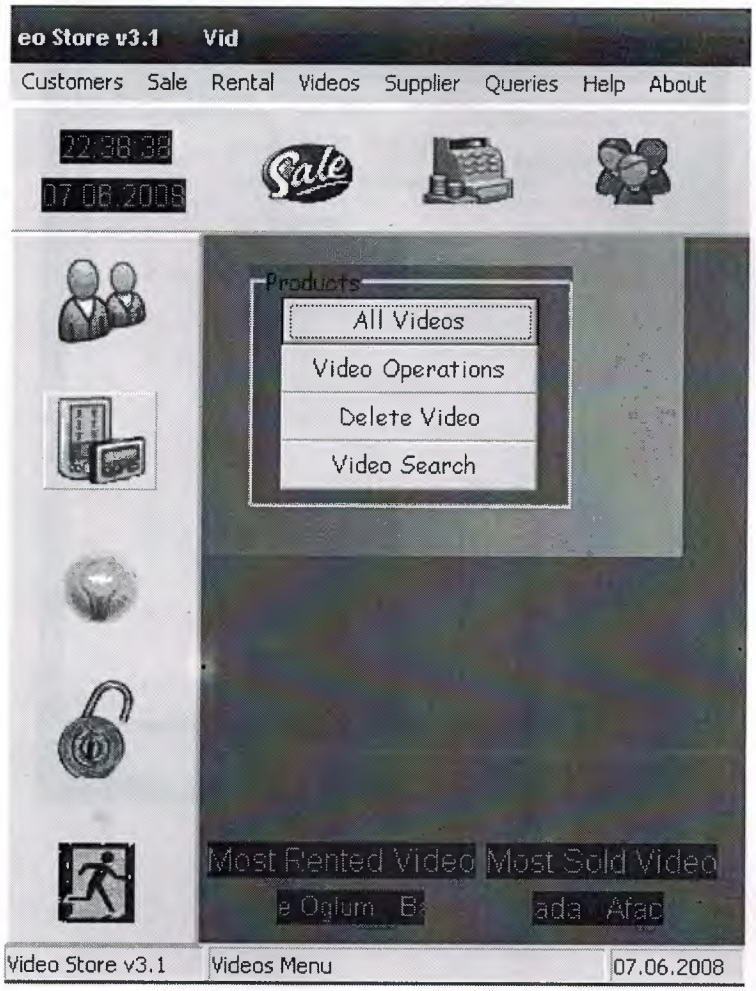



Figure 3.14

### 3.4.2 List of Video

This form contains the videos informations. List of all videos with video id, video name, video type, video description and video year in database.


Figure 3.15

### 3.4.3 Video Operator

In this form program will show video informations of data. The video can be searched, add and purchased an existing video from the database. The video store information will be entered by the user.

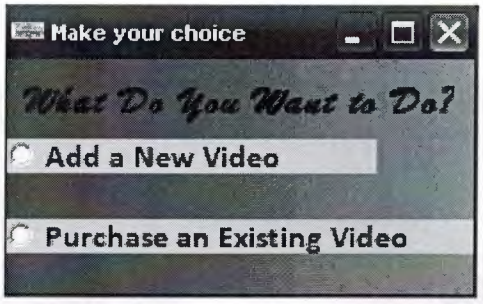



Figure 3.16

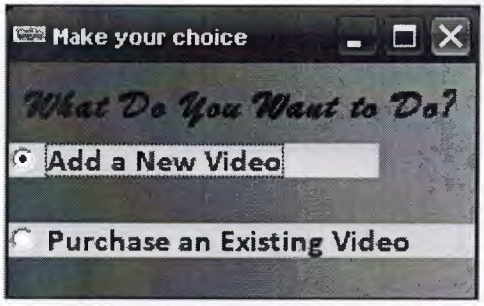



Figure 3.17

### 3.4.3.1 Add New Operations

The page is the video operation options page.In this page user can search an existing video and add a new video record. The video store informations will be entered by the user.


Figure 3.18

### 3.4.3.2 Purchase an Existing Video

User can search and purchase an existing video. Entering video price and video amount.

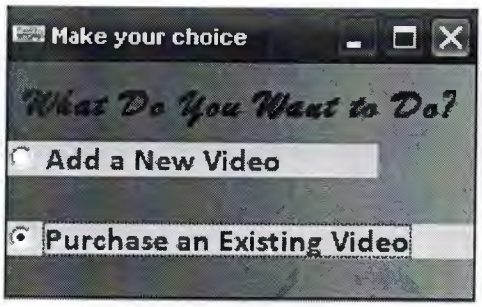



Figure 3.18

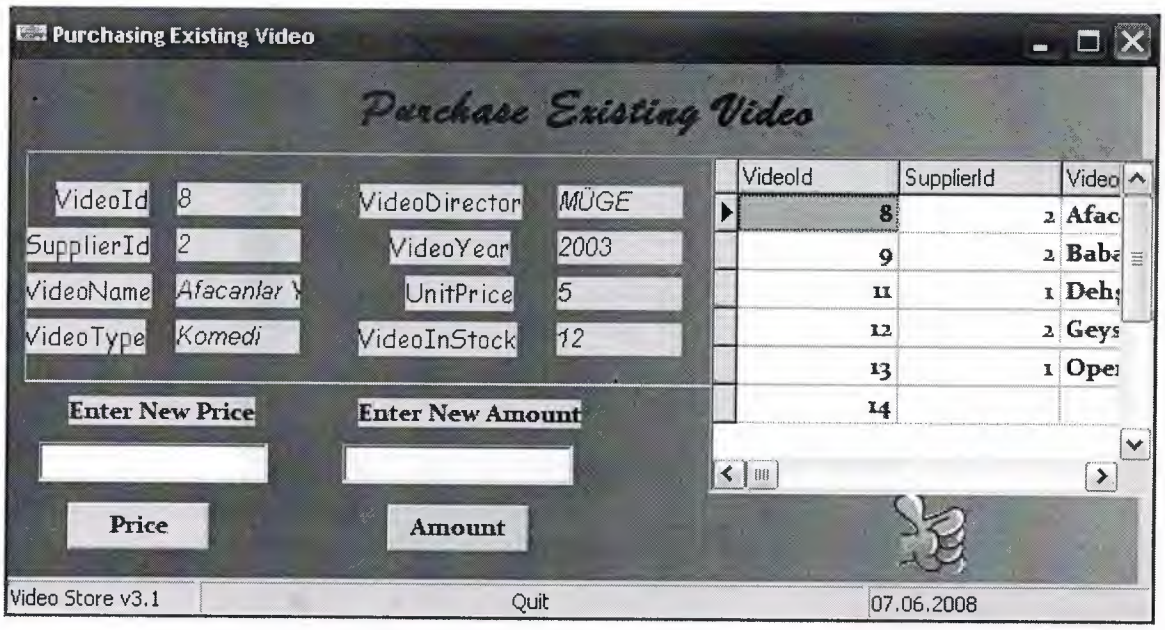



Figure 3.19

### 3.4.4 Delete Video

User can access list of all videos and delete video.


Figure 3.20

### 3.4.5 Video Search

The aim of the form is to use the searching videos. The form allows to reach the video informations easily and faster.


Figıre 3.21

### 3.4.5.1 Searching Video by Name

This form is searching videos by name. When user choose the video name search option,the program will show the video datas according to the their video names from the data base.

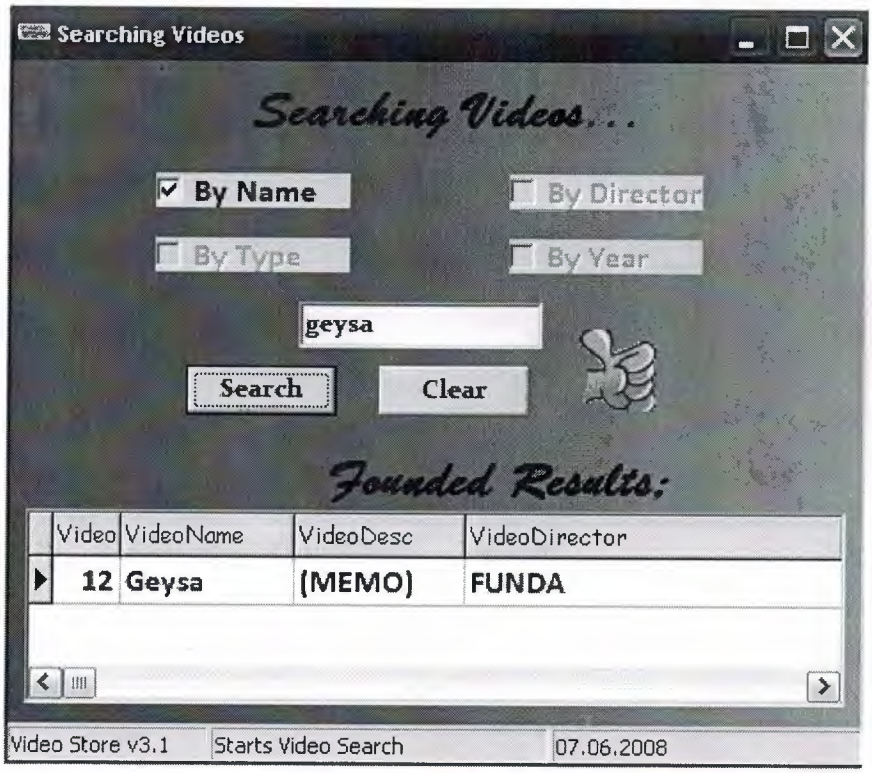



Figure 3.22

### 3.4.5.2 Searching videos by Director

In this form user searchs videos by director.When user choose the video director search option,the program will show the video datas according to the their video director from the data base.

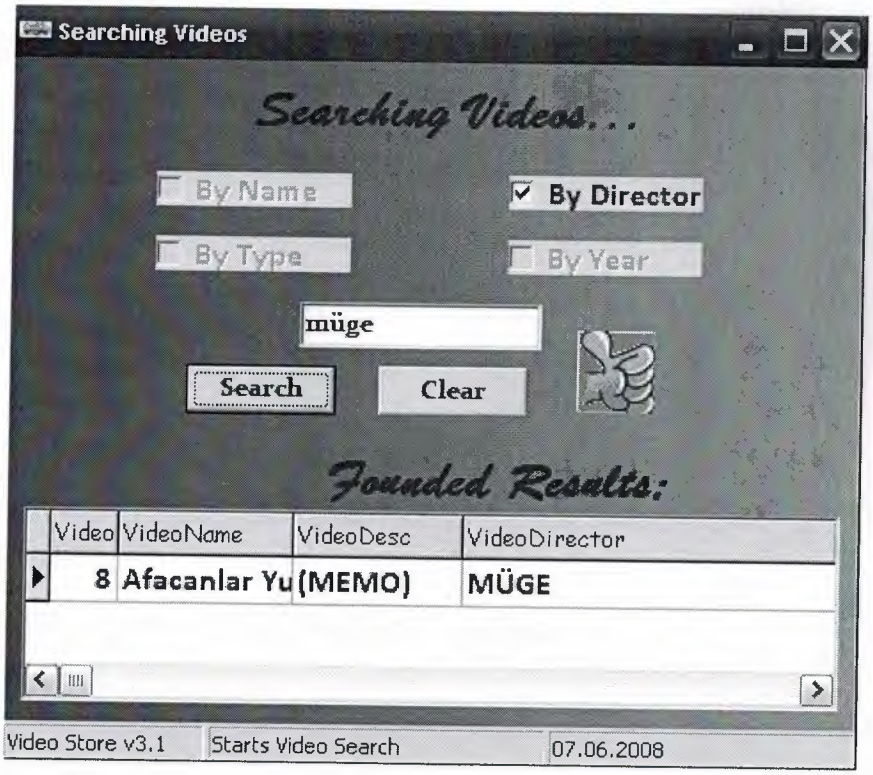



Figure 3.23

### 3.4.5.3 Searching Videos by Type

In this form user searchs videos by type.When user choose the video type search option,the program will show the video datas according to the their video type from the data base.

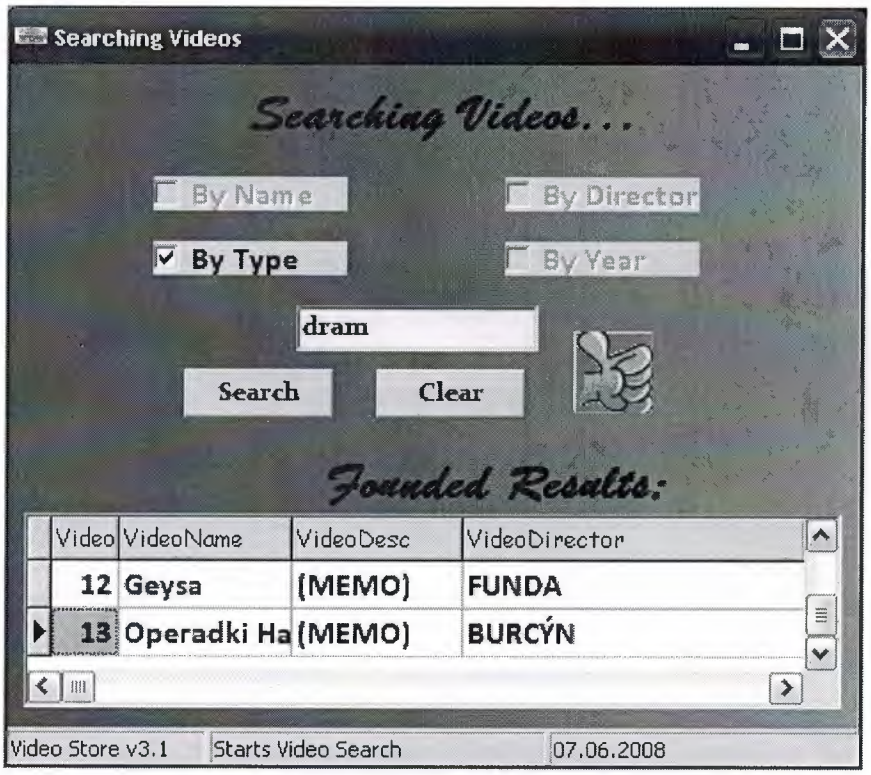



Figure 3.24

### 3.4.5.4 Searching Videos by Year

In this form user searchs videos by year.When user choose the video year search option,the program will show the video datas according to the their video year from the data base.

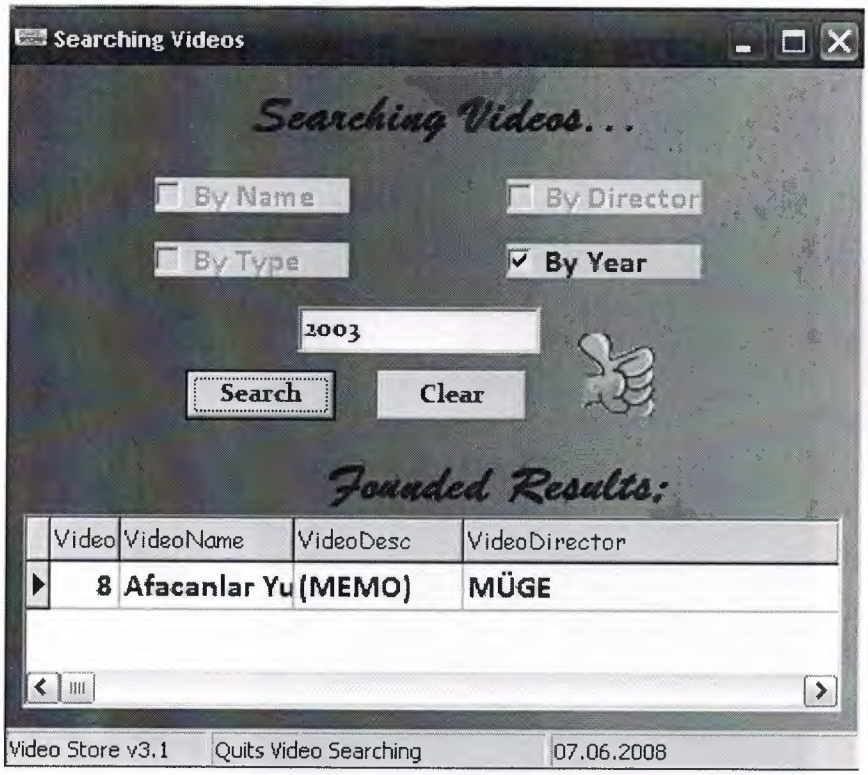



Figure 3.25

### 3.5 Suppliers

This form contains informations all suppliers, suppliers operations,delete supplier and supplier search.

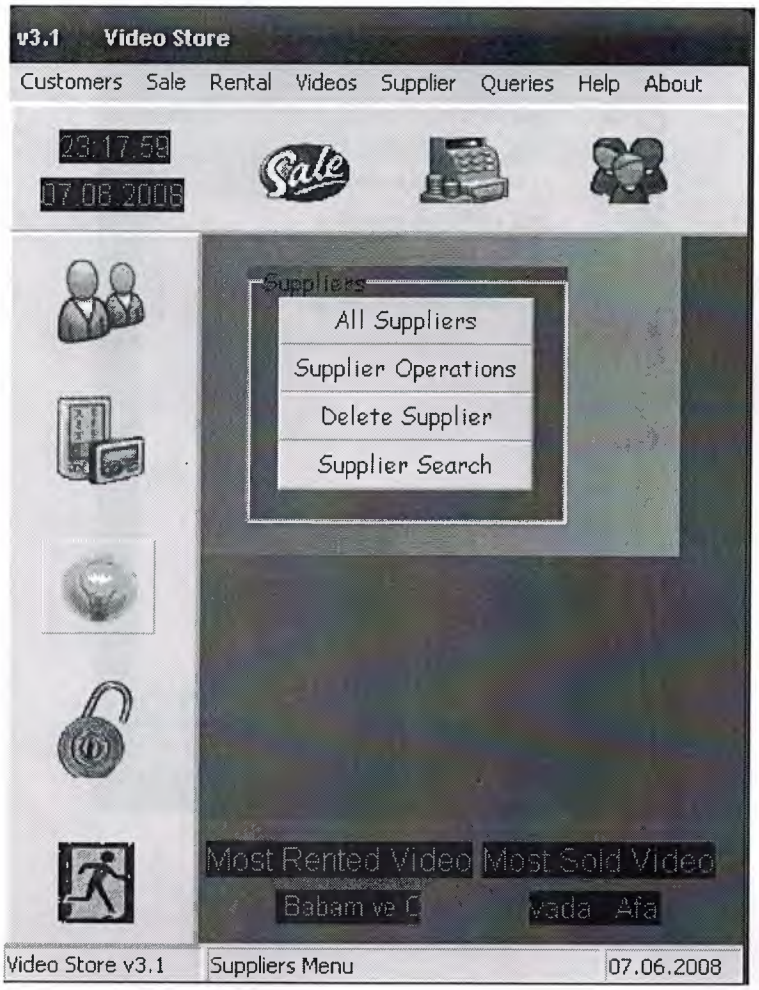



Figure 3.26

### 3.5.1 All Suppliers

This window contains the suppliers informations. List of all suppliers belong supplier id, supplier name, telephone, city, address and county in database.

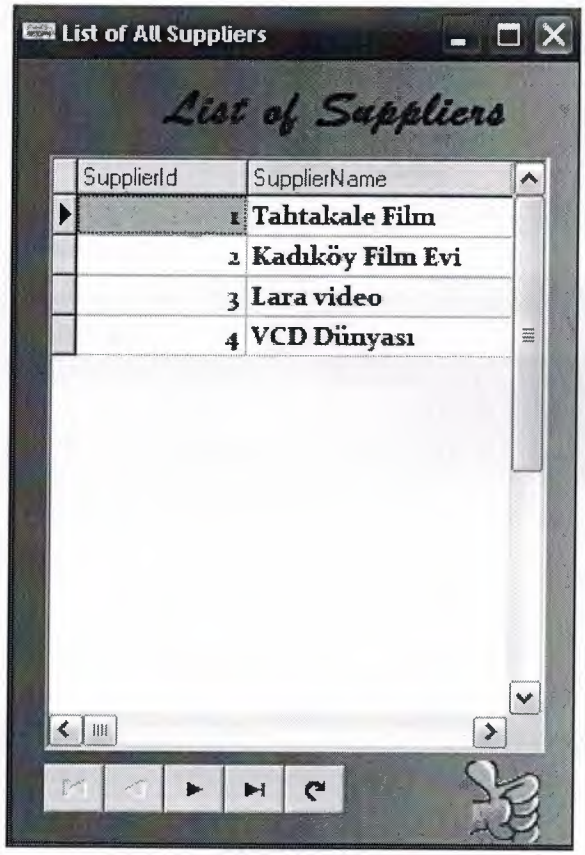



Figure 3.27

### 3.5.2 Suppliers Operations

The page is the supplier operations options page. In this page user can search datas. The supplier operation informations will be entered by the user.


Figure 3.28

### 3.5.3 Deleting Suppliers

User can access list of all suppliers and delete videos.


Figure 3.29

### 3.5.4 Supplier Searching

The aim of the form is to use the searching suppliers. The form allows to reach the supplier informations easily and faster.


Figure 3.30

### 3.5.4.1 Searching Supplier by Name

In this form user searchs supplier by name. When user choose the supplier name search option,the program will show the supplier datas according to the their supplier names from the data base.


Figure 3.31

### 3.5.4.2 Searching Supplier by City

In this form user searchs supplier by city. When user choose the supplier city search option,the program will show the supplier datas according to the their supplier city from the data base.


Figure 3.32

### 3.6 Security process

The user enters the password required to login to process.If the information provided by the user is correct the main form will be opened else it will not be displayed.There is an option for the user to change login information.When the change password button is pressed a form will be displayed for the user to change the password.

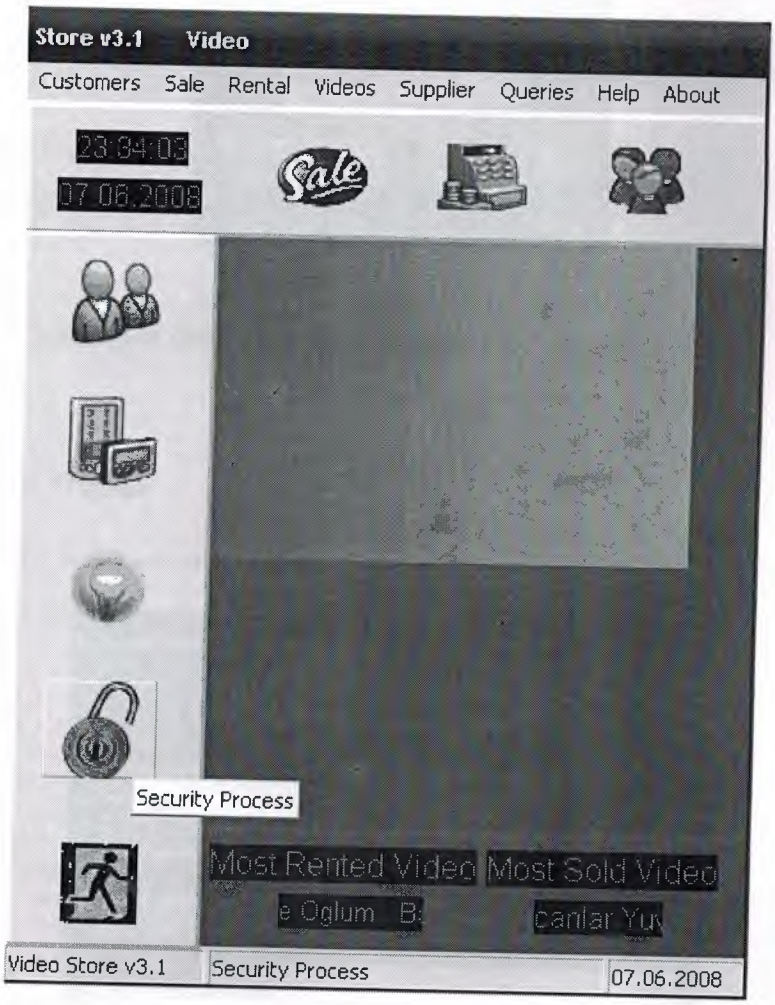



Figure 3.33

### 3.6.1 Password Required

User will enter the password to see the security processes.


Figure 3.34

If the correct password is given, the form in below will appear.

### 3.6.2 Password Process

Password process form will allow to user to change and update the user name and password.


Figure 3.35

### 3.7 Exist

It will be used to close the menu.

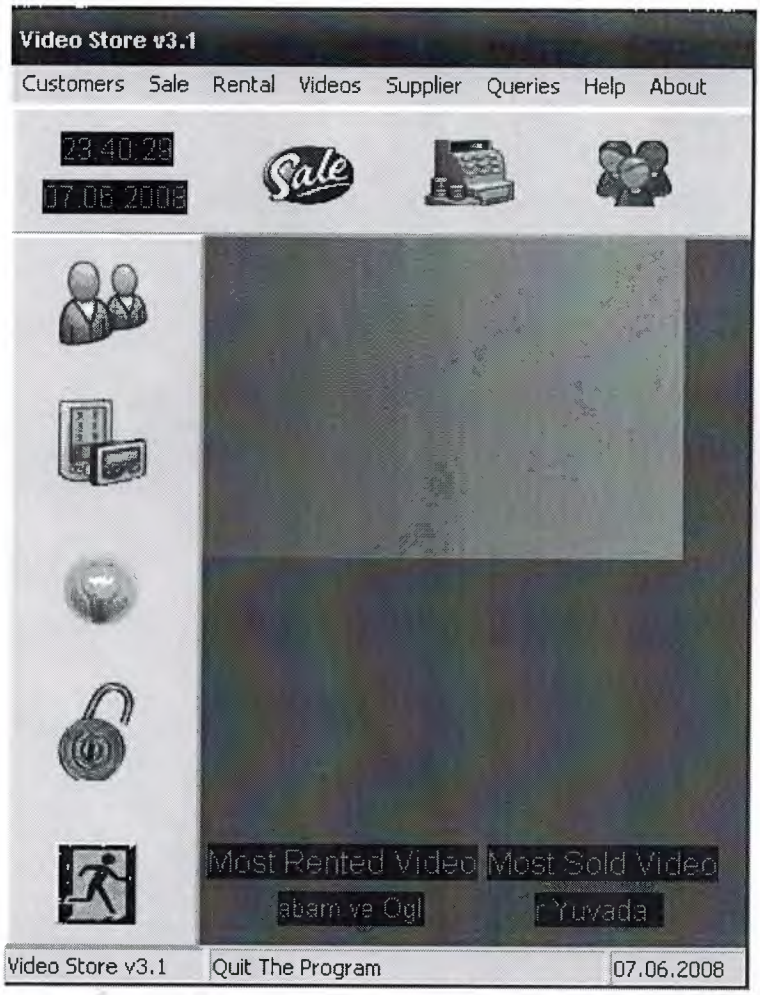



Figure 3.36

### 3.8 Sale

The sale button in this form will guide to the video sales form.


Figure 3.37

### 3.8.1 Video Sales

The form calculates the quantity, unit price and price according to the video id and customer id.

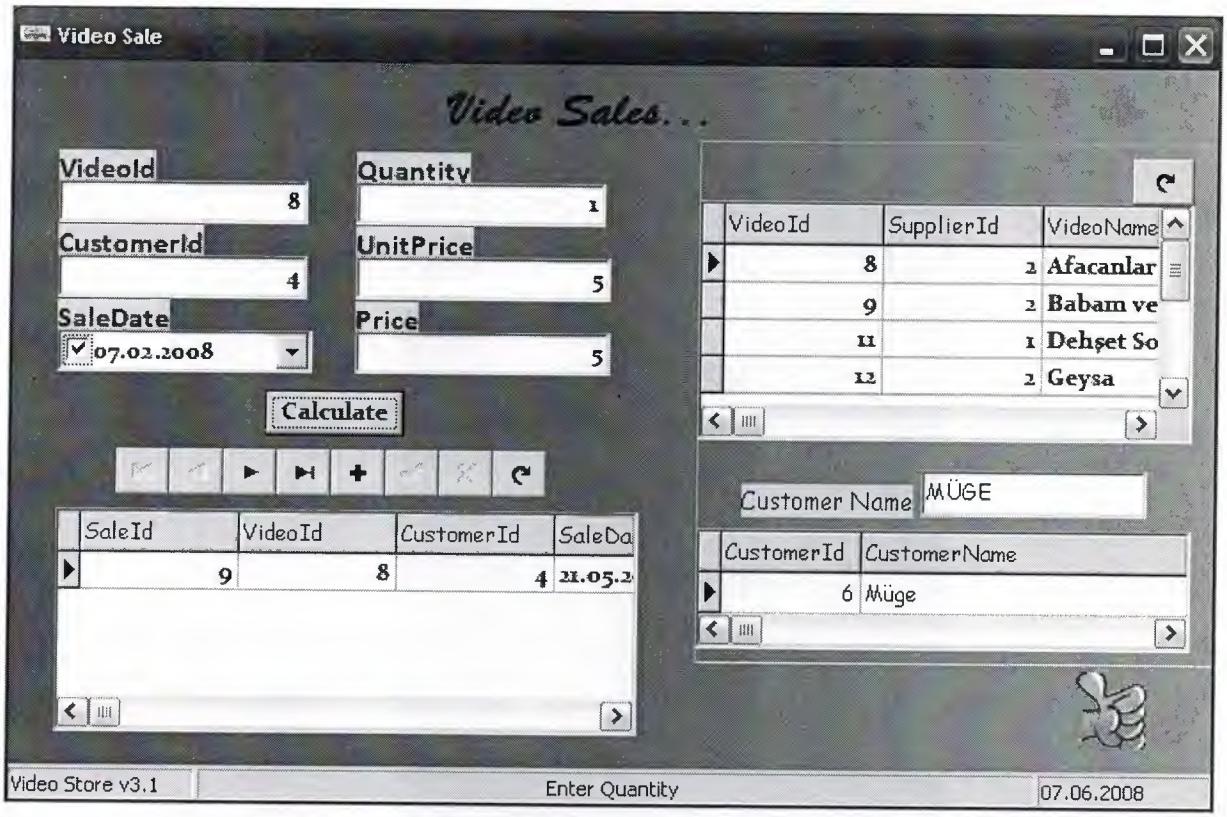



Figure 3.38

### 3.9 Rental

This form contains informations new rental and turned rentals search.

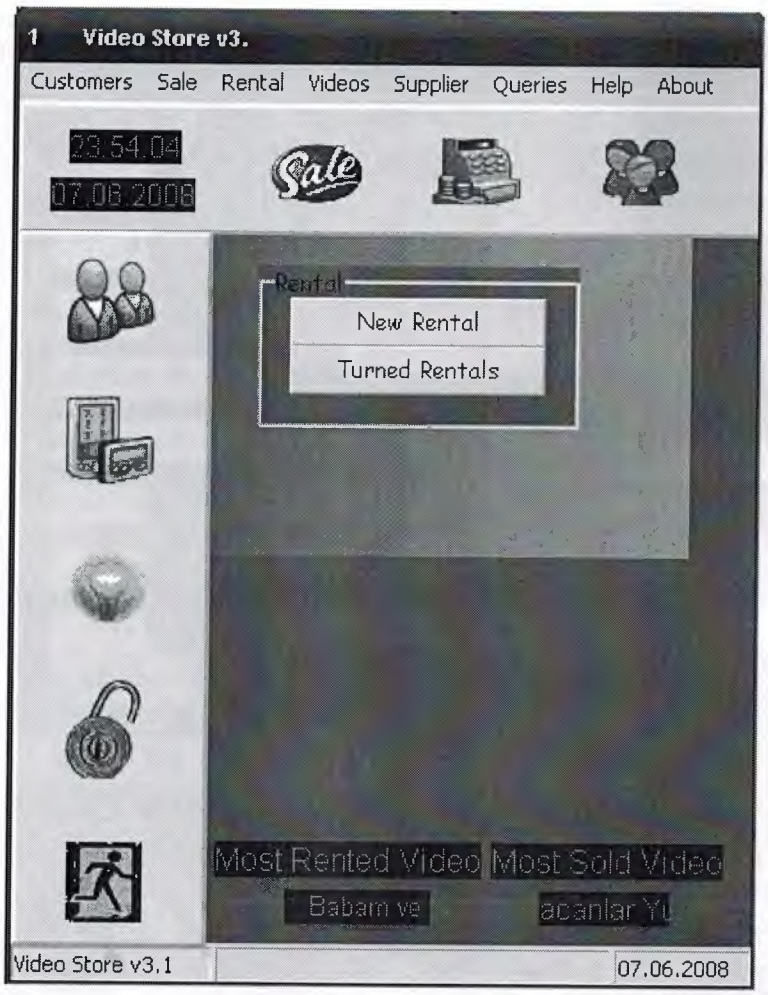



Figure 3.39

### 3.9.1 New Rental

In this form user can rent videos which is wanted .And see the video quantities and total prices

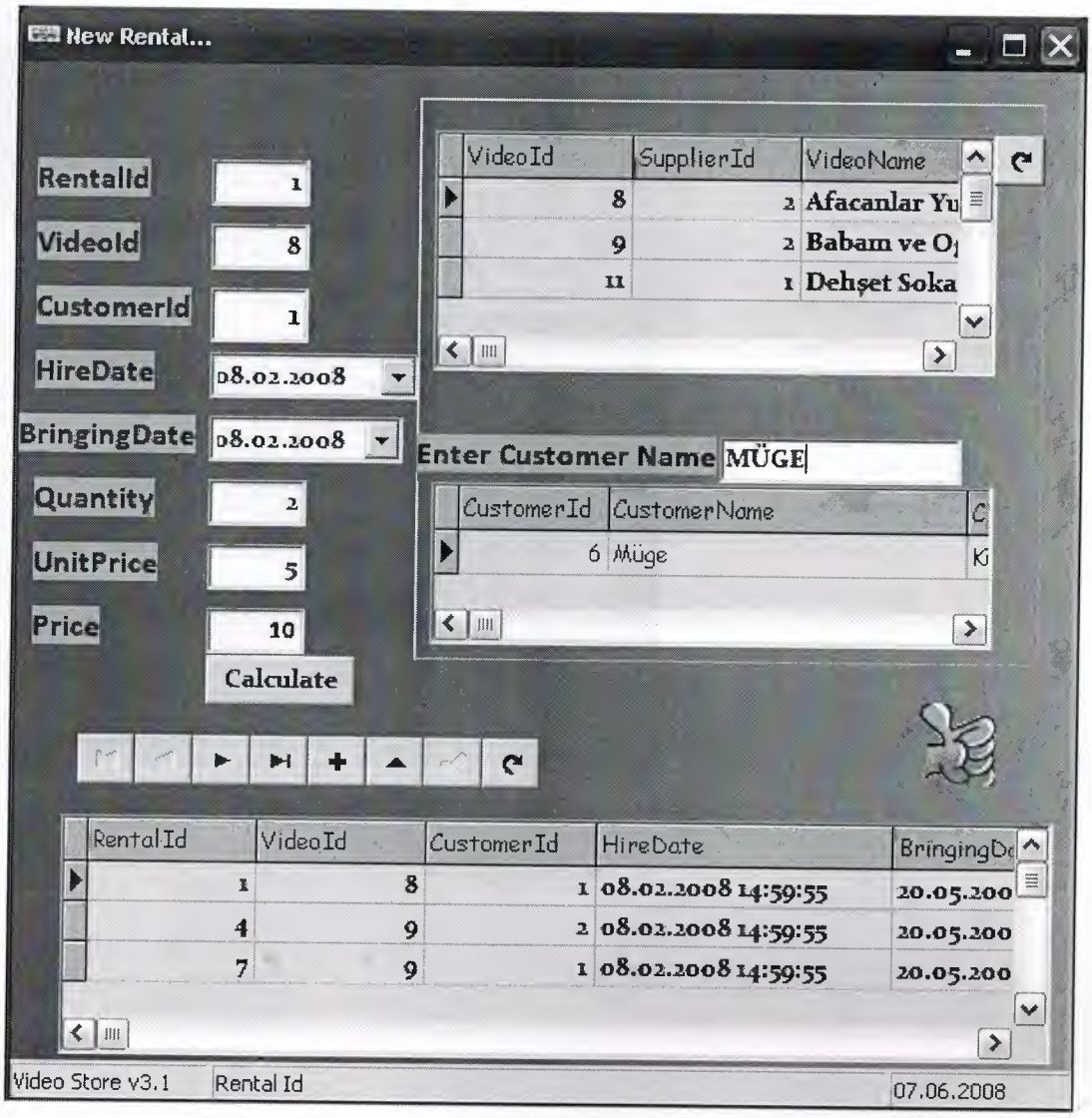



Figure 3.40

### 3.9.2 Turned Rental

In this page the videos which are rented before will be returned to the video store.

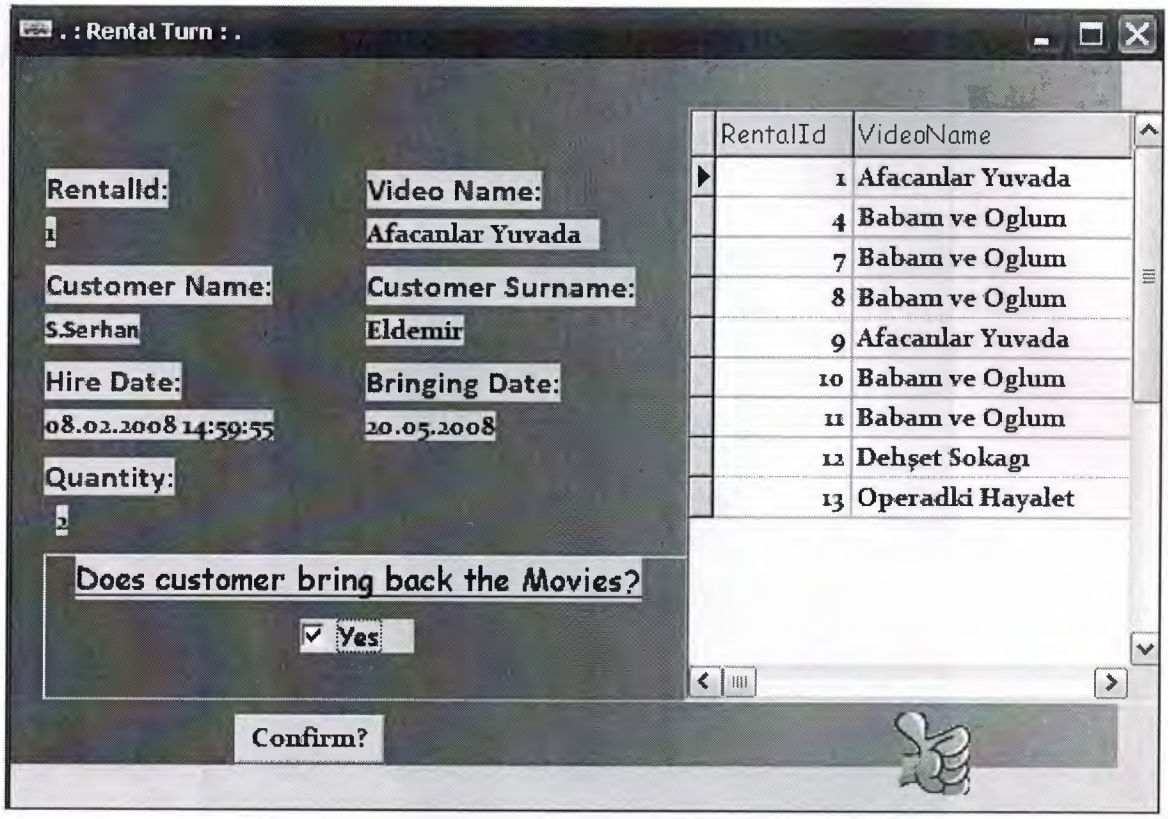



Figure 3.41

### 3.10 Queries

User can search beetwen two date, bringing date, most rented three videos, most sold three videos in date base.

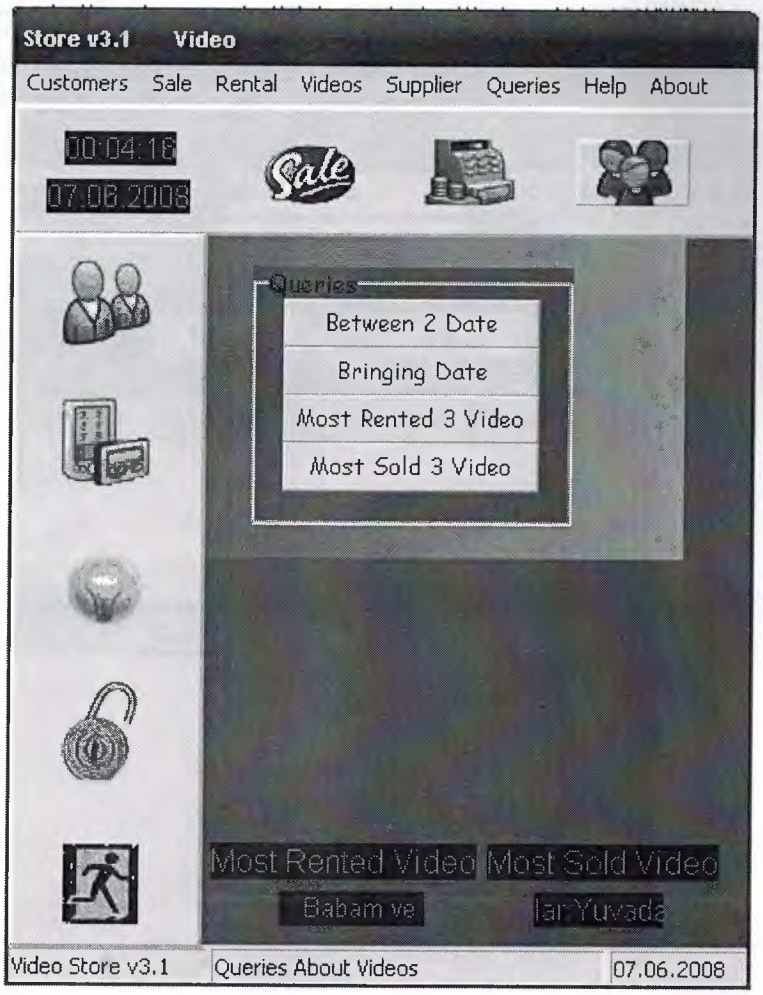



Figure 3.42

### 3.10.1 Between 2 Date

User can search the given video in between two date.


Figure 3.43

### 3.10.1.1 Searching for Bringing Date

If there is an empty space on the form program will show a message box.

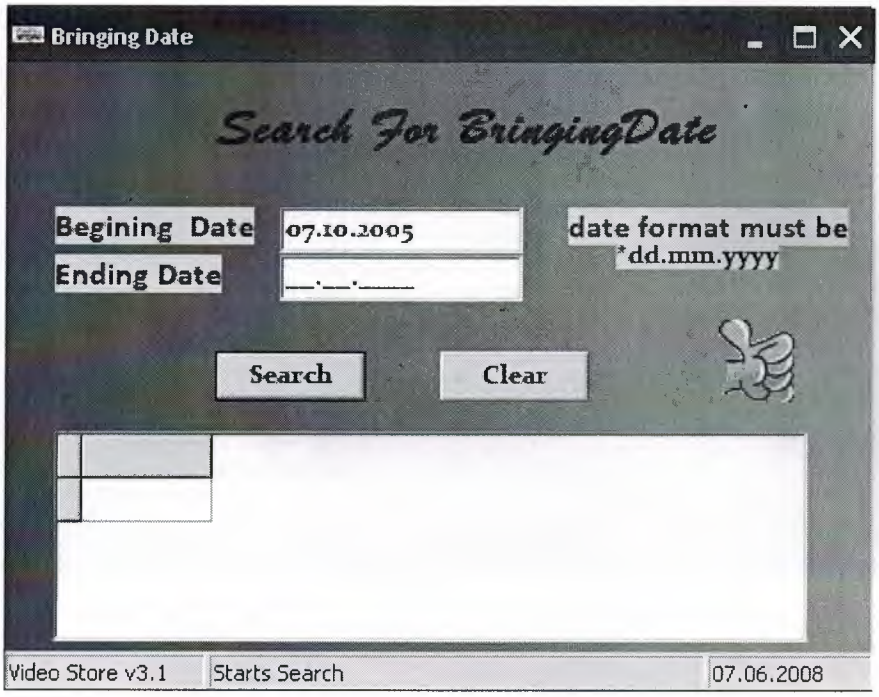



Figure 3.44

**Error:**

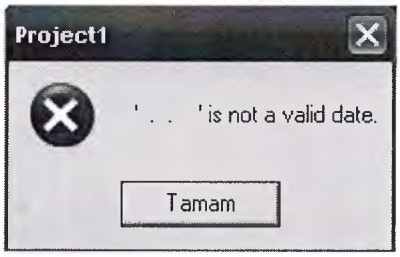



Figure 3.45

### 3.10.2 Bringing Date

It is the checking form which controls whether the video is bringed or not.

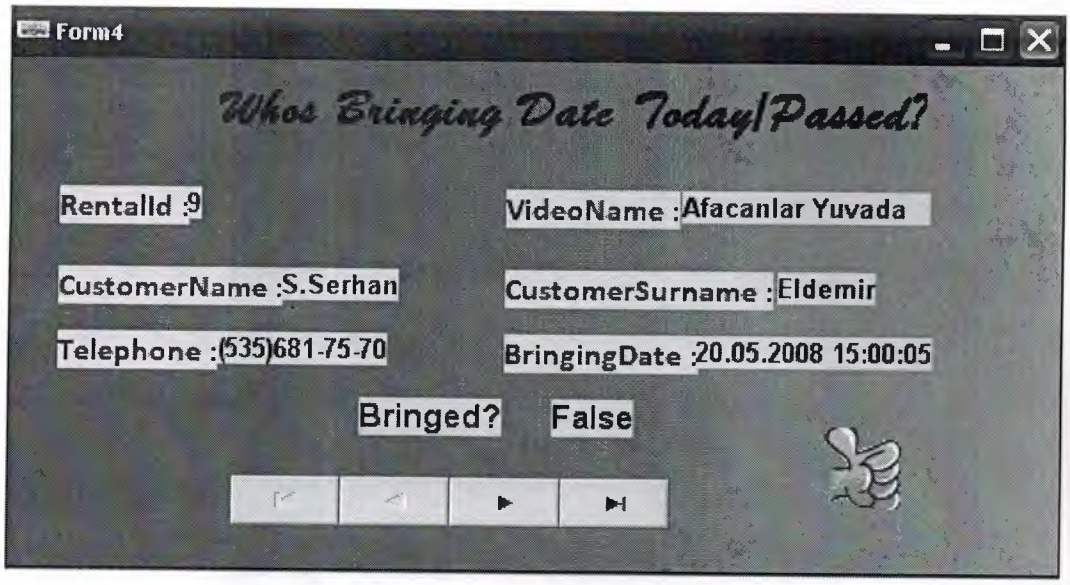



Figure 3.46

### 3.10.3 Most Rented 3 Video

This form contains most rented three videos.


Figure 3.47

### 3.10.4 Most Sold 3 Video

This form contains most sold three videos.

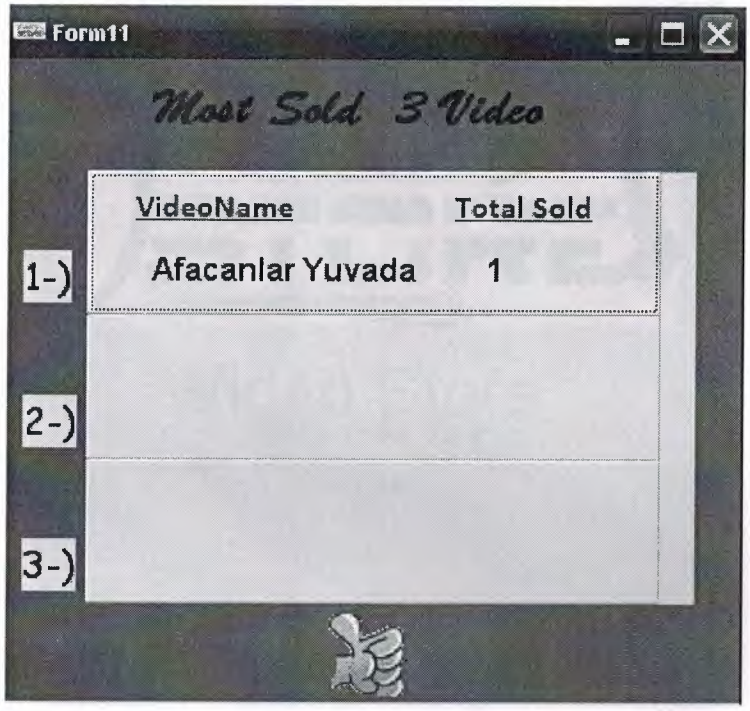



Figure 3.48

### 3.11 About

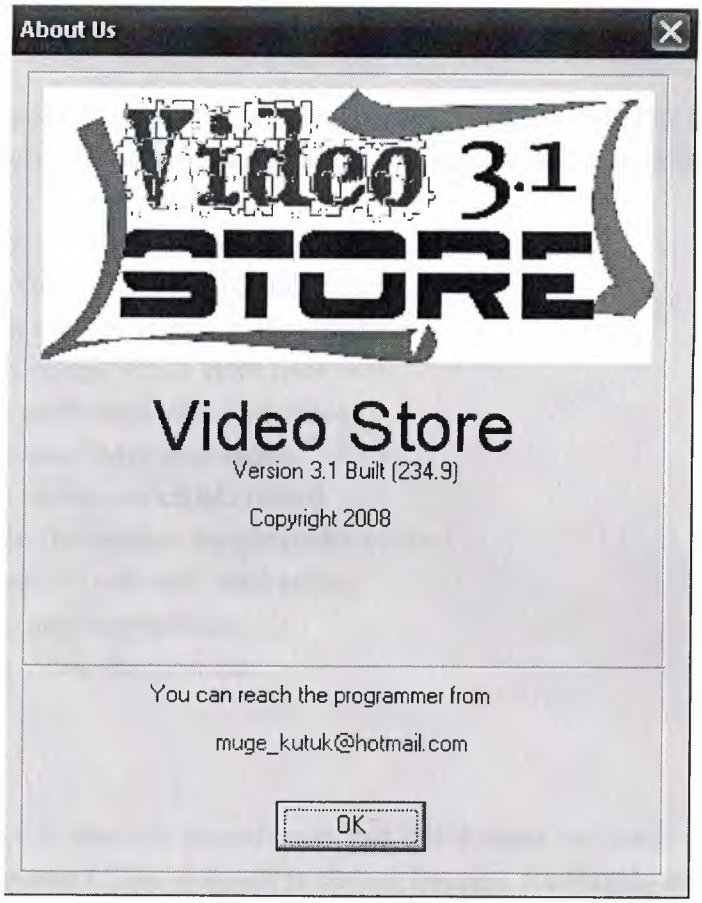



Figure 3.49

# CONCLUSION

Particular needs and demands of a video store management made this software program possible.The features of the program that the video store needs as fallows:

- To give the videos to the rent.
- To return videos which is rented before
- To register thenew videos at the database
- To see the videos which store has
- To list the customers who rent video
- To list the users who lend video
- To list the videos which are rented
- To keep the transaction details under control
- To calculate the sale and total price
- To list the customer actions
- To see the video discriptions

Borland Delphi 6.0 is used for the software that will answer the needs of aneed of a video store management.This program is chosen because it is highly secure and easy software to use in terms of both the user and the programmer when trying to fulfill the needs and demands of thestore.Access is the database that is used for this software as it is the most applicable database for Delphi.

This software holds the all necessary aspects to support the heavy work load of a video store and it is being tested in video store department and .There have been no problems faced with the program coping with the store's load so far.The program requires further development to handle remote access perhaps through internet to facilitate remote access by users and borrowers for better communications.

# References

**Referenca to Book:**

[1] Steve Teixeira,Delphi programming: Delphi 6 Developer's Guide,2000

**Referance to electronic book:**

[1] Delphi 7 Delphi QuickStart .PDF
[2] Delphi_2005_Reviewers_Guide.PDF
[3] Delphi_Database_Application_Developers_Book.PDF
[4] Office.microsoft.com/access
[5]tr.wikipedia.org/wiki/microsoft access

**Referance to Electronic source:**

[1] http://www.borland.com
[2]www.delphiturk.com
[3]www.delphidunyasi.net
[4]www.delphiturkiye.com/

# APPENDIX

## Login

```
unit Unit24;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, DB, ADODB, jpeg;

type
  TForm24 = class(TForm)
    Image1: TImage;
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit2: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Button1: TButton;
    Button2: TButton;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    Timer1: TTimer;
    Image2: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);

    procedure Timer1Timer(Sender: TObject);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form24: TForm24;
  tur: integer;
implementation
```

```
uses Unit2, Unit1;

{$R *.dfm}

procedure TForm24.Button1Click(Sender: TObject);
var
GirisPassword: String;

begin
    if tur=0 then
                Begin
                showmessage  ('The program Will Be Terminated');
                application.terminate;
                 End;
  If Self.edit1.Text = '' Then Begin
    Application.MessageBox('You must enter user name...','Error!!!');
    edit1.SetFocus;
    Exit;
  End;
  AdoQuery1.SQL.Text := 'SELECT * FROM Lock WHERE
UserName='''+Trim(edit1.Text)+'''';
  AdoQuery1.Open;
  GirisPassword := Trim(AdoQuery1.FieldByName('Pass').AsString);
  If AdoQuery1.RecordCount = 0 Then
    Begin
    Application.MessageBox('There arent any users like that...','Error!!!');
    tur:=tur-1;
    AdoQuery1.Close;

   if tur=0 then
                Begin
                showmessage  ('The program Will Be Terminated');
                application.terminate;
                 End; Exit;
   End
    Else
     Begin

      If GirisPassword = edit2.Text Then
       Begin
        form1.visible := True;
        form24.Visible := False;
       End
        Else
         Begin

          tur:=tur-1;
          Application.MessageBox('Wrong Password...','Error!!!');
             if tur=0 then
             Begin
```

```
                showmessage  ('The program Will Be Terminated');
                application.terminate;
                 End;
      end;
   End;
   AdoQuery1.Close;

end;

procedure TForm24.FormCreate(Sender: TObject);
begin
tur:=3;
form24.Caption :='Video Store v3.1 Security Check      ';
button1.Font.Size:=9;
button2.Font.Size:=9;
end;

procedure TForm24.Button2Click(Sender: TObject);
begin
close;
end;


procedure TForm24.Timer1Timer(Sender: TObject);
begin
Caption:=copy(caption,2,length(caption)-1)+caption[1];
end;

procedure TForm24.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
button1.Font.Size:=11;
end;

procedure TForm24.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
button2.Font.Size:=11;
end;

procedure TForm24.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
    button1.Font.Size:=9;
button2.Font.Size:=9;
image2.Stretch:= False;
end;

procedure TForm24.Image2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
image2.Stretch :=True;
end;

end.
```

## Main Menu

```
procedure TForm1.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=30;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);



LABEL1.Caption:=timetostr(time) ;
label1.Transparent :=false;
LABEL2.Caption:=datetostr(date) ;
label2.Transparent :=false;
form1.Caption :='   Video Store v3.1    ';
dbtext2.Transparent:=False;
dbtext1.Transparent:=False;
label3.Transparent :=false;
label4.Transparent :=false;


button9.caption:='All Customers';
button10.caption:='Customer Operations';
button12.caption:='Customer Search';
button15.caption:='All Videos';
button16.caption:='Video Operations';
button17.caption:='Delete Video';
button18.caption:='Video Search ';
button23.Caption :='All Suppliers';
button24.Caption :='Supplier Operations';
button25.Caption :='Delete Supplier';
button26.Caption :='Supplier Search';
button29.Caption :='Between 2 Date';
button30.Caption :='Bringing Date';
button11.Caption :='New Rental';
button13.Caption :='Turned Rentals';
Button31.Caption :='Most Rented 3 Video';
Button32.Caption :='Most Sold 3 Video';
```

```
groupbox1.caption:='Customers';
groupbox2.caption:='Products';
groupbox3.caption:='Suppliers';
groupbox4.caption:='Queries';
groupbox5.caption:='Rental';

groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;

speedbutton4.Hint:='Customer Menu';
speedbutton1.Hint:='Sales';
speedbutton3.Hint:='Rentals';
speedbutton2.Hint:='Videos Menu';
speedbutton5.Hint:='Queries Menu';
speedbutton6.Hint:='Supplier Menu';
speedbutton7.Hint:='Exit';
speedbutton8.Hint:='Security Process';

speedbutton1.showhint:=true;
speedbutton2.showhint:=true;
speedbutton3.showhint:=true;
speedbutton4.showhint:=true;
speedbutton5.showhint:=true;
speedbutton6.showhint:=true;
speedbutton7.showhint:=true;
speedbutton8.showhint:=true;


end;



procedure TForm1.Button9Click(Sender: TObject);
begin
form2.show;

form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
```

```
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button10Click(Sender: TObject);
begin
form3.show;

form2.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;


procedure TForm1.Button12Click(Sender: TObject);
begin
form5.show;
form2.hide;
form3.hide;
form4.hide;
```

```
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;


procedure TForm1.Button15Click(Sender: TObject);
begin
form7.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button16Click(Sender: TObject);
begin
form17.show;
```

```
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button17Click(Sender: TObject);
begin
form9.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button18Click(Sender: TObject);
```

```
begin
form10.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;
procedure TForm1.Button23Click(Sender: TObject);
begin
form14.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;
```

```
procedure TForm1.Button24Click(Sender: TObject);
begin
form15.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;


procedure TForm1.Button25Click(Sender: TObject);
begin
form18.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
```

```
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button26Click(Sender: TObject);
begin
form19.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;


procedure TForm1.Button29Click(Sender: TObject);
begin
form23.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
```

```
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
end;

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;


procedure TForm1.Button6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Help';
end;

procedure TForm1.Button7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='About Us';
end;

procedure TForm1.Button27MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rental Turns';
end;


procedure TForm1.Button9MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='All Customers';
end;

procedure TForm1.Button10MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Customer Operations';
end;
```

```
procedure TForm1.Button11MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='New Rental';
end;

procedure TForm1.Button12MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Customer Search';
end;

procedure TForm1.Button15MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='List of All Videos';
end;

procedure TForm1.Button16MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Video Operations';
end;

procedure TForm1.Button17MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Deleting Video';
end;

procedure TForm1.Button18MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Videos Search';
end;

procedure TForm1.Button23MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='List of All Suppliers';
end;

procedure TForm1.Button24MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Supplier Operations';
end;

procedure TForm1.Button25MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='Deleting Supplier';
end;

procedure TForm1.Button26MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Supplier Search';
end;

procedure TForm1.Button30Click(Sender: TObject);
begin
form4.show;
form2.hide;
form3.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button11Click(Sender: TObject);
begin
form21.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
```

```
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button13Click(Sender: TObject);
begin
form22.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form23.hide;
end;

procedure TForm1.CustomerOperations1Click(Sender: TObject);
begin
form3.show;

form2.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
```

```
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;end;

procedure TForm1.Button14Click(Sender: TObject);
begin
form6.Show;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
form20.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;
```

```
procedure TForm1.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Sales';
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=True;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.SpeedButton2MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Videos Menu';
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
```

```
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=True;

end;

procedure TForm1.SpeedButton3MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rentals';
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
```

```
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=True;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.SpeedButton4MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Customer Menu';
end;

procedure TForm1.Button31Click(Sender: TObject);
begin
form6.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
```

```
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=True;
groupbox5.Visible:=False;
end;

procedure TForm1.SpeedButton5MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Queries About Videos';
end;

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=True;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
```

```
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SpeedButton6MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Suppliers Menu';
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
label1.Caption :=timetostr(time);
end;

procedure TForm1.SpeedButton7Click(Sender: TObject);
var
c:word;
begin
c:=application.MessageBox('Do You Really Want to Exit from The
program?','Exit',mb_yesno+mb_iconquestion);
case c of
idyes :halt;
end;

end;

procedure TForm1.SpeedButton7MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit The Program';
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
Caption:=copy(caption,2,length(caption)-1)+caption[1];
end;


procedure TForm1.Timer3Timer(Sender: TObject);
begin
```

```
 dbtext1.Caption :=copy(dbtext1.Caption,2,length(dbtext1.Caption)-1
)+dbtext1.Caption[1];
 dbtext2.Caption :=copy(dbtext2.Caption,2,length(dbtext2.Caption)-1
)+dbtext2.Caption[1];

 end;

procedure TForm1.Button32Click(Sender: TObject);
begin
form11.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button13MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rental Turn';
end;

procedure TForm1.Button29MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search For Bringing Date';
end;

procedure TForm1.Button30MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Bringing date passed/today?';
end;
```

```
procedure TForm1.Button31MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Most Rented 3 Video';
end;

procedure TForm1.Button32MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Most Sold 3 Video';
end;


procedure TForm1.Help2Click(Sender: TObject);
begin
application.HelpFile :='help.hlp';
application.HelpJump('') ;
end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
form13.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;
```

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
form24.Show;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
halt;
end;

procedure TForm1.SpeedButton8MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Security Process';
end;

procedure TForm1.AboutUS1Click(Sender: TObject);
begin
form25.show;
end;

procedure TForm1.AllCustomers1Click(Sender: TObject);
begin
form2.show;

form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;
```

```
procedure TForm1.SearchingCustomers1Click(Sender: TObject);
begin
form5.show;
form2.hide;
form3.hide;
form4.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Sales1Click(Sender: TObject);
begin
form20.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form21.hide;
form22.hide;
```

```
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.Rentals1Click(Sender: TObject);
begin
form21.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.RentalTurn1Click(Sender: TObject);
begin
form22.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
```

```
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form23.hide;
end;

procedure TForm1.VideoOperations1Click(Sender: TObject);
begin
form7.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.DeletingVideo1Click(Sender: TObject);
begin
form8.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
```

```
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SearchingVideos1Click(Sender: TObject);
begin
form16.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.DeletingVideo2Click(Sender: TObject);
begin
form9.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
```

```
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SearchingVideos2Click(Sender: TObject);
begin
form10.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.ListOfSuppliers1Click(Sender: TObject);
begin
form14.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
```

```
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SupplierOperations1Click(Sender: TObject);
begin
form15.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.DeletingSupplier1Click(Sender: TObject);
begin
form18.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
```

```
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SearchingSupplier1Click(Sender: TObject);
begin
form19.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Between2Dates1Click(Sender: TObject);
begin
form23.Show;
form2.hide;
form3.hide;
form4.hide;
```

```
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
end;

procedure TForm1.BringingDatePassedAndToday1Click(Sender: TObject);
begin
form4.show;
form2.hide;
form3.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.N3MostRented1Click(Sender: TObject);
begin
form6.Show;
form2.hide;
```

```
form3.hide;
form4.hide;
form5.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.N3MostSold1Click(Sender: TObject);
begin
form11.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form9.hide;
form10.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form18.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

end.
```

## LIST OF ALL CUSTOMER

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, Buttons,
  jpeg;

type
  TForm2 = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    ADODataSet1: TADODataSet;
    ADODataSet1CustomerId: TAutoIncField;
    ADODataSet1CustomerName: TWideStringField;
    ADODataSet1CustomerSurname: TWideStringField;
    ADODataSet1Telephone: TWideStringField;
    ADODataSet1Address: TWideStringField;
    ADODataSet1City: TWideStringField;
    ADODataSet1Country: TWideStringField;
    DataSource1: TDataSource;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Label1Click(Sender: TObject);
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.FormCreate(Sender: TObject);
begin
form2.Caption :='All customers';
end;
```

```
procedure TForm2.Button1Click(Sender: TObject);
begin
form2.Hide;
end;

procedure TForm2.SpeedButton1Click(Sender: TObject);
begin
form2.Hide;
end;

procedure TForm2.Label1Click(Sender: TObject);
begin

end;

procedure TForm2.Image1Click(Sender: TObject);
begin

end;

end.
```

## CUSTOMER OPERATIONS

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, Mask, DB, ADODB,
  ComCtrls, Buttons, jpeg;

type
  TForm3 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    ADOTable1CustomerId: TAutoIncField;
    ADOTable1CustomerName: TWideStringField;
    ADOTable1CustomerSurname: TWideStringField;
    ADOTable1Telephone: TWideStringField;
    ADOTable1Address: TWideStringField;
    ADOTable1City: TWideStringField;
    ADOTable1Country: TWideStringField;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit4: TDBEdit;
```

```
    Label5: TLabel;
    DBEdit5: TDBEdit;
    Label6: TLabel;
    DBEdit6: TDBEdit;
    Label7: TLabel;
    DBEdit7: TDBEdit;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    StatusBar1: TStatusBar;
    Label1: TLabel;
    Label8: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1Click(Sender: TObject);
     private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm3.FormCreate(Sender: TObject);
begin
```

```
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=170;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);


form3.caption:='Customer Operations';
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
form3.Hide;
end;

procedure TForm3.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Name';
end;

procedure TForm3.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Surname';
end;

procedure TForm3.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Phone Number';
end;

procedure TForm3.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter City';
end;

procedure TForm3.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Address';
end;

procedure TForm3.DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='Enter Country';
end;

procedure TForm3.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;

procedure TForm3.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm3.SpeedButton1Click(Sender: TObject);
begin
form3.Hide;
end;

procedure TForm3.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;

procedure TForm3.Image1Click(Sender: TObject);
begin

end;

end.
```

## BRINGING DATE

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, Mask, DB, ADODB,
  ComCtrls, Buttons, jpeg;

type
  TForm3 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    ADOTable1CustomerId: TAutoIncField;
```

```
  ADOTable1CustomerName: TWideStringField;
  ADOTable1CustomerSurname: TWideStringField;
  ADOTable1Telephone: TWideStringField;
  ADOTable1Address: TWideStringField;
  ADOTable1City: TWideStringField;
  ADOTable1Country: TWideStringField;
  Label2: TLabel;
  DBEdit2: TDBEdit;
  Label3: TLabel;
  DBEdit3: TDBEdit;
  Label4: TLabel;
  DBEdit4: TDBEdit;
  Label5: TLabel;
  DBEdit5: TDBEdit;
  Label6: TLabel;
  DBEdit6: TDBEdit;
  Label7: TLabel;
  DBEdit7: TDBEdit;
  DBGrid1: TDBGrid;
  DBNavigator1: TDBNavigator;
  StatusBar1: TStatusBar;
  Label1: TLabel;
  Label8: TLabel;
  SpeedButton1: TSpeedButton;
  Image1: TImage;
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure Image1Click(Sender: TObject);
  private
  { Private declarations }
public
```

```
  { Public declarations }
  end;

var
  Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm3.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=170;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);


form3.caption:='Customer Operations';
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
form3.Hide;
end;

procedure TForm3.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Name';
end;

procedure TForm3.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Surname';
end;

procedure TForm3.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Phone Number';
end;

procedure TForm3.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='Enter City';
end;

procedure TForm3.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Address';
end;

procedure TForm3.DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Country';
end;

procedure TForm3.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;

procedure TForm3.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm3.SpeedButton1Click(Sender: TObject);
begin
form3.Hide;
end;

procedure TForm3.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;

procedure TForm3.Image1Click(Sender: TObject);
begin

end;

end.
```

## SEARCING CUSTOMER

```
unit Unit5;

interface
```

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, ComCtrls, Buttons, jpeg,
  ExtCtrls;

type
  TForm5 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    DBGrid1: TDBGrid;
    Button2: TButton;
    Label1: TLabel;
    StatusBar1: TStatusBar;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    Label2: TLabel;
    CheckBox4: TCheckBox;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure CheckBox3Click(Sender: TObject);
    procedure CheckBox4Click(Sender: TObject);
    procedure CheckBox1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure CheckBox3MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure CheckBox4MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
  private
    { Private declarations }
  public
```

```
  { Public declarations }
 end;

var
 Form5: TForm5;

implementation

{$R *.dfm}

procedure TForm5.Button1Click(Sender: TObject);
begin
if (checkbox1.Checked=false) and (checkbox2.Checked=false) and
(checkbox3.Checked=false)and (checkbox4.Checked=false)then
begin
showmessage('You have to Select one of Them');
end;

if checkbox1.checked then
begin
dbgrid1.visible:= true;
label1.Visible :=true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where customername=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;

end;
if checkbox2.Checked then
begin
label1.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where customersurname=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;
if checkbox3.checked then
begin
label1.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where city=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
```

```
adoquery1.Open;
end;
if checkbox4.checked then
begin
label1.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where country=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;


if edit1.text='' then
begin
showmessage('You Have to Enter Customers info!!');
dbgrid1.Visible :=false;
label1.Visible :=false;
end;
end;

procedure TForm5.FormCreate(Sender: TObject);
begin

statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=170;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);


edit1.Text:='';
label1.Caption:='Founded Results';
button1.Caption:='Search';
button2.Caption:='Clear';

form5.Caption:='Searching Customers ';

edit1.Text:='';
edit1.Visible :=False;
dbgrid1.Visible :=False;
label1.Visible :=false;

end;
```

```
procedure TForm5.Button2Click(Sender: TObject);
begin
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox3.Checked:=False;
checkbox4.Checked:=False;

checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Enabled :=True;
checkbox4.Enabled :=True;

label1.Visible :=False;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';
end;

procedure TForm5.CheckBox1Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox2.Enabled :=FAlse;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;

end;

procedure TForm5.CheckBox2Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;

end;

procedure TForm5.CheckBox3Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox4.Enabled :=False;

end;

procedure TForm5.CheckBox4Click(Sender: TObject);
begin
```

```
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox3.Enabled :=False;
end;

procedure TForm5.CheckBox1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by Name';
end;

procedure TForm5.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';

end;

procedure TForm5.CheckBox3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by City';

end;

procedure TForm5.CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by Surname';

end;

procedure TForm5.CheckBox4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by Country';

end;

procedure TForm5.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Searhing';

end;

procedure TForm5.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='Clears Form';

end;



procedure TForm5.SpeedButton1Click(Sender: TObject);
begin
form5.Hide;
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Checked:=False;
checkbox3.Enabled :=True;
checkbox4.Checked:=False;
checkbox4.Enabled :=True;
label1.Visible :=False;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';

end;

procedure TForm5.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;

end.
```

## LIST OF VIDEO

```
unit Unit7;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, DB, ADODB, Buttons,
  jpeg;

type
  TForm7 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    ADOTable1VideoId: TAutoIncField;
```

```
    ADOTable1SupplierId: TIntegerField;
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoDirector: TWideStringField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
    ADOTable1VideoInStock: TIntegerField;
    ADOTable1VideoPicture: TBlobField;
    Label1: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
     private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form7: TForm7;

implementation

{$R *.dfm}

procedure TForm7.Button1Click(Sender: TObject);
begin
form7.hide;
end;

procedure TForm7.FormCreate(Sender: TObject);
begin
form7.Caption :='List of Videos';
end;



procedure TForm7.SpeedButton1Click(Sender: TObject);
begin
form7.Hide;
end;

end.
```

## VIDEO OPERATION

```
unit Unit8;
```

```
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtDlgs, DBCtrls, StdCtrls, Mask, DB, ADODB, ExtCtrls, Grids,
  DBGrids, DBTables, ComCtrls, Buttons, jpeg;

type
  TForm8 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    ADOTable1VideoId: TAutoIncField;
    ADOTable1SupplierId: TIntegerField;
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
    ADOTable1VideoInStock: TIntegerField;
    ADOTable1VideoPicture: TBlobField;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBMemo1: TDBMemo;
    Label5: TLabel;
    DBEdit4: TDBEdit;
    Label6: TLabel;
    DBEdit5: TDBEdit;
    Label7: TLabel;
    DBEdit6: TDBEdit;
    Label8: TLabel;
    DBEdit7: TDBEdit;
    Label9: TLabel;
    DBImage1: TDBImage;
    OpenPictureDialog1: TOpenPictureDialog;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Edit1: TEdit;
    Button2: TButton;
    Label10: TLabel;
    ADOTable2: TADOTable;
    DataSource2: TDataSource;
    DBGrid2: TDBGrid;
    Bevel1: TBevel;
    DBNavigator2: TDBNavigator;
```

```
ADOTable1VideoDirector: TWideStringField;
Label11: TLabel;
DBEdit8: TDBEdit;
Label12: TLabel;
Label13: TLabel;
StatusBar1: TStatusBar;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure DBImage1DblClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBImage1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBMemo1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit8MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBGrid1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure Image1Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form8: TForm8;

implementation

uses Unit7;

{$R *.dfm}

procedure TForm8.DBImage1DblClick(Sender: TObject);
begin
 OpenPictureDialog1.Title:='Pick up an Picture';

 if (OpenPictureDialog1.Execute)then
 begin
 adotable1.Edit ;
 dbimage1.Picture.LoadFromFile(OpenPictureDialog1.FileName);
 adotable1.Post;
 end;
end;

procedure TForm8.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=350;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=90;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);


speedbutton1.Caption:='' ;
button2.Caption:='Search';
Label10.Caption:='Supplier Name';
edit1.Text :='';
dbgrid2.Visible :=False;
form8.Caption :='Video Operations';
end;

procedure TForm8.Button1Click(Sender: TObject);
begin
```

```
form8.Hide;
form7.hide;
end;
procedure TForm8.Button2Click(Sender: TObject);
var
src:boolean;
begin
if edit1.text='' then
begin
showmessage('Please enter a record');
edit1.setfocus
end
else
      begin
  src:=ADOtable2.Locate('SupplierName',edit1.Text,[lopartialkey]);
  dbgrid2.canvas.font.Color:=clred;
  dbgrid2.Visible :=true;
  end;

if not src=true then
 begin
  showmessage('Record Not Found');
  dbgrid2.Visible :=false;
  edit1.setfocus;
  dbgrid2.Visible :=false;
end;

end;



procedure TForm8.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm8.DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Id';
end;

procedure TForm8.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Name';
end;

procedure TForm8.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Type';
end;

procedure TForm8.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Year';
end;

procedure TForm8.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Price';
end;

procedure TForm8.DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Amount of Video';
end;

procedure TForm8.DBImage1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Add a Picture to Video';
end;

procedure TForm8.DBMemo1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Description For Video';
end;

procedure TForm8.DBEdit8MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Director';
end;

procedure TForm8.DBGrid1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='List Of Videos In Stock';
end;

procedure TForm8.Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
```

```
statusbar1.Panels.Items[1].Text:='Enter Supplier Name For Search';
end;

procedure TForm8.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Search';
end;

procedure TForm8.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits the Page';
end;

procedure TForm8.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits the Page';
end;

procedure TForm8.SpeedButton1Click(Sender: TObject);
begin
form8.Hide;
end;

procedure TForm8.Image1Click(Sender: TObject);
begin

end;

end.
```

## DELETING VIDEO

```
unit Unit9;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, Grids, DBGrids, StdCtrls, Mask, DBCtrls, Buttons,
  jpeg, ExtCtrls;

type
  TForm9 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    ADOTable1VideoId: TAutoIncField;
    ADOTable1SupplierId: TIntegerField;
```

```
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoDirector: TWideStringField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
    ADOTable1VideoInStock: TIntegerField;
    ADOTable1VideoPicture: TBlobField;
    DBEdit1: TDBEdit;
    DBGrid1: TDBGrid;
    Button1: TButton;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    Label1: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form9: TForm9;

implementation

{$R *.dfm}

procedure TForm9.FormCreate(Sender: TObject);
begin
dbedit1.Visible :=false;
button1.Caption :='Delete';
form9.Caption:='Deleting Videos';

end;

procedure TForm9.Button1Click(Sender: TObject);
var
x:integer;
begin
x:=strtoint(dbedit1.Text);

if x>0 then
showmessage('!!You Cant Delete Stock is not Empty!!')
else
adotable1.Delete;
adotable1.Next;
```

```
adotable1.Refresh;
end;


procedure TForm9.SpeedButton1Click(Sender: TObject);
begin
form9.Hide;
end;

procedure TForm9.Image1Click(Sender: TObject);
begin

end;

end.
```

## SEARCHING VIDEO

```
unit Unit10;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB, ComCtrls, Buttons, jpeg,
  ExtCtrls;

type
  TForm10 = class(TForm)
    DBGrid1: TDBGrid;
    Edit1: TEdit;
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
    Label2: TLabel;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    StatusBar1: TStatusBar;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
```

```
    procedure CheckBox3Click(Sender: TObject);
    procedure CheckBox4Click(Sender: TObject);
    procedure CheckBox1MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure CheckBox1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure CheckBox3MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure CheckBox4MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1Click(Sender: TObject);


  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form10: TForm10;

implementation

{$R *.dfm}

procedure TForm10.Button1Click(Sender: TObject);
begin
if (checkbox1.Checked=false) and (checkbox2.Checked=false) and
(checkbox3.Checked=false)and (checkbox4.Checked=false)then
begin
showmessage('You have to Select one of Them');
end;

if checkbox1.checked then
begin
dbgrid1.visible:= true;
label2.Visible :=true;
adoquery1.Close;
```

```
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
adoquery1.SQL.Add('where videoname=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;

end;
if checkbox2.Checked then
begin
label2.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
adoquery1.SQL.Add('where VideoDirector=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;
if checkbox3.checked then
begin
label2.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
adoquery1.SQL.Add('where VideoType=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;
if checkbox4.checked then
begin
label2.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
adoquery1.SQL.Add('where Videoyear=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=strtoint(edit1.text);
adoquery1.Open;
end;
end;
procedure TForm10.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=170;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
```

```
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);


button1.caption:='Search';
button2.caption:='Clear';
speedbutton1.caption:='';
Label1.caption:= 'Searching Videos...';
label2.Visible :=false;

dbgrid1.Visible :=False;

edit1.Text:='';
form10.Caption :='Searching Videos';
edit1.Text:='';
edit1.Visible :=False;

end;

procedure TForm10.Button2Click(Sender: TObject);
begin
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox3.Checked:=False;
checkbox4.Checked:=False;

checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Enabled :=True;
checkbox4.Enabled :=True;

label2.Visible :=False;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';

end;

procedure TForm10.CheckBox1Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox2.Enabled :=FAlse;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;

end;

procedure TForm10.CheckBox2Click(Sender: TObject);
begin
edit1.Visible :=True;
```

```
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;

end;

procedure TForm10.CheckBox3Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox4.Enabled :=False;
end;

procedure TForm10.CheckBox4Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox3.Enabled :=False;
end;

procedure TForm10.CheckBox1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Name';

end;

procedure TForm10.CheckBox1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Name';
end;

procedure TForm10.CheckBox2MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Director';
end;

procedure TForm10.CheckBox3MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Type';
end;
```

```
procedure TForm10.CheckBox4MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Year';
end;

procedure TForm10.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Video Search ';
end;

procedure TForm10.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Clears Form';
end;

procedure TForm10.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm10.SpeedButton1Click(Sender: TObject);
begin
form10.Hide;

checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox3.Checked:=False;
checkbox4.Checked:=False;

checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Enabled :=True;
checkbox4.Enabled :=True;
label2.Visible :=False;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';
end;

procedure TForm10.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits Video Searching';
end;

procedure TForm10.Image1Click(Sender: TObject);
```

```
begin

end;

end.
```

## LIST OF SUPPLIER

```
unit Unit14;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, DB, ADODB, Buttons,
  jpeg;

type
  TForm14 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form14: TForm14;

implementation

{$R *.dfm}

procedure TForm14.FormCreate(Sender: TObject);
begin
speedbutton1.Caption:='';
form14.Caption:='List of All Suppliers';
end;

procedure TForm14.SpeedButton1Click(Sender: TObject);
begin
form14.Hide;
```

```
end;

end.
```

## SUPPLIER OPERATIONS

```
unit Unit15;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, Mask, DB, ADODB,
  ComCtrls, Buttons, jpeg;

type
  TForm15 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    ADOTable1SupplierId: TAutoIncField;
    ADOTable1SupplierName: TWideStringField;
    ADOTable1Telephone: TWideStringField;
    ADOTable1Address: TWideStringField;
    ADOTable1City: TWideStringField;
    ADOTable1Country: TWideStringField;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit4: TDBEdit;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    Label6: TLabel;
    DBEdit6: TDBEdit;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label7: TLabel;
    StatusBar1: TStatusBar;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
    Y: Integer);
  procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure Image1Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form15: TForm15;

implementation

{$R *.dfm}

procedure TForm15.Button1Click(Sender: TObject);
begin
form15.Hide;
end;

procedure TForm15.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=150;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);
form15.caption:='Supplier Operations';
end;

procedure TForm15.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';

end;

procedure TForm15.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Name';

end;

procedure TForm15.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Phone';
end;

procedure TForm15.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Address';
end;

procedure TForm15.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier City';
end;

procedure TForm15.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Country';
end;

procedure TForm15.SpeedButton1Click(Sender: TObject);
begin
form15.Hide;
end;

procedure TForm15.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;

procedure TForm15.Image1Click(Sender: TObject);
begin

end;

end.
```

## DELETING SUPPLIER

```
unit Unit18;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, Grids, DBGrids, DB, ADODB, Buttons,
  jpeg, ExtCtrls;

type
  TForm18 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBEdit1: TDBEdit;
    Button1: TButton;
    Label1: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form18: TForm18;

implementation

{$R *.dfm}

procedure TForm18.Button1Click(Sender: TObject);
var
x:integer;
begin
x:=strtoint(dbedit1.Text);

if x>0 then
showmessage('!!You Cant Delete This Supplier, it still Supplies!!')
else
adotable1.Delete;
adotable1.Next;
```

```
end;

procedure TForm18.FormCreate(Sender: TObject);
begin
button1.Caption:='Delete';
end;

procedure TForm18.Button2Click(Sender: TObject);
begin
form18.Hide;
end;

procedure TForm18.SpeedButton1Click(Sender: TObject);
begin
form18.Hide;
end;

procedure TForm18.Image1Click(Sender: TObject);
begin

end;

end.
```

## SEARCH SUPPLIERS

```
unit Unit18;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, Grids, DBGrids, DB, ADODB, Buttons,
  jpeg, ExtCtrls;

type
  TForm18 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBEdit1: TDBEdit;
    Button1: TButton;
    Label1: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Image1Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form18: TForm18;

implementation

{$R *.dfm}

procedure TForm18.Button1Click(Sender: TObject);
var
x:integer;
begin
x:=strtoint(dbedit1.Text);

if x>0 then
showmessage('!!You Cant Delete This Supplier, it still Supplies!!')
else
adotable1.Delete;
adotable1.Next;


end;

procedure TForm18.FormCreate(Sender: TObject);
begin
button1.Caption:='Delete';
end;

procedure TForm18.Button2Click(Sender: TObject);
begin
form18.Hide;
end;

procedure TForm18.SpeedButton1Click(Sender: TObject);
begin
form18.Hide;
end;

procedure TForm18.Image1Click(Sender: TObject);
begin

end;

end.
```

**MAKE YOUR CHOICE**

```
unit Unit17;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, jpeg;

type
  TForm17 = class(TForm)
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Image1: TImage;
    Label1: TLabel;
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form17: TForm17;

implementation

uses Unit8, Unit16;

{$R *.dfm}

procedure TForm17.RadioButton1Click(Sender: TObject);
begin
form8.show;
form17.hide;

end;

procedure TForm17.RadioButton2Click(Sender: TObject);
begin
form16.show;
form17.Hide;
end;

end.
```

**PURCHASE EXITING VIDEO**

```
unit Unit16;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, DB, Grids, DBGrids, ADODB, ExtCtrls,
  ComCtrls, Buttons, jpeg;

type
  TForm16 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1VideoId: TAutoIncField;
    ADOTable1SupplierId: TIntegerField;
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoDirector: TWideStringField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
    ADOTable1VideoInStock: TIntegerField;
    ADOTable1VideoPicture: TBlobField;
    DataSource1: TDataSource;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    DBText1: TDBText;
    DBText2: TDBText;
    DBText3: TDBText;
    DBText5: TDBText;
    DBGrid1: TDBGrid;
    Label5: TLabel;
    DBText4: TDBText;
    Label7: TLabel;
    DBText6: TDBText;
    DBText7: TDBText;
    DBText8: TDBText;
    Label8: TLabel;
    Label11: TLabel;
    Button1: TButton;
    Edit1: TEdit;
    Label12: TLabel;
    Edit2: TEdit;
    Button2: TButton;
    Bevel1: TBevel;
    StatusBar1: TStatusBar;
```

```
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form16: TForm16;

implementation

{$R *.dfm}

procedure TForm16.Button1Click(Sender: TObject);
var
x,y,z: integer;
begin
if edit1.Text='' then
showmessage ('Enter Amount')

else
begin
x:= strtoint(edit1.Text);
y:= strtoint(dbtext8.caption);
z:=x+y;
label12.Caption :=inttostr(z);
adotable1.Edit;
adotable1.FieldValues['videoinstock']:=label12.caption;
adotable1.Refresh ;
adotable1.next;
```

```
end;
end;


procedure TForm16.Button2Click(Sender: TObject);
begin
if edit2.Text='' then
showmessage ('Enter New Price')

else
begin

adotable1.Edit;
adotable1.FieldValues['unitprice']:=edit2.text;
adotable1.Refresh ;
adotable1.next;

end;

end;

procedure TForm16.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=350;
statusbar1.panels.items[1].Alignment:=tacenter;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

speedbutton1.caption:='';
form16.Caption:='Purchasing Existing Video';
end;

procedure TForm16.Button3Click(Sender: TObject);
begin
form16.Hide;
end;

procedure TForm16.Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter New Price';
end;

procedure TForm16.Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter New Amount';
end;

procedure TForm16.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
 statusbar1.Panels.Items[1].Text:='Apply Changes on New Price';
end;

procedure TForm16.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Apply Changes on New Amount';
end;

procedure TForm16.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm16.SpeedButton1Click(Sender: TObject);
begin
form16.Hide;
end;

procedure TForm16.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;

procedure TForm16.Image1Click(Sender: TObject);
begin

end;

end.
```

## PASSWORD PROCESS

```
unit Unit16;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, DB, Grids, DBGrids, ADODB, ExtCtrls,
```

```
  ComCtrls, Buttons, jpeg;

type
  TForm16 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1VideoId: TAutoIncField;
    ADOTable1SupplierId: TIntegerField;
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoDirector: TWideStringField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
    ADOTable1VideoInStock: TIntegerField;
    ADOTable1VideoPicture: TBlobField;
    DataSource1: TDataSource;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    DBText1: TDBText;
    DBText2: TDBText;
    DBText3: TDBText;
    DBText5: TDBText;
    DBGrid1: TDBGrid;
    Label5: TLabel;
    DBText4: TDBText;
    Label7: TLabel;
    DBText6: TDBText;
    DBText7: TDBText;
    DBText8: TDBText;
    Label8: TLabel;
    Label11: TLabel;
    Button1: TButton;
    Edit1: TEdit;
    Label12: TLabel;
    Edit2: TEdit;
    Button2: TButton;
    Bevel1: TBevel;
    StatusBar1: TStatusBar;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
      Y: Integer);
    procedure Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form16: TForm16;

implementation

{$R *.dfm}

procedure TForm16.Button1Click(Sender: TObject);
var
x,y,z: integer;
begin
if edit1.Text='' then
showmessage ('Enter Amount')

else
begin
x:= strtoint(edit1.Text);
y:= strtoint(dbtext8.caption);
z:=x+y;
label12.Caption :=inttostr(z);
adotable1.Edit;
adotable1.FieldValues['videoinstock']:=label12.caption;
adotable1.Refresh ;
adotable1.next;

end;
end;


procedure TForm16.Button2Click(Sender: TObject);
begin
```

```
if edit2.Text='' then
showmessage ('Enter New Price')

else
begin

adotable1.Edit;
adotable1.FieldValues['unitprice']:=edit2.text;
adotable1.Refresh ;
adotable1.next;

end;

end;

procedure TForm16.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=350;
statusbar1.panels.items[1].Alignment:=tacenter;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

speedbutton1.caption:='';
form16.Caption:='Purchasing Existing Video';
end;

procedure TForm16.Button3Click(Sender: TObject);
begin
form16.Hide;
end;

procedure TForm16.Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter New Price';
end;

procedure TForm16.Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter New Amount';
end;

procedure TForm16.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
 begin
  statusbar1.Panels.Items[1].Text:='Apply Changes on New Price';
 end;

 procedure TForm16.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
   Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='Apply Changes on New Amount';
 end;

 procedure TForm16.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
   Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='';
 end;

 procedure TForm16.SpeedButton1Click(Sender: TObject);
 begin
 form16.Hide;
 end;

 procedure TForm16.SpeedButton1MouseMove(Sender: TObject;
   Shift: TShiftState; X, Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='Quit';
 end;

 procedure TForm16.Image1Click(Sender: TObject);
 begin

 end;

 end.
```

## PASSWORD REQUIRES

```
unit Unit13;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls;

type
  TForm13 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
```

```
  Image1: TImage;
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure Image1Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form13: TForm13;
 tur: integer;
implementation

uses Unit12;

{$R *.dfm}

procedure TForm13.FormCreate(Sender: TObject);
begin
form13.Caption:='Security!!';
button1.Caption:='Confirm?';
button2.Caption:='Cancel';
end;

procedure TForm13.Button1Click(Sender: TObject);
 begin
if edit1.Text ='serhan' then
begin
edit1.Text:='';
Application.MessageBox('Access Granted...', '  Congratulation ');
form12.Show;
form13.Hide;

end
else
begin
Application.MessageBox('Wrong Password !'#13'Please Enter Again!','Try Again' );
tur:=tur+1;
edit1.Text:='';
edit1.SetFocus;
        if tur=3 then
        begin
        edit1.Text:='';
        Application.MessageBox('Sorry You cant Enter this time!','Ooppss!!!');
         form13.Hide;
        end;
```

```
end;
end;

procedure TForm13.Button2Click(Sender: TObject);
begin
form13.Hide;
end;

procedure TForm13.Image1Click(Sender: TObject);
begin

end;

end.
```

## VIDEO SALE

```
unit Unit20;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ADODB, Grids, DBGrids, StdCtrls, Mask, DBCtrls, DB, ComCtrls,
  ExtCtrls, Buttons, jpeg;

type
  TForm20 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1SaleId: TAutoIncField;
    ADOTable1VideoId: TIntegerField;
    ADOTable1CustomerId: TIntegerField;
    ADOTable1SaleDate: TDateTimeField;
    ADOTable1Quantity: TIntegerField;
    ADOTable1Price: TBCDField;
    DataSource1: TDataSource;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    Label6: TLabel;
    DBEdit6: TDBEdit;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    ADOQuery2: TADOQuery;
    DataSource3: TDataSource;
    Edit2: TEdit;
```

```
Label7: TLabel;
DBGrid3: TDBGrid;
DateTimePicker1: TDateTimePicker;
ADOTable2: TADOTable;
ADOTable2VideoId: TAutoIncField;
ADOTable2SupplierId: TIntegerField;
ADOTable2VideoName: TWideStringField;
ADOTable2VideoDesc: TMemoField;
ADOTable2VideoDirector: TWideStringField;
ADOTable2VideoType: TWideStringField;
ADOTable2VideoYear: TIntegerField;
ADOTable2UnitPrice: TBCDField;
ADOTable2VideoInStock: TIntegerField;
ADOTable2VideoPicture: TBlobField;
Label1: TLabel;
DBEdit1: TDBEdit;
DataSource2: TDataSource;
DBEdit4: TDBEdit;
DBNavigator1: TDBNavigator;
Button1: TButton;
Label9: TLabel;
DBNavigator2: TDBNavigator;
Label8: TLabel;
Bevel1: TBevel;
StatusBar1: TStatusBar;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
  procedure Image1Click(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 Form20: TForm20;

implementation

{$R *.dfm}

procedure TForm20.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=450;
statusbar1.panels.items[1].Alignment:=tacenter;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=90;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);



form20.caption:= 'Video Sale';
button1.Caption :='Calculate';

end;



procedure TForm20.Button2Click(Sender: TObject);
begin
form20.Hide;
end;

procedure TForm20.Edit2Change(Sender: TObject);
begin
adoquery2.Close;
adoquery2.SQL.Clear ;
adoquery2.SQL.Add('select * from customers');
adoquery2.SQL.Add('where customername=:edit2.text');
adoquery2.Parameters.ParamByName('edit2.text').Value:=(edit2.text);
adoquery2.Open;
```

```
dbgrid3.visible:= true;

end;

procedure TForm20.Button1Click(Sender: TObject);
 var
uni,x,y,z,a: integer ;

begin

uni:=strtoint(dbedit4.Text);
x:= strtoint(dbedit5.Text);
y:= strtoint(dbedit1.Text);

if  x>uni then
begin
showmessage('Less Unit in Stock');
exit;
dbedit5.setfocus
end
else

z:= x*y;
a:=uni-x;
dbedit6.text:=inttostr(z) ;
label9.Caption:=inttostr(a);
datasource1.DataSet.FieldValues['saledate']:=datetimepicker1.Date;

datasource2.Edit;
datasource2.DataSet.FieldValues['videoinstock']:=label9.Caption;


end;

procedure TForm20.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm20.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video ID';
end;

procedure TForm20.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Customer ID';
```

```
end;

procedure TForm20.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Quantity';
end;

procedure TForm20.DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Unit Price';
end;

procedure TForm20.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Total Price';
end;

procedure TForm20.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Calculates The TOTAL Price';
end;

procedure TForm20.Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Customer Name';
end;

procedure TForm20.SpeedButton1Click(Sender: TObject);
begin
form20.Hide;
end;

procedure TForm20.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Close the Page';
end;

procedure TForm20.Image1Click(Sender: TObject);
begin

end;

end.
```

## NEW RENTAL

```
unit Unit21;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, DB, StdCtrls, Mask, DBCtrls, ADODB, Grids, DBGrids,
  ExtCtrls, Buttons, jpeg;

type
  TForm21 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1RentalId: TAutoIncField;
    ADOTable1VideoId: TIntegerField;
    ADOTable1CustomerId: TIntegerField;
    ADOTable1HireDate: TDateTimeField;
    ADOTable1BringingDate: TDateTimeField;
    ADOTable1Quantity: TIntegerField;
    ADOTable1Price: TBCDField;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    DataSource1: TDataSource;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    DBEdit6: TDBEdit;
    Label7: TLabel;
    DBEdit7: TDBEdit;
    DateTimePicker1: TDateTimePicker;
    DateTimePicker2: TDateTimePicker;
    Button1: TButton;
    ADOTable2: TADOTable;
    ADOTable2VideoId: TAutoIncField;
    ADOTable2SupplierId: TIntegerField;
    ADOTable2VideoName: TWideStringField;
    ADOTable2VideoDesc: TMemoField;
    ADOTable2VideoDirector: TWideStringField;
    ADOTable2VideoType: TWideStringField;
    ADOTable2VideoYear: TIntegerField;
    ADOTable2UnitPrice: TBCDField;
    ADOTable2VideoInStock: TIntegerField;
    ADOTable2VideoPicture: TBlobField;
    Label8: TLabel;
    DBEdit4: TDBEdit;
```

```
    DataSource2: TDataSource;
    DBEdit5: TDBEdit;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    Edit1: TEdit;
    ADOQuery1: TADOQuery;
    DataSource3: TDataSource;
    Label10: TLabel;
    DBGrid3: TDBGrid;
    DBNavigator1: TDBNavigator;
    Bevel1: TBevel;
    Label11: TLabel;
    DBNavigator2: TDBNavigator;
    StatusBar1: TStatusBar;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form21: TForm21;

implementation
```

```
{$R *.dfm}

procedure TForm21.Button1Click(Sender: TObject);
var
uni,x,y,z,a: integer ;

begin

uni:=strtoint(dbedit5.Text);
x:= strtoint(dbedit6.Text);
y:= strtoint(dbedit4.Text);

if  x>uni then
begin
showmessage('Less Unit in Stock');
exit;
dbedit5.setfocus
end
else

z:= x*y div 2;
a:=uni-x;
dbedit7.text:=inttostr(z) ;
label10.Caption:=inttostr(a);
datasource1.DataSet.FieldValues['hiredate']:=datetimepicker1.Date;
datasource1.DataSet.FieldValues['bringingdate']:=datetimepicker2.Date;

datasource2.Edit;
datasource2.DataSet.FieldValues['videoinstock']:=label10.Caption;

end;

procedure TForm21.Edit1Change(Sender: TObject);
begin
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where customername=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;

procedure TForm21.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=340;
statusbar1.Panels.Add;
```

```
statusbar1.panels.items[2].width:=80;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);


edit1.text:='';
button1.caption:='Calculate';
end;

procedure TForm21.Button2Click(Sender: TObject);
begin
form21.Hide;
end;

procedure TForm21.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video ID';
end;

procedure TForm21.DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rental Id';
end;

procedure TForm21.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Customer ID';
end;

procedure TForm21.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Quantity';
end;

procedure TForm21.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Unit Price';
end;

procedure TForm21.DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Total Price';
end;
```

```
procedure TForm21.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Calculates The Total Price';
end;

procedure TForm21.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm21.SpeedButton1Click(Sender: TObject);
begin
form21.Hide;
end;

procedure TForm21.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits the Page';
end;

procedure TForm21.Image1Click(Sender: TObject);
begin

end;

end.
```

## TURNED RENTAL

```
unit Unit22;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, Mask, DBCtrls, ExtCtrls,
  Buttons, jpeg;

type
  TForm22 = class(TForm)
    ADOConnection1: TADOConnection;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
```

```
  Label7: TLabel;
  DBCheckBox1: TDBCheckBox;
  DBText1: TDBText;
  DBText2: TDBText;
  DBText3: TDBText;
  DBText4: TDBText;
  DBText5: TDBText;
  DBText6: TDBText;
  DBText7: TDBText;
  Label8: TLabel;
  Bevel1: TBevel;
  ADODataSet1: TADODataSet;
  ADODataSet1RentalId: TAutoIncField;
  ADODataSet1VideoName: TWideStringField;
  ADODataSet1CustomerName: TWideStringField;
  ADODataSet1CustomerSurname: TWideStringField;
  ADODataSet1HireDate: TDateTimeField;
  ADODataSet1BringingDate: TDateTimeField;
  ADODataSet1Quantity: TIntegerField;
  ADODataSet1Turned: TBooleanField;
  ADODataSet1VideoInStock: TIntegerField;
  Label10: TLabel;
  DataSource1: TDataSource;
  Button1: TButton;
  ADOTable1: TADOTable;
  ADOTable1VideoId: TAutoIncField;
  ADOTable1SupplierId: TIntegerField;
  ADOTable1VideoName: TWideStringField;
  ADOTable1VideoDesc: TMemoField;
  ADOTable1VideoDirector: TWideStringField;
  ADOTable1VideoType: TWideStringField;
  ADOTable1VideoYear: TIntegerField;
  ADOTable1UnitPrice: TBCDField;
  ADOTable1VideoInStock: TIntegerField;
  ADOTable1VideoPicture: TBlobField;
  DataSource2: TDataSource;
  DBGrid1: TDBGrid;
  DBEdit2: TDBEdit;
  SpeedButton1: TSpeedButton;
  Image1: TImage;
  procedure Button1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
  procedure Image1Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  Form22: TForm22;

implementation

{$R *.dfm}


procedure TForm22.Button1Click(Sender: TObject);
var
x,y,z: integer ;
begin

x:= strtoint(dbedit2.Text);
y:= strtoint(dbtext7.caption);

if dbcheckbox1.Checked  then
z:= x+y;
label10.Caption:=inttostr(z);
datasource1.Edit;
datasource1.DataSet.FieldValues['videoinstock']:=label10.Caption;

end;


procedure TForm22.FormCreate(Sender: TObject);
begin
button1.Caption:='Confirm?';
end;

procedure TForm22.SpeedButton1Click(Sender: TObject);
begin
form22.Hide;
end;

procedure TForm22.Image1Click(Sender: TObject);
begin

end;

end.
```

## 3 MOST RENTED PRODUCTS

```
unit Unit6;

interface
```

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, dbcgrids, Grids, DBGrids, DB, ADODB,
  Buttons, jpeg, ExtCtrls;

type
  TForm6 = class(TForm)
    ADODataSet1: TADODataSet;
    ADODataSet1Deyim1: TIntegerField;
    ADODataSet1VideoName: TWideStringField;
    DataSource1: TDataSource;
    DBCtrlGrid1: TDBCtrlGrid;
    Label2: TLabel;
    DBText1: TDBText;
    DBText2: TDBText;
    Label4: TLabel;
    Label1: TLabel;
    Label3: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure SpeedButton1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;

implementation

{$R *.dfm}

procedure TForm6.SpeedButton1Click(Sender: TObject);
begin
form6.Hide;
end;

procedure TForm6.FormCreate(Sender: TObject);
begin
  form6.Caption :='3 Most Rented Products';
  label6.Caption :='3 Most Rented Products';
end;

procedure TForm6.Image1Click(Sender: TObject);
```

```
begin

end;

end.
```

**MOST SOLD 3 VIDEO**

```
unit Unit11;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Buttons, DBCtrls, StdCtrls, DB, dbcgrids, ADODB, jpeg, ExtCtrls;

type
  TForm11 = class(TForm)
    ADODataSet1: TADODataSet;
    DataSource1: TDataSource;
    DBCtrlGrid1: TDBCtrlGrid;
    ADODataSet1Deyim1: TIntegerField;
    ADODataSet1VideoName: TWideStringField;
    Label1: TLabel;
    DBText1: TDBText;
    DBText2: TDBText;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure SpeedButton1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form11: TForm11;

implementation

{$R *.dfm}

procedure TForm11.SpeedButton1Click(Sender: TObject);
begin
form11.Hide;
```

```
end;

procedure TForm11.FormCreate(Sender: TObject);
begin
form11.Caption:='Most Sold  3 Video';
end;

end.
```

## BRINGING DATE

```
unit Unit23;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, ComCtrls, Grids, DBGrids, Mask, Buttons,
  jpeg, ExtCtrls;

type
  TForm23 = class(TForm)
    DBGrid1: TDBGrid;
    Button1: TButton;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    MaskEdit1: TMaskEdit;
    MaskEdit2: TMaskEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Button2: TButton;
    StatusBar1: TStatusBar;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure MaskEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure MaskEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
    Y: Integer);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 Form23: TForm23;

implementation

{$R *.dfm}

procedure TForm23.Button1Click(Sender: TObject);
begin
adoQuery1.Close;
adoQuery1.Sql.Clear;
adoQuery1.Sql.Add('Select * From rental');
adoQuery1.Sql.Add('Where bringingdate between:br1 AND :br2 and turned=false');
adoquery1.Parameters.ParamByName('br1').Value:=strtodate(maskedit1.text);
adoquery1.Parameters.ParamByName('br2').Value:=strtodate(maskedit2.text);
adoQuery1.open;

end;

procedure TForm23.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=250;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=90;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);



maskedit1.text:='';
maskedit2.text:='';
maskedit1.EditMask:='99.99.9999';
maskedit2.EditMask:='99.99.9999';
end;

procedure TForm23.Button2Click(Sender: TObject);
begin
```

```
maskedit1.text:='';
maskedit2.text:='';
maskedit1.EditMask:='99.99.9999';
maskedit2.EditMask:='99.99.9999';
end;

procedure TForm23.Button3Click(Sender: TObject);
begin
form23.Hide;
end;

procedure TForm23.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm23.MaskEdit1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter First Date as dd.mm.yyyy';
end;

procedure TForm23.MaskEdit2MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Second Date as dd.mm.yyyy';
end;

procedure TForm23.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Search';
end;

procedure TForm23.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Clears The Page';
end;

procedure TForm23.SpeedButton1Click(Sender: TObject);
begin
form23.Hide;
end;

procedure TForm23.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits The Page';
```

```
end;

end.
```

# References

**Referenca to Book:**

[1] Steve Teixeira,Delphi programming: Delphi 6 Developer's Guide,2000

**Referance to electronic book:**

[1] Delphi 7 Delphi QuickStart .PDF
[2] Delphi_2005_Reviewers_Guide.PDF
[3] Delphi_Database_Application_Developers_Book.PDF
[4] Office.microsoft.com/access
[5]tr.wikipedia.org/wiki/microsoft access

**Referance to Electronic source:**

[1] http://www.borland.com
[2]www.delphiturk.com
[3]www.delphidunyasi.net
[4]www.delphiturkiye.com/