



Database Programming With VB

Unit V

Dr. A. Malathi M.Sc.,M.Phil.,Ph.D.,M.Sc(Psych)

Assistant Professor

PG and Research Department of Computer Science

Government Arts College

Coimbatore - 641018

Syllabus

- UNIT–V: Database Programming With VB: Understanding Database Management Systems –Understanding Relation Concepts – Using Visual Data Manager- Validating Data – Enter Data –Accessing Fields In Record Sets – SQL – Advanced Data –Bound Controls – Managing Databases. Active Data Objects: Crating Data Project –Designing With The Data Environment Active x Designer –Ado Data Control

Understanding Database Management Systems –

- An Introduction to Databases: The smallest unit in a database is called a field (data item).
- A record is a collection of fields.
- A table (file) is a collection of records.
- A database is a collection of one or more tables.
- Visual Basic can manage, revise, and analyze a database that has been created with database management products such as Access, Btrieve, dBase, FoxPro, and Paradox.

 **Customers** ← **File of Customer Data**

	ID	First Name	Last Name	Street Address
+	52	Denver	Ferguson	856 Cook St.
+	53	John	Emory	99 Hillsborough St.
+	54	Ebony	Farmer	872 W. Morgan St.
+	55	Kim	Doe	553 Wayne St.
+	56	Coretta	Diaz	781 Bloodsworth St.
+	57	Victor	Denver	31 St. Mary's St.
+	58	Hamish	David	21 Cameron Ct.

A record stored in a file

A field of the Record

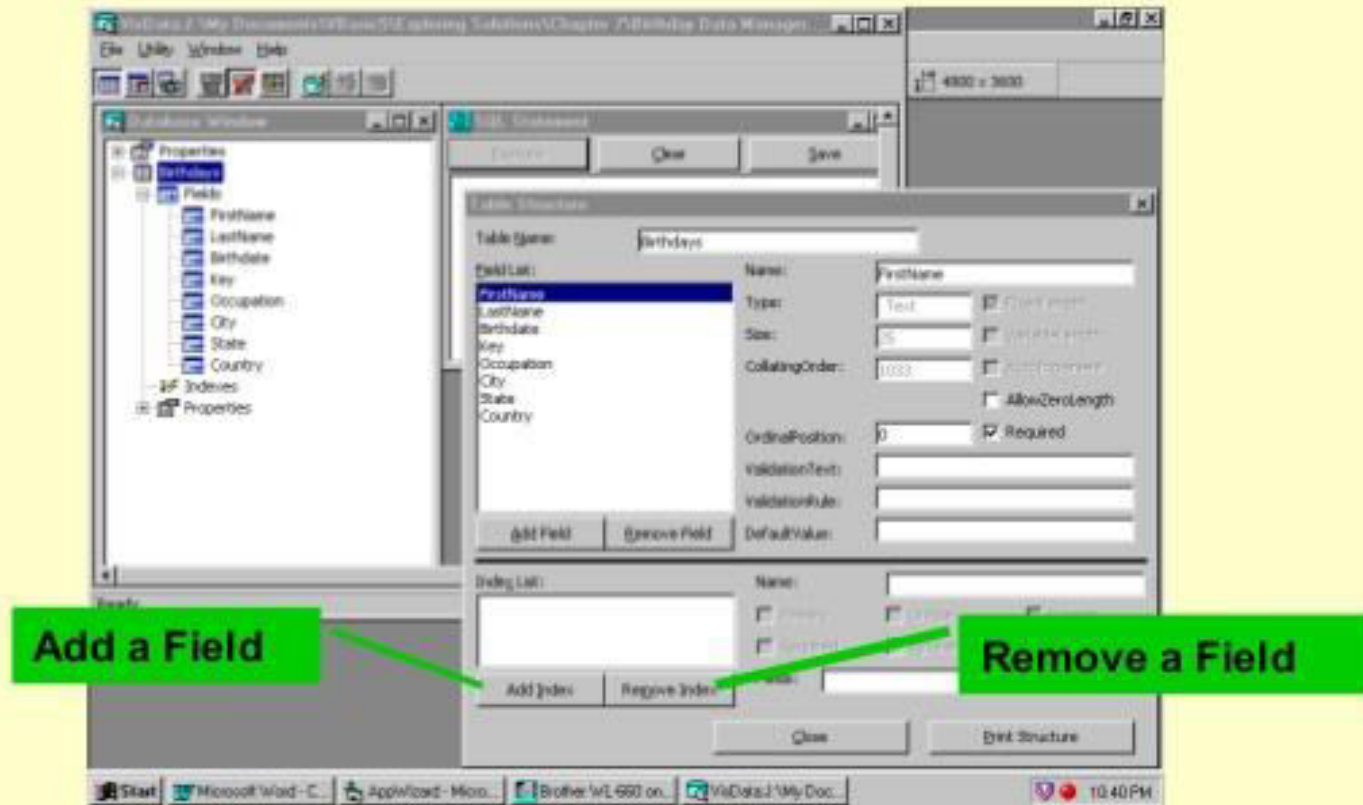
Understanding Relation Concepts

- Microsoft Visual Basic provides tools for creating and accessing a variety of RDBMS (Relational Database Management System). An RDBMS stores and retrieves information according to the relationship defined. In a RDBMS, the data is the container of the tables in which all data is stored in the relationships is formed by data values.
- A database is a collection of data that is related one to another to support a common application. For example Employee details - Name, Address, etc. Each of these collections of data continue a database.

database accessing methods are as follows;

- Jet Engine - Accessing Microsoft Access and Visual Basic databases.
- ODBC (Open Database Connectivity) - Allow access to the client server databases on a network.
- ISAM (Index Sequential Access Method) - Used to access flat databases such as dBase, FoxPro, ParaDox.

Visual Data Manager



Visual Data Manager

- The power of Visual Basic lies in its ability to manipulate records in code. Such tasks as determining the values of particular fields, adding records, deleting records, and moving from record to record are easily done.
- This allows us to build a complete database management system (DBMS).
- The Visual Data Manager is a Visual Basic Add-In that allows the creation and management of databases.
- It is simple to use and can create a database compatible with the Microsoft Jet (or Access) database engine.

Visual Data Manager

- Select **Visual Data Manager** from Visual Basic's **Add-In** menu
- Select **New** from the Data Manager **File** menu. Choose database type (Microsoft Access, Version 7.0), then select a directory and enter a name for your database file. Click **OK**.
- The Database window will open. Right click the window and select **New Table**.
- In the **Name** box, enter the name of your table. Then define the table's fields, one at a time, by clicking **Add Field**, then entering a field name, selecting a data type, and specifying the size of the field, if required.
- Once the field is defined, click the **OK** button to add it to the field box. Once all fields are defined, click the **Build the Table** button to save your table.

Visual Data Manager

- To examine an existing database using the Data Manager, follow these steps:
- Select Visual Data Manager from Visual Basic's Add-In menu (you may be asked if you want to add SYSTEM.MDA to the .INI file - answer No.)
- Select Open Database from the Data Manager File menu.
- Select the database type and name you want to examine.

Validating Data

- Validating user input can be quite a pain, especially if you do not know what techniques and what namespaces are at your disposal. Use regular Expressions to validate user input.
- In simple terms, validation refers to ensuring entered data is well, valid. Determining whether or not data is valid can sometimes be a painstaking process as there are numerous tests that need to be thought of.

These tests include:

- **Format test** - This test determines if the entered string is in the correct format.
- **Type test** - This test checks if the input is the correct type. **For example:** Integers or dates that do not need conversion afterwards.
- **Permitted Character test** - This test ensures that no illegal characters are entered. **For example:** If a user is supposed to enter a name, numbers and some special symbols must not be allowed.
- **Range test** - This test checks to see if entered values are within a specified range. **For example :** If you are only supposed to type in 50 characters, you must not be able to enter more than fifty.

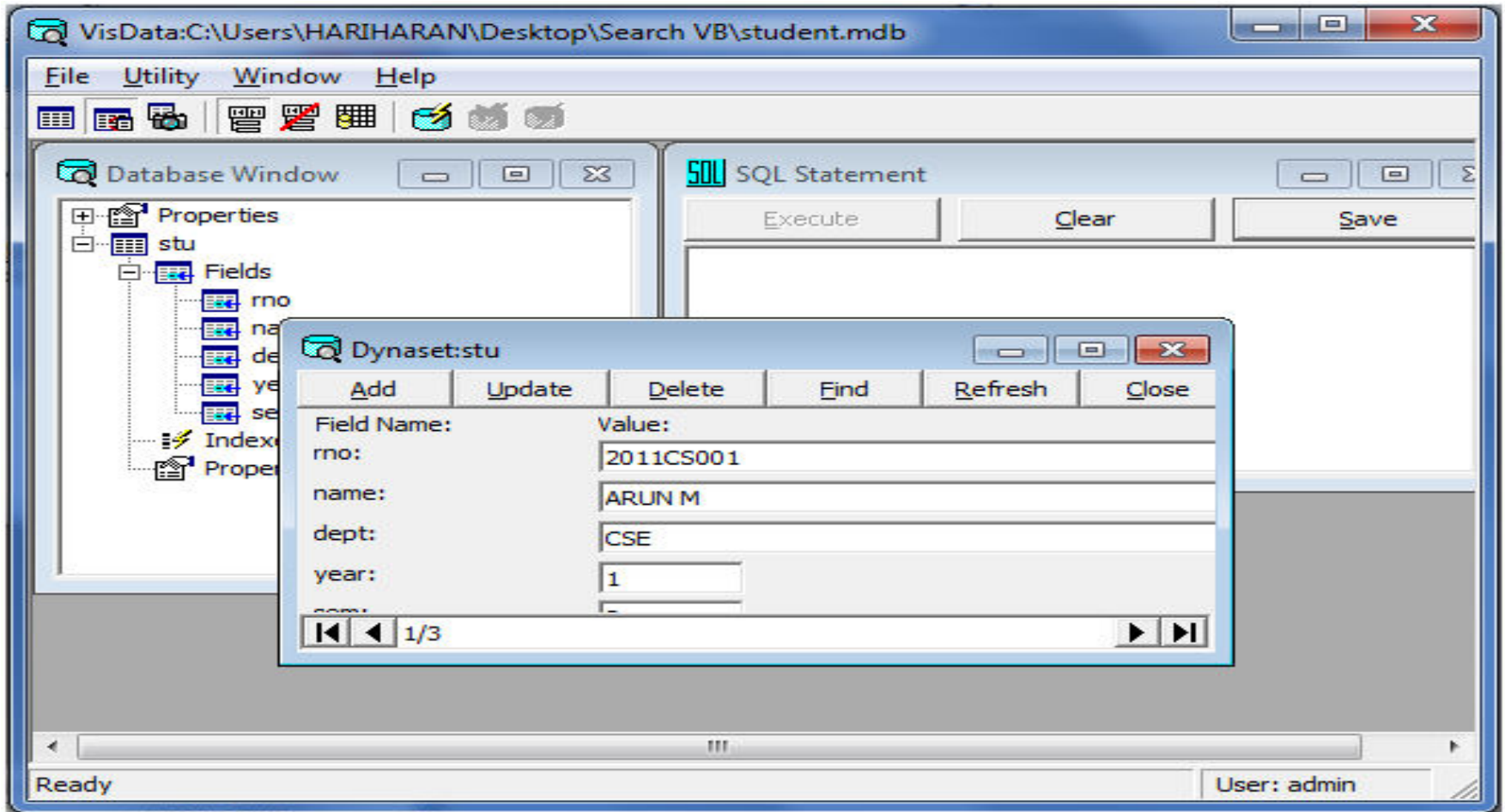
Validating Data

- **Validating data** Before the data you are entering get permanently transferred to the database, you often want to make sure they are correct. That is called **data validation**. We look here at two simple ways of validating data.

Entering data

- **Entering Data with the Visual Data Manager**
- In addition, use the Visual Data Manager for data entry. Double-clicking the name of a table in the Database window opens-the table in data entry ,mode, it can be edited, added, and deleted records.

Entering Data



SQL

- SQL (Structured Query Language) SQL is a query language which is used with relational databases.
- SQL is a very powerful language
- Using SQL you can request specialized information from an existing database and/or have the information presented in a specified order.

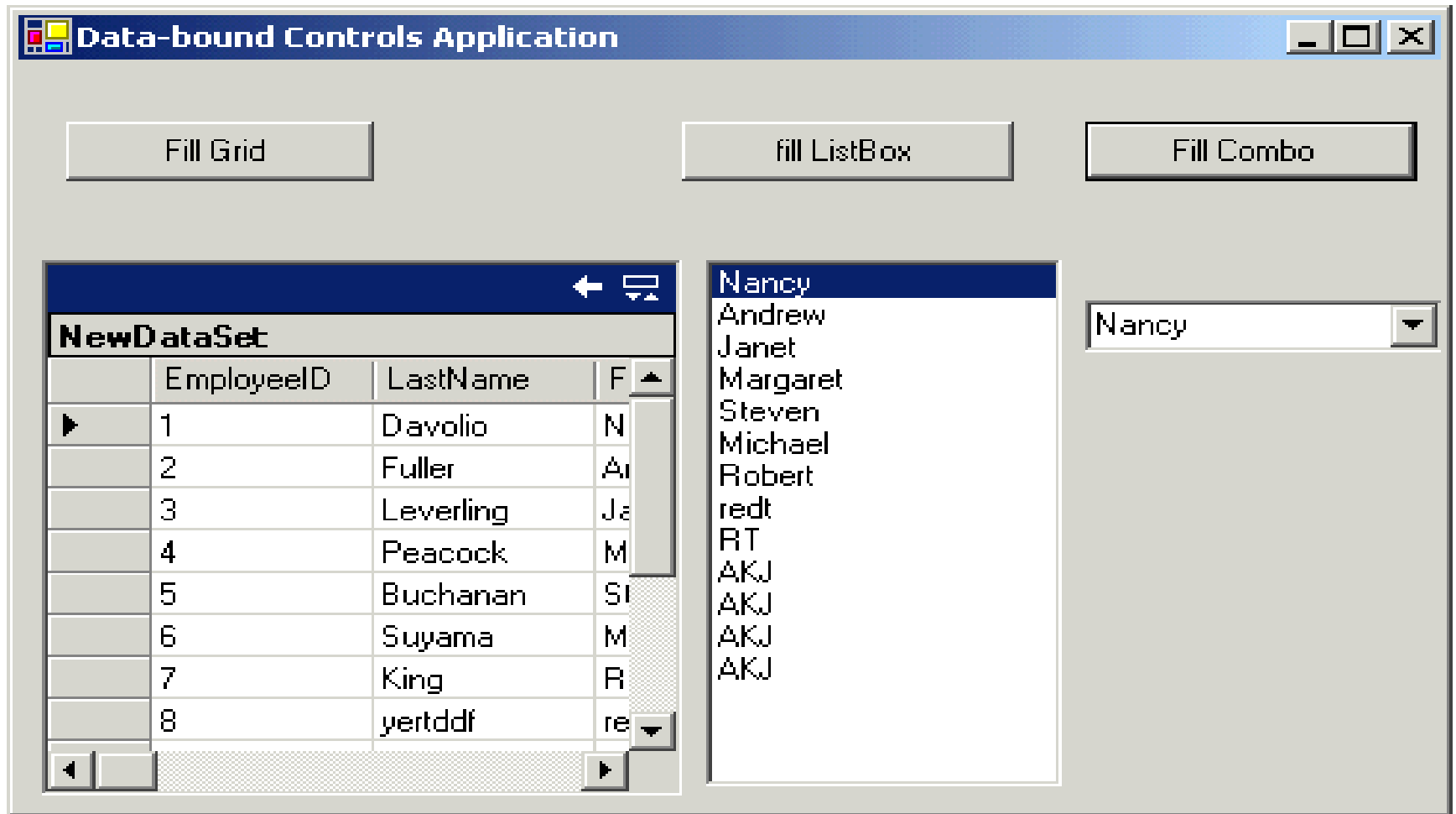
Advanced Data-Bound Controls

- There are three data-bound controls that are quite different from others
 1. The data-bound List control
 2. The data-bound ComboBox control
 3. The data-bound Grid control
- Unlike the other data-bound controls, these controls can display fields from multiple rows.
- The data-bound List control is similar to the regular ListBox control and can be populated with an entire column of a RecordSet.
- The data-bound List control has all the functionality of the regular ListBox control, and in addition, allows the user to enter new values (records). The data-bound Grid control is similar to the Grid control and can display an entire RecordSet.
- Each row of the grid holds a row (record) of the RecordSet, and the RecordSet's columns correspond to the columns (fields) of the RecordSet,

Advanced Data-Bound Controls

- These two data-bound controls aren't installed by default. To use them in your projects, you must first add them to the Toolbox. To do so, follow these steps:
- Right-click the Toolbox and select Components to open the Components dialog .box .
- Check the Microsoft Data Bound Grid Control and the Microsoft Data Bound List Control boxes.
- Click the Close button to close the Components dialog box.

Advanced Data-Bound Controls



DATA Control



Data Control [-] [□] [✕]

Name

Roll No

Class

[<<] [<] Data1 [>] [>>]

Data Control

- The ADO Data Control Two types of data controls
- DAO, stands for Data Access Objects, is the intrinsic data control found on the Toolbox
- ADO, stands for ActiveX Data Objects, is a custom control that must be invoked from the Project Menu
- Provides communication with databases such as Access
- Using a data control you can read, modify, delete and add records to databases
- Use prefix dat for naming a data control.

Managing Databases

- Microsoft Visual Basic provides tools for creating and accessing a variety of RDBMS (Relational Database Management System). An RDBMS stores and retrieves information according to the relationship defined. In a RDBMS, the data is the container of the tables in which all data is stored in the relationships is formed by data values.
- A database is a collection of data that is related one to another to support a common application. For example Employee details - Name, Address, etc. Each of these collections of data continue a database.

database accessing methods are as follows;

- Jet Engine - Accessing Microsoft Access and Visual Basic databases.
- ODBC (Open Database Connectivity) - Allow access to the client server databases on a network.
- ISAM (Index Sequential Access Method) - Used to access flat databases such as dBase, FoxPro, ParaDox.

More Database Tools...

- Different ways to access data
 - ActiveX Data Objects (ADO)
 - Remote Data Objects (RDO)
 - Data Access Objects (DAO)

Active Data Objects: Creating Data Project

- Setting up an ADO project and writing ADO code is similar whether you use Visual Basic or Visual Basic for Applications. This topic addresses using ADO with both Visual Basic and Visual Basic for Applications and notes any differences.
- **Referencing the ADO Library**
- The ADO library must be referenced by your project.
- **To reference ADO from Microsoft Visual Basic**
- In Visual Basic, from the **Project** menu, select **References....**
- Select **Microsoft ActiveX Data Objects x.x Library** from the list. Verify that at least the following libraries are also selected:
 - Visual Basic for Applications
 - Visual Basic runtime objects and procedures
 - Visual Basic objects and procedures
 - OLE Automation

Active Data Objects: Creating Data Project

- use ADO just as easily with Visual Basic for Applications, by using Microsoft Access, for example, **To reference ADO from Microsoft Access**
- In Microsoft Access, select or create a module from the **Modules** tab in the **Database** window. On the **Tools** menu, select **References...**
- Select **Microsoft ActiveX Data Objects x.x Library** from the list. Verify that at least the following libraries are also selected:
 - Visual Basic for Applications
 - Microsoft Access 8.0 Object Library
 - Microsoft DAO 3.5 Object Library
- Click **OK**.
- **Creating ADO Objects in Visual Basic**
- To create an automation variable and an instance of an object for that variable, use two methods: **Dim** or **CreateObject**.
- **Dim** to declare and create instances of ADO objects in one step
- Dim conn As New ADODB.Connection Alternatively, the **Dim** statement declaration and object instantiation can also be two steps:

Designing With The Data Environment Active x Designer

- The Data Environment designer provides an interactive, environment for creating programmatic data access. At design time, you set property values for Connection and Command objects, write code to respond to ActiveX® Data Object (ADO) events, execute commands, and create aggregates and hierarchies. You can also drag Data Environment objects onto forms or reports to create data-bound controls.
- With the Data Environment designer, you can accomplish the following tasks:
 - Add a Data Environment designer to a Visual Basic project.
 - Create Connection objects.
 - Create Command objects based on stored procedures, tables, views, synonyms, and SQL statements. Create hierarchies of commands based on a grouping of Command objects, or by relating one or more Command objects together.
 - Write and run code for Connection and Recordset objects.
 - Drag fields within a Command object from the Data Environment designer onto a Visual Basic form or the Data Report designer.

ADO Data control

- The ADO (ActiveX Data Object) data control is the primary interface between a Visual Basic application and a database.
- It can be used without writing any code at all! Or, it can be a central part of a complex database management system.
- This icon may not appear in your Visual Basic toolbox. If it doesn't, select Project from the main menu, then click Components. The Components window will appear. Select Microsoft ADO Data Control, then click OK. The control will be added to your toolbox.
- The data control (or tool) can access databases created by several other programs besides Visual Basic (or Microsoft Access). Some other formats supported include Btrieve, dBase, FoxPro, and Paradox databases.

ADO Data control

- The data control can be used to perform the following tasks:
 1. Connect to a database.
 2. Open a specified database table.
 3. Create a virtual table based on a database query.
 4. Pass database fields to other Visual Basic tools, for display or editing. Such tools are bound tools (controls), or data aware.
 5. Add new records or update a database.
 6. Trap any errors that may occur while accessing data.
 7. Close the database.

References

- SCHAUM'S OUTLINE OF THEORY AND PROBLEMS OF PROGRAMMING WITH VISUAL BASIC
- http://media.pearsoncmg.com/ph/esm/deitel/vb_hpt_2010/codeexamples.html
- <https://labprograms.wordpress.com/2011/03/21/working-with-database-in-vb6-0-without-using-data-control/>
- <https://www.c-sharpcorner.com/article/working-with-data-bound-controls/>
- <https://www.slideshare.net/MuralidharanRadhakrishnan/vb6-ch8>

The End