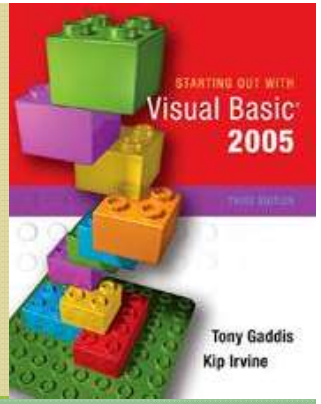
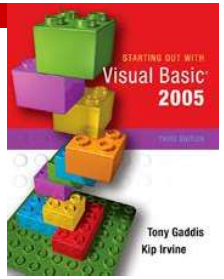


10.2 Using the DataGridView

The DataGridView Control Allows you to Display a Database Table in a Grid Which Can be Used at Runtime to Sort and Edit the Contents of the Table

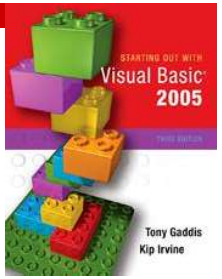
Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Connecting VB to a Database



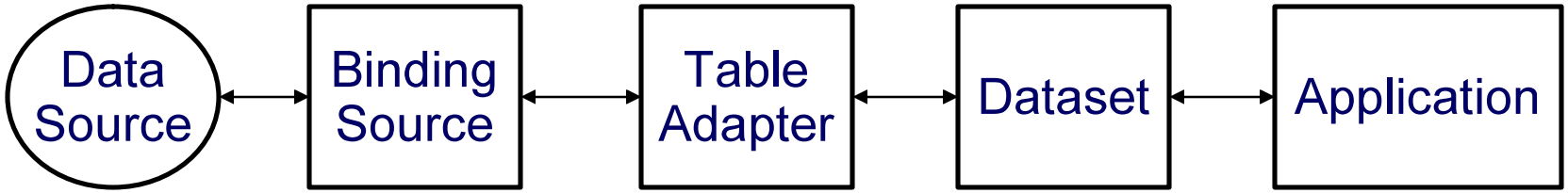
- VB provides tools to display database tables
- *Data binding* links tables to form controls
  - Controls called components establish the link
  - A wizard guides you through the process
- We'll use these data-related components:
  - *Data source* – usually a database
  - *Binding source* – holds database name, location, and other connection information
  - *Table adapter* – uses SQL to select data
  - *Dataset* – in-memory copy of data from tables

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley



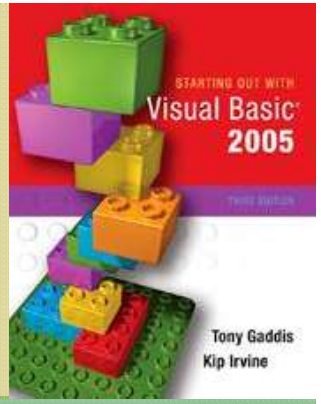
Connecting VB to a Database

- The flow of data from database to application



  - Note that data moves in both directions
  - Data travels from data source to application
  - Application can view/change dataset contents
  - Changes to dataset can be written back to the data source
- Tutorial 10-1 demonstrates how to connect a database table to a DataGridView control

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

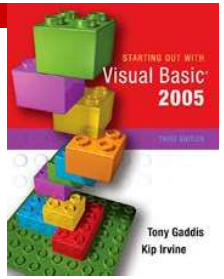


10.3 Selecting Dataset Rows

Visual Basic Provides Convenient Tools for Selecting Which Rows from the Dataset You Want to Display

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

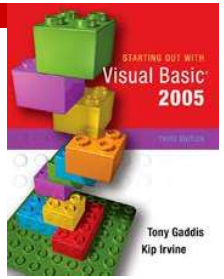
Structured Query Language (SQL)



- Often need to select certain rows in a table using *Structured Query Language* (*SQL*)
  - Standard method of working with a database
  - Adopted by most all database software
  - *Not* a general purpose programming language
  - Defines how to construct *queries* that return selected data from the database

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

SQL Query Example



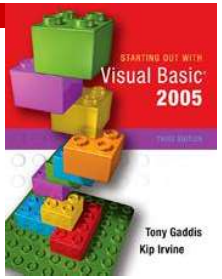
- You might want to select all employees hired before 1998 and earning less than $45,000

```
Select ID, Name, Full_Time, Hire_Date, Salary
  From employees
  Where Hire_Date < 1/1/1998 and Salary < 45000
```

  - **`Select`**, **`From`**, and **`Where`** are keywords
  - Fields to be returned listed after **`Select`**
  - Table containing the data listed after **`From`**
  - Conditions affecting row selection after **`Where`**
- SQL query is a property of the *TableAdapter*

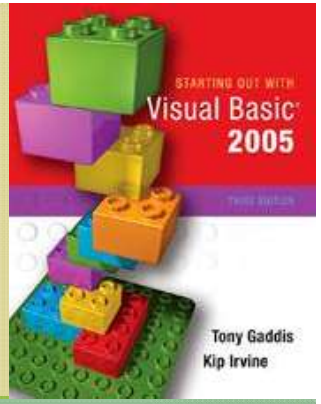
Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Configuring the TableAdapter



- Right-click on dataset and select *Configure* to start *TableAdapter Configuration Wizard*
  - Can modify a simple query directly in wizard
  - Or use *Query Builder* for more complex query
- Query Builder is a tool to create or modify queries with minimal knowledge of SQL
- To add a query to a DataGridView
  - Right-click Table Adapter icon attached to grid
  - Select *Add Query* from shortcut menu
- Tutorial 10-2 demonstrates how this is done

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

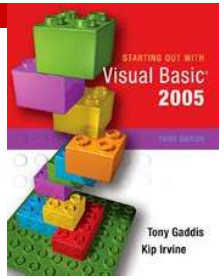


10.4 Data-Bound Controls

Some Controls Can Be Bound to a Dataset. A Data-bound Control Can be Used to Display and Edit the Contents of a Particular Row and Co.lumn

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

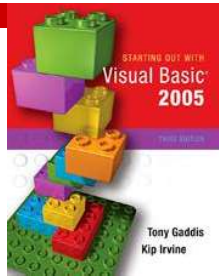
Advantages of Data-Binding



- Can add multiple data sources to a project
- Can bind fields in a data source to controls:
  - Text boxes
  - Labels
  - List boxes
- Contents of *data-bound controls* change automatically when moving from row to row
- Updating the contents of a database field from a *data-bound control* is also very easy

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

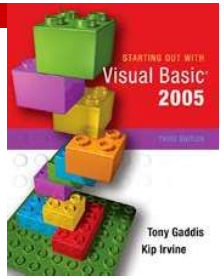
Binding Existing Dataset to DataGrid



- Use the *Data Sources* window
  - Locate the dataset table
  - Drag table to an open area of a form
  - Creates a data grid bound to the data source
- Automatically adds a navigation bar to form
- Set *Dock* property to *Center Docking* to make the data grid fill the entire form

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

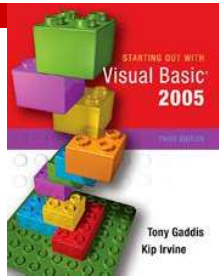
Binding Individual Fields to Controls



- Use the *Data Sources* window
  - Locate the dataset table
  - Select Details from the table drop-down list
  - Drag table to an open area of a form
  - Creates a separate control for each field
  - Can also drag columns individually
- Adds automatic navigation bar as before
- Text and numeric fields added as text boxes
- Yes/No fields added as checkboxes
- May wish to change some control properties

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

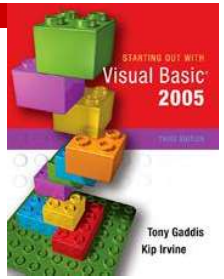
Binding to List and Combo Boxes



- List and combo boxes are frequently used to supply a list of items for a user to select from
- Such lists are often populated from a table
- Must set two list/combo box properties
  - *DataSource* identifies a table within a dataset
  - *DisplayMember* identifes the table column to be displayed in the list/combo box
- If table column dragged onto a list/combo box
  - Visual Studio creates the required dataset, table adapter, and binding source components
- Tutorial 10-3 shows this type of binding

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Adding New Rows to a Dataset

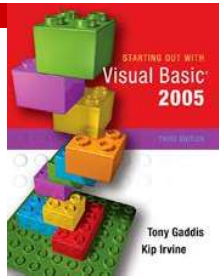


- *NewRow* method creates a new, empty row
- Assign values to columns in the empty row
- *Add* method appends new row to the dataset
- The following example adds a new row to the Payments table of the dsPayments dataset

```
Dim row as dsPayments.PaymentsRow
Row = CType(DsPayments.Payments.NewRow(), _
          dsPayments.PaymentsRow)
With row
  row.Member_ID = 5
  row.Payment_Date = #5/15/2006#
  row.Amount = 500D
End With
DsPayments.Payments.Rows.Add(row)
```

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Another Means to Add Dataset Rows



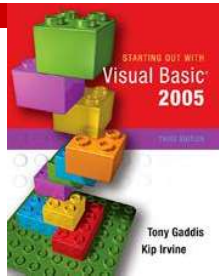
- Can call the dataset *Add* method directly
- This approach is more straightforward but:
  - Must specify values in correct column order
  - Previous example becomes far simpler:

```
DsPayments.Payments.Rows.Add( _
                Nothing, 5, #5/15/2006#, 500D)
```

- Payments table 1st column is autonumber so:
  - Must have an argument for this column
  - Pass *Nothing* as the value for this column

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Removing a Row from a Dataset

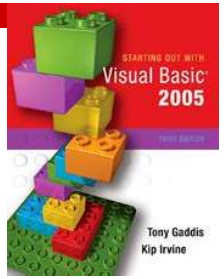


- Two steps to remove a row
  - Get a reference to the row to remove
  - Call *Remove* method on Rows collection
- Following example calls *FindByID* with the primary key to get a reference to the row

```
Dim row as DataRow = _
        DsPayments.Payments.FindByID(36)
DsPayments.Payments.Rows.Remove(row)
```

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Updating the Database



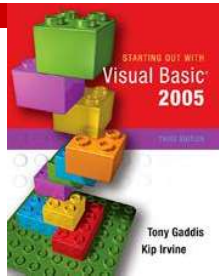
- Previous add and remove examples change the dataset but not the underlying database
- Must call TableAdapter *Update* method to save dataset changes to the database
- To write changes in the DsPayments dataset to the underlying database:

```
PaymentsTableAdapter.Update(DsPayments)
```

- Tutorial 10-4 demonstrates adding new rows to the Payments table

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Reading Dataset Rows with For-Each



- A *For-Each* statement can be used to iterate over all rows of a dataset
- Usually use a strongly typed dataset for this
- Sum Amount column of dsPayments dataset

```
Dim row as dsPayments.PaymentsRow
Dim decTotal as Decimal = 0
For Each row in DsPayments.Payments.Rows
    decTotal += row.Amount
Next
```

- Tutorial 10-5 demonstrates this technique

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley