

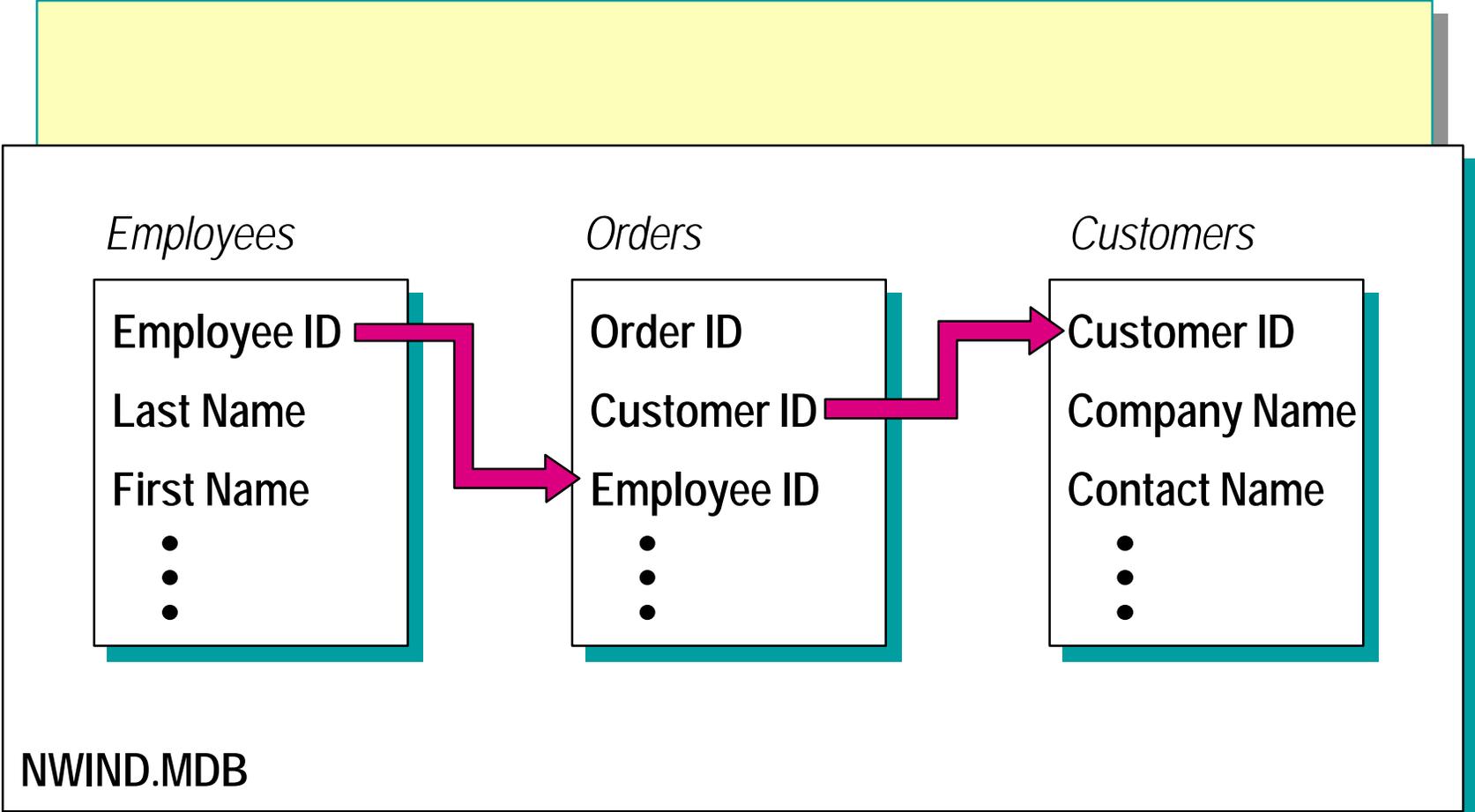
Data Access Using the Data Control

12

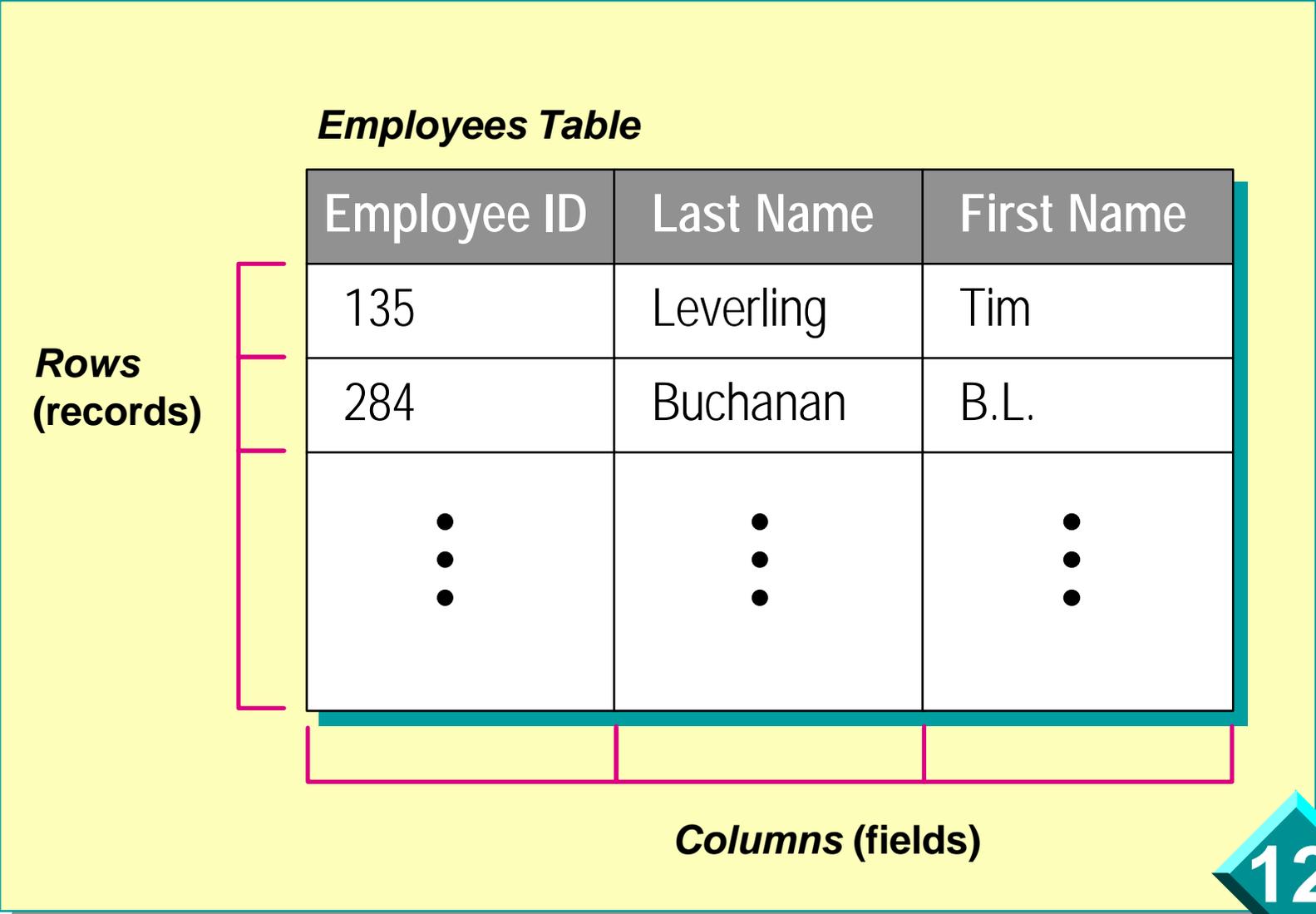
◆ Overview

- Overview of a Database
- How Does Visual Basic Access Databases?
- What Is the Data Control?
- The Current Row and the Dynaset
- Finding Records
- Updating, Deleting, and Adding Records
- Data Control Events
- How Locking Is Handled
- Filling a Grid with Data from a Table

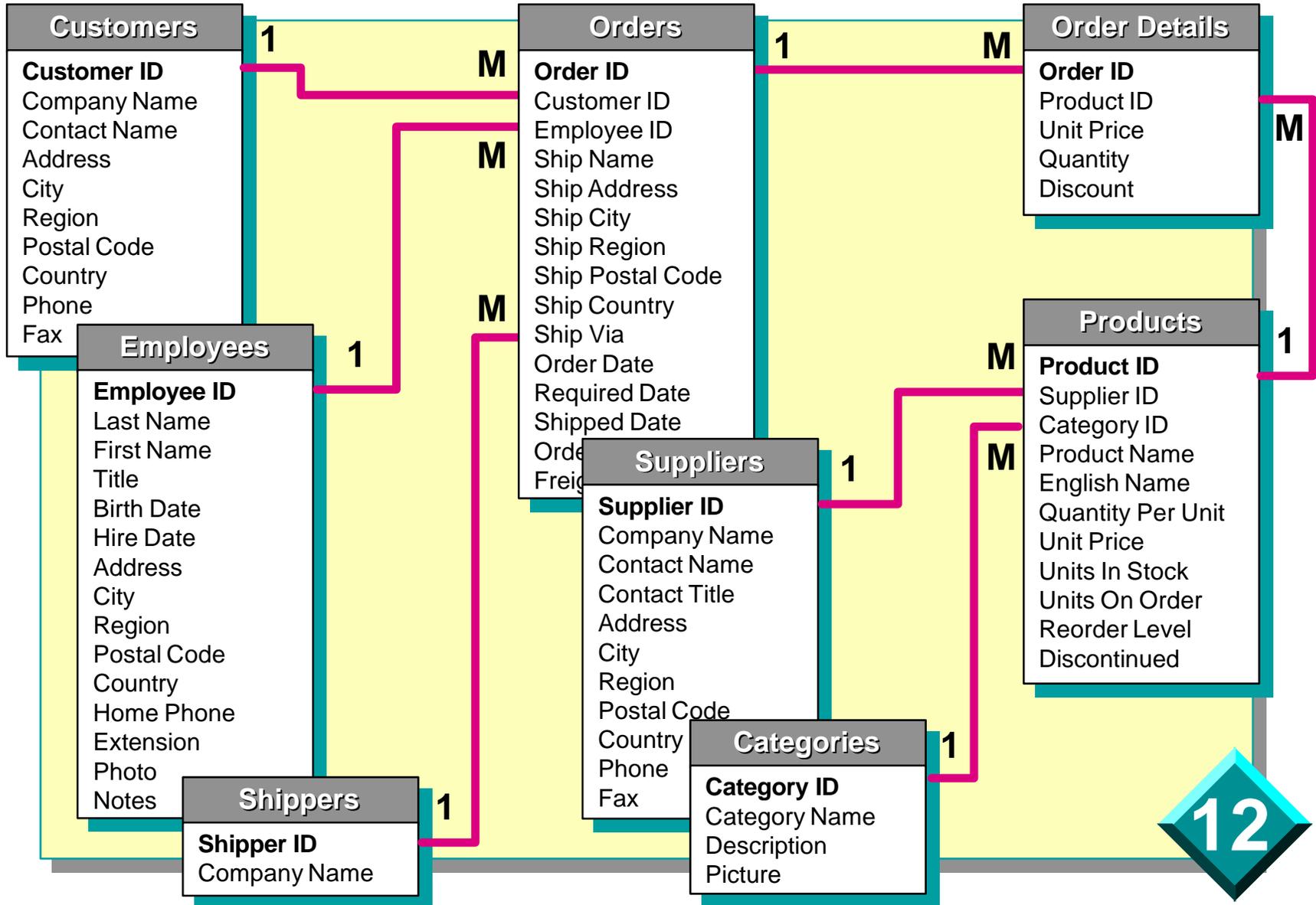
Overview of a Database



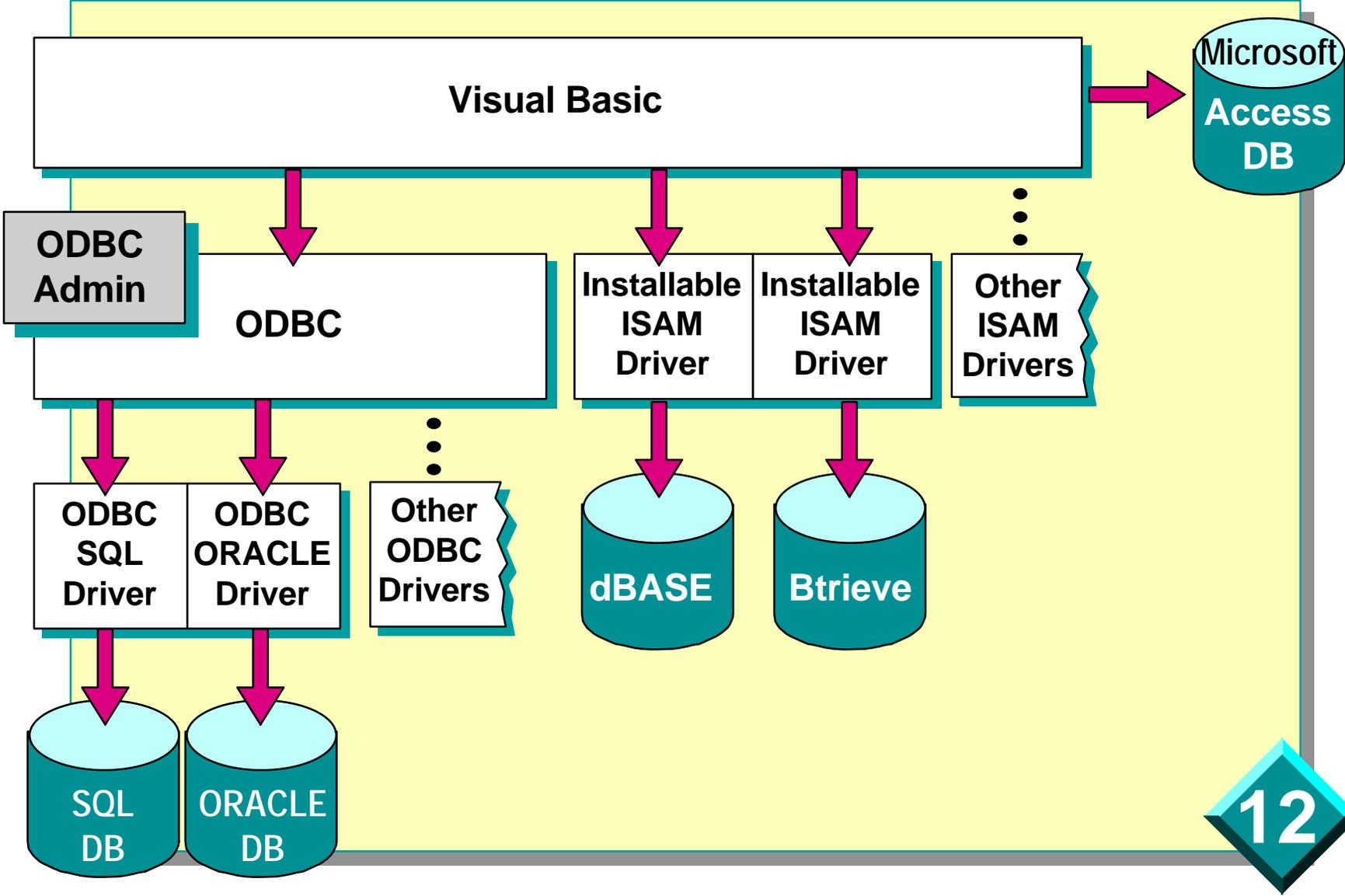
Overview of a Table



Northwind Database Diagram

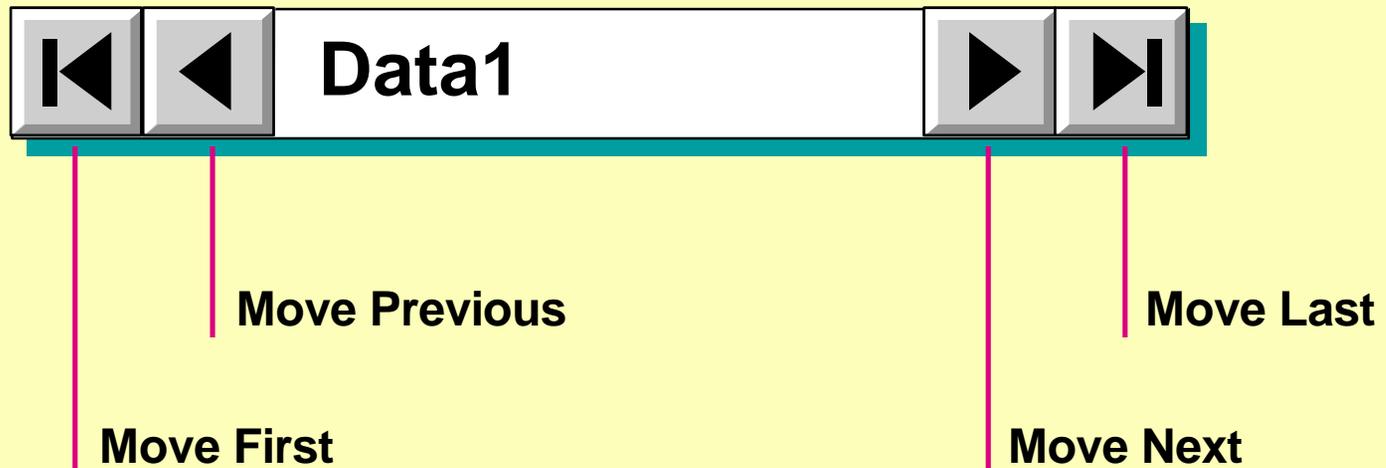


How Does Visual Basic Access Databases?



The Data Control

- Creates DYNASET
- Reflects DB Table or SQL Query



Binding Controls to the Data Control

```
.DataSource = Data1  
.DataField = 'Last Name'
```

The diagram illustrates the binding of a control to a data control. It shows a window titled "Form1" containing a text box labeled "Last Name:" and a data control labeled "Data1" with navigation buttons. A pink arrow points from the code above to the text box, indicating the binding process.

Data Control Walkthrough

The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area of the window contains two text input fields. The first field is labeled "Last Name:" and the second is labeled "First Name:". Below these fields is a data control labeled "Data1". The "Data1" control has four navigation buttons: a "Home" button (represented by a vertical bar and a left-pointing arrow), a "Previous" button (represented by a left-pointing arrow), a "Next" button (represented by a right-pointing arrow), and an "End" button (represented by a right-pointing arrow and a vertical bar).

Displaying a Calculated Field

Form1

Quantity:

Price:

Total:

Navigation buttons: Data1

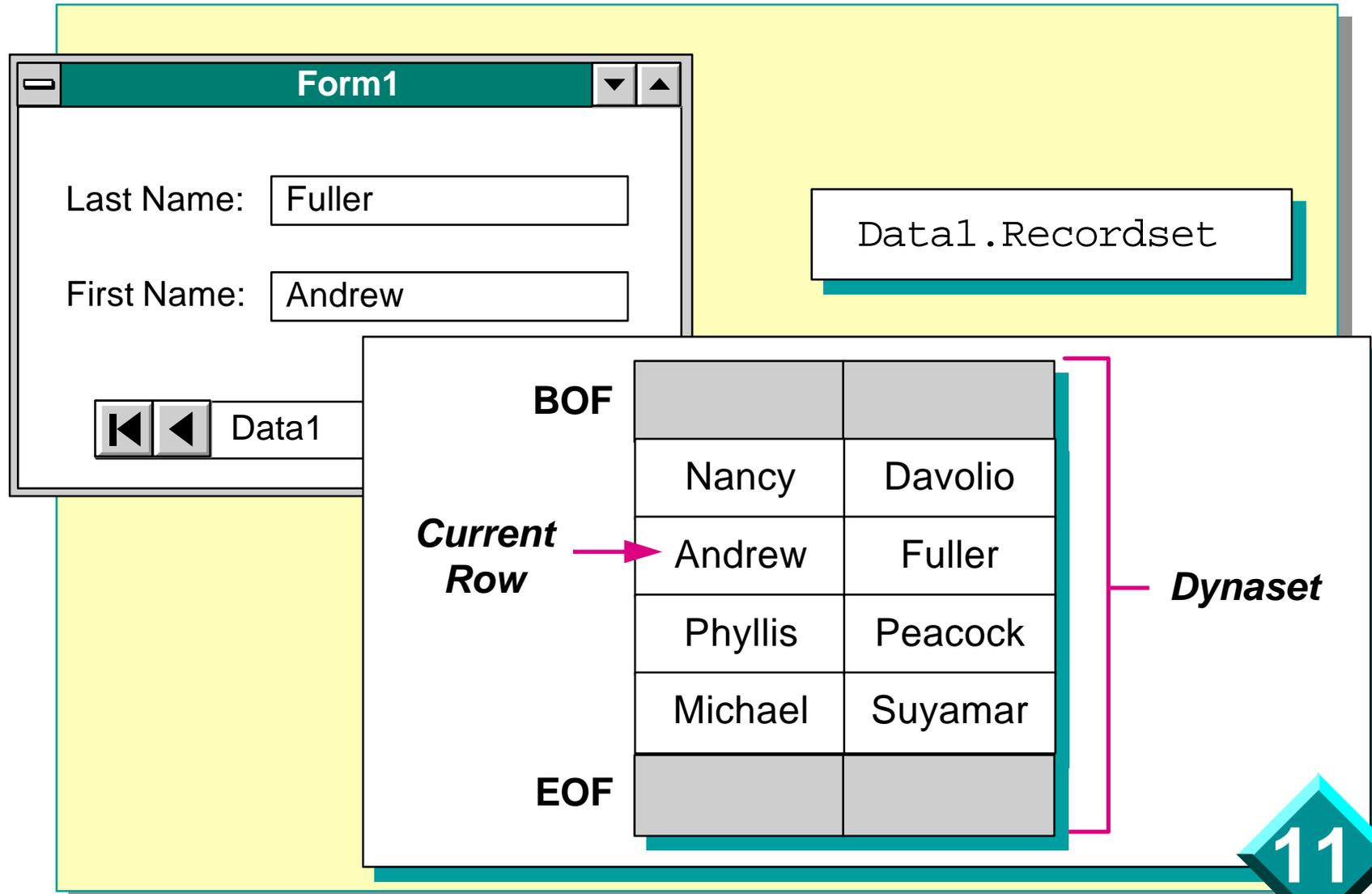
`total = [Unit Price] * Quantity`

```
.RecordSource = "Select [Unit Price], quantity,  
↪ [Unit Price] * quantity As total from [Order Details]"
```

Data Control Methods and Properties

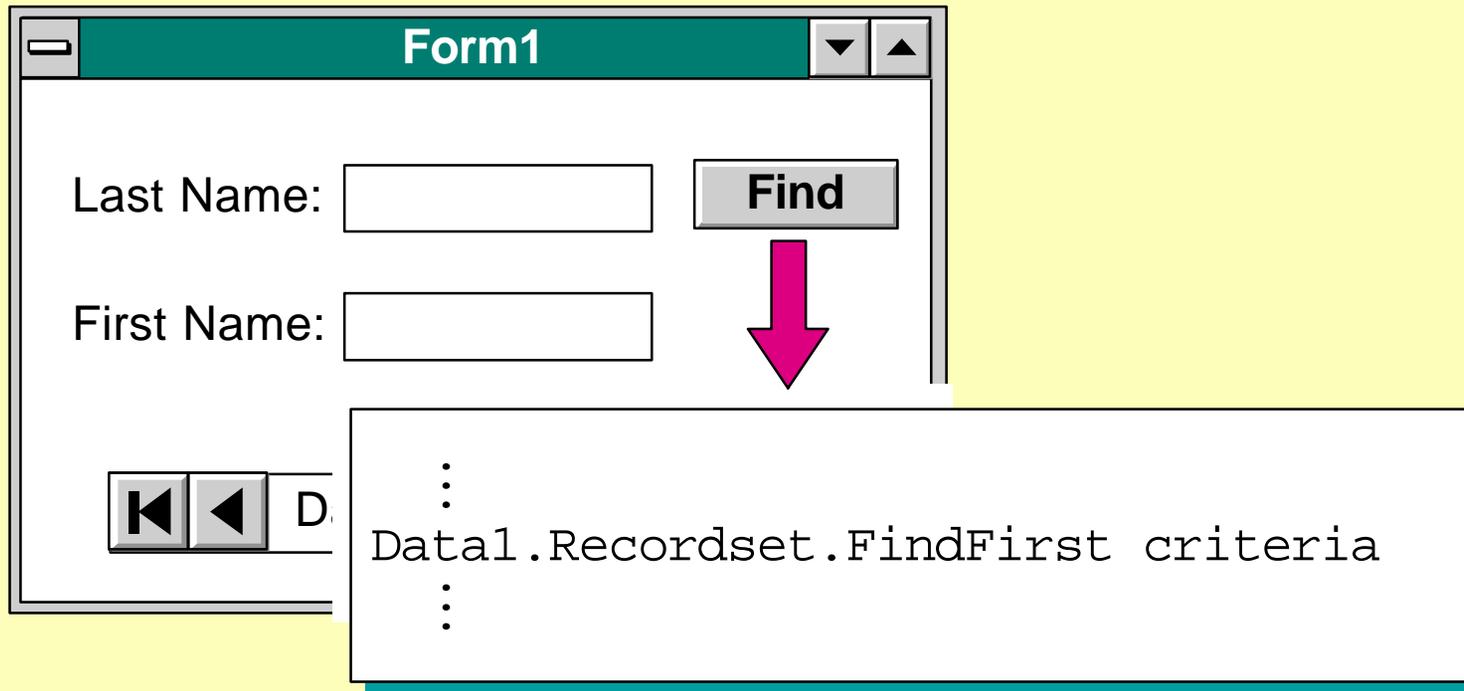
- Refresh Method
- Connect Property
- Exclusive Property
- ReadOnly Property

The Current Row and The Dynaset



Finding Records

- Find Methods: `FindFirst`, `FindLast`, `FindPrevious`, `FindNext`

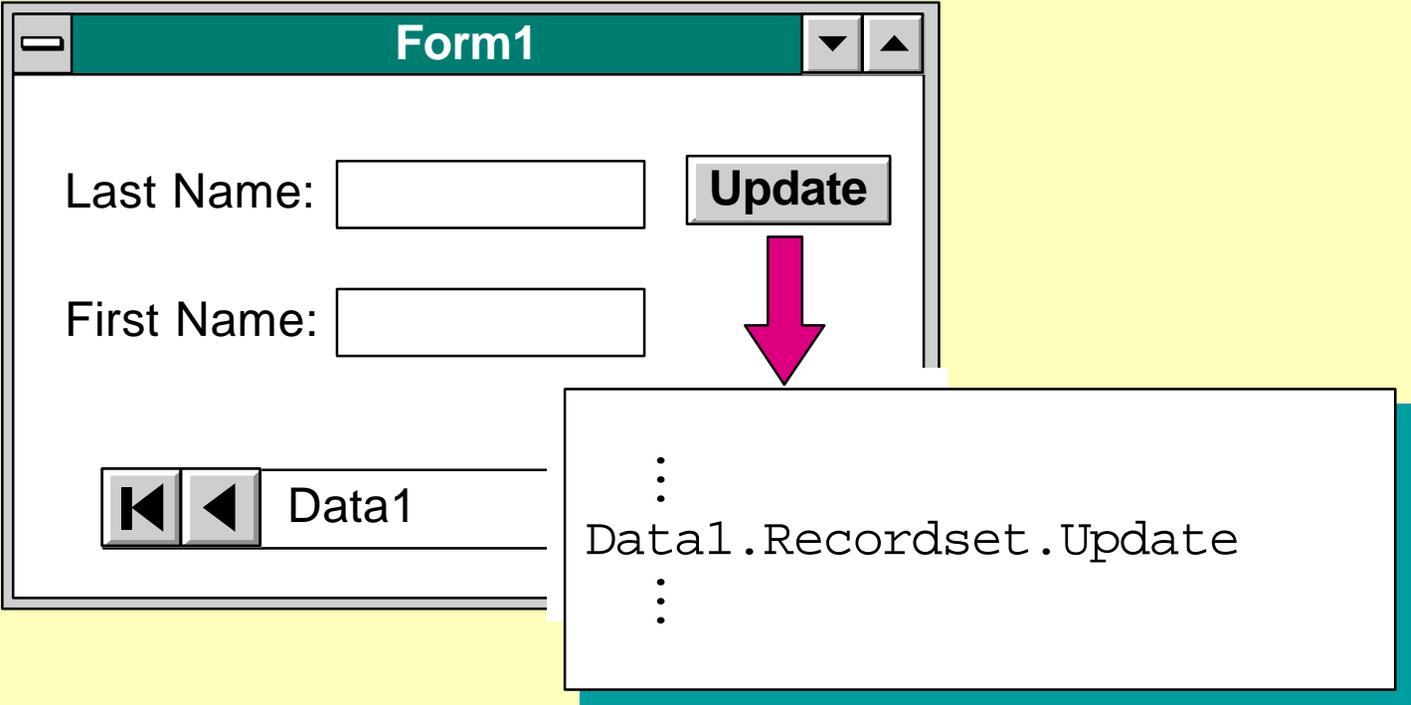


The screenshot shows a form window titled "Form1". It contains two text input fields labeled "Last Name:" and "First Name:". To the right of the "Last Name:" field is a button labeled "Find". A pink arrow points from the "Find" button to a code block. The code block contains the following text:

```
⋮  
Data1.Recordset.FindFirst criteria  
⋮
```

- Criteria: Any Valid SQL "Where" Clause
- Use Bookmark Property to Save Current Position

Updating Records



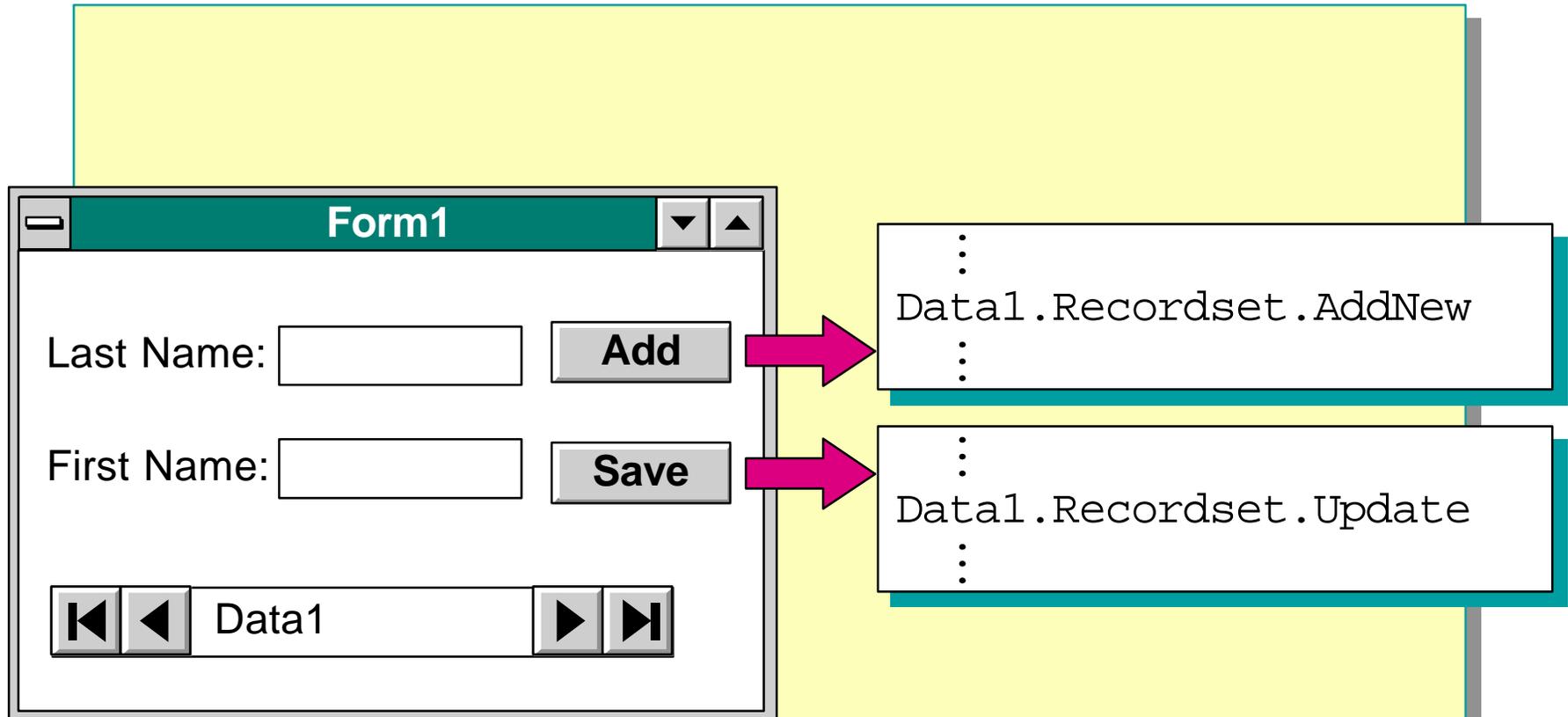
Deleting Records

The image shows a Windows form titled "Form1" with a dark green title bar. Inside the form, there are two text boxes: "Last Name:" and "First Name:". To the right of the "Last Name:" box is a button labeled "Delete". A pink arrow points from the "Delete" button to a white box containing the following code:

```
⋮  
Data1.Recordset.Delete  
⋮
```

At the bottom of the form, there is a data grid labeled "Data1" with navigation buttons (back, forward, first, last).

Adding New Records



◆ Data Control Events

- Validate
- Reposition
- Error

The Validate Event

Form1

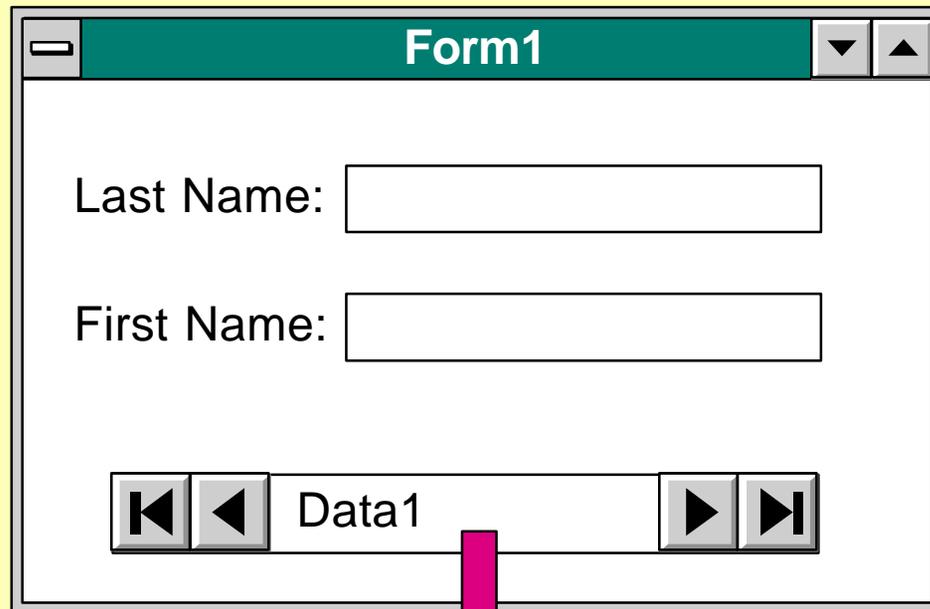
Last Name:

First Name:

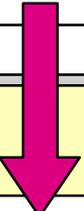
◀◀ Data1 ▶▶

```
Sub Data1_Validate (Action As Integer, Save As Integer)  
    'Occurs when current record changes
```

The Reposition Event



The screenshot shows a window titled "Form1" with a standard Windows title bar. Inside the window, there are two text input fields. The first is labeled "Last Name:" and the second is labeled "First Name:". Below these fields is a control labeled "Data1" which includes four navigation buttons: a double left arrow, a single left arrow, a single right arrow, and a double right arrow.



```
Sub Data1_Reposition()  
    'Occurs when current record changes
```

The Error Event

Form1

Employee ID:

Last Name:

First Name:

Data1

```
Sub cmd Update  
  On Error...  
  ...  
  'Handle Duplicate Key
```

```
Sub Data1_Error ()  
  'If error occurs, for example: Duplicate Key
```

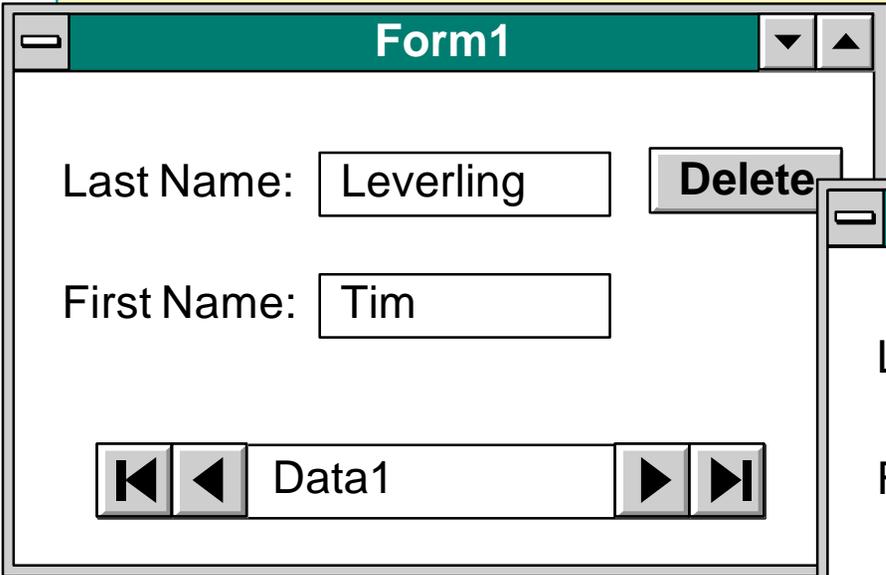
Handling Multiple Users

User #1

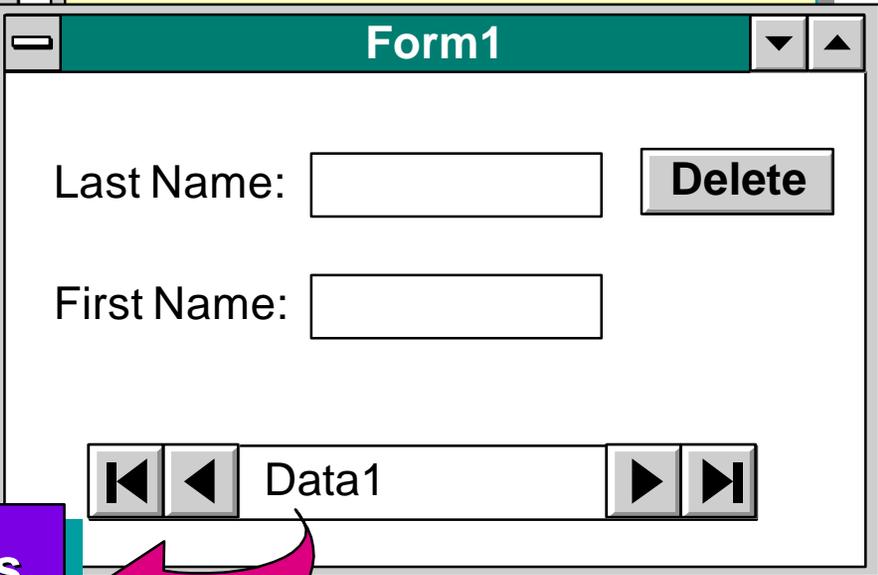
- Deletes "Leverling"

User #2

- "Record is Deleted" message when moving to "Leverling"

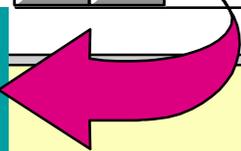


A screenshot of a Windows-style window titled "Form1". It contains two text input fields: "Last Name:" with the text "Leverling" and "First Name:" with the text "Tim". To the right of the "Last Name" field is a "Delete" button. Below the input fields is a data control bar with a "Data1" label and navigation buttons (back, forward, first, last).

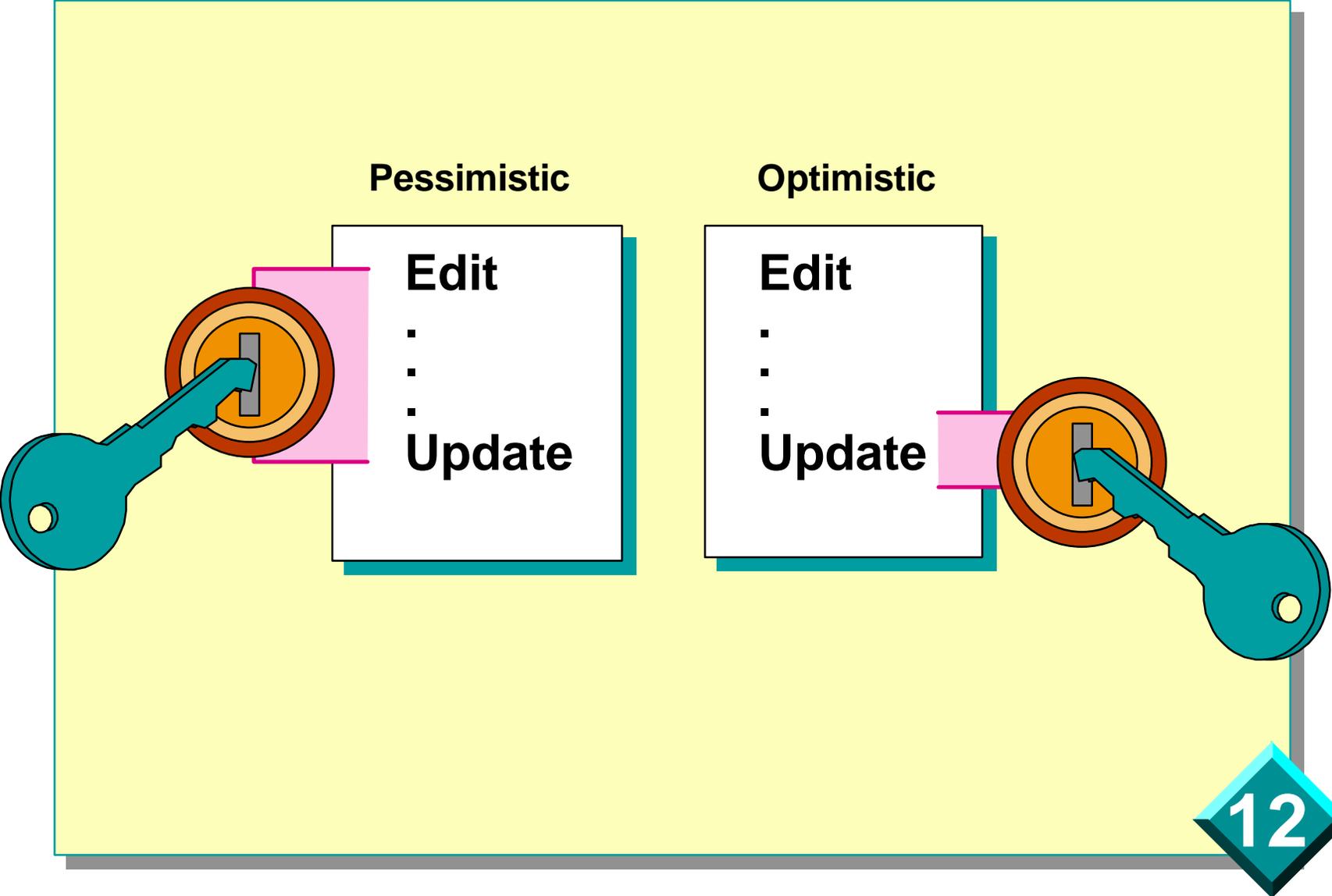


A screenshot of a Windows-style window titled "Form1". It contains two empty text input fields: "Last Name:" and "First Name:". To the right of the "Last Name" field is a "Delete" button. Below the input fields is a data control bar with a "Data1" label and navigation buttons (back, forward, first, last).

"Record is Deleted"



How Locking Is Handled



Accessing Data Programmatically

```
Data1.Recordset("Last Name")  
Data1.Recordset![Last Name]  
Data1.Recordset.Fields(0).Value
```

Filling a Grid with Data

Single Record

Last Name:

First Name:

Multiple Records

Last Name	First Name
Leverling	Tim
Buchanan	B.L.
⋮	⋮

Note: Many 3rd Party Grid Control are Bound Controls

Determining the Files You Need to Distribute

- **Special DLLs Are Required For**
 - Data control
 - Accessing Btrieve, Paradox, or Microsoft FoxPro® files
 - Accessing ODBC format databases
 - Accessing SQL Server (Professional Edition only)

Summary

- Overview of a Database
- How Does Visual Basic Access Databases?
- What Is the Data Control?
- The Current Row and the Dynaset
- Finding Records
- Updating, Deleting, and Adding Records
- Data Control Events
- How Locking Is Handled
- Filling a Grid with Data from a Table