CHAPTER 18

DATABASE MANGEMENT

INTRODUCTION

- Database systems provide file-processing capabilities, but organize data in a manner to facilitate satisfying sophisticated queries.
- Most popular type of database is relational database.
- A language called Structured Query Language (SQL) is used in relational databases to make queries.
- One of the most important aspects of database systems is data independence- applications need not be concerned with how the data is physically stored or accessed.

RELATIONAL DATABASE MODEL

- Logical representation of data that allows relationships between data to be considered without concerning oneself with physical implementation of data structures.
- Relational database is composed of Tables.(Figure 18.1)
- Any particular row of table is called a record (or row).
- Primary key is a unique field (i.e., ssn or employee id) used for searching or matching other records in another table.
- Each column of table represents a different field.

RELATIONAL DATABASE MODEL

- Records are normally unique(by primary key) within a table, but particular field values may be duplicated between records.
- In VB, a table is manipulated as a **Recordset** object.
- ADO Data Control & Data Grid Control
- VB provides a variety of data-aware controls for displaying and manipulating data in databases.
- Data-aware controls in Figure 18.3, p.771-772.
- Database application use ADO(Active X Data Object) Data Control and Data Grid control.

ADO DATA CONTROL & DATA GRID CONTROL

- DataGrid control- allows an entire table or the result of a query to be displayed and manipulated.
- DataSource- used to set or get the source of data from a control.
  - This can be set at design time or at run time to any valid data source.
  - Valid sources include:Recordsets produced via queries through ADO Data Control, ADO code-based queries, DataEnvirionment objects created with DataEnvironment Designer, a user-defined class that is created as a data source or a data source control.

ADO DATA CONTROL & DATA GRID CONTROL

- Data-aware controls must be bound to the DataControl.
- These controls have the ability to access information from databases.
- Example: DataControl.Recordset.Fields("FieldName").Value
- lastName = datPhonebook.Recordset.Fields("LastName").Value
- DataMember- used to set or get the data set that contains data to be manipulated.
- Example: ADO Data Control specifies a table or query or stored procedure(a frequently executed query than can receive parameters) in the database that will produce the Recordset; and the Data Environment Designer allows creation of multiple command objects that each specify a table, query or stored procedure in database will produce the Recordset.

ADO DATA CONTROL & DATA GRID CONTROL

- VB displays a list of the available data sets.
- DataField- used to set or get the specific field in Recordset to which a data-aware control is bound.
  - VB displays a list that contains all fields in Recordset.
- DataFormat- used to specify automatic data formats for data retrieved from data source.
- Program of Figure 18.5 uses ADO DataControl and DataGrid.
- ADO DataControl manages the connection between an application and a database.

ADO DATA CONTROL & DATA GRID CONTROL

- Allows data-aware controls to view and manipulate data in database.
- DataGrid control provides easy access to contents of a Recordset for both viewing and editing data in database.
- Setting AllowUpdate property of DataGrid to False disables user's ability to edit data.
- Connecting DataGrid to ADO DataControl, automatically displays rows, columns, and Field names for table attached.
- Adding ADO DataControl and DataGrid:
  - Project->Components->Microsoft ADO DataControl 6.0, check to select

ADO DATA CONTROL & DATA GRID CONTROL

  - Scroll down to Microsoft DataGrid Control 6.0, check to select, click ok, two controls will appear in toolbox.
- Creating an OLE DB data source:
  - 1. Right on ADO DataControl and select ADODC properties from property pages.
  - 2. In source of connection from, select Use Connection String option and press Build button to display Data Link Properties dialog.
    - Allows programmer to build a connection string.
      - Connection string provides information the ADO Data Control used to connect to the database.
      - Connection string includes database provider database management system type(name and location).
      - Access databases normally use Microsoft jet 4.0 OLE DB Provider.

ADO DATA CONTROL & DATA GRID CONTROL

  - 3. On Data Link Properties dialog's Provider tab, select Microsoft Jet 4.00 OLE DB Provider.
  - 4. Click Data Link Properties dialog's Connection tab
    - A. Select or enter a database name.
    - B. Can select the database to which ADO Data Control will connect.
    - C. Press the button at right side of Textbox to display Select Access Database dialog.
  - 5. Locate database file; Open; select or enter a database name in Textbox; OK–Data Link Property.
  - 6. In Property Pages dialog's RecordSource tab, select 2-adCmdtable from the CommandType comboBox to specifiy a table in database will be source of data

ADO DATA CONTROL & DATA GRID CONTROL

    - Select desired TableName from Table or Stored Procedure Name comboBox to specify data will be retrieved specifically from desired table in database.
  - Note: Properties can be set using code in program.
  - 7. Click Test Connection button to determine if connection succeeded.
    - If connection is successful, a dialog box appears with message "Test connection succeeded" .
- DataGrid
  - 1. Left-arrow and right-arrow keys used to move from field to field.
  - 2. Up-arrow and down-arrow keys used to move from record to record.

ADO DATA CONTROL & DATA GRID CONTROL

  - 3. Page-Up and Page-Down keys used to jump through a page of records.
  - 4. Vertical scrollbar allows movement up and down records.
  - 5. Resize columns by positioning cursor between columns, drag to size.
  - 6. Resize rows by positioning cursor horizontal bar between rows.
  - 7. Navigation buttons to move between records.
- If AllowUpDate is set to True for DataGrid, modifying data in a cell causes data in database to change when another record becomes current record.

STRUCTURED QUERY LANGUAGE

- Once connection string is specified, set properties for controls via property sheet (Figure 18.6).
- **Structured Query Language(SQL)**
- Keywords listed in Figure 18.13)
  - SELECT- retrieve fields from a table or several tables.
  - WHERE- criteria for selection that determines rows to be retrieved.
  - FROM- Tables from which to get fields. Required in every SELECT.
  - GROUP BY- How to group records.

STRUCTURED QUERY LANGUAGE

  - GROUP BY- How to group records.
  - ORDER BY- Criteria for sorting records.
- **Basic SELECT Query** Format:
  - SELECT * FROM TableName
    - Asterisk indicates that all fields from TableName should be selected and TableName specifies table in database from which the fields will be selected.
    - Example: SELECT * FROM Company
    - Example: SELECT [Ticker Symbol] FROM Company
- **WHERE Clause**: only records that match selection criteria are equally selected.
- SQL uses optional WHERE clause to specify selection criteria for query.

STRUCTURED QUERY LANGUAGE

  - Format: SELECT * FROM TableName WHERE criteria
  - Example: SELECT * FROM Company WHERE [Portfolio Percent] > .006
- Note: Square braces are required when field name has a space.
- Conditions: <,>,>=,<=, =,<>, and Like
- Operator **Like** is used for pattern matching with wildcard characters * and ?
- Asterisk in pattern indicates any number of characters in a row at asterisk's location in pattern.

STRUCTURED QUERY LANGUAGE

  - Example: SELECT * FROM Company WHERE [Industry Description] Like 'A*'
  - Example: Like 'A?L*' or Like 'A[B-D]*'
- **ORDER BY Clause** sorts by ascending(ASC) or descending(DESC) order. Format:
  - SELECT * FROM TableName ORDER BY field ASC
  - SELECT * FROM TabeName ORDER BY field DESC
  - Multiple sort fields: ORDER BY field1 SortingOrder, field 2 SortingOrder
  - Example: SELECT * FROM Company WHERE [Industry Description] Like 'A*' ORDER BY [Ticker Symbol] ASC

STRUCTURED QUERY LANGUAGE

- Queries created for VB are usually written as one long string containing the entire query.
- Using **INNER JOIN** to merge data from multiple tables.
  - SELECT * FROM Table1 INNER JOIN Table2 ON Table1.field = Table2.field
  - Example: SELECT [Industry Code], [Industry Description] FROM Company INNER JOIN [Stock Info] ON Company.[Ticker Symbol] = [Stock Info].[Ticker Symbol] ORDER BY company ASC
- TableName.FieldName specifies fields from table that should be compared to join the tables.

STRUCTURED QUERY LANGUAGE

  - Example: SELECT [Industry Code], [Industry Description] FROM Company INNER JOIN [Stock Info] ON Company.[Ticker Symbol] = [Stock Info].[Ticker Symbol] ORDER BY company DESC
- The same syntax can be used in a query anytime it is necessary to distinguish between fields in different tables that happen to have the same name.
- Revisiting ADO Data Control and DataGrid Control
- Figure 18.14 describes some key properties, methods, and events for the ADO Data Control

ADO DATA CONTROL & DATAGRID CONTROL

- Key feature ADO Data Control is that SQL statements can be specified for the RecordSource property at execution time.
- Program of Figure 18.15 allows the users to type an SQL query into a textbox and display the results of query in a DataGrid control.
- Recordset produced via an SQL query of database can limit the fields that are actually displayed in the DataGrid.
- **Hierarchical FlexGrid Control**
  - Similar to DataGrid control.
  - User can only view data through a FlexGrid.

HIERARCHICAL FLEXGRID CONTROL

- Example Figure 18.17
- Properties Figure 18.18
- Requires ADO Data Control and Hierarchical FlexGrid Control.
- ADO Data Control has same settings as Figure 18.5.
- DataList and DataCombo Controls are capable of being able to accept database records via Recordset.
- Operate like ComboBox and ListBox but contain properties suitable for database use.
  - The properties are listed in Figure 18.19.
  - Figure 18.21 contains property settings for controls in program of Figure 18.20.