CSIS 4135

Intro to RDBMS & SQL
Accessing a database from ASP.NET

Databases

- Virtually all interesting applications require a structured, persistent data store
- Examples:
  - E-Commerce: placing an order, fulfilling an order
  - Discussion forums
- Database needs vary with the type of application

What Is a Database?

Files optimized to store data in a structured way

- Organization helps to ensure that data is accurate

Database files are managed by software to

- Make it accessible to applications and users
  - e.g. concurrent access
- Manage and maintain the data

Database Requirements

- Store, view, and modify data
- Move, copy, and transform data
- Backup and restore data
- Enforces data integrity rules
  - Not corrupt, not redundant, up-to-date
- Scalable and available
  - High number of users
  - Lots of data
  - High throughput with low response time
- Secure
- Facilitates application development

Evolution of Database Technology

- File-based
- Hierarchical
- Network
- Relational (RDBMS) ⇐ Most everyone uses this
- Object-oriented ⇐ Most programmers want this
- XML
- NoSQL

What Is a Relational Database?

- The most often used database management system
- Helps maintain data integrity by
  - Ensuring data type correctness
  - Eliminating redundant data that could lead to inaccurate versions of an entity
  - Eliminating unwanted deletion of related data
  - Optimizing efficiency of data retrieval and data manipulation

Relational Databases (Tables)

- Table (relation, entity)
  - Organized in columns
- Column (attribute, field)
  - Describes part of an entity (e.g. FirstName)
  - Has a data type (e.g. integer, character)
  - Can be null
- Row (tuple, record)
  - A single instance of data in a table (values for each column)
  - Each row is unique

AuthID	FirstName	LastName
1	Joe	Smith
2	Diane	Jones

Column Properties

Used to maintain data integrity

- Correct type
- Ensure data is in appropriate range of values
- Don't allow missing data
- Special constraints and rules
  - e.g. referential integrity
- Etc.

Data Types

Many available in SQL Server, including

SQL Data Type	.NET Data Type
date	System.DateTime
datetime	System.DateTime
float	System.Double
int	System.Int32
varchar	System.String
money	System.Decimal?

Transact-SQL Data Types

Relating Data

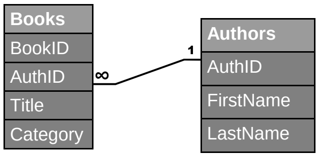
Tables can be related through primary/foreign key relationships (e.g., a book has an author)

- Primary key
  - Guarantees the uniqueness of a row (ensures entity integrity)
  - Can be composed of one or more columns
- Foreign key
  - Establishes logical relationship between tables
  - One or more columns of another table that match the primary or alternate key of a table
  - Referential integrity

Relational Databases

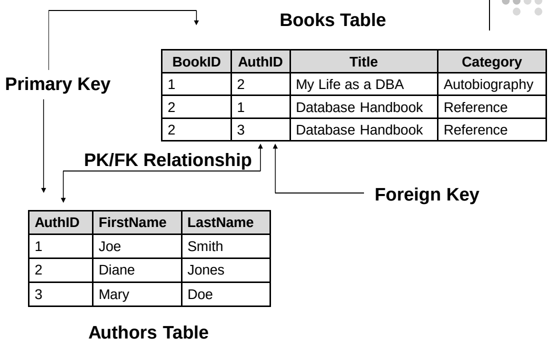
Relating Data

Schema diagram depicts tables, columns, primary keys, foreign keys



Schema Diagram

Relating Data



Books Table

BookID	AuthID	Title	Category
1	2	My Life as a DBA	Autobiography
2	1	Database Handbook	Reference
2	3	Database Handbook	Reference

Primary Key

PK/FK Relationship

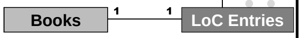
Foreign Key

AuthID	FirstName	LastName
1	Joe	Smith
2	Diane	Jones
3	Mary	Doe

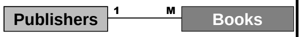
Authors Table

Types of Relationships

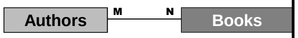
- One-to-One (1:1)



  - One row in table X matches one row in table Y
  - A book has at most one Library of Congress entry
- One-to-Many (1:M)



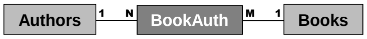
  - One row in table X matches 0+ rows in table Y
  - A publisher publishes one or more books
- Many-to-Many (M:N)



  - 1+ rows in table X matches 1+ rows in table Y
  - An author writes one or more books;
    A book is written by one or more authors

Many-to-many Relationships

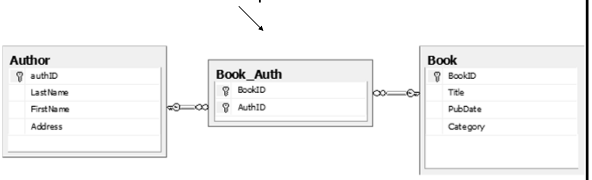
- More complex
- Can result in very large tables (repeated data)
- Difficult to ensure data integrity
- The remedy: Create a third table
  - The third table contains the primary key of the two original tables in a composite key
  - Data is repeated in the third table, but not in the two original tables



Relational Databases

Many-to-many Relationships

Data is duplicated here



Normalization/Denormalization

Normalization

- Breaking large tables into multiple smaller tables
- Goal: minimize redundant data, maximize correctness
- Improves performance for updates
- Desirable in transaction-based applications

Denormalization

- Combining smaller tables into fewer larger tables
- Goal: improve performance
- Introduces redundant data
- Improves performance for reads
- Desirable in data warehouse applications

Structured Query Language (SQL)

- Standard language for accessing a relational database - American National Standards Institute (ANSI); SQL-92
- Open, but not really
  - Common functions are mostly the same across products
  - Most vendors have proprietary extensions
- Subsets of SQL
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Control Language (DCL)

Relational Databases

DDL Examples

Used to create and modify database objects

```
CREATE DATABASE Bookstore
```

```
CREATE TABLE Books
(
   BookID        INT IDENTITY(1,1) PRIMARY KEY,
   AuthID        INT REFERENCES Authors,
   Title         VARCHAR(30) NOT NULL,
   PubDate       DATETIME NOT NULL,
   Description   VARCHAR(50),
   Category      INT NOT NULL
)
```

DML Examples

Select data to view

```
SELECT * FROM Authors
```

```
SELECT AuthID, FirstName, LastName
   FROM Authors
```

```
SELECT AuthID, FirstName, LastName, Address
   FROM Authors
   WHERE City = 'Boston'
```

```
SELECT FirstName, LastName, Address
   FROM Authors
   WHERE AuthID = 249
```

Joins

A join is a way of combining data from multiple tables, usually by resolving primary key/foreign key relationships

DML Examples

Using SELECT to join tables

```
SELECT Authors.AuthID, FirstName,
       LastName, Title, PubDate, Category
   FROM Authors
   INNER JOIN Books
   ON Authors.AuthID = Books.AuthID
```

DML Examples

Insert, update and delete data

```
INSERT INTO Books
   (Title, PubDate, Description, Category)
VALUES
   ('Database Design', GETDATE(),
    'How to design a database', 3)
```

```
UPDATE Authors
   SET Phone = '617-555-1234'
   WHERE AuthID = 5
```

```
DELETE FROM Authors
   WHERE AuthID = 5
```

DCL Examples

Set security options on database objects

```
GRANT INSERT, UPDATE, DELETE
   ON Authors
   TO Mary, John
```

```
REVOKE CREATE TABLE FROM Joe
```

```
DENY ALL
   ON Authors, Books
   TO Sally
```

Visual Studio Database Tools

Supports most typical database operations

- Attaching an existing data base file
- Creating databases
- Creating relationships between tables
- Querying
- Programming

App_Data Folder

This is where a project can keep its own database files

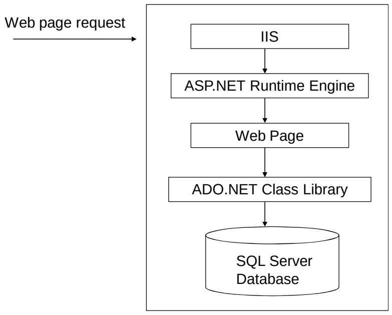
- Attach existing database file (.mdf) by drag-n-drop
- Add new item to create a new database
  - Define columns including types, primary key, default values, identity (sequential numeric values generated automatically)

Database Driven Web Sites

Most dynamic web sites

- Retrieve data from a database and present it on a web page for viewing, editing, etc.
- Collect data from web pages and insert into a database

Single Server Scenario



ADO.NET Library

This is the bridge between your code and the database

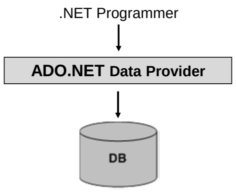
Handles

- Handshake with the database (connection)
- Execution of commands (insert, delete, update)
- Execution of queries (retrieve info, filtering, etc.)

ADO.NET provides an object oriented view of CRUD operations (Create, Read, Update, Delete)

Architecture

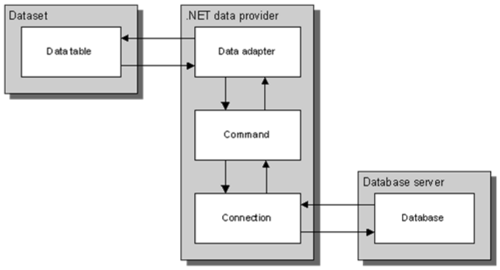
ADO.NET architecture uses data providers to encapsulate DB-specific implementation details



.NET ships with data providers for Microsoft SQL Server and others, like Oracle

Basic ADO.NET Objects

Disconnected Data Architecture



How to use datasets

A *dataset* contains one or more *data tables* that store the data retrieved from the database.

To retrieve data from a database and store it in a data table:

1. a *data adapter* object issues a Select statement that's stored in a *command* object
2. the command object uses a *connection* object to connect to the database and retrieve the data
3. the data is passed back to the data adapter, which stores it in the dataset

How to use datasets (continued)

To update the data in a database based on the data in a data table:

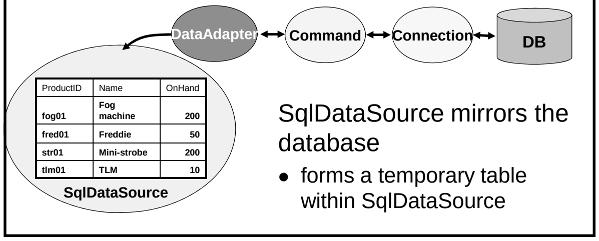
1. the data adapter object issues an Insert, Update, or Delete statement that's stored in a command object
2. the command object uses a connection to connect to the database and update the data

- When you use a data adapter, the data provider remains connected to the database only long enough to retrieve or update the specified data and then it disconnects from the database. This is referred to as a *disconnected data architecture.*

SqlDataSource

SqlDataSource is an in-memory data structure

- easily filled with data from a database
- easily displayed in a GUI app



SqlDataSource mirrors the database

- forms a temporary table within SqlDataSource

Why use a SqlDataSource?

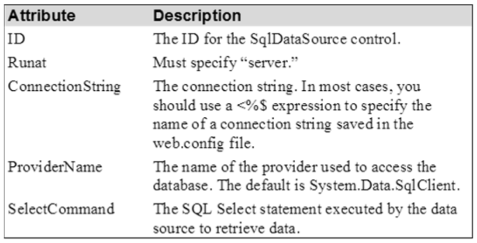
- SqlDataSource is filled with a local copy of requested data
- This allows you to:
  - reduce load on DB server since data is now available locally
  - close connection to DB & work off-line
- These are important advantages in a networked environment…

SqlDataSource is really an in-memory database

A SqlDataSource:

- holds as much data as you want
- is searchable
- is sortable
- is updateable

Basic SqlDataSource control attributes



Attribute	Description
ID	The ID for the SqlDataSource control.
Runat	Must specify "server."
ConnectionString	The connection string. In most cases, you should use a <%$ expression to specify the name of a connection string saved in the web.config file.
ProviderName	The name of the provider used to access the database. The default is System.Data.SqlClient.
SelectCommand	The SQL Select statement executed by the data source to retrieve data.

Connection Strings

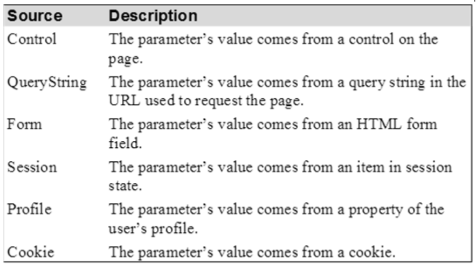
- A connection string is used to specify the server, database, and login details
- This is usually stored in the `web.config` file for the application
- The SQLDataSource references this when it needs to make a connection

Elements used to define select parameters

A SQLDataSource needs at least a SELECT statement that specifies how to retrieve data from the database

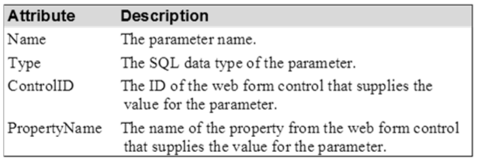
- SelectParameters
- ControlParameter
- QueryStringParameter
- FormParameter
- CookieParameter

Parameter sources



Source	Description
Control	The parameter's value comes from a control on the page.
QueryString	The parameter's value comes from a query string in the URL used to request the page.
Form	The parameter's value comes from an HTML form field.
Session	The parameter's value comes from an item in session state.
Profile	The parameter's value comes from a property of the user's profile.
Cookie	The parameter's value comes from a cookie.

The ControlParameter element



Attribute	Description
Name	The parameter name.
Type	The SQL data type of the parameter.
ControlID	The ID of the web form control that supplies the value for the parameter.
PropertyName	The name of the property from the web form control that supplies the value for the parameter.

Other DML Statements

- The advanced options for a SQLDataSource can be used to define DELETE and UPDATE commands

The GridView Control

A powerful control for formatting, displaying and updating the data in a data source

- Full-featured list output
- Default look is a grid
- Can specify a subset of columns to display
- Columns can be formatted with templates
- Optional paging, sorting, etc.