

# C#程序设计及应用

唐大仕

dstang2000@263.net

北京大学

Copyright © by ARTCOM PT All rights reserved.



# 第13章 数据访问

唐大仕

[dstang2000@263.net](mailto:dstang2000@263.net)

<http://www.dstang.com>



# 本讲内容

- 数据库基础知识
- ADO.NET概述
- 访问数据库的两种基本方法
- 数据库应用程序

# 本讲内容



功能

ADO. NET概述

功能

功能

Connect ion和Command

功能

功能

DataReader

功能

功能

DataAdapter和DataSet



## I. ADO.NET概述



# 数据库访问







- 数据库(Database)
- 数据库管理系统( DBMS )
  - 如Oracle, MS SQL Server
- 数据库系统的优点：
  - 共享性、独立性、完整性和冗余数据少
- 管理功能
  - 数据定义/操纵/完整/安全/并发



# 常用的数据库管理系统

- 微软的
  - Microsoft SQL Server
  - Microsoft Access ( 小型桌面数据库 )
- 其他常用数据库
  - Oracle
  - MySql





- 关系型数据库(RDB)

- 以二维表的形式组织数据
- 表 DataTable
- 行 DataRow : 记录 Record
- 列 DataColumn: 字段 Field
- 主键: primary key





# 数据库表的设计

- 示例：

- 书籍表： 编号(ID), 书名, 作者, 出版社, 关键字
- 院系表： 编号(ID), 名称, 办公地址, 办公电话
- 读者表： 人号编号 ( ID), 姓名, 性别, 院系编号, email
  - ( 院系与读者是1对多的关系 )
- 借阅表： 人员编号, 书号, 日期
  - 读者与书籍是 多对多的关系
- 选课表： 人, 课, 成绩
  - 2501, A03, 80
  - 2501, A04, 83
  - 2502, A04, 87

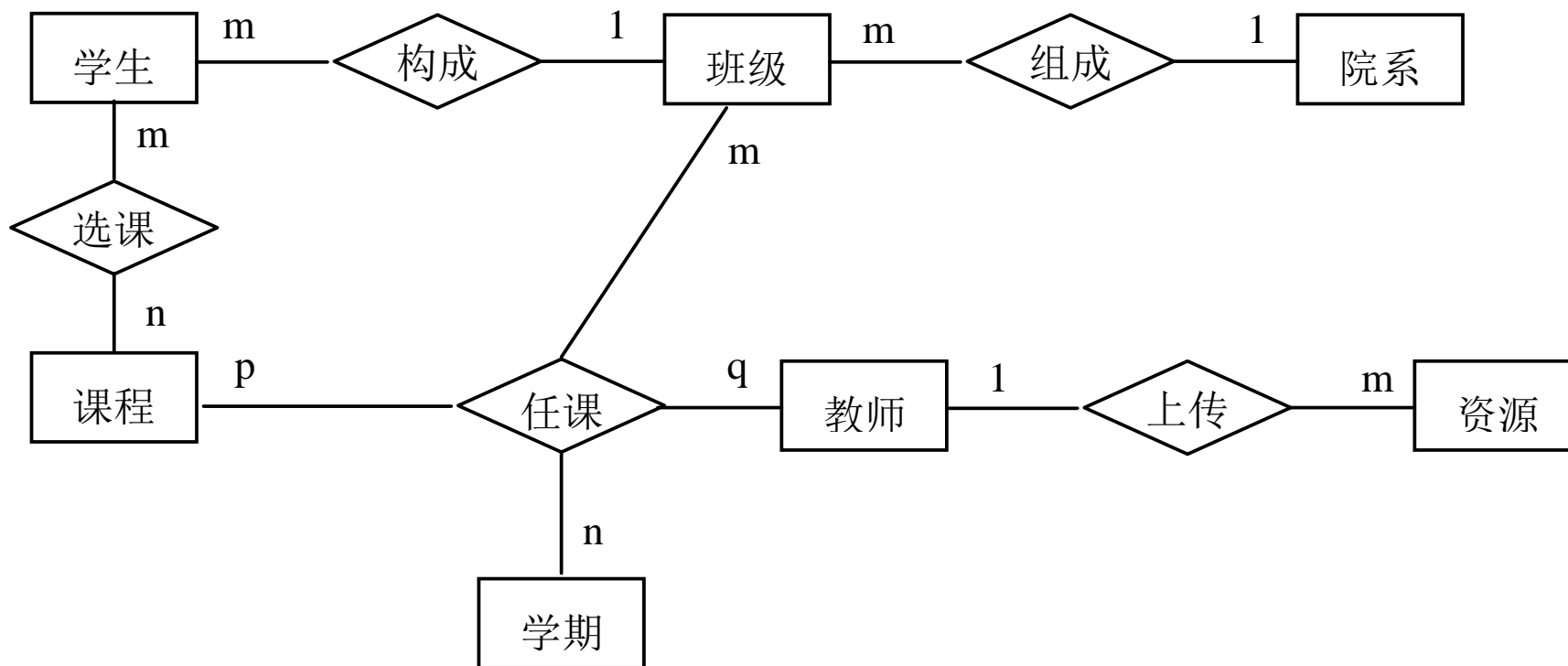


- 院系（院系编号，院系名称）
- 班级（班号，班级名，学生数，班主任）
- 学生（学号，登陆密码，姓名，性别，状态，入校时间，毕业时间，退学时间，注册时间，最后登陆时间，登陆次数）  
这里“状态”有3种取值：在读、退学、毕业
- 教师（教师号，登陆密码，姓名，电话，邮箱，QQ号，找回密码问题，找回密码答案，注册时间，最后登陆时间，登陆次数）
- 课程（课程号，课程名称，学分）
- 课程资源（资源号，资源名，链接地址，更新时间，资源类型，说明）
- 学期（学期号，学期名称，是否当前学期，学期起始时间，学期结束时间）

# 实体关系图 (E-R图)

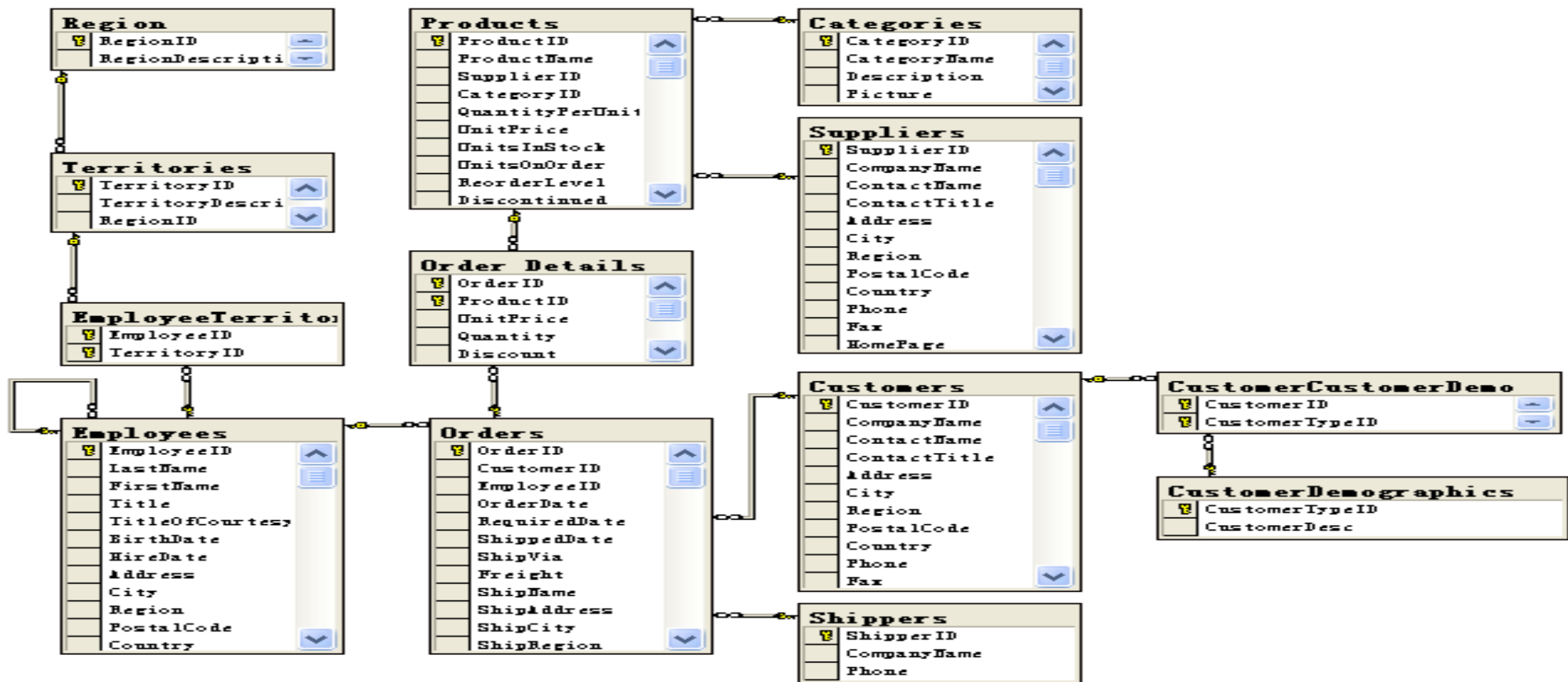


- 网上成绩查询管理系统的 E-R图 ( 实体-关系图 Entity-Relation)





# 实体关系图举例







- 用可视化的界面
  - 如：使用SQL Server企业管理器
- 通过SQL语句
  - 在管理界面中使用
  - 或者编程来使用





- SQL ( Structured Query Language ) , 即结构化查询语言
- 是数据库的标准操作语言
- SQL包含数据定义、数据查询、数据操纵和数据控制等多种功能
- 最常用的语句就是：增删改查
  - ▣ 英文中叫 CRUD——Create、Retrieve、Update、Delete



- `SELECT * FROM [publishers]`
- `SELECT name, age, sex, salary + bonus`  
`FROM employee`  
`WHERE depart= '销售部' and title= '经理'`  
`ORDER BY age DESC`
- `SELECT avg( salary ), max(age), sum(salary)`  
`FROM employee`  
`GROUP BY title`



# 常用的SQL的聚合函数

函数	返回数值
AVG（表达式）	返回表达式中所有值的平均值。仅用于数字列并自动忽略 null 值。
COUNT（表达式）	返回表达式中非 null 值的数量。可用于数字和字符列。
COUNT(*)	返回表中的行数（包括有 null 值的行）。
MAX（表达式）	返回表达式中的最大值，忽略 null 值。可用于数字、字符和日期时间列。
MIN（表达式）	返回表达式中最小值，忽略 null 值。可用于数字、字符和日期时间列。
SUM（表达式）	返回表达式中所有值的总和，忽略 null 值。仅用于数字列。



# SQL语句（续）

- 增加数据

```
INSERT INTO [employee] ( name, age ) VALUES ( '李明', 18)
```

- 更新数据

```
UPDATE [employee] SET salary = salary +500
```

- 删除数据

```
DELETE FROM [employee] WHERE age>80
```

- 创建及删除数据表

```
CREATE TABLE [employee]  
    ( id integer, name char(10), age integer )  
DROP TABLE [employee]
```



# 通过编程来操作数据库

- Client/Server模式 (客户/服务模式)
  - 数据库提供服务，称为Server
  - 程序是客户端，称为Client
  - 客户端通过SQL命令来向服务端发出请求
- 编程的任务
  - 通过用户界面获取参数，并形成SQL语句
  - 建立与数据库的连接
  - 向数据库管理系统发出命令 ( SQL )
  - 从数据库得到数据并显示到用户界面



# 数据库访问







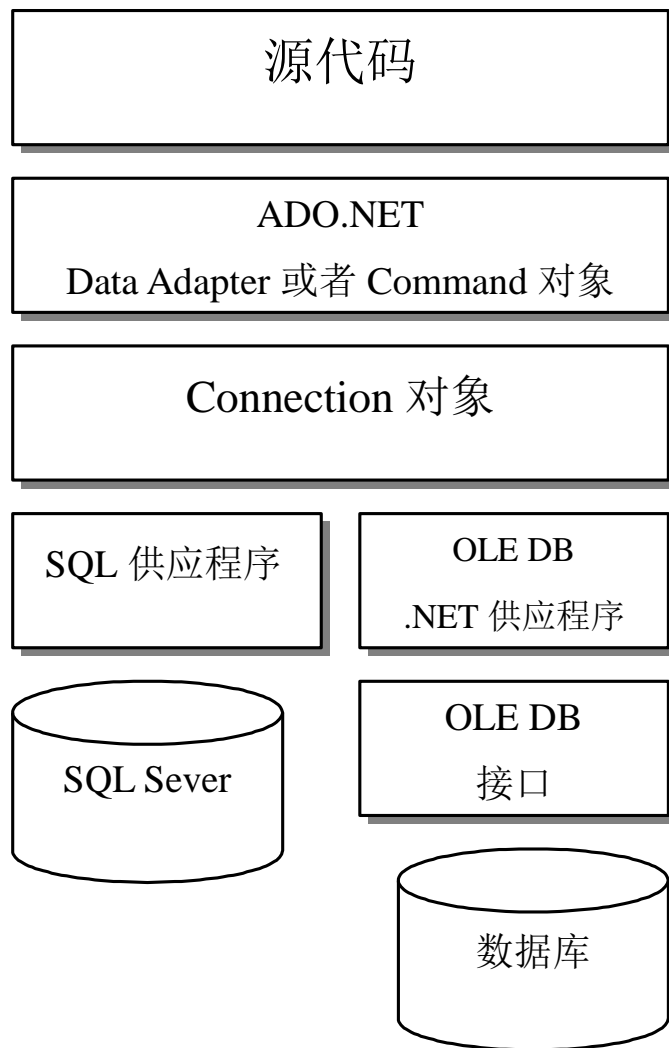
- System.Data名称空间
- 这种访问数据库的技术叫ADO.NET
- ADO.NET实现数据库的访问
  - 提供标准的CRUD接口
  - 对不同的数据库提供统一的访问接口

# ADO.NET技术的发展



- ODBC ( Open Database Connection)
- DAO ( Data Access Object )
- ADO ( ActiveX Data Object )
- ADO.NET

# ADO.NET中的数据访问层



# ADO.NET的四个重要对象

对象	说明
Connection	建立与特定数据源的连接。
Command	对数据源执行命令。公开 Parameters，并且可以从 Connection 在 Transaction 的范围内执行。
DataReader	从数据源中读取只进且只读的数据流。
DataAdapter	用数据源填充 DataSet 并解析更新。



# 供应程序Provider

- Provider是一些托管的应用程序集
  - ▣ 包含了对特定数据元的访问代码
- SQL provider ( 访问SqlServer)
  - ▣ System.Data.**SqlClient**名称空间
- OLE/DB provider ( 访问Access等)
  - ▣ System.Data.**OleDb**
- ODBC provider ( 访问ODBC )
  - ▣ System.Data.**Odbc**
- Oracle provider ( 访问Oracle )
  - ▣ System.Data.**OracleClient**

# 不同Provider下的类也不同

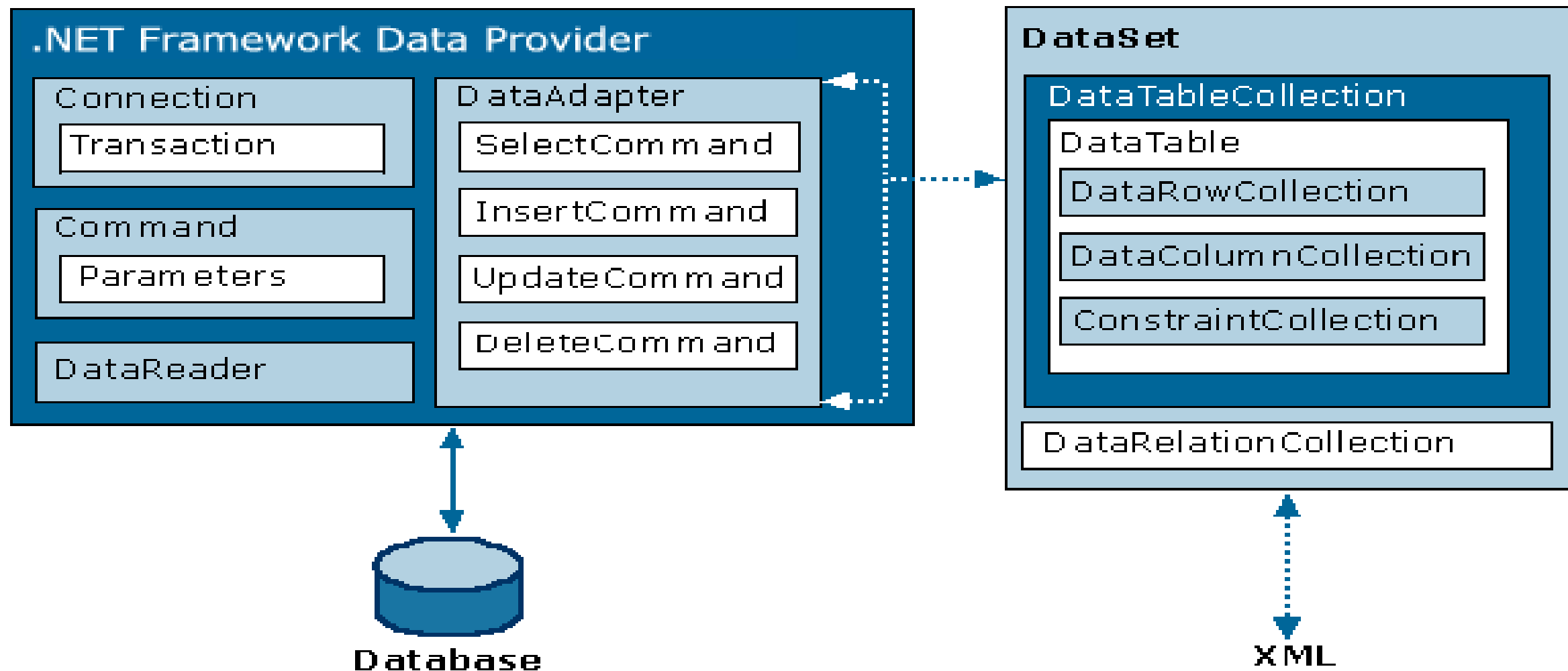


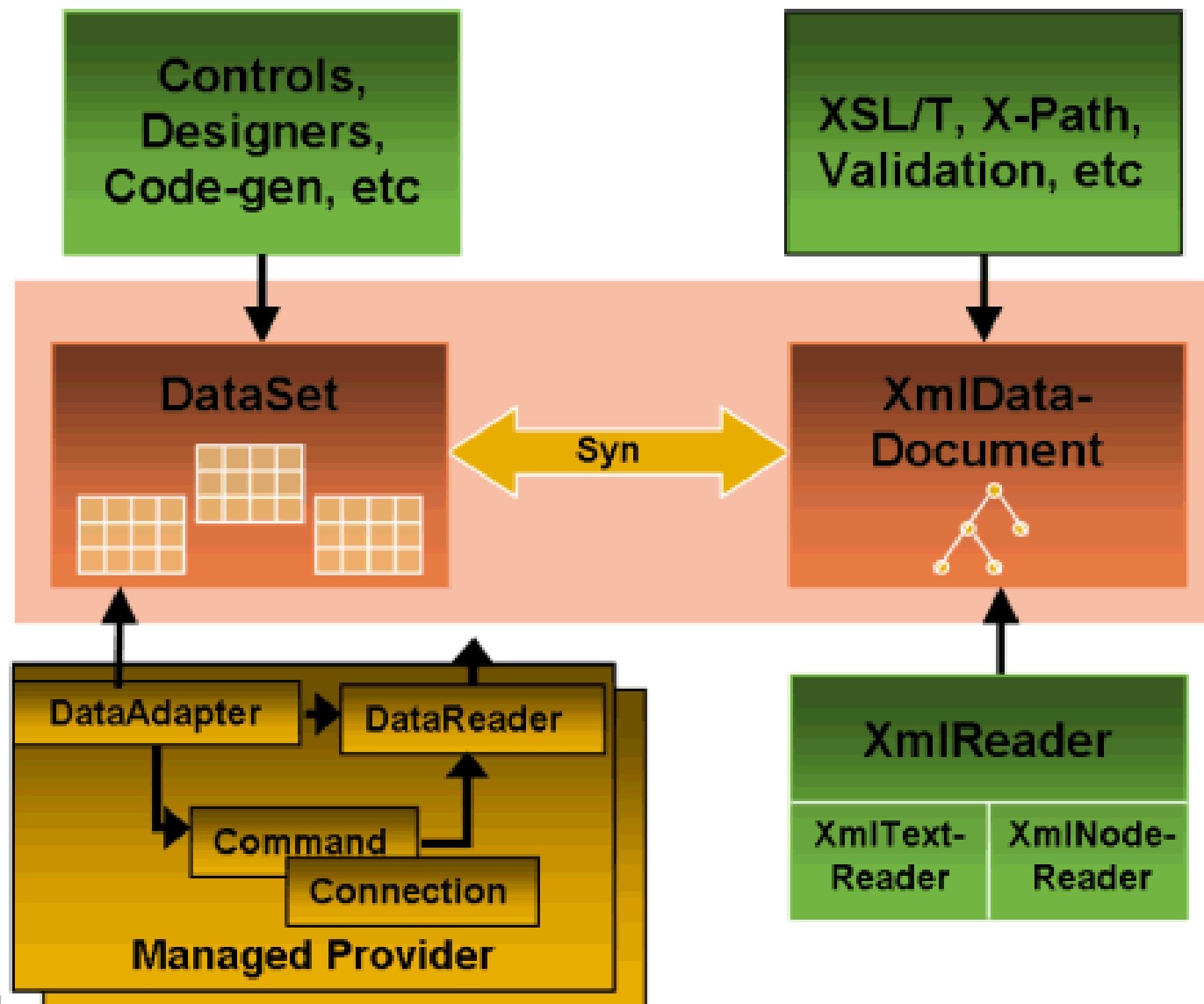
对象	OLE DB供应程序 System.Data.OleDb名字空间	SQL供应程序 System.Data.SqlClient名字空间
DataAdapter	System.Data.OleDb.OleDbDataAdapter	System.Data.SqlClient.SqlDataAdapter
Connection	System.Data.OleDb.OleDbConnection	System.Data.SqlClient.SqlConnection
Command	System.Data.OleDb.OleDbCommand	System.Data.SqlClient.SqlCommand
DataReader	System.Data.OleDb.OleDbDataReader	System.Data.SqlClient.SqlDataReader





# ADO.NET体系结构







# 数据库访问





# 访问数据的两种基本方式

- 方式1：DataAdapter及DataSet
  - 适合于“离线”处理
  - 自动建立Command对象，
- 方式2：DataReader
  - 适合于只读数据，效率较高
- 它们都要使用Connection及Command



- 对于不同的Data Provider有不同Connection对象
- System.Data.SqlClient
- SqlConnection
  - Open()方法——SQL Provider使用它来与SQL Server 数据库进行连接
  - ConnectionString属性——指定连接字符串参数

# SQL Server导入的命名空间



- 语法如下：
- `using System.Data;`
- `using System.Data.SqlClient;`
- 连接SQL Server的连接串的书写格式
- `string connString="server=localhost;database=pubs;uid=sa;pwd=aaa";`
- `SqlConnection Conn = new SqlConnection(connString);`





(1) 连接**Sql Server**数据库，使用**SqlServer Provider**

- data source=MyServer;initial catalog=MyDataBase;  
user id=MyUser;password=MyPassword

(2) 连接早期的**Access**数据库mdb文件或Excel文件，使用**Microsoft.Jet.OLEDB.4.0**

Provider=Microsoft.Jet.OLEDB.4.0;Password="xxx";User ID=Admin;  
Data Source=D:\CsExample\ch10\BIBLIO.MDB

(3) 连接新版的**Access**数据库accdb文件或**Excel**文件，使用Microsoft.ACE.OLEDB.12.0

Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\myFile.xlsx;Excel 12.0 Xml;HDR=YES;

(4) 连接**Oracle**数据库，使用**Oracle Provider**

Provider=MSDAORA;;DataSource=oracle\_db;User ID=scott;Password=tiger

(5) 连接**Mysql**数据库，使用MySql.Data.dll

Data Source='localhost'; Database='wp'; User Id='root';Password='root';charset='utf8';pooling=true



# 使用Command对象

- 建立数据连接以后，可以利用Command对象来执行命令并从数据源返回结果。
  - ExecuteReader()
    - 得到Reader对象，单向只读
  - ExecuteScalar()
    - 得到单一的量，如sum,avg等的结果
  - ExecuteNonQuery()
    - 执行非查询性的命令（如Update/Delete/Insert）
    - 返回的是它所影响的记录数



```
cmd.CommandText = "INSERT INTO Nations(CName, EName, FName) VALUES  
(@CName, @EName, @FName);  
cmd.Connection = this.sqlConnection1;  
cmd.Parameters.Add("@CName", SqlDbType.VarChar, 60).Value = "aaaaaa";
```

使用参数比直接用字符串相加更安全，更清晰

```
cmd.CommandText = "Insert into Nataions(CName) Values(' " + name + " ' )";
```



# Command和DataReader

- 1)使用Command来获取DataReader
- 2)使用Command来获取一个数据
- 3)直接使用数据库命令



# DataReader对象

- ADO.NET中最高效的得到数据的方法
- SqlDataReader
  - SqlCommand.ExecuteReader()方法
  - SqlDataReader.Read()方法可以遍历结果集中的行
  - SqlDataReader按照当前行的列索引可得到每一列的数据，GetString()，GetBoolean()，GetInt16()，GetInt32()，GetDateTime()，GetDecimal()，GetDouble()，GetFloat()，GetGuid()

# 关键代码示例



- CommandDataReader.cs





# DataAdapter和DataSet

- 1)使用DataAdapter来填充DataSet
- 2)取得DataSet中的数据
- 3)修改DataSet中的数据
- 4)添加和删除行
- 5)在DataTable中查找数据
- 6)接受和拒绝更改
- 7)保存对DataSet的改变返回数据库



# 可以自动产生Command

- `OleDbCommandBuilder cmdbld = new OleDbCommandBuilder( daAdapter );`

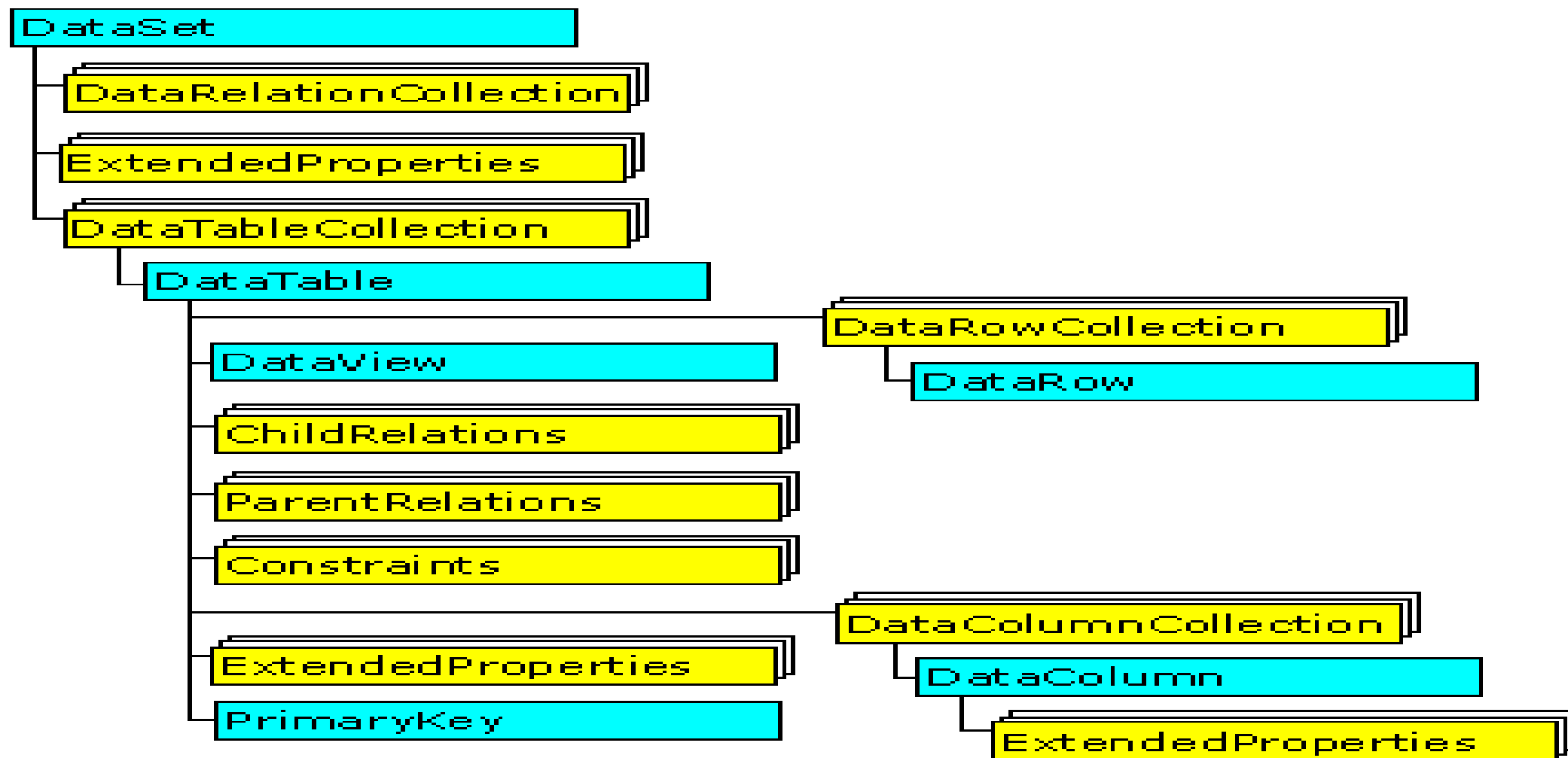


# DataSet及相关对象

- DataSet
- DataTable
- DataRow
- DataColumn
- 表之间的联系 Relation
- 约束



# DataSet对象模型





- DataSet可以直接的序列化为XML，同时XML也可以直接的反序列化为DataSet对象
  - ReadXml()方法
  - WriteXml()方法
- SqlCommand.ExecuteXmlReader()返回一个XmlReader对象

# 关键代码示例



- AdapterDataSet.cs





# 数据库访问





# 建立自己的实用库

- 建立实用工具类
  - AccessDB.cs
  - SqlServerDB.cs
- 主要有两个方法
  - GetDataSet(sql)
  - ExecuteNonQuery(sql)



# 界面及数据绑定

- 使用DataGridView
- 其.DataSource=myDataTable;



- CrudDemo.rar
  - CRUD (Create, Retrieve, Update, Delete)
  - 增删改查
- 界面与数据库的关系
  - 界面—sql语句—数据库



# 应用示例：酒店管理系统

- 分层架构
  - UI: 用户界面层 ( user graphical interface)
  - BLL: 业务逻辑层 ( business logic layer )
  - DAL: 数据访问层 ( data access layer )
  - Model: 数据模型
- ORM Mapping (Object-Relation Mapping)
  - 对象-关系 ( 数据库表 ) 映射



# 问题与讨论

[dstang2000@263.net](mailto:dstang2000@263.net)