

# Datenbanken

Proseminar *Objektorientiertes Programmieren mit .NET und C#*

Sebastian Pinteä

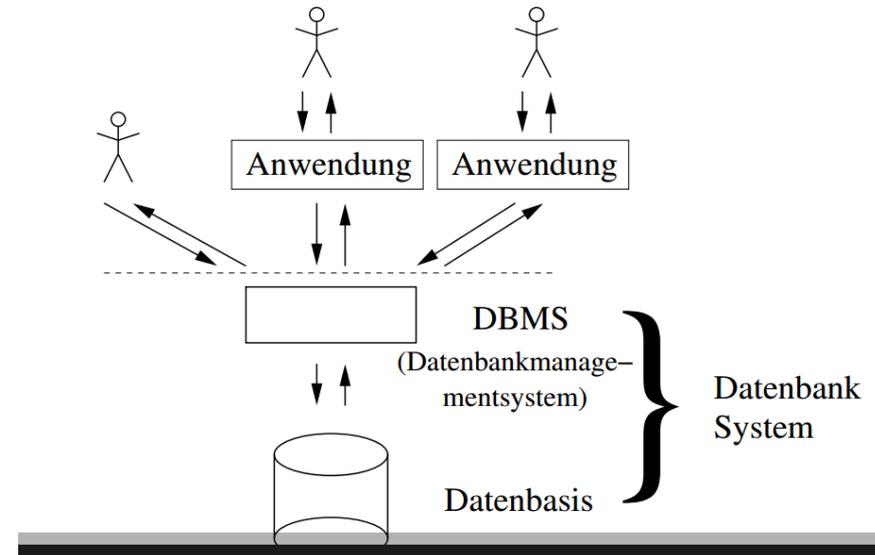
Institut für Informatik  
Software & Systems Engineering

# Agenda

- 1. Datenbanken
- 2. SQL
- 3. ADO.NET
  - DataProvider (providerabhängig)
    - Connection object
    - Command object
    - DataAdapter
    - DataReader
  - DataSet (providerunabhängig)
    - Data Set
    - Data Table
    - Data Row
    - Data Column
    - DataView
    - DataRelation
    - DataGrid

# 1. Datenbanken

- Was sind Datenbanken (DBS)?
  - Systeme zum Speichern und Verwalten von Daten
  - Besteht aus DBMS und der eigentlichen DB
- Warum kein normales Dateisystem?
  - Ausfallsicherheit
  - Skalierbarkeit



# 1. Datenbanken

- Wieso verwendet man Datenbanken?
  - Große Datenmengen, wertvolle Daten
  - Datenunabhängigkeit
  - Mehrbenutzersynchronisation
  - Fehlerbehandlung
  - Deklarative Anfragesprachen
  - Effizienz und Skalierbarkeit
  - Datenintegrität
- Probleme ohne DBS
  - Redundanz und Inkonsistenz
  - Verschiedene Datenformate
  - Probleme für Mehrbenutzerbetrieb
  - Verlust von Daten
  - Sicherheitsprobleme

# 1. Datenbanken

- Wo verwendet man Datenbanken?
  - Buchhaltung
  - Verwaltung
  - Geschäftsdaten

aber auch sehr verbreitet in

- Websuche
- Informationsintegration
- Data Mining
- Wissenschaftliche oder medizinische Daten

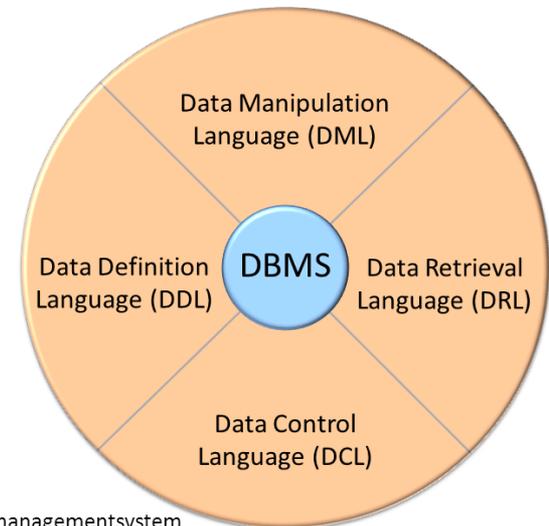
# 1. Datenbanken

- Beispiele
  - Oracle
  - IBM DB2
  - Microsoft SQL Server
  
  - PostgreSQL
  - MySQL
  - MonetDB
  
  - Noch viele weitere, teils stark spezialisierte Systeme.

## 2. SQL

- SQL: Structured Query Language
- Früherer Name war SEQUEL
- Standardisierte Anfragesprache fuer relationale DBMS: SQL-89,
- SQL-92, SQL-99
- SQL ist eine deklarative Anfragesprache
  
- Vier große Teile:
  - DML : Data Manipulation Language
  - DDL: Data Definition Language
  - DCL: Data Control Language
  - DRL : Data Retrieval Language

### Structured Query Language (SQL)



DBMS =  
Datenbankmanagementsystem

## 2. SQL

- DML : Data Manipulation Language
  - Enthält Befelhe um
    - Daten einzufügen (insert)
    - Daten zu löschen (delete)
    - Daten zu ändern (update)
  
- DDL: Data Definition Language
  - Definiert das Schema einer DB
  - Enthält Befehle, um den Zugriff auf Daten zu kontrollieren

## 2. SQL

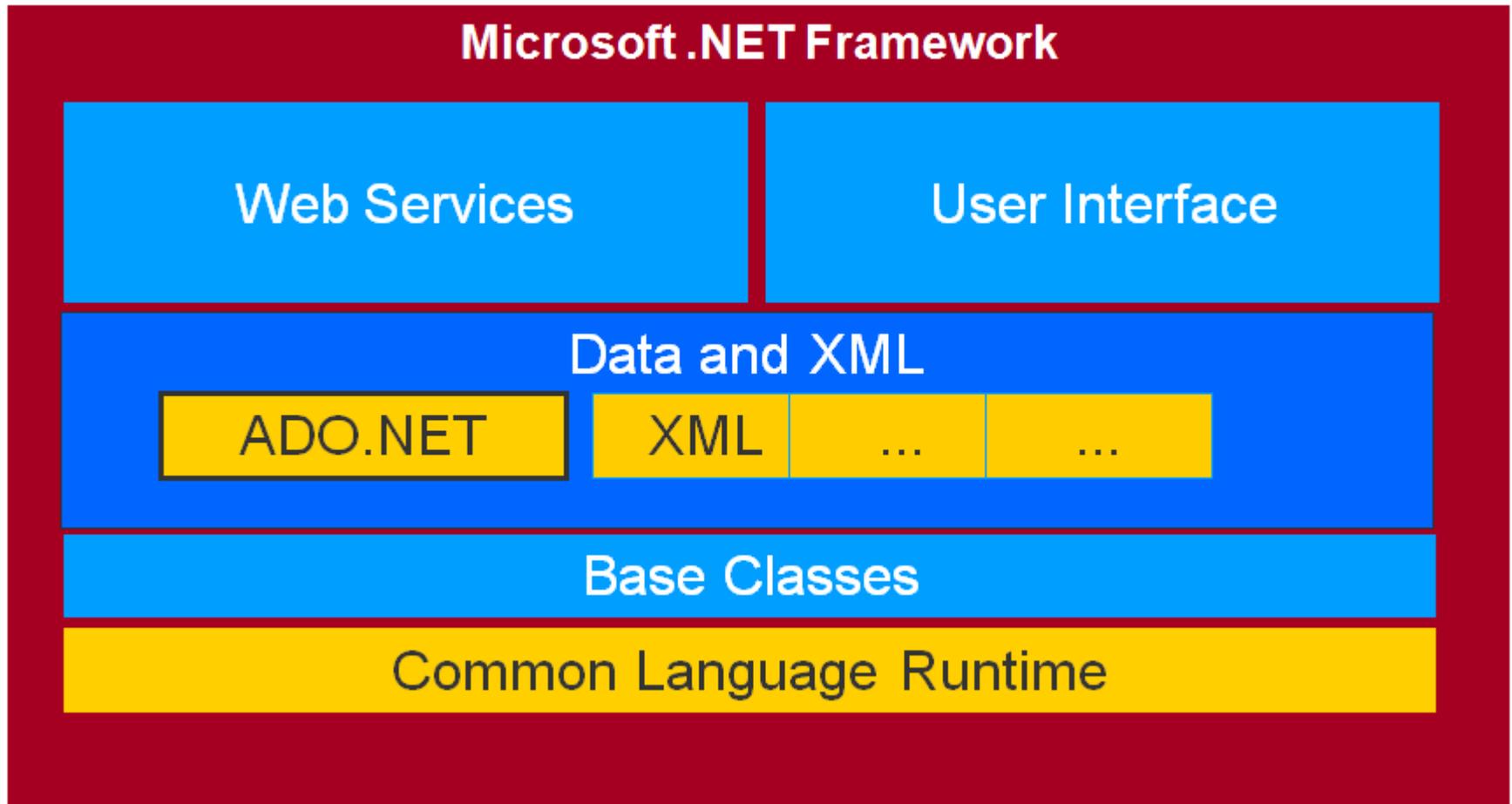
- DCL: Data Control Language
  - Datenüberwachungssprache einer DB
  - Befehle um Berechtigungen zu vergeben oder zu entziehen
  - Befehle um den Fluß von Transaktionen zu steuern
  
- DRL : Data Retrieval Language
  - Enthält Kommandos um Anfragen zu stellen
  - Besteht aus drei Klauseln
    - select
    - from
    - where

## 3. ADO.NET

### Definition

- „ADO.NET ist ein Teil der von Microsoft entwickelten .NET-Plattform. Es handelt sich um eine Sammlung von Klassen, die den Zugriff auf die relationale Datenbank gewährleisten.“
- „ADO.NET is ein objekt-orientierter Satz von Bibliotheken, welches erlaubt mit Datenquellen zu interagieren. Üblicherweise ist die Datenquelle eine Datenbank, aber sie könnte auch eine Text Datei sein, eine Exel Tabelle, oder eine XML Datei.“

# ADO.NET und die .NET Framework



## 3. ADO.NET

- DataProvider
  - Satz von Klassen, die Zugriff auf bestimmten Datenspeichertyp ermöglichen
  - Schnittstelle zur Datenbank
  - Besitzt Informationen über Datenbank

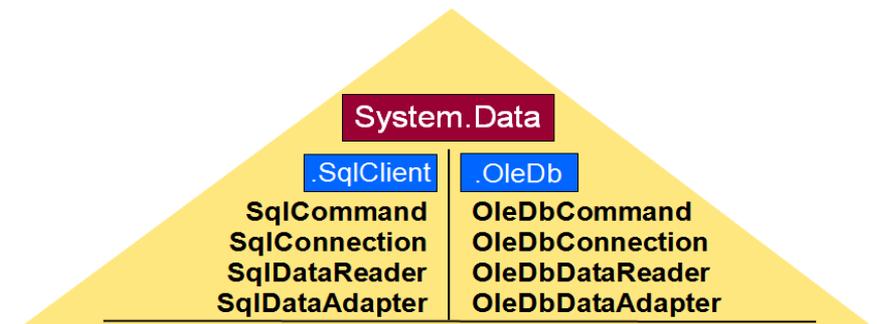
### Im .NET Framework enthaltene DataProvider

- MS SQL Server (System.Data.SqlClient)
- OLE DB (System.Data.OleDb)

### Basisklassen

- Command
- Connection
- DataReader
- DataAdapter

### .NET Data Providers Hierarchie



## 3. ADO.NET

- DataProvider
  - Die Connection Klasse (OleDbConnection, SqlConnection, OracleConnection, OdbcConnection) repräsentiert die Verbindung zur DB und besitzt Methoden zum Öffnen bzw. Schließen der DB.

```
using System.Data.SqlClient
```

```
SqlConnection sqlConn = new SqlConnection("Data Source=localhost; Initial  
Catalog=SQL2008; User Id=user;  
Password=password;");
```

```
sqlConn.Open();  
// Ausführen von Abfragen  
sqlConn.Close();
```

## 3. ADO.NET

- DataProvider
  - Die DataAdapter Klasse (OleDbDataAdapter, SqlDataAdapter, OracleDataAdapter, OdbcDataAdapter) stellt die Verbindung zwischen Datenverbindung und dem DataSet dar; beinhaltet Methoden zum Füllen des DataSets mit Daten und gleicht Aktualisierungen mit der Datenquelle ab.

```
String queryString = "select Name, MatrNr from Studenten";  
SqlDataAdapter adapter = new SqlDataAdapter(queryString, sqlConn);  
DataSet student = new DataSet();  
adapter.Fill(student, "Studenten");
```

## 3. ADO.NET

- DataProvider
  - Die Command Klasse (OleDbCommand, SqlCommand, OracleCommand, OdbcCommand) führt Anweisungen, gespeicherte Prozeduren und Aktionsabfragen aus wie z.B. SELECT-, UPDATE-, DELETE- Kommandos.

Wichtige Methoden:

- ExecuteReader: Führt Befehle aus, die Tabellenzeilen zurückliefern
- ExecuteNonQuery: Führt Befehle aus, die keine Tabellenzeilen zurückliefern
- ExecuteScalar: Führt Befehle aus, die einen einzelnen Wert zurückliefern

## 3. ADO.NET

- DataProvider
  - Command Beispiel:
    - ExecuteReader

```
SqlCommand cmd = new SqlCommand("select Name from Studenten", sqlConn);  
cmd.ExecuteReader();
```

- ExecuteNonQuery

```
string insertString = "insert into Studenten(Name, Nachname) values (` Max`, `Mustermann`)";  
SqlCommand cmd = new SqlCommand(insertString, sqlConn);  
cmd.ExecuteNonQuery();
```

- ExecuteScalar

```
SqlCommand cmd = new SqlCommand("select count(*) from Studenten, sqlConn);  
int count = (int)cmd.ExecuteScalar();
```

## 3. ADO.NET

- DataProvider
  - Die DataReader Klasse (OleDbDataReader, DataAdapter, OracleDataReader, OdbcDataReader) liefert einen Strom von Datenbankzeilen. Ein DataReader kann aus einem Command Objekt erzeugt werden und benötigt eine offene Verbindung.
    - Vorwärts gerichtet
    - Nur lesenden Zugriff
    - Es kann nur einmal gelesen werden

## 3. ADO.NET

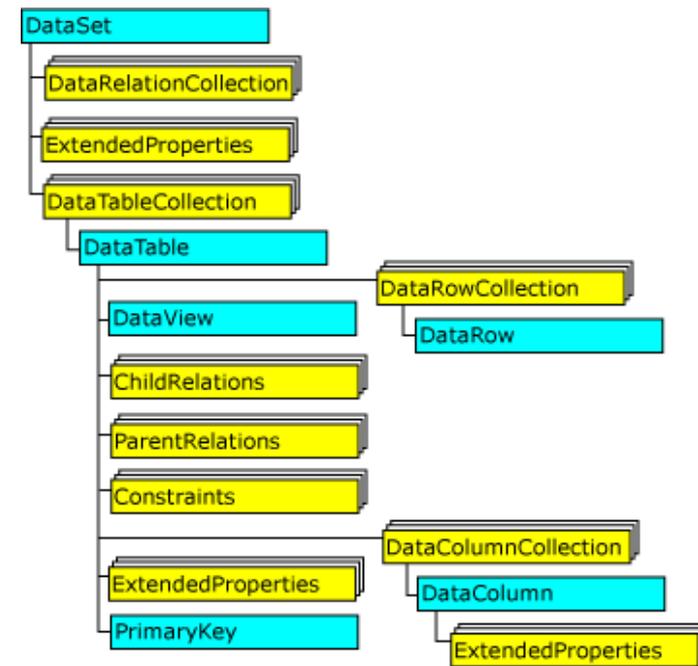
- DataProvider
  - Die DataReader Beispiel

```
SqlConnection sqlConn = new SqlConnection("Data Source=localhost; Initial
                                          Catalog=SQL2008; User Id=user;
                                          Password=password;");

sqlConn.Open();
SqlCommand cmd = new SqlCommand("select Name from Studenten", sqlConn);
SqlDataReader reader = cmd.ExecuteReader();
String value;
While (reader.Read()) {
    value = reader.GetString(0);
}
reader.Close();
```

## 3. ADO.NET

- DataSet
  - Zentrale Stelle, wo Daten der Anwendung gespeichert werden
  - Speicherabbild der eigentlichen DB
  - Speichert Struktur und Beziehungen der Datenbank
  - Daten werden über den DataAdapter im DataSet gespeichert
  - Hat keinerlei Wissen über die Herkunft der Daten
  - Änderungen der Daten werden im DataSet durchgeführt
  - Für das Zurückschreiben in die Datenbank ist ein DataAdapter nötig
  - Kann XML verwenden



## 3. ADO.NET

- DataSet
  - DataTable
    - Bestandteil vom DataSet
    - Stellt eine Datenbanktabelle dar
    - Besitzt eine DataRow-Auflistung
    - Besitzt eine DataColumn-Auflistung
  - DataRow
    - Bestandteil vom DataSet
    - Repräsentiert eine Zeile innerhalb einer DataTable
    - Ist stets an ein DataTable-Objekt gebunden
  - DataColumn
    - Bestandteil vom DataSet
    - Repräsentiert eine Spalte innerhalb der DataTable
    - Ist stets an ein Data-Table-Objekt gebunden

## 3. ADO.NET

```
using System;
using System.Data;

class Program
{
    static void Main()
    {
        // Get DataTable
        DataTable table = GetTable();
        // Die DataTable benutzen...
    }
    // Erzeugen einer DataTable
    static DataTable GetTable()
    {
        // Erzeugen einer DataTable mit vier Spalten
        DataTable table = new DataTable();
        table.Columns.Add("Semester", typeof(int));
        table.Columns.Add("Name", typeof(string));
        table.Columns.Add("Nachname", typeof(string));
        table.Columns.Add("Datum", typeof(DateTime));

        // Einfügen von vier Zeilen
        table.Rows.Add(1, "Max1", "Mustermann1", DateTime.Now);
        table.Rows.Add(2, "Max2", "Mustermann2", DateTime.Now);
        table.Rows.Add(3, "Max3", "Mustermann3", DateTime.Now);
        table.Rows.Add(4, "Max4", "Mustermann4", DateTime.Now);
        return table;
    }
}
```

## 3. ADO.NET

- DataSet
  - DataView
    - Dient zum Filtern und Sortieren der Daten einer DataTable
    - Dient zur Erstellung einer bestimmten Sicht auf die DataTable
  - DataRelation
    - Stellt eine Eltern-Kind Beziehung zwischen zwei DataTable
    - Verknüpft zwei DataTable durch ihre DataColumnn
  - DataGrid
    - Zeigt Daten in Tabellenform auf dem Bildschirm an
    - Datenquelle können eine DataTable, ein DataSet oder ein DataView sein

# Fragen?

Vielen Dank!