

# Window Presentation Foundation

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- Raum 2.202
- Tel. 03943 / 659 338



FB Automatisierung und Informatik: Windows Presentation Foundation

## · Inhalt

- Einführung WPF
- **Layouts**
- C# Sprache
- Dialog-Elemente, Menüs
- 2D- / 3D-Grafik, Audio, Video, Animation
- Eigene Komponenten
- Threads
- DataBinding
- **Datenbanken**
- Routet Events, Dependency Properties, Command
- Textdarstellung (Flow-FixedDocuments)



FB Automatisierung und Informatik: Windows Presentation Foundation

·2

## Datenbanken

### ■ Mögliche Varianten

- ODBC Providers (Excel, Access, Textdatei, AS 400, DB2)
- OLE DB Data Providers (Excel, Access, Textdatei, AS 400, DB2)
- OLE DB Data Link (Data Link File UDL)
- **Net-Provider**
- ADO.net
- LINQ (Language Integrated Query)
- Windows Forms vs. Windows Presentation Foundation

### ■ Datenbank-Unterstützung

- Access
- MS SQL Server (Express Edition bis 2GB)
- MySQL
- Firebird (Server und Embed Version)
- Oracle (Microsoft, Oracle, Corelab)
- Postgre (Corelab)
- Sybase
- VistaDB



## Datenbanken

### ■ Mögliche GUI-Varianten

- DataGridView
- Einbau der FK in das Grid (ComboBox)
- Master / Details (DetailsView, nur ein Datensatz)
- Gruppenwechsel mit Editzeile, ComboBox, Checkbox, ListView etc.

### ■ Weitere Eigenschaften

- Aufruf eines Filters
- VirtualMode für sehr große Datenmengen
- Auslesen der Tabellen in der Datenbank
- Auslesen der Attribute mit Datentypen pro Tabelle
- Auslesen der Fremdschlüssel etc. pro Tabelle



## Datenbanken mit .net Provider erstellen

### ■ Struktur

- Der .net Provider für Firebird und andere DBS kapselt alle Zugriffe auf die Datenbank
- Vollständig in C# geschrieben
- Funktionen:
  - Connection
  - GetSchema
  - Open
  - Close
  - BeginTransaction
  - ChangeDatabase
  - CreateCommand
  - Dispose
  - State (Status: Open Close)
  - FbConnection.CreateDatabase
  - FbConnection.DropDatabase
  - FbConnection.ClearAllPools
  - FbConnection.ClearPool



## Datenbanken erstellen mit Firebird

### ■ Voraussetzungen

- Firebird Server installiert
- Firebird Client-Installation
- Firebird Embed-Version, einfache DLL
- .net Provider für Firebird
- ab Visual Studio 2005

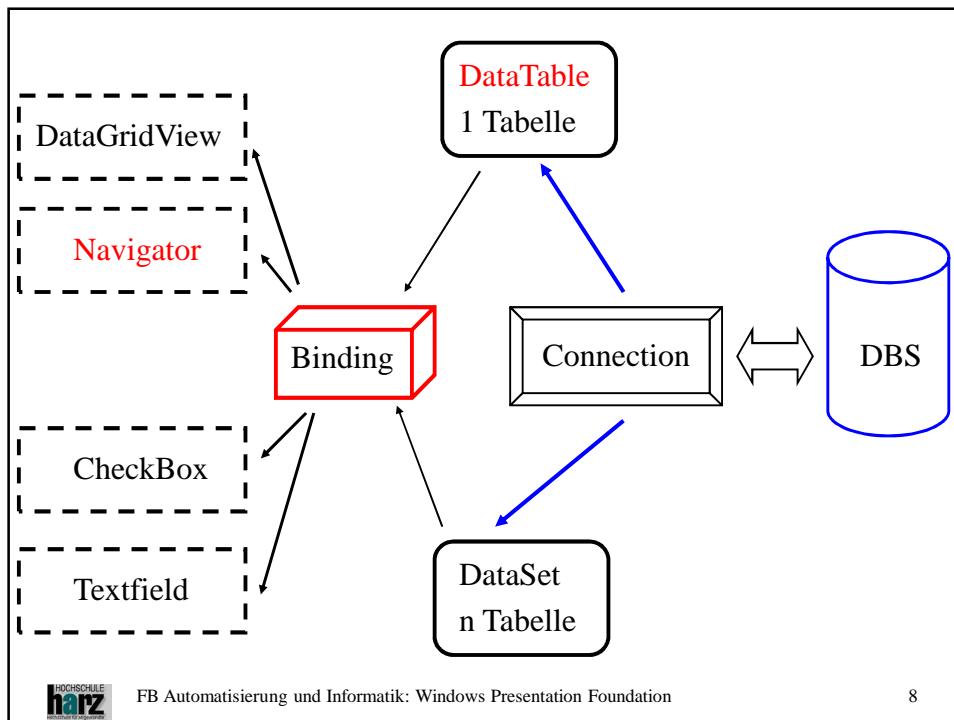
### ■ Ablauf

- Erstellen eines Projektes
- Einbinden des Providers
- Erstellen einer DBS-Klasse (Kapselung)
- Erstellen einer Query-Klasse (Kapselung)
- Aufbau der GUI (DataGridView)
- Erstellen einer Verbindung (connection)
- SQL-Befehl holen
- Query-Abfrage starten
- DataTable dem Grid zuweisen



## Vergleich Firebird und .net Datentypen

Firebird type	.NET type	FbDataReader Methode	Example
BIGINT	System.Int64	GetInt64()	BIGINT Reading Example (C#)
BLOB	System.Byte[]	GetBytes()	BLOB Reading Example (C#)
BLOB SUB_TYPE 1	System.String	GetString()	BLOB SUB_TYPE 1 Reading Example (C#)
CHAR	System.String	GetString()	CHAR Reading Example (C#)
DATE	System.DateTime	GetDateTime()	DATE Reading Example (C#)
DECIMAL	System.Decimal	GetDecimal()	DECIMAL Reading Example (C#)
DOUBLE PRECISION	System.Double	GetDouble()	DOUBLE PRECISION Reading Example (C#)
FLOAT	System.Float	GetFloat()	FLOAT Reading Example (C#)
INTEGER	System.Int32	GetInt32()	INTEGER Reading Example (C#)
NUMERIC	System.Decimal	GetDecimal()	NUMERIC Reading Example (C#)
SMALLINT	System.Int16	GetInt16()	SMALLINT Reading Example (C#)
TIME	System.DateTime	GetDateTime()	TIME Reading Example (C#)
TIMESTAMP	System.DateTime	GetDateTime()	TIMESTAMP Reading Example (C#)
VARCHAR	System.String	GetString()	VARCHAR Reading Example (C#)



```

FbConnection cnn = new FbConnection(sArg);

public DataTable getDBSData(string sSQL) {
    //FbCommand command = new FbCommand(sSQL, _cnn);
    FbDataAdapter da = new FbDataAdapter(sSQL, _cnn);
    DataTable dt = new DataTable("RELATIONS");
    da.Fill(dt); // füllt die DataTable mit Daten
    da.Fill(dt, von, anzahl, dt); // Big Data

    return dt;
}

// hier kann man mehrere Tabellen erhalten, für WPF
public DataSet getDBSDataSet(string sSQL, string sTable) {
    //FbCommand command = new FbCommand(sSQL, _cnn);
    FbDataAdapter da = new FbDataAdapter(sSQL, _cnn);
    DataSet ds = new DataSet();
    da.Fill(ds, sTable); // z. B. "Employee", wichtig da es mehrere gibt
    return ds;
}

```



## Abfrage einer SQL-Anweisung: Direkte Zuweisung zum Grid

```

string sFilename = "EMPLOYEE.FDB";
string sSQL1 = "select * from EMPLOYEE";
_dbs.setDbs(sFilename);
DataSet ds = dbs.getDBSDataSet(sSQL, "EMPLOYEE");
dataGridView1.DataContext = ds;

```



## 1. Datenbankbeispiel: DataBase1



## 1. Datenbankbeispiel: DataBase1

- **1. Register**
  - Einfaches DataGridView
  - Editor für SQL-Befehle
  - Schalter zum Ausführen
  - Automatisches DataBinding
- **2. Register**
  - Einfaches DataGridView
  - manuelles DataBinding, feste Abfrage
- **3. Register**
  - Einfaches DataGridView
  - DataGridViewComboBoxColumn für die Fachbereiche
- **4. Register**
  - Einfaches DataGridView
  - Master-Detail

## 1. Datenbankbeispiel: DataBase1

### ■ Ablauf (1)

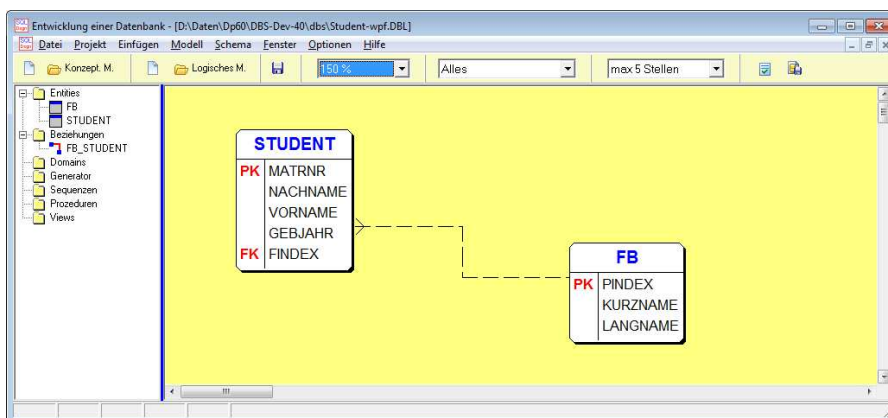
#### ○ Datenbankdesigner

- Erstellen des logischen ER-Modell
- Erstellen der DDL-Befehle

#### ○ Erstellen der Datenbank

- Aufrufen von FBConsole
- Erstellen einer neuen Datenbank
- Einfügen der DDL-Befehle (create Table)
- Einfügen der DML-Befehle (insert into)

## 1. Datenbankbeispiel: DatenbankDesigner



## Tabelle Student

Einstellungen eines Entitys: STUDENT

Allgemein | Attribute | Indizes | Triggers | Constraints  
 Tabellen-Skripte | Beschreibungen | zu Erledigen | Farben | Position

	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Wert	Check-Bedingung	Unique-Bedingung	Beschreibung
1	PK	MATRNR	INTEGER				NOT NULL		CH_MATRNR	-	
2		NACHNAME	VARCHAR	50						-	
3		VORNAME	VARCHAR	50						-	
4		GEBJAHR	INTEGER						CH_GEBJAHR	-	
5	FK	FINDEX	INTEGER				NOT NULL			-	

Add Edit Delete Copy

Ok Abbruch

## Tabelle FB

Einstellungen eines Entitys: FB

Allgemein | Attribute | Indizes | Triggers | Constraints  
 Tabellen-Skripte | Beschreibungen | zu Erledigen | Farben | Position

	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Wert	Check-Bedingung	Un
1	PK	FINDEX	INTEGER				NOT NULL			-
2		KURZNAME	VARCHAR	5						-
3		LANGNAME	VARCHAR	50						-

Add Edit Delete Copy

Ok Abbruch



## Tabellen:

### TABLE FB

- int PINDEX
- char(5) KURZNAME
- varchar(50) LANGNAME

### TABLE STUDENT

- int MATRNR MATRNR muss größer 0 sein
- varchar(50) NACHNAME
- varchar(50) VORNAME
- int GebJahr GebJahr<1995

- Beziehung Student mit FB



FB Automatisierung und Informatik: Windows Presentation Foundation

17

## Ablauf (2): FBConsole

### Erstellen der Datenbank;

- Aufrufen von FBConsole
- Erstellen einer neuen Datenbank
- Einfügen der DDL-Befehle (create Table)
- Einfügen der DML-Befehle (insert into)
  - Datei „WPF-DML.txt“



FB Automatisierung und Informatik: Windows Presentation Foundation

18

## Projekt DataBase1

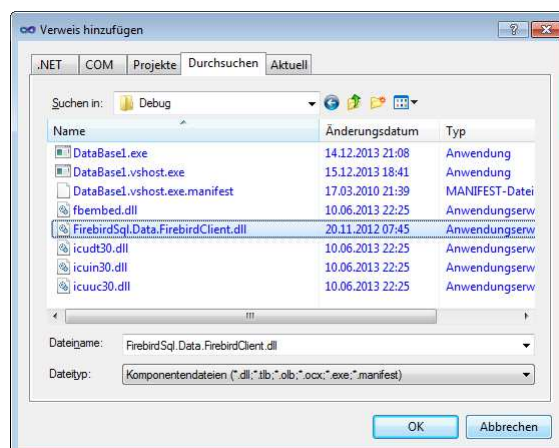
### ■ Ablauf (3)

- „Erstellen“ des WPF-Projektes:
  - Homepage, Laden der ZIP-Datei
- Verändern des Projektes:
  - Einfügen der Klasse DBS, siehe Homepage
  - Einfügen eines Verweises zum Firebird-Provider

## Laden des Verweises auf den Provider

### Ablauf:

- Projekt-Explorer
- Verweise
- Rechte Maustaste
- Verweis hinzufügen
- Register „Durchsuchen“
- Wechseln ins Debug-Verzeichnis
- Auswahl der Fireb....dll



## Projekt DataBase1

### ■ Ablauf (4)

#### ◦ Verändern des Projektes: main.cs

- Einfügen einer globalen Variable dbs
- In load:
  - Feste Zuweisung zur Studenten-Datenbank „STUDENT-WPF.FDB“
  - grid1.ItemsSource = dt1.DefaultView;
  - grid1.AutoGenerateColumns = true;
  - grid1.IsReadOnly = true;
  - grid1.CanUserAddRows = false;
  - grid1.CanUserDeleteRows = false;
  - grid1.CanUserResizeColumns = true;
  - grid1.CanUserSortColumns = true;

## Projekt DataBase1

### ■ Ablauf (5)

- bnGrid1\_SQL1\_Click:
  - String sql = "Select \* from FB";
  - grid1.ItemsSource = dt1.DefaultView;
- bnGrid1\_SQL2\_Click:
  - String sql = "Select \* from Student";
  - grid1.ItemsSource = dt1.DefaultView;
- bnGrid1\_SQL3\_Click:
  - String sql = "select \* from student order by Nachname";
  - grid1.ItemsSource = dt1.DefaultView;
- bnSQL\_Click:
  - Auslesen des Editors
  - Ausführen des SQL-Befehls
  - Ausgabe in grid1

## ■ Ablauf (6)

- Manuelles Definieren der Felder im zweiten DataGrid „grid2“
  - Jeweils eine „DataGrid.Columns“
    - Feld: MATRNR
      - Width="100"
      - Header="Matrnr"
    - Feld NACHNAME
      - Width="100"
      - Header="Nachname"
    - Feld VORNAME
      - Width="100"
      - Header="Vorname,,
    - Feld GEBJAHR
      - Width=„80"
      - Header=„Geburts-Jahr"
    - Feld FINDEX
      - Width="100"
      - Header="Findex"
- «Binding="{Binding FINDEX}"

## Projekt DataBase1

### ■ Ablauf (7)

- bnGrid2\_SQL1\_Click:
  - String sql = "Select \* from Student";
  - grid1.ItemsSource = dt1.DefaultView;

Matrnr	Nachname	Vorname	Geburts-Jahr	Findex
12345	Müller	Andreas	1987	1
12346	Schmidt	Petra	1992	2
12347	Meier	Ralf	1990	3
12348	Brandt	Ute	1982	1
12349	MacDonald	Matthew	1991	1
12350	Braun	Dieter	1988	3
12351	Schulze	Rolf	1983	3
12352	Fleischer	Susanne	1992	3
12355	Huber	Thomas	1991	2
12356	Staller	Frank	1981	1
12358	Meyer	Freya	1985	3

## Projekt DataBase1: 4. Register

System.Data.DataRowView

Grid1: SQL1 Grid1: SQL2 Grid1: SQL3 Grid2: SQL1 Grid3: SQL1 Grid4: SQL1 Insert Student

Grid1 Grid2 Grid3 Grid4

PINDEX	KURZNAME	LANGNAME
1	AI	Automatisierung und Informatik
2	VW	Verwaltungswissenschaften
3	W	Wirtschaftswissenschaften

MATNR	NACHNAME	VORNAME	GEBJAHR	FINDEX
12345	Müller	Andreas	1987	1
12417	Turgut	Emine	1985	1
12356	Staller	Frank	1981	1
12373	Nubert	Günther	1992	1
12349	MacDonald	Matthew	1991	1

## Projekt DataBase1

### ■ Ablauf (8)

- Viertes DataGrid „grid4a“ und „grid4b“
- Master/Details
- „grid4a“
  - Automatisches Anzeigen der Felder à la grid1
  - Es darf nur ein Eintrag ausgewählt werden
  - Event „grid4a\_SelectionChanged“ erzeugen
- „grid4b“
  - Abfragen des „SelectedIndex“ von „grid4a“
  - Wenn  $\geq 0$  dann
    - Abfrage des „SelectedItem“ von „grid4a“
    - **KARSTEN**
    - `int pindex = (int) 42;`
    - `String sql = "Select * from Student where FINDEX=" + pindex;`
    - Abfrage und Eintragen

## Projekt DataBase1

### ■ Ablauf (9)

- Schalter „Insert Student“
- Eintragen eines neuen Studenten
  - Matrikelnummer: 88888
  - Nachname: Garret
  - Vorname: David
  - FB Wirtschaftswissenschaften

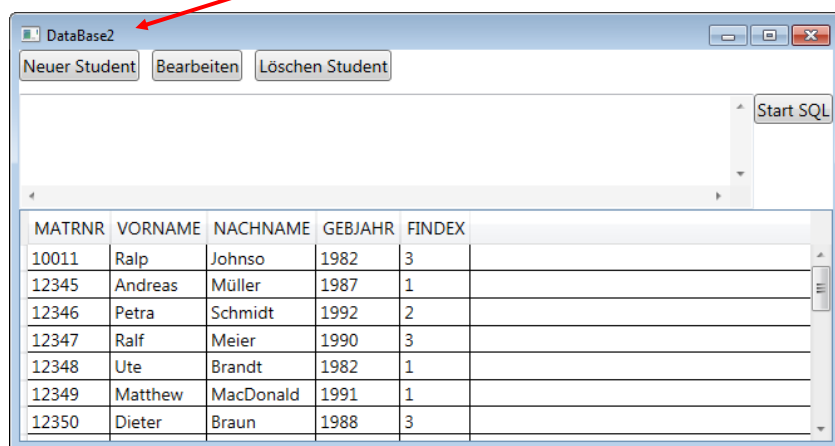
## Datenbank-ComboBox: 3. Register

EMP_NO	FIRST_NAME	LAST_NAME	PHONE_EXTENSION	HIRE_DATE	DEPT_NO	Abteilungen	JOB_CODE	JOB_GRADE	JOB_COUNTRY	SALARY
2	Robert	Nelson	250	28.12.19...	600	Engineering	VP	2	USA	105900
4	Bruce	Young	233	28.12.19...	621	Software Development	Enq	2	USA	97500
5	Kim	Lambert	22	06.02.19...	130	Corporate Headquarters	Enq	2	USA	102750
8	Leslie	Johnson	410	05.04.19...	180	Sales and Marketing	Mktg	3	USA	64635
9	Phil	Forest	229	17.04.19...	622	Engineering	Mnqr	3	USA	75060
11	K. J.	Weston	34	17.01.19...	130	Finance	SRep	4	USA	86292,9...
12	Terri	Lee	256	01.05.19...	000	Marketing	Admin	4	USA	53793
14	Stewart	Hall	227	04.06.19...	900	Software Products Div.	Finan	3	USA	69482,625
15	Katherine	Young	231	14.06.19...	623	Software Development	Mnqr	3	USA	67241,25
20	Chris	Papado...	887	01.01.19...	671	Quality Assurance	Mnqr	3	USA	89655
24	Pete	Fisher	888	12.09.19...	671	Customer Support	Enq	3	USA	81810,1...
28	Ann	Bennet	5	01.02.19...	120	Consumer Electronics	Admin	5	England	22935
29	Roger	De Souza	288	18.02.19...	623	Research and Development	Enq	3	USA	69482,625
34	Janet	Baldwin	2	21.03.19...	110	Customer Services	Sales	3	USA	61637,8...
36	Roger	Reeves	6	25.04.19...	120	Field Office: East Coast	Sales	3	England	33620,625
37	Willie	Stansbury	7	25.04.19...	120	Field Office: Canada	Enq	4	England	39224,0...
44	Leslie	Phong	216	03.06.19...	623	Pacific Rim Headquarters	Enq	4	USA	56034,375
45	Ashok	Ramanat...	209	01.08.19...	621	Field Office: Japan	Enq	3	USA	80689,5
46	Walter	Steadman	210	09.08.19...	900	Field Office: Singapore	CFO	1	USA	116100
52	Carol	Nordstrom	420	02.10.19...	180	European Headquarters	PRel	4	USA	42742,5
61	Luke	Leung	3	18.02.19...	110	Field Office: Switzerland	SRep	4	USA	68805
65	Sue Anne	O'Brien	877	23.03.19...	670	Field Office: France	Admin	5	USA	31275
71	Jennifer	Burbank	289	15.04.19...	622	Field Office: Italy	Enq	3	USA	53167,5
72	Claudia	Suthera		20.04.19...	140		SRep	4	Canada	100914

## Projekt DataBase2:

- DataGrid-Spalten werden automatisch erstellt
- DataGrid ist readonly
- Editieren mittels Dialogelemente
- Update / Insert / Delete mit externen SQL-Befehlen

## Projekt DataBase2:



## Projekt DataBase2: neuer Student, bearbeiten

Two side-by-side screenshots of a Windows application titled "Bearbeiten eines Studenten".

The left window is titled "Neuer Student" and shows empty input fields for Matnr, Nachname, Vorname, Geburtsjahr (1990), and Fachbereich. The right window is titled "Bearbeiten eines Studenten" and shows the same fields filled with: Matnr: 12345, Nachname: Müller, Vorname: Andreas, Geburtsjahr: 1987, and Fachbereich: AI. Both windows have "Ok" and "Abbrechen" buttons at the bottom.

## Projekt DataBase2: Eintrag löschen

A screenshot of a Windows dialog box titled "Student löschen". It contains a question mark icon and the text: "Wollen Sie wirklich den Datensatz löschen? Nachname: Müller Vorname: Andreas Matrnr: 12345 FB:". At the bottom are "Ja" and "Nein" buttons.



## Projekt DataBase2: Editieren

```
private void bnEdit_Click(object sender, RoutedEventArgs e) {
    Student std = new Student();
    int i = grid1.SelectedIndex;
    if (i >= 0) {
        // DataRowView row
        Object obj = grid1.SelectedItem;
        Title = obj.ToString();
        DataRowView drvview = (DataRowView)grid1.SelectedItem;
        DataRow dRow = drvview.Row;
        // Matrnr, Vorname, Nachname, gebJahr, findex
        int matrnr = (int)dRow["MATRNR"];
        std.Matrnr = matrnr;
        std.Vorname = (String)dRow["VORNAME"];
        std.Nachname = (String)dRow["NACHNAME"];
        std.Gebjahr = (int)dRow["GEBJAHR"];
        std.FB_Index = (int)dRow["FINDEX"];
    }
}
```



## Projekt DataBase2: Editieren

```
Student std_Ori = new Student(std);
FStudent dialog = new FStudent(this, "Bearbeiten eines Studenten", std, true);
dialog.ShowDialog();
if (dialog.Retcode) {
    _dbs.setDbs(Konstanten.STUDENT_WPF);
    String sql;
    if (!std.Nachname.Equals(std_Ori.Nachname)) {
        sql = "UPDATE student SET Nachname="+qt(std.Nachname)
            +" WHERE Matrnr="+matrnr;
        DataTable dt1 = _dbs.getDBSData(sql);
    }
    if (!std.Vorname.Equals(std_Ori.Vorname)) {
        sql = "UPDATE student SET Vorname=" + qt(std.Vorname) +
            " WHERE Matrnr=" + matrnr;
        DataTable dt1 = _dbs.getDBSData(sql);
    }
}
```



## Projekt DataBase2: Editieren

```
if (std.Gebjahr!=std_Ori.Gebjahr) {
    sql = "UPDATE student SET Gebjahr=" + std.Gebjahr +
        " WHERE Matrn=" + matrn;
    DataTable dt1 = _dbs.getDBSData(sql);
}
if (std.FB_Index != std_Ori.FB_Index) {
    sql = "UPDATE student SET FIndex=" + std.FB_Index +
        " WHERE Matrn=" + matrn;
    DataTable dt1 = _dbs.getDBSData(sql);
}
_dbs.Close();
showTable();
}
}
} // bnEdit_Click
```

## Weitere Eigenschaften

- DataAdapter
  - da.Fill(dt); // füllt die DataTable mit Daten
  - **da.Fill(dt, von, anzahl, dt); // für große Daten**

- **SqlBulkCopy: Massenkopieren**

- `SqlConnectionStringBuilder cb = new SqlConnectionStringBuilder();`
- `cb.DataSource = "Produktion";`
- `cb.InitialCatalog = "Verkauf";`
- `cb.IntegratedSecurity = true;`
- `SqlConnection cnn = new SqlConnection(cb.ConnectionString);`
  
- `SqlCommand cmd = new SqlCommand("SELECT * FROM Rechnungen",cnn);`
- `cnn.Open();`
- `SqlDataReader rdr = cmd.ExecuteReader();`
  
- `SqlBulkCopy sbc = new SqlBulkCopy("server=.;database=PTest;" +<BR>"Integrated Security=SSPI");`
  
- `sbc.DestinationTableName = "Temp";`
- `sbc.WriteToServer(rdr);`
  
- `sbc.Close();`                      `rdr.Close();`                      `cnn.Close();`