Fachbereich Automatisierung und Informatik	HOCHSCHULE harz Hochschule für angewandte Wissenschaften (FH) Wernigerode

4. Labor WPF

„WPF und Datenbanken“

Dipl. Inf., Dipl.-Ing. (FH) Michael Wilhelm
Friedrichstraße 57 - 59
38855 Wernigerode

Raum: 2.202
Tel.: 03943/659-338
Fax: 03943/659-399
Email: mwilhelm@hs-harz.de

1 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

1.1 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

1.2 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

1.3 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#)

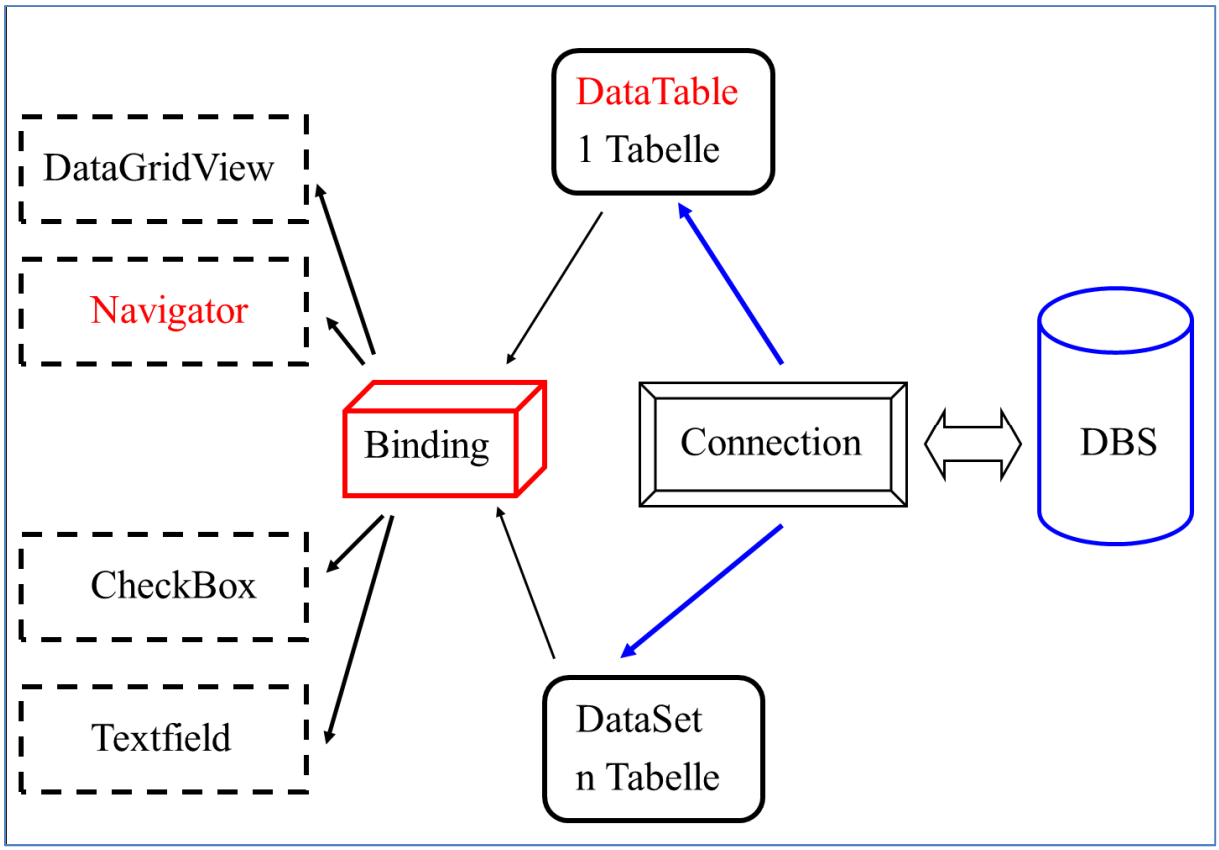


Abbildung 1 Überblick

1.4 Beispielcode:

```
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
     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;
}
```

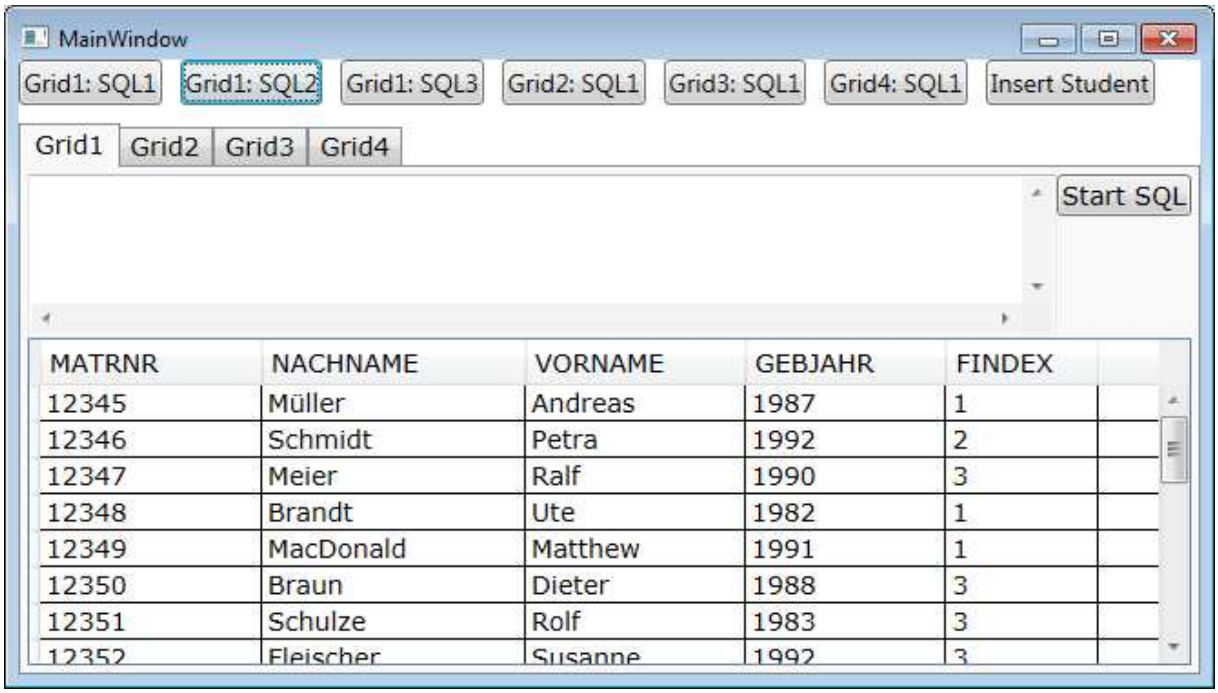
1.4.1 Abfrage einer SQL-Anweisung: Direkte Zuweisung zum Grid:

```
string sFilename = "EMPLOYEE.FDB";
string sSQL1 = "select * from EMPLOYEE";
_dbs.setDbs(sFilename);
```

```
DataSet ds = dbs.getDBSDataSet(sSQL, "EMPLOYEE");
dataGrid1.DataContext = ds;
```

2 Datenbankbeispiel: DataBase1

2.1 Oberfläche



2.1.1 Erläuterung

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

2.2 Ablauf

2.2.1 Datenbankdesigner

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

2.2.2 Erstellen der Datenbank

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

Ergebnis:

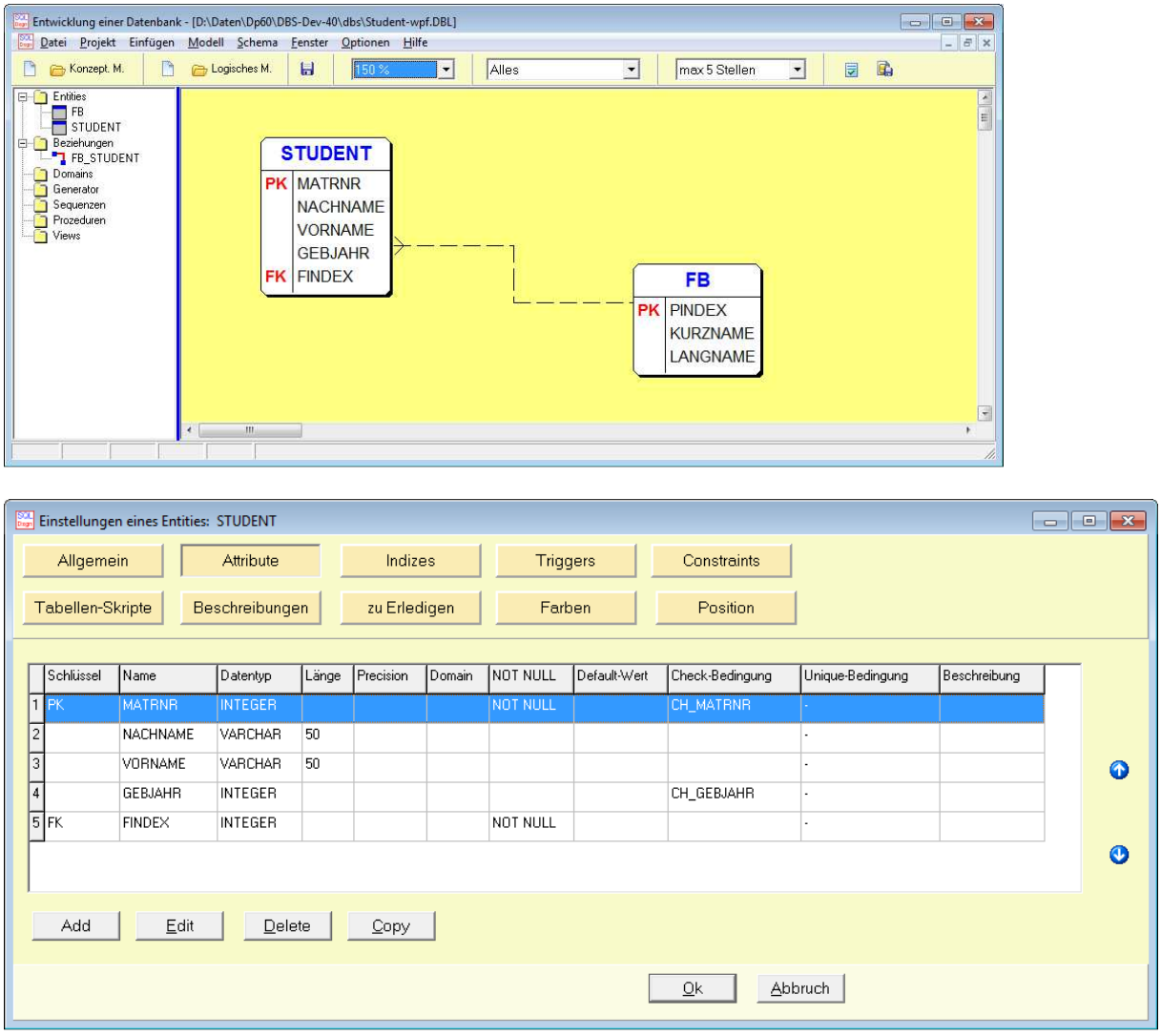


Abbildung 2 Tabelle Student

Details:

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

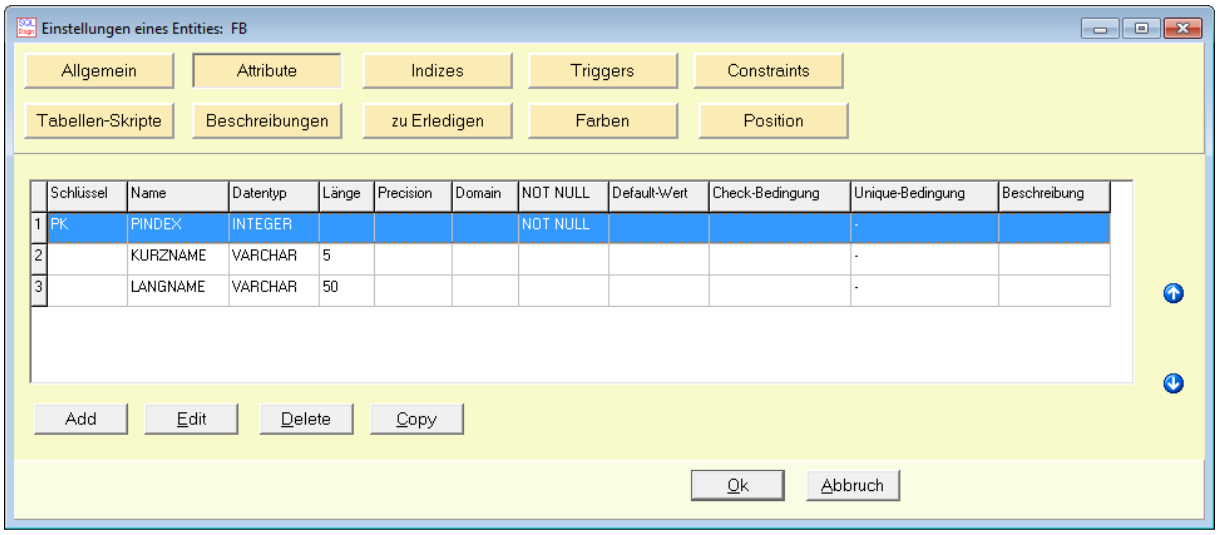


Abbildung 3 Tabelle FB

2.2.3 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“

2.2.4 „Erstellen“ des WPF-Projektes:

- Homepage, Laden der ZIP-Datei

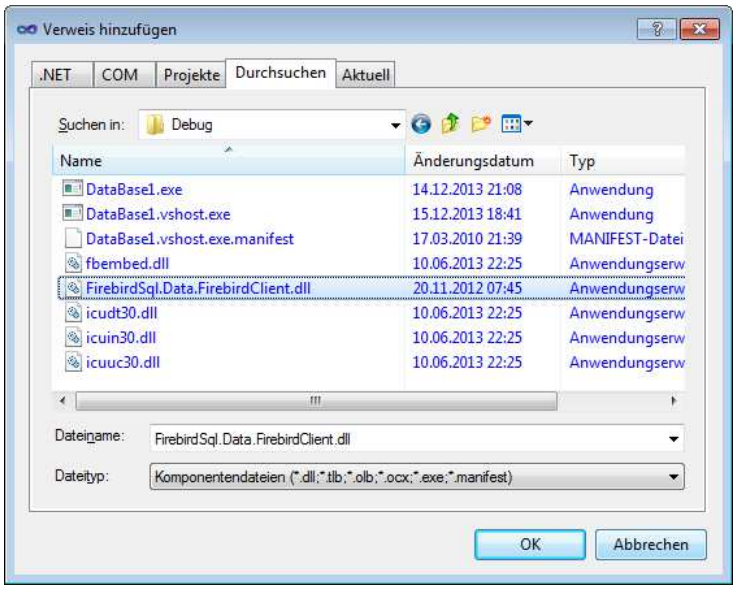
2.2.5 Verändern des Projektes:

- Einfügen der Klasse DBS, siehe Homepage
- Einfügen eines Verweises zum Firebird-Provider

2.2.6 Laden des Verweises auf den Provider

Ablauf:

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



2.2.7 Verändern der Datei 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;

2.2.8 Register Grid1

- 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 = dt1.DefaultView;

- bnSQL_Click:
    - Auslesen des Editors
    - Ausführen des SQL-Befehls
    - Ausgabe in grid1

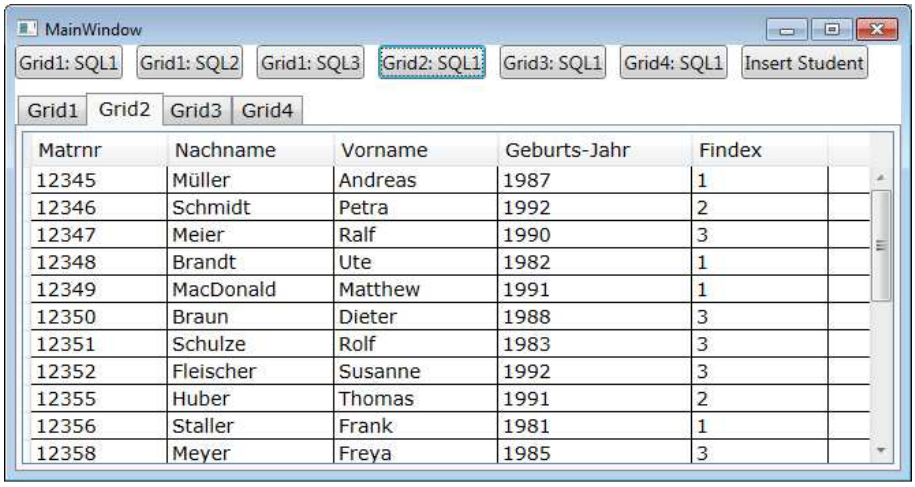
2.2.9 Register Grid2

- 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"

Hinweis: Binding="{Binding FINDEX}"

- bnGrid2_SQL1_Click:
    - String sql = "Select * from Student";
    - grid1.ItemsSource = dt1.DefaultView;

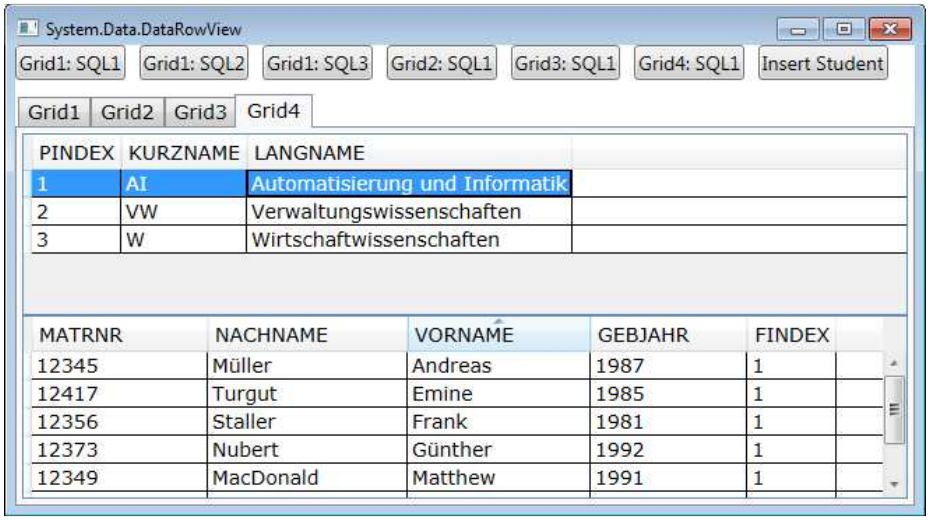
Ergebnis:



2.2.10 Register Grid4

- 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 >=0 dann
        - Abfrage des „SelectedItems“ von „grid4a“
        - **KARSTEN**
        - int pindex = (int) 42;
        - String sql = "Select * from Student where FINDEX=" + pindex;
        - Abfrage und Eintragen



2.2.11 Insert Into

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

Test mit dem ersten Register.