

2. BACHELORARBEIT

EIN DATENMODELL FÜR AUSWERTUNGEN VON KUNDENBEFRAGUNGEN UND DESSEN VERWENDUNG IN EINEM INFORMATIONSSYSTEM

FH-Bachelorstudiengang

Informatik

Software Engineering

Name: Patrick Felber

Matrikelnummer: 0810276007

FH-Betreuer: Mag. Thomas Woltron

Datum: 30.07.2012

Titel der Bachelorarbeit:

**EIN DATENMODELL FÜR AUSWERTUNGEN VON
KUNDENBEFRAGUNGEN UND DESSEN VERWENDUNG IN
EINEM INFORMATIONSSYSTEM**

2. Bachelorarbeit

Eingereicht von: Patrick Felber

am: Fachhochschul-Bachelorstudiengang
Informatik

Vertiefung: Software Engineering

Begutachter: Mag. Thomas Woltron

Wiener Neustadt am: 30.07.2012

Ich versichere,
dass ich die Bachelorarbeit selbständig verfasst, andere als die angegebenen
Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten
Hilfe bedient habe und diese Bachelorarbeit bisher weder im In- noch im Ausland
in irgendeiner Form als Prüfungsarbeit vorgelegt habe. Die von mir eingereichte
schriftliche Version stimmt mit der digitalen Version der Arbeit überein.

Datum

Unterschrift

Kurzzusammenfassung:

Diese Arbeit hat sich als Ziel gesetzt, ein System zu entwickeln um Kundenzufriedenheitsumfragen automatisiert auswerten zu können. Dabei stößt man auf die Notwendigkeit ein Datenmodell zu modellieren, welches Daten von Kundenbefragungen einheitlich speichern kann. Ein weiterer Punkt dieser Arbeit beschäftigt sich mit der Anwendung des Datenmodells. Dazu wird eine Systemarchitektur erstellt, welche drei Programme enthält.

Schlagwörter: Kundenzufriedenheit, Umfragen, Befragung, Datenmodell

Abstract:

It is the goal of this paper, to create a system to automatically evaluate customer satisfaction surveys. During this process a database model is designed, that can uniformly save the data gathered from the surveys. Another chapter of this thesis deals with the application of this database model. For this purpose a system architecture, that contains three programs, is created.

Keywords: Customer Satisfaction, Surveys, Interview, Database Model, Polling

INHALTSVERZEICHNIS

1	Einleitung.....	3
2	Datenmodell	3
2.1	Rahmenbedingungen	3
2.2	Informationen sammeln	3
2.2.1	Fragebögen.....	4
2.2.2	Unternehmen	4
2.2.3	Ergebnis einer Umfrage	4
2.3	Datenmodell	5
2.3.1	Fragenstellung.....	5
2.3.2	Fragen.....	6
2.3.3	Antwort und Antwortmöglichkeiten.....	6
2.3.4	Segmente	6
2.3.5	Studie	7
2.3.6	Fragenkatalog	7
2.3.7	Abhängige Fragen	7
2.3.8	Interviewer	7
2.3.9	Kunden- und Firmendaten.....	7
3	Programme	7
3.1	Programmarchitektur	7
3.2	Entwicklungsumgebung.....	8
3.3	Verwendete Datenbank Schreibe- und Lesetechniken.....	9
3.4	User Experience	9
3.5	Administrationsprogramm	9
3.5.1	Beschreibung	9

3.5.2	Funktionsumfang	9
3.5.3	Kassendiagramm	13
3.5.4	Implementation	13
3.6	Import-Programm	13
3.6.1	Beschreibung und Anforderungen	13
3.6.2	Klassendiagramm	14
3.6.3	Userinterface.....	15
3.6.4	Ablauf des Importvorganges	15
3.7	Mafo-Dashboard.....	16
3.7.1	Beschreibung und Anforderungen	17
3.7.2	Userinterface.....	17
3.7.3	Funktionsumfang	17
3.7.4	Klassendiagramm	18
4	User Test.....	18
5	Zusammenfassung und Ausblicke	18
6	Abbildungsverzeichnis	19
7	Literaturverzeichnis.....	20
Anhang A	21

1 Einleitung

In dieser Arbeit wird ein Datenmodell zur Auswertung von Kundenbefragungen beschrieben. Da Kundenbefragungen auf verschiedenen Arten durchgeführt werden können, liegt die Anstrengung darin ein einheitliches Datenmodell zu entwerfen.

Die Arbeit umfasst zwei Hauptpunkte. Im ersten Punkt (2. Datenmodell) wird der Aufbau des Datenmodells beschrieben. Im zweiten Punkt (3. Programme) wird der praktische Einsatz des Datenmodells beschrieben. Dazu wurden drei *Programme programmiert* die das Datenmodell verwenden. Diese Programme werden von der Entstehung, über verwendete Techniken, Aufbau und Strukturierung, bis hin zur Test- und Einsatzphase beschrieben.

2 Datenmodell

In diesem Punkt wird beschrieben wie das Datenmodell erstellt wird. Dazu wird der Database Systems Development Lifecycle von Connolly T. und Begg C. (2009) angewandt.

2.1 Rahmenbedingungen

Dieses Datenmodell ist für interne und externe Studien entwickelt. Es interagieren maximal 10 Personen gleichzeitig mit der Datenbank. Daten für das Datenmodell kommen aus mehreren Ergebnissen der Umfragen. Das Datenmodell muss bei großen Datenmengen performant sein.

Das Datenmodell unterstützt das Importieren, das Verwalten und die Erhaltung der Daten.

2.2 Informationen sammeln

Um herauszufinden welche Informationen im Datenmodell gespeichert werden, wird mit der Analyse bei den Fragebögen begonnen.

2.2.1 Fragebögen

Umfragen können auf verschiedene Arten durchgeführt werden. In der Arbeit von Küttler (2011) wird die Erstellung eines Fragebogens für ein Online-Umfrageprogramm erklärt. Gräf (o. Jg.) befasst sich mit der Optimierung von Onlinebefragungen. In beiden Arbeiten setzt sich ein Fragebogen aus diesen Punkten zusammen:

- Frage
- Antwort
- Antwortmöglichkeit
- Frageart
- Befragter

Gräf (o. Jg.) kategorisiert Fragen nach Fragetypen. Zusammengefasst ergeben sich diese 4 Fragetypen:

- Freitext
- Einfachauswahl
- Mehrfachauswahl
- Kombination aus den ersten 3 Typen

Jede Frage kann einem dieser 4 Typen zugeordnet werden.

2.2.2 Unternehmen

Ein Unternehmen unterteilt seine Aufgaben in Segmente. Segmente sind Geschäftsbereiche, wie zum Beispiel. Befragungen eines Unternehmens werden für bestimmte Segmente durchgeführt. Die Segmente haben verschiedene Wichtigkeitsgrade für das Unternehmen. Segmente eines Unternehmens können Untersegmente haben.

2.2.3 Ergebnis einer Umfrage

Das Datenmodell verwendet zwei Quellen von Befragungsergebnissen. Einerseits Programme, die Befragungen erstellen, durchführen und das Ergebnis ausgeben. Andererseits externe Firmen, die auf Umfragen spezialisiert sind. Für diese Arbeit wird das Programm Ransys verwendet.

Ransys ist ein Programm zum Verwalten und Durchführen von Umfragen. Fragebögen können erstellt und dem Kunden auf verschiedene Arten gestellt werden. Ransys gibt ein Excelformat aus, das alle Informationen in einem Blatt enthält.

2.3 Datenmodell

In Abbildung 1 wird das Datenmodell beschrieben.

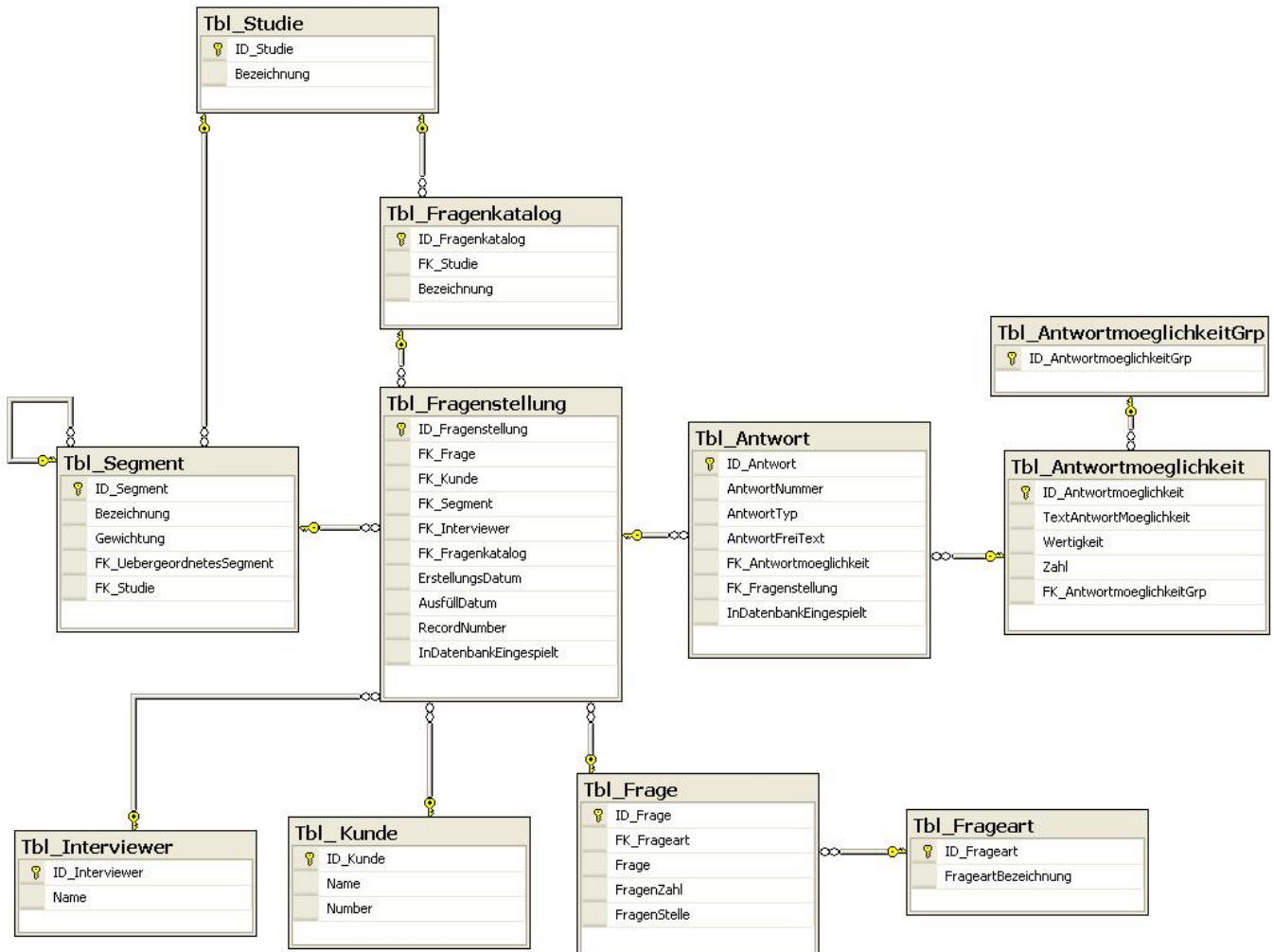


Abbildung 1 - Darstellung des Datenmodells

2.3.1 Fragenstellung

Die *Tbl_Fragenstellung* ist das Zentrum des Datenmodells. Sie ist die Verknüpfung von den Fragen und den Antworten des Kunden.

Zusätzliche Information wird an der *Tbl_Fragenstellung* angehängt. Der *Tbl_Interviewer* ist eine optional zusätzliche Information.

Die Spalte *InDatenbankEingespield* definiert den Zeitpunkt des Imports des Datensatzes. *Recordnumber* speichert die Zeilennummer des importierten Excelldokuments. *ErstellDatum* und *AusfüllDatum* geben den Zeitpunkt der Frage und Antwort an.

2.3.2 Fragen

Eine Frage wird im Datenmodell nur einmal gespeichert, sofern sie dieselbe Fragenart hat. Zusätzlich zum Fragentext wird die Nummer der Frage gespeichert. Die *Fragenstelle* gibt die Position im Fragebogen an.

2.3.3 Antwort und Antwortmöglichkeiten

Die Antwort wird über die Spalte *Antworttyp* verwaltet. Wenn in der Spalte *Antworttyp* der Wert „AntwortFreiText“ steht, dann steht in der Spalte *AntwortFreiText* die Antwort der befragten Person. Steht in der Spalte *AntwortFreiText* der Wert „Antwortmehrfachauswahl“ oder „Antworteinfachauswahl“, dann wird über den *FK_Antwortmoeglichkeit* auf mehrere Antwortmöglichkeiten verwiesen.

Jede Antwortmöglichkeit gehört zu einer einmalig in der Datenbank vorkommenden Antwortmöglichkeitengruppe.

Die Zahl der Antwortmöglichkeit beschreibt ihre Position in der Befragung. Die Wertigkeit gibt der Antwortmöglichkeit ein Ranking.

2.3.4 Segmente

Segmente sind Teile oder Produkte eines Unternehmens. Segmente können auf ihre Übersegmente verweisen. Jedes Segment gehört einer Studie an und wird in den Fragenstellungen verwendet. Weiters kann dem Segment eine Gewichtung zugewiesen werden.

2.3.5 Studie

Die *Tbl_Studie* ist allen anderen Tabellen übergeordnet.

2.3.6 Fragenkatalog

Eine Studie kann mehrere Fragebögen umfassen. In diesem Datenmodell sind Studien oder Fragebögen auswertbar.

2.3.7 Abhängige Fragen

Die Abhängigkeit von Fragen wird in diesem Datenmodell nicht berücksichtigt. Das Datenmodell wird verwendet um beantwortete Fragen auszuwerten. Dafür wird die Information der Abhängigkeit von Fragen nicht benötigt.

2.3.8 Interviewer

Die *Tbl_Interviewer* ist zusätzliche Information. Zusätzliche Information wird als Tabelle an die Fragenstellung angefügt.

2.3.9 Kunden- und Firmendaten

Die Kunden und Firmen werden im Datenmodell als eindeutige Nummer dargestellt. Diese Tabelle kann zum Erkennen von schon befragten Personen benutzt werden.

3 Programme

In diesem Punkt wird das Datenmodell eingesetzt und getestet.

3.1 Programmarchitektur

Die Programme müssen dem User die Daten der Datenbank nutzerfreundlich zur Verfügung stellen.

Die Zielgruppe sind Personen mit Erfahrung in Statistik und Umgang mit Windowsprogrammen.

Die Übersicht über die Programmarchitektur ist in der Abbildung 2 wiedergegeben.

Das erste Programm ist das Admin-Programm. Es ermöglicht dem User Daten in der Datenbank einfach zu verwalten.

Das zweite Programm ist das Importer-Programm. Es importiert Daten aus dem Umfrageergebnis in die Datenbank. In dieser Arbeit wird der Importer für das Ransysformat entwickelt. Soll ein anderes Format importiert werden, muss dazu der Importer abgeändert werden. Die Funktionalität des restlichen Systems bleibt dabei unverändert

Das dritte Programm wird benötigt um Daten aus der Datenbank auszulesen und als Statistik auszugeben.

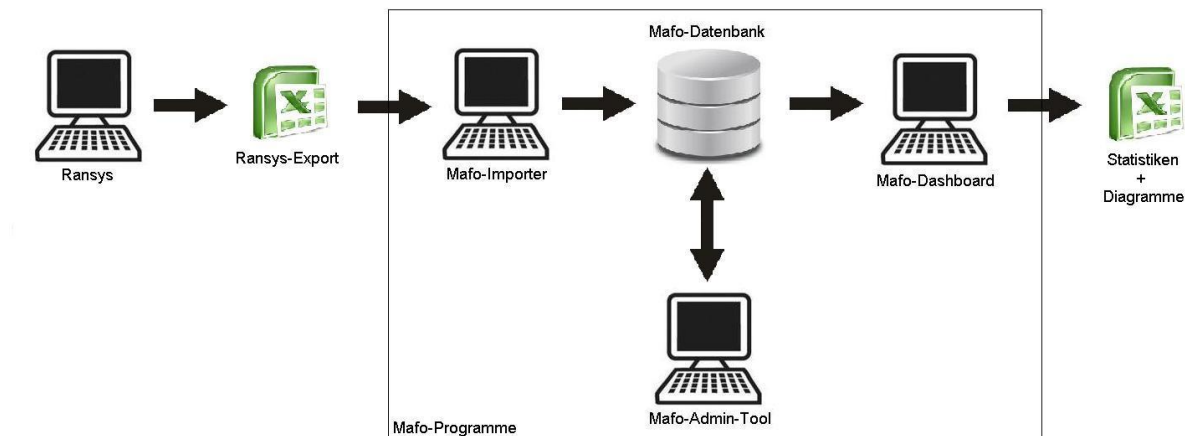


Abbildung 2 – Übersicht der Programmarchitektur

3.2 Entwicklungsumgebung

In dieser Arbeit wurde folgende Entwicklungsumgebung verwendet:

- SQL 2008 Server
- SQL 2008 Server Manager
- Visual Studio 2008, C#, Framework 3.5, ADO.NET 3.5
- Windows XP SP3
- Microsoft Office 2007

3.3 Verwendete Datenbank Schreibe- und Lesetechniken

Für die Lese und Schreibvorgänge in den Programmen werden drei Techniken verwendet. Die erste Technik ist ADO.NET. Diese Technik wird verwendet um kleine Mengen an Daten zu schreiben oder zu lesen. Die zweite Technik ist SQLBulkCopy. Diese Technik wird benötigt um große Datenmengen effizient in die Datenbank zu schreiben. Die dritte Technik sind SQL Befehle oder Stored Procedures.

In den Tabellen „Fragenstellung“ und „Antwort“ wird für alle Foreign Keys ein Index verwendet. Dadurch werden die Datenbanken bei Join-Abfragen performanter.

Bei Abfragen der Programme werden nach jedem Join temporäre Zwischentabellen verwendet. Dadurch werden beim mehrfachen Joinen nur Teile der Tabellen verbunden.

3.4 User Experience

Die benutzerfreundliche Bedienung der Programme wird durch Einhaltung den Kriterien der ISO 9241-110 garantiert.

3.5 Administrationsprogramm

In diesem Punkt wird beschrieben wie ein Programm zur Verwaltung von Daten aus dem Datenmodell aufgebaut wird.

3.5.1 Beschreibung

Das Administrationsprogramm ermöglicht es Studien und zugehörige Daten in der Datenbank anzulegen und zu verwalten.

3.5.2 Funktionsumfang

Die ersten Funktionen des Administrationsprogramms sind das Anlegen, Bearbeiten und Löschen einer Studie. In Abbildung 3 wird das Userinterface dieser drei Funktionen dargestellt.

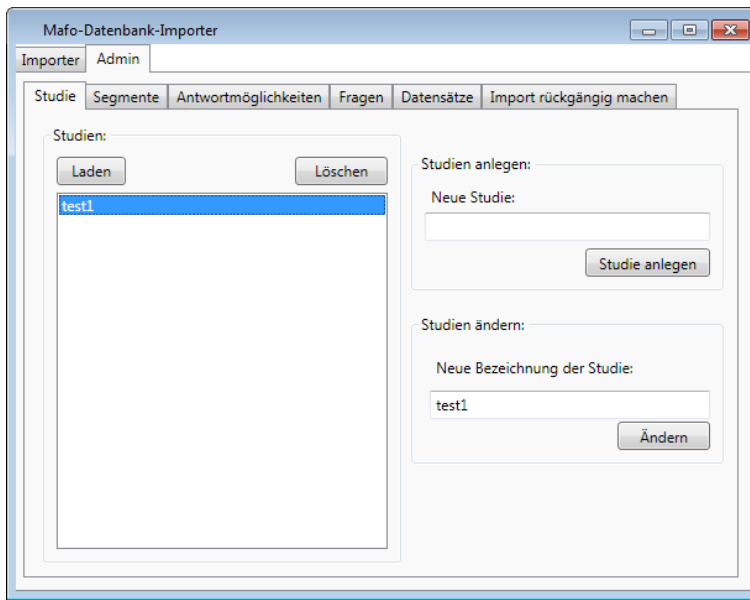


Abbildung 3 – Userinterface des Administrationsprogramms für die Studie

Die nächste Funktion ist das Verwalten der Segmente für eine Studie. Es beinhaltet das Strukturieren der Segmente. Den Segmenten können Übersegmente zugeordnet werden. Das Userinterface dieser Funktionen ist in Abbildung 4 dargestellt.

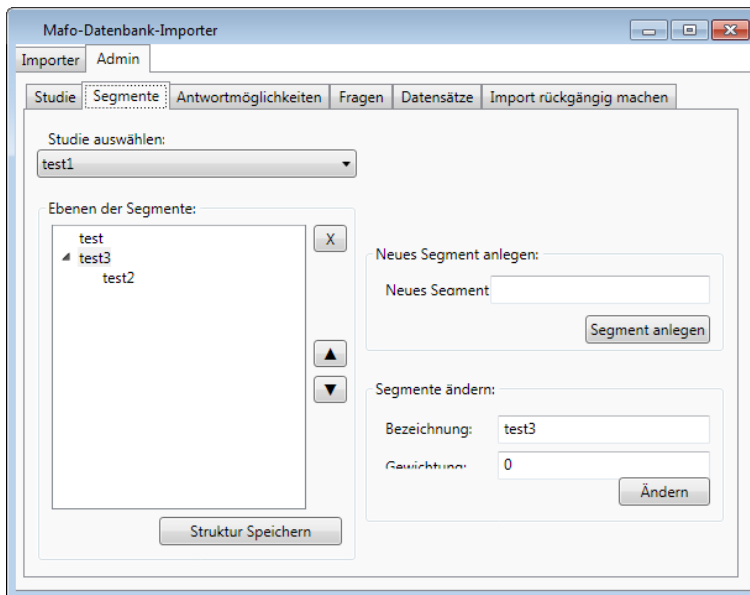


Abbildung 4 - Userinterface des Administrationsprogramms für die Segmente

Die Information über Wertigkeit oder Wichtigkeit der Antwortmöglichkeiten ist optional und daher bei den meisten Ergebnissen einer Umfrage nicht enthalten. So kann nach einem Import den Antwortmöglichkeiten eine Reihenfolge

zugewiesen werden. Eine Antwortmöglichkeit muss nach dem Import editierbar sein. Das Userinterface dieser Funktionalitäten ist in Abbildung 5 dargestellt.

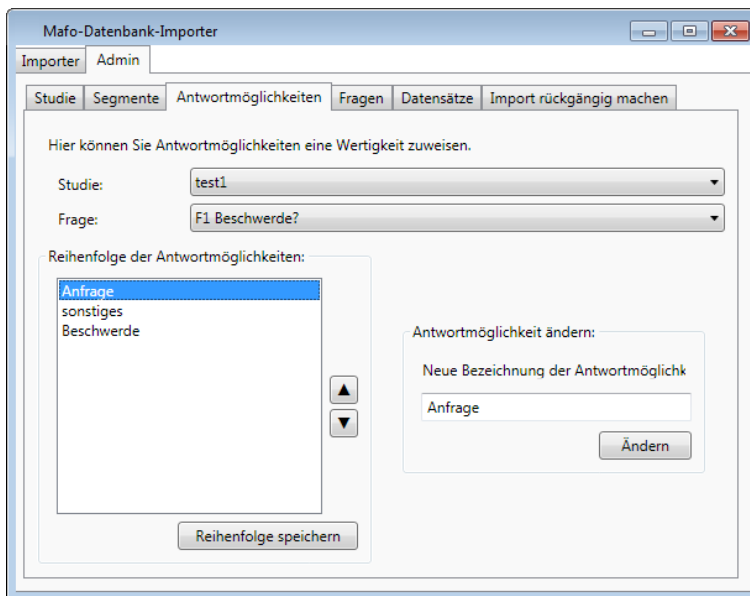


Abbildung 5 - Userinterface des Administrationsprogramms für die Antwortmöglichkeiten

Während einer Umfrage können sich die Fragenbezeichnung oder die Antwortmöglichkeiten ändern. Dadurch existiert eine Frage nach einem Teilimport zweimal in der Datenbank. Das Administrationsprogramm kann diese Fragen zusammenfügen. Das Userinterface dieser Funktionalitäten ist in Abbildung 6 dargestellt.

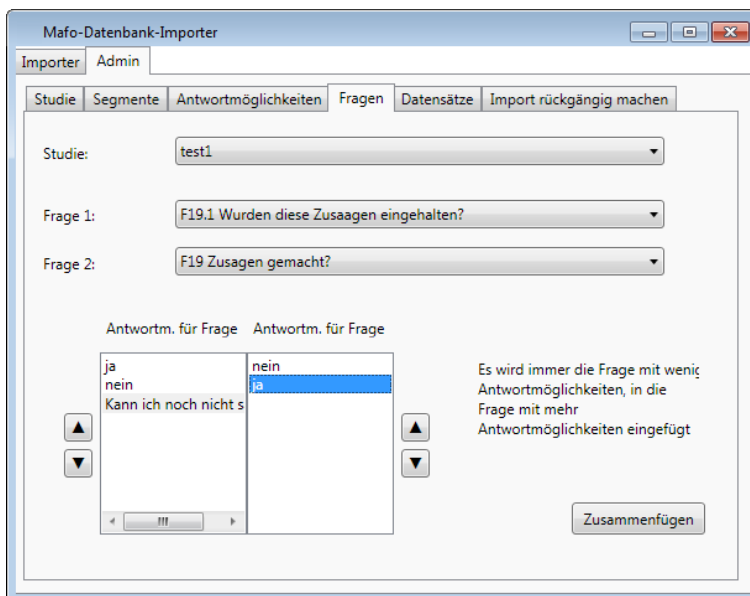


Abbildung 6 - Userinterface des Administrationsprogramms für die Fragen

Das Administrationsprogramm kann Datensätze nach dem Import einer Umfrage löschen. Das Userinterface dieser Funktionalitäten ist in Abbildung 7 dargestellt.

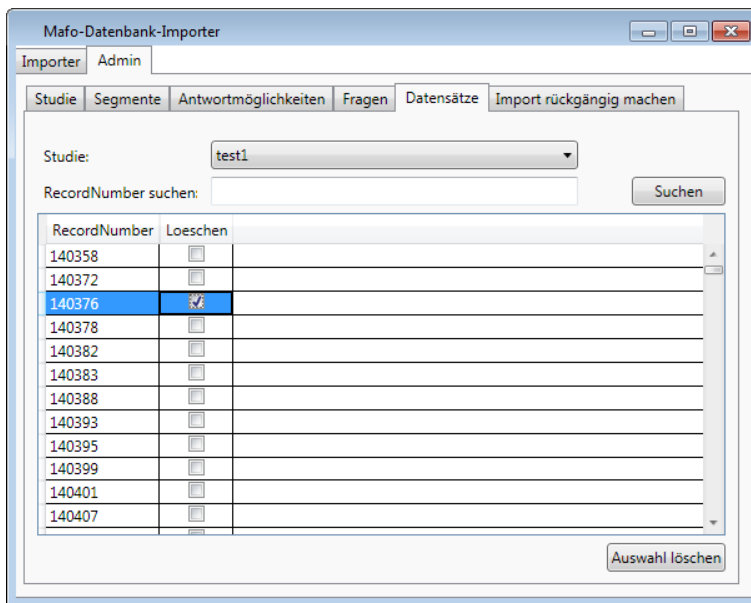


Abbildung 7 - Userinterface des Administrationsprogramms für die Datensätze

Die Datenbank muss mittels Administrationsprogramm nach einem Import wieder rekonstruiert werden können. Das Userinterface dieser Funktionalitäten ist in Abbildung 8 dargestellt.

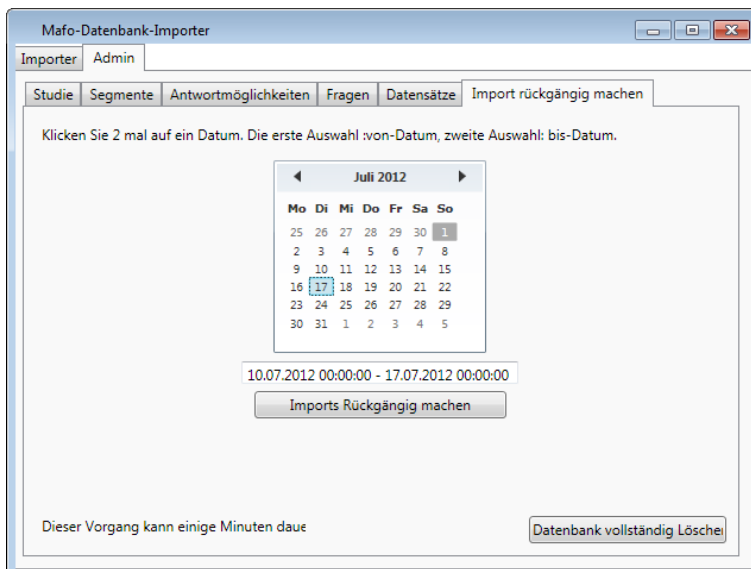


Abbildung 8 - Userinterface des Administrationsprogramms um Imports rückgängig zu machen

3.5.3 Klassendiagramm

Wie für die übrigen kommenden Klassendiagramme wurde die Model-View-Controller-Struktur von Sanderson (2009) angewendet. Die Klassen von ADO.NET entsprechen dem Model und das View ist das Fenster.

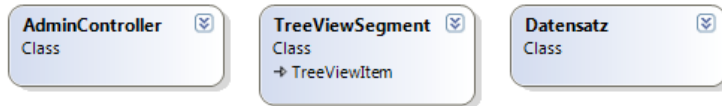


Abbildung 9 – Klassendiagramm des Administrationsprogramms

3.5.4 Implementation

Alle Funktionen werden auf dieselbe Art und Weise implementiert. Dabei stellt ADO.NET Teile der Funktionen zur Verfügung. Eine Aufgabe beginnt mit einem vom User ausgelösten Event. Dieses Event wird im AdminController erkannt. Der AdminController ruft die Objekte von der Datenbank ab. Danach editiert der User ein Objekt. Anschließend wird das geänderte Objekt in der Datenbank gespeichert.

Eine Ausnahme bilden die Segmente. Diese müssen als Hierarchie gespeichert werden. Diese Hierarchie wird rekursiv durch das TreeView-Control ausgelesen.

3.6 Import-Programm

In diesem Punkt wird ein Programm beschrieben, welches die Daten aus dem Ergebnis einer Umfrage ausliest und anschließend in die Datenbank importiert.

3.6.1 Beschreibung und Anforderungen

Die Hauptkriterien des Importvorganges sind Geschwindigkeit und Benutzerfreundlichkeit. Die zu importierenden Daten stammen aus einem Excelldokument im importierbaren Format. Es sollen eine oder mehrere Studien importiert werden. Der User wird über den Fortschritt des Imports informiert und entscheidet bei Importproblemen, wie fortgefahren wird.

3.6.2 Klassendiagramm

Das Klassendiagramm in Abbildung 10 orientiert sich am Datenmodell. Die Controllerklasse steuert den Import.

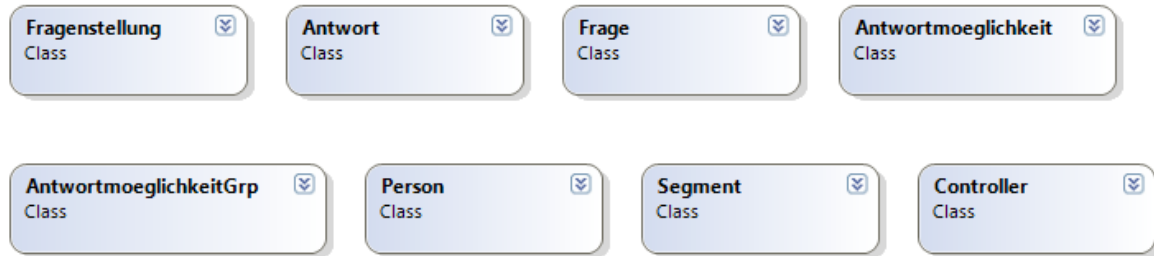


Abbildung 10 – Basisklassen des Importer-Programms

Die drei Fenster in Abbildung 11 sind Dialogfenster für auftretende Probleme.

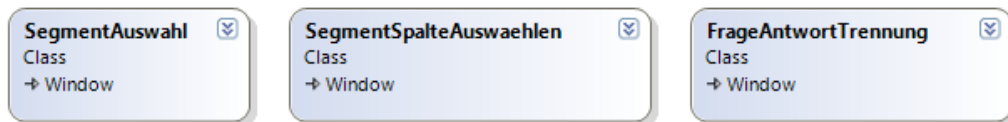


Abbildung 11 – Fensterklassen des Importer-Programms

In Abbildung 12 sind die Klassen die das Exceldokument abstrahierten. Das ExcelFile-Objekt speichert die Daten des Exceldokumentes, wobei die Spalte die Werte als Liste speichert.

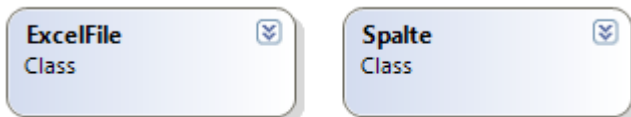


Abbildung 12 – Excelklassen des Importer-Programms

Die Funktionalitäten für den BulkCopy-Vorgang werden als Klassen in Abbildung 13 zur Verfügung gestellt. Diese Klassen können Fragenstellungen und Antworten importieren.

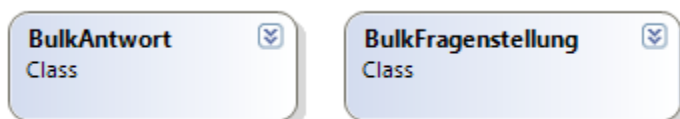


Abbildung 13 – Hilfsklassen des Importer-Programms

Zusätzlich kommen noch die abstrahierten Klassen des Datenmodells von ADO.NET hinzu.

3.6.3 Userinterface

Das Userinterface ist funktional. Die Funktion ist es ausgewertete Fragebögen zu importieren. Diese Auswertungen gehören einer Studie an und werden in einem Exceldokument festgehalten.

Während des Importvorganges werden im Userinterface alle Schritte angezeigt. Ein Ladenbalken informiert über den Gesamtfortschritt. Bei Fehlern oder fehlender Information werden Dialogfenster geöffnet.

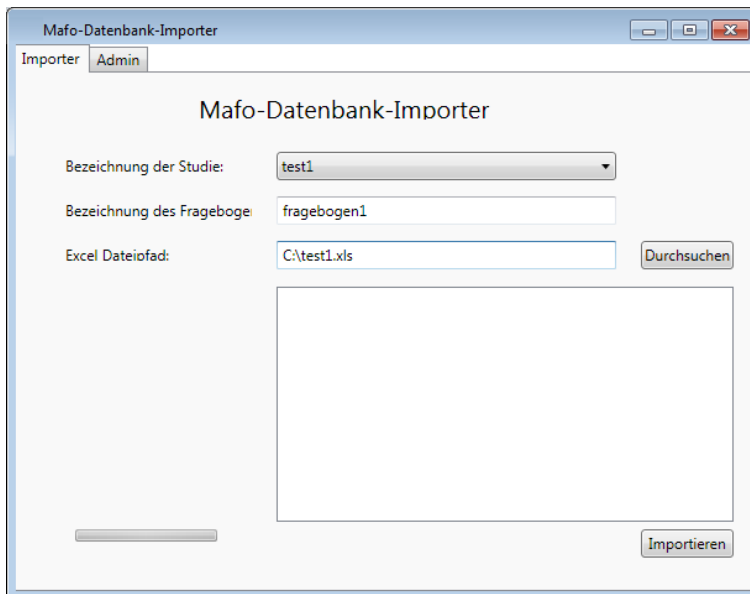


Abbildung 14 – Userinterface des Importer-Programms

3.6.4 Ablauf des Importvorganges

In diesem Punkt wird ein Import beschrieben.

3.6.4.1 Threading

Der Importvorgang startet in einem neuen Thread. Das garantiert, dass der Importvorgang von anderen gleichzeitigen Importvorgängen abgekapselt ist. Ein weiterer Grund ist, dass der Importvorgang nicht abhängig vom Fenster ist. Importvorgänge werden auch abgeschlossen wenn das Fenster nicht existiert.

3.6.4.2 Excel einlesen

Zum Einlesen von Exceldokumenten wird die .NET Bibliothek „Microsoft.Office.Interop.Excel“ mit der Version 14.0.0.0 verwendet.

3.6.4.3 Überprüfung der eingelesenen Daten

Nach dem Einlesen werden die Daten auf ihre Korrektheit überprüft. Bei behebbaren Problemen wird mit dem User interagiert. Ist das Problem kritisch, wird der Vorgang abgebrochen und der User informiert.

3.6.4.4 Zusammenfassen der eingelesenen Daten

Die eingelesenen Daten werden als Objekte strukturiert.

3.6.4.5 Überprüfen auf Existenz in der Datenbank

Die strukturierten Objekte werden auf ihre Existenz in der Datenbank überprüft. Bereits existierenden Daten werden in den Objekten vermerkt. Der Primary Key wird ebenfalls im Objekt gespeichert.

3.6.4.6 Einspielen der Daten in die Datenbank

Es werden alle zu importierenden Objekte in folgender Reihenfolge eingespielt:

- Fragebogen
- Kunden
- Fragen
- Fragestellung
- Antwortmöglichkeitengruppen
- Antwortmöglichkeiten
- Antworten

Die Einhaltung der Reihenfolge ist wichtig, um die Foreign Keys bereits beim Import zuzuweisen zu können.

3.7 Mafo-Dashboard

In diesem Punkt wird beschrieben, wie man auf Basis des Datenmodells ein Dashboard entwickelt. Ein Dashboard stellt Informationen aus einer Quelle dar. Darin enthalten ist die Verwendung der Auswertungsmöglichkeiten, die das Datenmodell zur Verfügung stellt.

3.7.1 Beschreibung und Anforderungen

Das Marktforschungsdashboard soll dem User eine einfach zu bedienende Möglichkeit geben, Daten aus der Datenbank nach eigenen Kriterien auszuwerten. Als Ergebnis wird eine Excelstatistik mit einheitlichem Standard ausgegeben.

3.7.2 Userinterface

Das Userinterface in der Abbildung 15 ist an die Funktionen des Dashboards angepasst.

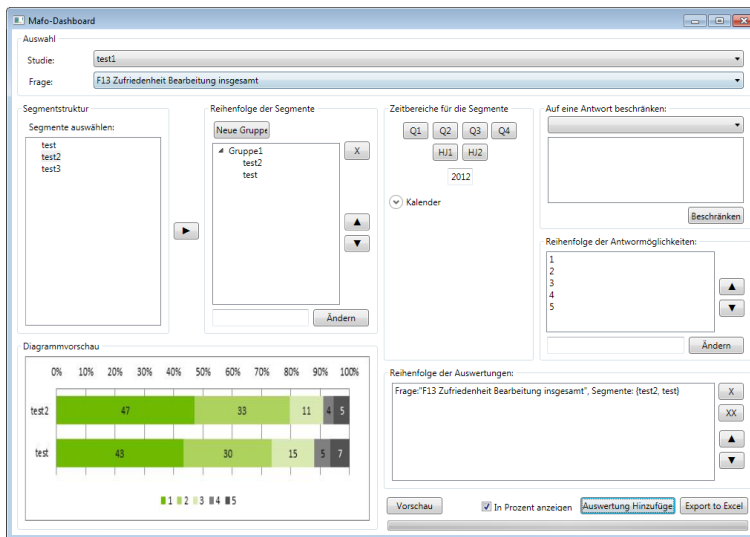


Abbildung 15 – Userinterfaces des Dashboards

3.7.3 Funktionsumfang

Das Dashboard erstellt Auswertungen. Eine Auswertung besteht aus einer Reihe von Auswahlmöglichkeiten. Die erste notwendige Auswahl ist die Studie. Danach muss eine Frage der gewählten Studie ausgewählt werden. Als letzter notwendiger Punkt, werden die Segmente ausgewählt. Dazu werden die Segmente in der gespeicherten Hierarchie der Datenbank angezeigt.

Ausgewählte Segmente können optional nach Zeitraum und gegebener Antwort beschränkt werden. Die Reihung der Antwortmöglichkeiten und Segmente ist optional. Segmente können gereiht und gruppiert werden.

Es können mehrere Auswertungen vorgemerkt und durchgeführt werden.

3.7.4 Klassendiagramm

Das Klassendiagramm in Abbildung 16 ist am Userinterface des Dashboards orientiert. Die Controls des Fensters verwenden die Objekte direkt.

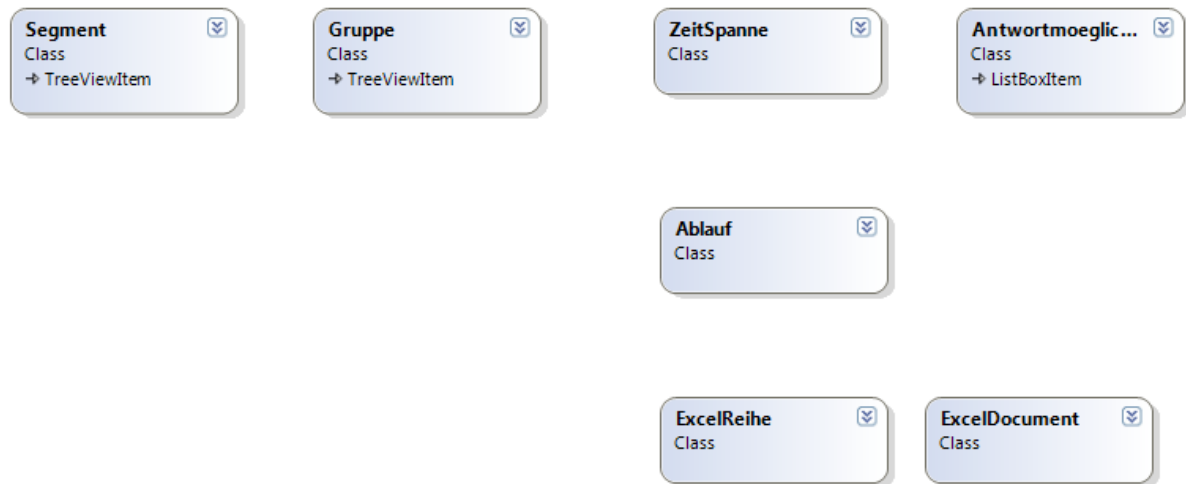


Abbildung 16 – Klassendiagramm des Dashboards

4 User Test

Es wurde auf 3 Arten getestet:

Während der Entwicklungsphase der Programme, wurde das Test-Driven Development von Pugh (2010) angewendet. Während dieser Phase wurden User Acceptance Tests von Pugh (2010) durchgeführt, um die Usability zu testen. Die Programme wurden mittels Blackbox-Test auf Ihre Funktionalitäten getestet. Jeder Test wurde mit Echtdaten durchgeführt.

5 Zusammenfassung und Ausblicke

Das Datenmodell wurde mit verschiedenen Studien in verschiedenem Umfang eingesetzt und hat sich als praktikable Methode für Auswertungen von Umfragen herausgestellt. Das Datenmodell ist flexibel und erweiterbar. Ein Blick in die Zukunft zeigt, dass es voraussichtlich möglich wäre, das Datenmodell für Echtzeitauswertungen einzusetzen. Ein Blick in die Zukunft zeigt eine mögliche Einsetzbarkeit des Datenmodells für Echtzeitauswertungen.

6 Abbildungsverzeichnis

Abbildung 1 - Darstellung des Datenmodells	5
Abbildung 2 – Übersicht der Programmarchitektur	8
Abbildung 3 – Userinterface des Administrationsprogramms für die Studie	10
Abbildung 4 - Userinterface des Administrationsprogramms für die Segmente...	10
Abbildung 5 - Userinterface des Administrationsprogramms für die Antwortmöglichkeiten	11
Abbildung 6 - Userinterface des Administrationsprogramms für die Fragen	11
Abbildung 7 - Userinterface des Administrationsprogramms für die Datensätze .	12
Abbildung 8 - Userinterface des Administrationsprogramms um Imports rückgängig zu machen	12
Abbildung 9 – Klassendiagramm des Administrationsprogramms	13
Abbildung 10 – Basisklassen des Importer-Programms.....	14
Abbildung 11 – Fensterklassen des Importer-Programms	14
Abbildung 12 – Excelklassen des Importer-Programms.....	14
Abbildung 13 – Hilfsklassen des Importer-Programms.....	14
Abbildung 14 – Userinterface des Importer-Programms.....	15
Abbildung 15 – Userinterfaces des Dashboards	17
Abbildung 16 – Klassendiagramm des Dashboards.....	18

7 Literaturverzeichnis

Connolly, T., & Begg, C. (2009). *Database Systems: A Practical Approach to Design, Implementation and Management*. Prentice Hall.

Gräf, L. (o. Jg.). *schoolpark*. Abgerufen am 29. 06 2012 von http://www.schoolpark.at/site/files/artikel_optimierung_www_umfragen.pdf

Küttler, L. (28. 04 2011). *tu-dresden*. Abgerufen am 29. 06 2012 von http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_mathematik_und_naturwissenschaften/fachrichtung_psychologie/i2/differentielle/Tut%20WisAr/Onlinefragebogen_1.pdf

Pugh, K. (2010). *Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration*. Addison Wesley Professional.

Sanderson, S. (2009). *Pro ASP.NET MVC Framework*. Apress.

ANHANG A

Das folgende Skript kann zum Generieren des fertigen Datenmodells verwendet werden.

```
/****** Object: Table [dbo].[Tbl_AntwortmoeglichkeitGrp]  Script Date: 04/20/2012 11:00:45  
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Tbl_AntwortmoeglichkeitGrp](  
    [ID_AntwortmoeglichkeitGrp] [int] IDENTITY(1,1) NOT NULL,  
    CONSTRAINT [Tbl_AntwortmoeglichkeitGrp_PK] PRIMARY KEY NONCLUSTERED  
(  
    [ID_AntwortmoeglichkeitGrp] ASC  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)  
)  
  
GO
```

```
/****** Object: Table [dbo].[Tbl_Antwortmoeglichkeit]  Script Date: 04/20/2012 11:00:31 *****/
```


SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[Tbl_Antwortmoeglichkeit](

[ID_Antwortmoeglichkeit] [int] IDENTITY(1,1) NOT NULL,

[TextAntwortMoeglichkeit] [varchar](200) NULL,

[Wertigkeit] [int] NULL,

[Zahl] [int] NULL,

[FK_AntwortmoeglichkeitGrp] [int] NULL,

CONSTRAINT [Antwortmoeglichkeit_PK] PRIMARY KEY NONCLUSTERED

(

[ID_Antwortmoeglichkeit] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)

)

GO

SET ANSI_PADDING OFF

GO

```
ALTER TABLE [dbo].[Tbl_Antwortmoeglichkeit] WITH CHECK ADD CONSTRAINT
[Tbl_Antwortmoeglichkeit_Tbl_AntwortmoeglichkeitGrp_FK] FOREIGN
KEY([FK_AntwortmoeglichkeitGrp])

REFERENCES [dbo].[Tbl_AntwortmoeglichkeitGrp] ([ID_AntwortmoeglichkeitGrp])

GO
```

```
ALTER TABLE [dbo].[Tbl_Antwortmoeglichkeit] CHECK CONSTRAINT
[Tbl_Antwortmoeglichkeit_Tbl_AntwortmoeglichkeitGrp_FK]

GO
```

```
/***** Object: Table [dbo].[Tbl_Kunde]  Script Date: 04/20/2012 10:59:10 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[Tbl_Kunde](
    [ID_Kunde] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](100) NULL,
    [Number] [varchar](15) NULL,
    CONSTRAINT [Tbl_Kunde_PK] PRIMARY KEY NONCLUSTERED
```

```
(  
    [ID_Kunde] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)  
)
```

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[Tbl_Frageart] Script Date: 04/20/2012 11:03:04 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[Tbl_Frageart](  
    [ID_Frageart] [int] IDENTITY(1,1) NOT NULL,  
    [FrageartBezeichnung] [varchar](30) NULL,  
    CONSTRAINT [Tbl_Antwortart_PK] PRIMARY KEY NONCLUSTERED  
)
```

```
        [ID_Frageart] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
)
```

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[Tbl_Frage] Script Date: 04/20/2012 11:02:48 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[Tbl_Frage](
    [ID_Frage] [int] IDENTITY(1,1) NOT NULL,
    [FK_Frageart] [int] NULL,
    [Frage] [varchar](500) NULL,
    [FragenZahl] [int] NULL,
```

```

[FragenStelle] [int] NULL,

CONSTRAINT [Tbl_Frage_PK] PRIMARY KEY NONCLUSTERED

(

    [ID_Frage] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)

)

GO


SET ANSI_PADDING OFF

GO


ALTER TABLE [dbo].[Tbl_Frage] WITH CHECK ADD CONSTRAINT [Tbl_Frage_Tbl_Frageart_FK]
FOREIGN KEY([FK_Frageart])

REFERENCES [dbo].[Tbl_Frageart] ([ID_Frageart])

GO


ALTER TABLE [dbo].[Tbl_Frage] CHECK CONSTRAINT [Tbl_Frage_Tbl_Frageart_FK]

GO


/***** Object: Table [dbo].[Tbl_Studie] Script Date: 04/20/2012 11:04:03 *****/

SET ANSI_NULLS ON

GO


SET QUOTED_IDENTIFIER ON

```

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[Tbl_Studie](
    [ID_Studie] [int] IDENTITY(1,1) NOT NULL,
    [Bezeichnung] [varchar](100) NULL,
    CONSTRAINT [Tbl_Studie_PK] PRIMARY KEY NONCLUSTERED
(
    [ID_Studie] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
)
```

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[Tbl_Fragenkatalog] Script Date: 04/20/2012 11:03:17 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[Tbl_Fragenkatalog](
    [ID_Fragenkatalog] [int] IDENTITY(1,1) NOT NULL,
    [FK_Studie] [int] NULL,
    [Bezeichnung] [varchar](100) NULL,
    CONSTRAINT [Tbl_Fragenkatalog_PK] PRIMARY KEY NONCLUSTERED
(
    [ID_Fragenkatalog] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
)
```

GO

SET ANSI_PADDING OFF

GO

```
ALTER TABLE [dbo].[Tbl_Fragenkatalog] WITH CHECK ADD CONSTRAINT
[Tbl_Fragenkat_Tbl_Studie_FK] FOREIGN KEY([FK_Studie])
REFERENCES [dbo].[Tbl_Studie] ([ID_Studie])
GO
```

```
ALTER TABLE [dbo].[Tbl_Fragenkatalog] CHECK CONSTRAINT [Tbl_Fragenkat_Tbl_Studie_FK]
```

GO

/***** Object: Table [dbo].[Tbl_Interviewer] Script Date: 04/20/2012 11:03:41 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[Tbl_Interviewer](

[ID_Interviewer] [int] IDENTITY(1,1) NOT NULL,

[Name] [varchar](100) NULL,

CONSTRAINT [Tbl_Interviewer_PK] PRIMARY KEY NONCLUSTERED

(

[ID_Interviewer] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)

)

GO

SET ANSI_PADDING OFF

GO

/***** Object: Table [dbo].[Tbl_Segment] Script Date: 04/20/2012 11:03:52 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[Tbl_Segment](

[ID_Segment] [int] IDENTITY(1,1) NOT NULL,

[FK_Ebene] [int] NULL,

[Bezeichnung] [varchar](100) NULL,

[Gewichtung] [int] NULL,

[FK_UebergeordnetesSegment] [int] NULL,

[FK_Studie] [int] NULL,

CONSTRAINT [Tbl_Segment_PK] PRIMARY KEY NONCLUSTERED

(

[ID_Segment] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)

)

GO

SET ANSI_PADDING OFF

GO

ALTER TABLE [dbo].[Tbl_Segment] WITH CHECK ADD CONSTRAINT [Tbl_Segment_Tbl_Segment_FK]
FOREIGN KEY([FK_UebergeordnetesSegment])

REFERENCES [dbo].[Tbl_Segment] ([ID_Segment])

GO

ALTER TABLE [dbo].[Tbl_Segment] CHECK CONSTRAINT [Tbl_Segment_Tbl_Segment_FK]

GO

ALTER TABLE [dbo].[Tbl_Segment] WITH CHECK ADD CONSTRAINT [Tbl_Segment_Tbl_Studie_FK]
FOREIGN KEY([FK_Studie])

REFERENCES [dbo].[Tbl_Studie] ([ID_Studie])

GO

ALTER TABLE [dbo].[Tbl_Segment] CHECK CONSTRAINT [Tbl_Segment_Tbl_Studie_FK]

GO

/***** Object: Table [dbo].[Tbl_Fragenstellung] Script Date: 04/20/2012 11:03:31 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```

CREATE TABLE [dbo].[Tbl_Fragenstellung](
    [ID_Fragenstellung] [int] NOT NULL,
    [FK_Frage] [int] NULL,
    [FK_Kunde] [int] NULL,
    [FK_Segment] [int] NULL,
    [FK_Interviewer] [int] NULL,
    [FK_Fragenkatalog] [int] NULL,
    [ErstellungsDatum] [datetime] NULL,
    [AusfüllDatum] [datetime] NULL,
    [RecordNumber] [int] NULL,
    [InDatenbankEingespielt] [datetime] NULL,
    CONSTRAINT [Tbl_Fragenstellung_PK] PRIMARY KEY NONCLUSTERED
(
    [ID_Fragenstellung] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
)

GO

ALTER TABLE [dbo].[Tbl_Fragenstellung] WITH NOCHECK ADD CONSTRAINT [Tbl_Fragenstel_Tbl_
Kunde_FK] FOREIGN KEY([FK_Kunde])
REFERENCES [dbo].[Tbl_Kunde] ([ID_Kunde])
GO

ALTER TABLE [dbo].[Tbl_Fragenstellung] CHECK CONSTRAINT [Tbl_Fragenstel_Tbl_Kunde_FK]
GO

```

```
ALTER TABLE [dbo].[Tbl_Fragenstellung] WITH NOCHECK ADD CONSTRAINT  
[Tbl_Fragenstel_Tbl_Frage_FK] FOREIGN KEY([FK_Frage])
```

```
REFERENCES [dbo].[Tbl_Frage] ([ID_Frage])
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Fragenstellung] CHECK CONSTRAINT [Tbl_Fragenstel_Tbl_Frage_FK]
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Fragenstellung] WITH NOCHECK ADD CONSTRAINT  
[Tbl_Fragenstel_Tbl_Fragenkat_FK] FOREIGN KEY([FK_Fragenkatalog])
```

```
REFERENCES [dbo].[Tbl_Fragenkatalog] ([ID_Fragenkatalog])
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Fragenstellung] CHECK CONSTRAINT [Tbl_Fragenstel_Tbl_Fragenkat_FK]
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Fragenstellung] WITH NOCHECK ADD CONSTRAINT  
[Tbl_Fragenstel_Tbl_Intervie_FK] FOREIGN KEY([FK_Interviewer])
```

```
REFERENCES [dbo].[Tbl_Interviewer] ([ID_Interviewer])
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Fragenstellung] CHECK CONSTRAINT [Tbl_Fragenstel_Tbl_Intervie_FK]
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Fragenstellung] WITH NOCHECK ADD CONSTRAINT  
[Tbl_Fragenstel_Tbl_Segment_FK] FOREIGN KEY([FK_Segment])
```

```
REFERENCES [dbo].[Tbl_Segment] ([ID_Segment])
```

GO

ALTER TABLE [dbo].[Tbl_Fragenstellung] CHECK CONSTRAINT [Tbl_Fragenstel_Tbl_Segment_FK]

GO

/***** Object: Table [dbo].[Tbl_Antwort] Script Date: 04/20/2012 11:00:18 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[Tbl_Antwort](
    [ID_Antwort] [int] IDENTITY(1,1) NOT NULL,
    [AntwortNummer] [int] NULL,
    [AntwortTyp] [varchar](50) NULL,
    [AntwortFreiText] [varchar](500) NULL,
    [FK_Antwortmoeglichkeit] [int] NULL,
    [FK_Fragenstellung] [int] NULL,
    [InDatenbankEingespelt] [datetime] NULL,
    CONSTRAINT [Tbl_Antwort_PK] PRIMARY KEY NONCLUSTERED
(
    [ID_Antwort] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
```

```
)
```

```
GO
```

```
SET ANSI_PADDING OFF
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Antwort] WITH NOCHECK ADD CONSTRAINT  
[Tbl_Antwort_Tbl_Antwortmoeglichkeit] FOREIGN KEY([FK_Antwortmoeglichkeit])  
REFERENCES [dbo].[Tbl_Antwortmoeglichkeit] ([ID_Antwortmoeglichkeit])
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Antwort] CHECK CONSTRAINT [Tbl_Antwort_Tbl_Antwortmoeglichkeit]
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Antwort] WITH NOCHECK ADD CONSTRAINT  
[Tbl_Antwort_Tbl_Fragenstellung_FK] FOREIGN KEY([FK_Fragenstellung])  
REFERENCES [dbo].[Tbl_Fragenstellung] ([ID_Fragenstellung])
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_Antwort] CHECK CONSTRAINT [Tbl_Antwort_Tbl_Fragenstellung_FK]
```

```
GO
```