

**Fachbereich**  
**Automatisierung und Informatik**

▲ Hochschule Harz  
Wernigerode

**Handbuch zum  
Datenbank-Designer  
und der  
FBConsole**

***Version 4,74***

**von**

**Dipl. Inf., Dipl.-Ing. (FH) Michael Wilhelm**

**Version 12. Februar 2016**

**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](mailto:mwilhelm@hs-harz.de)**

**Web: <http://www.miwilhelm.de>**

**<http://www.miwilhelm.de/datenbanken/designer/index.html>**

**<http://www.miwilhelm.de/datenbanken/dbspaket/index.html>**

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung .....</b>	<b>9</b>
1.1	Eigenschaften des Designer-Programms	9
1.2	Die Eigenschaften in Kurzform:	9
1.3	Wichtige neue Eigenschaften der Version 4	10
1.4	Ablauf bei der Erstellung eines konzeptionellen Modells	11
1.5	Ablauf bei der Erstellung eines logischen Modells	11
<b>2</b>	<b>FBConsole.....</b>	<b>12</b>
<b>3</b>	<b>Zip-Datei.....</b>	<b>12</b>
3.1	Installation	12
3.2	Inhalt der ZIP-Datei	12
<b>4</b>	<b>Beispiele .....</b>	<b>13</b>
4.1	Entwicklung eines konzeptionelles ER-Modells	13
4.1.1	Entitäten	13
4.1.2	Beziehungen:	13
4.1.3	Neues Projekt	13
4.1.4	Entitiy anlegen	14
4.1.4.1	Attribute des Entity „Kunde“	16
4.1.4.2	Attribute des Entity „Ort“	18
4.1.4.3	Attribute des Entity „Artikel“	18
4.1.5	Beziehungen anlegen	18
4.1.6	Weitere Schritte	20
4.1.6.1	Default-Werte definieren	21
4.1.6.2	Unique-Attribute definieren	22
4.1.7	Umwandlung in ein logisches Modell	23
4.1.8	Weitere Schritte	24
4.1.8.1	Check-Constraint definieren	24
4.1.8.2	Generator oder Sequenz definieren	28
4.1.8.3	Ergebnis des „Handbuch Bsp1.dbl“	31
4.2	Entwicklung eines konzeptionelles ER-Modells	33
4.2.1	Entitäten	33
4.2.2	Beziehungen:	33
4.2.3	Neues Projekt	33
4.2.4	Entitiy anlegen	34
4.2.4.1	Attribute des Entity „Mitarbeiter	35
4.2.4.2	Attribute des Entity „Abteilung	41
4.3	Entwicklung eines konzeptionelles ER-Modells	46
4.3.1	Entitäten	47
4.3.2	Beziehungen:	47
4.3.3	Neues Projekt	47
4.3.4	Entitiy anlegen	47
4.3.4.1	Attribute des Entity „Kunden	47
4.3.4.2	Attribute des Entity „Bestellung	48
4.4	Entwicklung eines konzeptionelles ER-Modells	52
4.4.1	Entitäten	52

4.4.2	Beziehungen:	52
4.4.3	Neues Projekt	52
4.4.4	Entitiy anlegen	53
4.4.4.1	Attribute des Entity „Lieferant	53
4.4.4.2	Attribute des Entity „Teil“	53
4.4.4.3	Attribute des Entity „Projekt“	54
4.5	Beispiel5, Entwicklung eines logischen ER-Modell	58
4.5.1	Aufgabenstellung des fünften Beispiels	58
4.5.2	Neues Projekt	58
4.5.3	Entities anlegen	60
4.5.4	Beziehungen anlegen	64
4.5.5	Beziehung der Vorlesungen	69
4.5.6	Weitere Schritte	74
4.5.7	Zusätzliche SQL-Befehle	74
4.5.8	Generierung einer Datenbank	75
<b>5</b>	<b>FBConsole.....</b>	<b>78</b>
5.1	Neue Datenbank erstellen	80
5.2	Daten eintragen	82
5.2.1	SQL-Befehle zum Eintragen:	82
5.3	Testabfragen	83
<b>6</b>	<b>Anhang.....</b>	<b>86</b>
6.1	Generator	86
6.2	Sequenz	86
6.3	Daten-Eintrag	87
<b>7</b>	<b>SQL-Beispiele.....</b>	<b>88</b>
7.1	Daten für die Tabelle „Vorlesung“	88
7.2	Daten für die Tabelle „Student“	88
7.3	Daten für die Tabelle „Belegt“	89
7.4	Update-Befehle	89
7.4.1.1	Delete-Befehle	90
<b>8</b>	<b>Beispiele .....</b>	<b>91</b>
8.1	Beispiel6, Entwicklung eines logischen ER-Modell	91
8.1.1	Aufgabenstellung des sechsten Beispiels	91
8.1.2	Neues Projekt	91
8.1.3	Entities anlegen	92
8.1.4	Beziehungen anlegen	95
8.1.5	Beziehung der Hobbies	98
8.1.6	Self-Relation	101
8.1.7	Weak-Relation	104
8.1.8	Weitere Schritte	106
8.1.9	Generierung einer Datenbank	108
8.1.10	Erstellen der Datenbank	111
8.1.11	Daten für die Tabelle „Dept“	112
8.1.12	Daten für die Tabelle „Emp“	114
8.2	Generator für die Mitarbeiternummer EMPNO	115

<b>9 Funktionen .....</b>	<b>116</b>
9.1 Menüfunktionen	116
9.1.1 Menü Datei	116
9.1.2 Menü Projekt	116
9.1.3 Menü Einfügen	118
9.1.4 Menü Modell	118
9.1.5 Menü Schema	118
9.1.6 Menü Optionen	119
9.2 Funktionen des Objekt-Baumes	120
9.3 Funktionen der Grafik	121
9.3.1 Funktionen des Grafik-Panels	121
9.3.2 Entity-Funktionen	122
9.4 Funktionen der Attribute	123
<b>10 Spezielle Funktionen.....</b>	<b>125</b>
10.1 Computed By	125
10.1.1 Syntax von Computed by	125
10.2 Weak-Attribute	126
10.3 Self-Relation	126
10.4 Ternäre-Relation	126
10.5 Erstellung einer Datenbank-Typdatei	126
10.5.1 Aufruf der Funktion	126
10.5.2 SQL-Syntax	127
10.6 Generator / Sequenz	129
10.6.1 Sequenz in der Datenbank Firebird	130
10.7 Sequenz für die Abteilungsnummer Deptno	130
10.7.1 Erzeugen eines Triggers mittels SQL-Code:	135
10.7.2 Sequenz in Oracle	136
<b>11 Literatur .....</b>	<b>137</b>
<b>12 Indexverzeichnis .....</b>	<b>139</b>

## Abbildungsverzeichnis

Abbildung 1	Überblick des Programms	9
Abbildung 2	Projekteigenschaften (Integer vs. Max Numeric)	14
Abbildung 3	Erzeugen eines Entities	14
Abbildung 4	Neue Entities	15
Abbildung 5	Haupt-Eigenschaften eines Entities im konzeptionellen Modells	15
Abbildung 6	Bearbeiten, Erzeugen, Löschen von Attributen	16
Abbildung 7	Kundennummer einfügen	16
Abbildung 8	Attribute des Entities Kunde	17
Abbildung 9	Entitäten mit Attributen des Kunden	17
Abbildung 10	ER-Modell des ersten Beispiels	18
Abbildung 11	Erstellen der Beziehung Ort zu Kunde	19
Abbildung 12	Korrekte Einstellung für das erste Beispiel	19
Abbildung 13	Definition der Beziehung Ort zu Kunde	19
Abbildung 14	Zweite Beziehung im ersten Beispiel	20
Abbildung 15	Konzeptionelles Modell des ersten Beispiels	20
Abbildung 16	Attribute des Entity Ort	21
Abbildung 17	Postleitzahl auf Länge 5	21
Abbildung 18	Postleitzahl als Default-Wert	22
Abbildung 19	Unique-Constraint für den Ortsnamen einfügen	22
Abbildung 20	Zieldatenbank Firebird	23
Abbildung 21	Portierungsmeldung	23
Abbildung 22	Das logische Modell des ersten Beispiels	24
Abbildung 23	Attribut GebDatum	25
Abbildung 24	Check-Bedingung	25
Abbildung 25	Neue Definition des GebDatums	26
Abbildung 26	Check-Bedingung	26
Abbildung 27	Erzeugen des Attributs Preis	27
Abbildung 28	Check-Bedingung des Preises	28
Abbildung 29	Definition einer Sequenz	29
Abbildung 30	Definition einer Sequenz	30
Abbildung 31	Logisches ER-Modell des ersten Beispiels	31
Abbildung 32	Erzeugen eines SQL-Skriptes	31
Abbildung 33	Erzeugen eines Entities	34
Abbildung 34	Erzeugen eines Entities	34
Abbildung 35	Neue Entities	34
Abbildung 36	Startdialog für das 2. Beispiel	35
Abbildung 37	Attributdialog	35
Abbildung 38	Einfügen des Attribut Mitarbeiter	36
Abbildung 39	Einfügen des Attribut Name	36
Abbildung 40	Einfügen des Defaultwerts für Name	37
Abbildung 41	Einfügen des Attribut Vorname	37
Abbildung 42	Einfügen des Defaultwerts Vorname	38
Abbildung 43	Einfügen des Attribut Strasse	38
Abbildung 44	Defaultwert für Strasse	39
Abbildung 45	Einfügen des Attribut PLZ	39
Abbildung 46	Defaultwert für das Attribut PLZ	40
Abbildung 47	Einfügen des Attribut Ort	40
Abbildung 48	Haupt-Eigenschaften des Entities Mitarbeiters	41

Abbildung 49	Neues Attributs "Nr"	41
Abbildung 50	Default-Wert für das Attribut "Nr"	42
Abbildung 51	Neues Attributs "Standorte" mit Multi-Attribut	42
Abbildung 52	Check-Bedingung für das Attribut "Standorte"	43
Abbildung 53	Haupt-Eigenschaften des Entities Abteilung	43
Abbildung 54	Attribut als Multiattribut	44
Abbildung 55	Anzeige der beiden Entities des 2. Beispiels	44
Abbildung 56	Auswahl der Entities der Relation	44
Abbildung 57	Relation Mitarbeiter zu Abteilung	45
Abbildung 58	Standorte als Multi-Attribute	45
Abbildung 59	Auswahl der Datenbank für das logische Modell	46
Abbildung 60	Multi-Attribute umgesetzt im logischen Modell	46
Abbildung 61	Neue Entities	47
Abbildung 62	Haupt-Eigenschaften des Entities Kunden	48
Abbildung 63	Neues Attribut "Menge"	48
Abbildung 64	Default-Wert für das Attribut "Menge"	49
Abbildung 65	Haupt-Eigenschaften eines Entities	49
Abbildung 66	Auswahl der Entities der Relation	50
Abbildung 67	Relation Kunden zu Bestellung	50
Abbildung 68	Attribut Datum und Rabatt in einer Relation	50
Abbildung 69	Standorte als Multi-Attribute	51
Abbildung 70	Hinweis, Hint, einer Relation	51
Abbildung 71	Relationen-Attribute umgesetzt im logischen Modell	51
Abbildung 72	Ternäre Beziehung im konzeptionellen Modells	52
Abbildung 73	Neue Entities	53
Abbildung 74	Haupt-Eigenschaften des Entities Lieferant	53
Abbildung 75	Haupt-Eigenschaften des Entities Teil	54
Abbildung 76	Haupt-Eigenschaften des Entities Teil	54
Abbildung 77	Auswahl der Entities der ternären Relation	55
Abbildung 78	Relation Kunden zu Bestellung	55
Abbildung 79	Attribut Datum und Menge in einer ternären Relation	56
Abbildung 80	ER-Modell einer ternären Beziehung	56
Abbildung 81	Hinweis, Hint, einer Relation	56
Abbildung 82	Relationen-Attribute umgesetzt im logischen Modell	57
Abbildung 83	Beispiel Handbuch Bsp5.DBL	58
Abbildung 84	Auswahl der Datenbank	59
Abbildung 85	Auswahl der Darstellungen	59
Abbildung 86	Projekteigenschaften (Integer vs. Max Numeric)	60
Abbildung 87	Erzeugen eines Entities	60
Abbildung 88	Entites des zweiten Beispiels	61
Abbildung 89	Neues Attribut "Mnr"	61
Abbildung 90	Neues Attribut "Name"	62
Abbildung 91	Default-Wert für Name eintragen	62
Abbildung 92	Neues Attribut "FB" als Primary Key" (Datentyp Zeichenkette)	63
Abbildung 93	Entities ohne die Beziehungen	63
Abbildung 94	Haupt-Eigenschaften des Entities Student	64
Abbildung 95	Erstellen einer Beziehung	64
Abbildung 96	Korrekte Einstellung für das Beispiel	65
Abbildung 97	Definition der Beziehung	66
Abbildung 98	Schalter "Link-Attribute" gedrückt, Frage nach der Übertragung des Attribut	67
Abbildung 99	Eingabe des Foreignkey-Namens	67

Abbildung 100	Übertragen eines Fremdschlüssels	68
Abbildung 101	Erste Beziehung im fünften Beispiel	68
Abbildung 102	Beziehung Student zu Belegt	69
Abbildung 103	Beziehung „Student“ zu „Belegt Vorlesung“	69
Abbildung 104	Automatische Übertragung des Primary-Keys als Foreign-Key	70
Abbildung 105	Name des Foreign-Keys	70
Abbildung 106	Anzeige der Eingaben der Beziehung	70
Abbildung 107	Aktuelle Fassung mit zwei Beziehungen	71
Abbildung 108	Aufruf Beziehung Vorlesung zu Belegt	71
Abbildung 109	Beziehung „Vorlesung“ zu „Belegt“	72
Abbildung 110	Automatische Übertragung des Primary-Keys als Foreign-Key	72
Abbildung 111	des Foreign-Keys	72
Abbildung 112	Anzeige der Eingaben der Beziehung	73
Abbildung 113	Aktuelle Fassung mit drei Beziehungen	73
Abbildung 114	ER-Modell nach den Beziehungen der Hobbies	74
Abbildung 115	zusätzliche SQL-Anweisungen	75
Abbildung 116	Erzeugung einer Datenbank	76
Abbildung 117	Hauptfenster der FBConsole	78
Abbildung 118	Erstellen der Datenbank "Firma.fdb"	80
Abbildung 119	Eintragen der SQL-Skripte	81
Abbildung 120	Anzeige der Datenbank-Tabellen	81
Abbildung 121	Daten mittels „Insert-Into“ eintragen	82
Abbildung 122	Abfrage bezüglich der Artikel	83
Abbildung 123	Abfrage bezüglich der Kunden	84
Abbildung 124	Abfrage bezüglich der Bestellungen	84
Abbildung 125	Abfrage bezüglich der Bestellungen	84
Abbildung 126	Definition eines Generators	86
Abbildung 127	Auswahl der Datenbank	92
Abbildung 128	Auswahl der Darstellungen	92
Abbildung 129	Erzeugen eines Entities	93
Abbildung 130	Entites des sechsten Beispiels	93
Abbildung 131	Haupt-Eigenschaften eines Entities	94
Abbildung 132	Bearbeiten, Erzeugen, Löschen von Attributen	94
Abbildung 133	Fertige Entities des sechsten Beispiels	95
Abbildung 134	Korrekte Einstellung für das Beispiel	96
Abbildung 135	Definition der Beziehung	96
Abbildung 136	Übertragen der Primär-Attribute	97
Abbildung 137	Übertragen eines Fremdschlüssels	98
Abbildung 138	Erste Beziehung im sechsten Beispiel	98
Abbildung 139	Beziehung Emp zu Has_Hobby	99
Abbildung 140	Beziehung „Emp“ zu „Has_Hobby“	99
Abbildung 141	Aufruf Beziehung Emp zu Hobbies	100
Abbildung 142	Eintrag der Beziehung "Emp" zu "Hobbies"	100
Abbildung 143	ER-Modell nach den Beziehungen der Hobbies	101
Abbildung 144	ER-Modell mit Primary-Key (Has_Hobby)	101
Abbildung 145	Erzeugen einer Self-Relation	102
Abbildung 146	Self-Relation	102
Abbildung 147	EmpNo zu Manager definieren	103
Abbildung 148	Einstellungen für die Darstellung der Self-Beziehung	104
Abbildung 149	Darstellung der Self-Relation	104
Abbildung 150	Beschreibung des Weak-Entities	105
Abbildung 151	Eintragen des Primarykeys als Fremdschlüssels	105

Abbildung 152	Aktivieren des Weak-Entity	106
Abbildung 153	ER-Modell des sechsten Beispiels	106
Abbildung 154	Defaultwert für Budget eingegeben	107
Abbildung 155	Check-Constraint für das Attribut Budget	107
Abbildung 156	Unique-Bedingung für den Abteilungsnamen	108
Abbildung 157	Erzeugung einer Datenbank, ohne Kommentare	109
Abbildung 158	Anzeige der Tabellen, linke Liste	112
Abbildung 159	Abfrage der Daten für die Abteilung	113
Abbildung 160	Test der Check-Bedingung Budget	114
Abbildung 161	Projekt-Eigenschaften	117
Abbildung 162	Grafiklinien	117
Abbildung 163	Generierung des SQL-Scriptes	119
Abbildung 164	Funktionen des Objekt-Baumes	120
Abbildung 165	Eigenschaftsdialog des Baumes	120
Abbildung 166	Eigenschaften des Grafikpanels	121
Abbildung 167	Popup-Funktionen des Grafik-Panels	121
Abbildung 168	Eigenschaften eines Entities	122
Abbildung 169	Register Farben in den Entity-Einstellungen	122
Abbildung 170	Einstellungen der Attribute. Primarykey, logische Modell	123
Abbildung 171	Check-Bedingung	124
Abbildung 172	Dialogfenster zum Erstellen einer Datenbank-Definition	126
Abbildung 173	Datenbank-Definitionsdialog	128
Abbildung 174	Sequenz, Generator definieren	129
Abbildung 145	Sequenz im Trigger	133
Abbildung 146	Fertiger Trigger	134
Abbildung 147	Ergebnis eines Insert-Befehls	135



# 1 Einführung

Das vorliegende Programm soll den Datenbankentwurf vereinfachen. Dazu werden die Elemente (Entities, Relationen etc), grafisch dargestellt. Mit einfachen Befehlen können die Elemente erzeugt und bearbeitet werden.

Es wird vorausgesetzt, dass Grundbegriffe in der Datenbanktechnologie vorhanden sind.

## 1.1 Eigenschaften des Designer-Programms

Die Abbildung 1 zeigt den Aufbau des Designers. Es hat die normale Menü- und Schalterstruktur. Alle Datenbankelemente werden in einem Baum links dargestellt. Die Breite dieses Baumes kann mit der blauen Linie (Splitter) verändert werden. Im rechten Hauptteil werden die Entities und Beziehungen dargestellt.

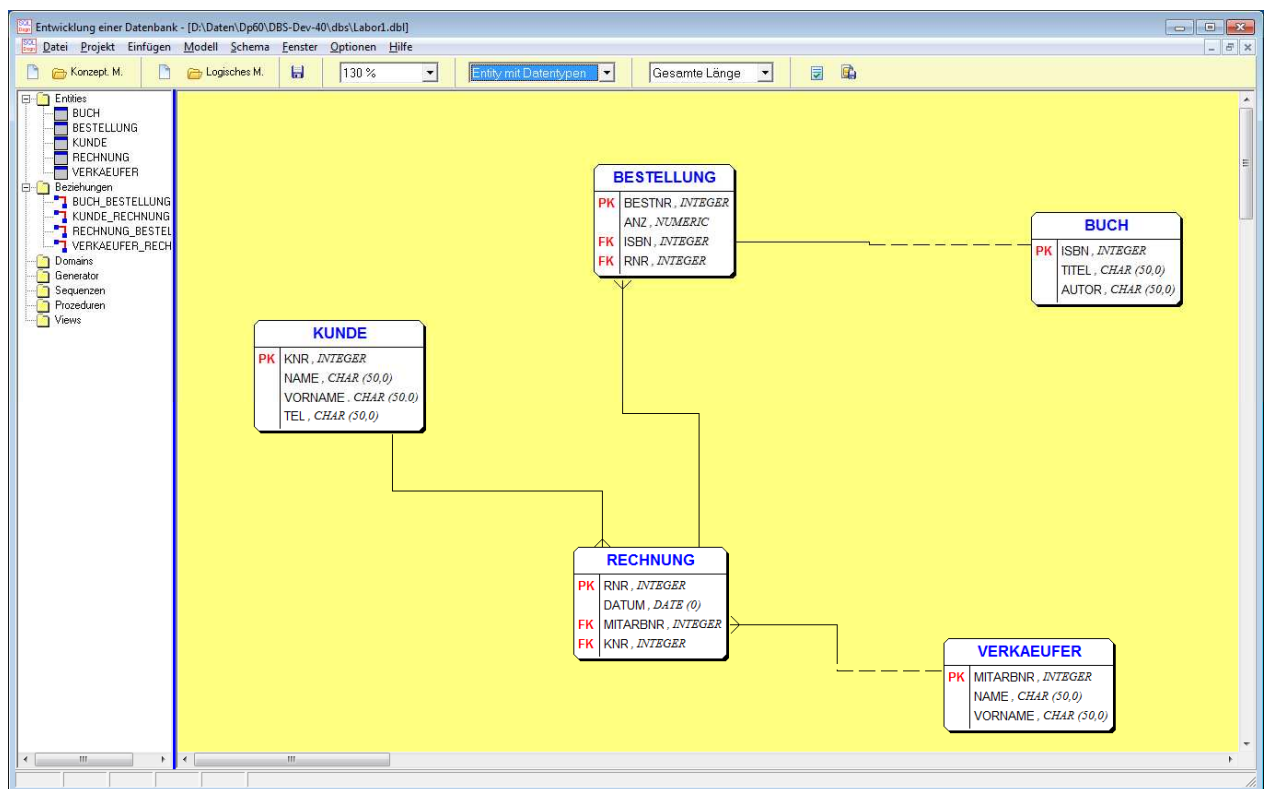


Abbildung 1 Überblick des Programms

## 1.2 Die Eigenschaften in Kurzform:

- Verwendung des konzeptionelles Modell
- Verwendung des logischen Modell
- Automatischer Transfer vom konzeptionellen zum logischen Modell
- Definition der Datenbanktypen (logische Modell)

- Vordefinierte Datenbanken
- Freie Definition einer Datenbank
  - Namen der Datentypen
  - Zuordnung zu den Grunddatentyp
  - SQL-Syntax definieren (Kommentare, Eckige Klammer etc.)
  - Counter definieren (Generator, Sequenz, Autonumber)
  - Liste der reservierten Wörter
- Erzeugung und Verwaltung von Entities
- Erzeugung und Verwaltung von Attributen
- Erzeugung und Verwaltung von Relationen
- Erzeugung und Verwaltung von Generatoren
- Erzeugung und Verwaltung von Sequenzen
- Generierung eines SQL-Skriptes
- Vielfältige Farb- und Schriftwahl
- Anzeige der Relationen mit Krähenfuss oder Punktnotation
- Export der Daten nach Winword
- Export der ERM-Grafik in die Zwischenablage

### 1.3 Wichtige neue Eigenschaften der Version 4

#### Allgemeines:

- Es kann eine zusätzliche Instanz aufgerufen werden (ALT-Tab Umschaltung)
- Kopieren von Entities über die Zwischenablage möglich
- Export in die Zwischenablage als Bild mit Randdefinition
- Überprüfung der Datenbank auf doppelte Namen, Checkbedingungen
- Check-Attribut in Entity-Einst eingetragen
- Es gibt **zusätzliche Anweisungen**, die für Initialisierungen benutzt werden können.
  - Automatische Eingabe von festen Werten in Tabellen

#### Konzeptionelles Modell:

- Es kann nun auch ein Defaultwert eingegeben werden
- Es kann nun auch eine Check-Bedingung eingegeben werden
- Die Länge der Kurznamen wird auf 8 Zeichen begrenzt
- Es gibt einen Test der Relationsnamen FK\_???? auf Länge <=31 Zeichen
- Es gibt eine Weak-Relation
- Die Anzeige erfolgt in der Chen-Notation und Martin-Notation
- Es gibt Multi-Attribute in den Entities
- Es gibt Ternäre Relationen
- Es gibt auch eine Self-Relation
- Rauten-Darstellung der Relation
- Die Relationen können Attribute haben

#### Logisches Modell:

- Es gibt nun auch ein Weak-Entity
- Die Anzeige erfolgt nun auch in der Martin-Notation

## Installation

Das Programm bzw. die Zip-Dateien können in jedes beliebige Verzeichnis kopiert werden. Beim ersten Aufruf wird das Unterverzeichnis „dbs“ erzeugt. In diesem werden zum Einen die Projekte gespeichert, und zum Anderen die Datenbank-Definitionsdateien (\*.def).

Interne Daten werden in der Registry abgespeichert.

### Eintrag:

HKEY\_CURRENT\_USER\Software\WILHELM\DESIGNER

## **1.4 Ablauf bei der Erstellung eines konzeptionellen Modells**

- Erzeugen eines neuen Fensters für ein konzeptionelles Modell (STRG+M)
- Speichern unter einem Dateinamen
- Einstellen der Projekteinstellungen (Projekt | Eigenschaften)
- Erstellen der Entities (STRG+E)
- Eintragen der Attribute, ohne der/des Fremdschlüssels
- Festlegen der Primärschlüssel
- Erstellen der Beziehungen (STRG+R)
- Zusätzliche Eintragungen (Check, Unique, Weak-Entities)
- Eintragen der Texte (Beschreibungen etc.)
- Aufruf der Übertragung vom konzeptionellen zum logischen Modell (F9)

## **1.5 Ablauf bei der Erstellung eines logischen Modells**

- Erzeugen eines neuen Fensters für ein logischen Modell (STRG+N).
- Einstellen der Projekteinstellungen (Darstellungen der Beziehungen, Texte)
- Erstellen der Entities (STRG+E)
- Eintragen der Attribute, ohne der/des Fremdschlüssels
- Festlegen der Primärschlüssel
- Erstellen der Beziehungen (STRG+R)
- Zusätzliche Eintragungen (Check, Unique, Weak-Entities, Generator, Sequenz)
- Eintragen der Texte (Beschreibungen etc.)
- Aufruf der Generierung zum SQL-Skript (F9)

## 2 FBConsole

## 3 Zip-Datei

Dieses Script bietet einen Einstieg in die Benutzung von Datenbanken unter Winform / WPF. Dabei wird ein Gesamtpaket vom Datenbank-Designer, über eine Datenbank-Console bis hin zu grafischen Programme bereitgestellt.

### 3.1 *Installation*

- Entzippen der Datei in einen beliebigen Ordner

### 3.2 *Inhalt der ZIP-Datei*

#### **Ordner Designer**

- Datenbank-Designer
  - Programm „DBW\_Designer.exe“
  - Datenbank-Designer-Handbuch.pdf
  - Ordner „dbs“
  - Ordner „def“

#### **Ordner Firebird**

- Datenbank\_Employee.pdf
- dbs\_Firebird\_Oracle.pdf
- Tabelle der Firebird-Beispieldatenbank
  - Employee.fdb
  - Oracle.fdb
  - Eisenbahn.fdb
- Firebird-Datenbank-Konsole, nur Programm
- DLL's
  - fbembed.dll
  - FirebirdSql.Data.FirebirdClient.dll
  - icudt30.dll
  - icuin30.dll
  - icuuc30.dll

## 4 Beispiele

Dieses Kapitel beschreibt die Vorgehensweise zur Erzeugung einer Datenbank. Es wird unter dem Namen „Handbuch Bsp1.dbk“ mitgeliefert.

### Hinweis:

Das konzeptionelle Modell erlaubt alle möglichen Beziehungstypen zwischen zwei Entitäten. Dazu gehört auch die n:m-Beziehung.

Mit dem Schritt zum logischen Modell müssen alle netzförmigen Beziehungen (n:m etc.) in hierarchische Beziehungen umgewandelt werden. Dabei wird jeweils ein neues Entity erzeugt. Der Name wird aus den beiden Entitäten gebildet.

### **4.1 Entwicklung eines konzeptionelles ER-Modells**

Das folgende Beispiel entwickelt eine einfache Kundendatenbank: „Handbuch Bsp1.DBK“

#### 4.1.1 Entitäten

- Kunde
- Ort
- Artikel

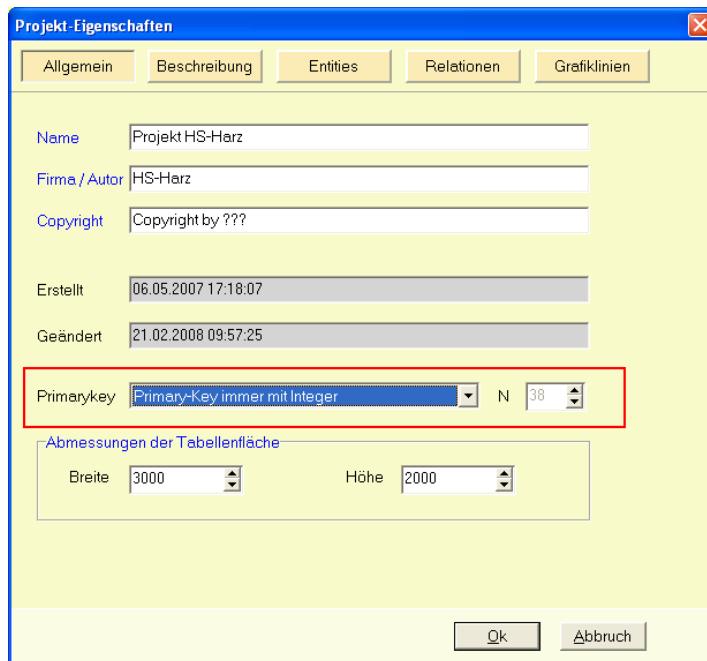
#### 4.1.2 Beziehungen:

Kunde zu Ort	cm:1
Kunde zu Artikel	cm:cm

#### 4.1.3 Neues Projekt

Mit der Taste „STRG+M“ wird ein neues Projekt angelegt. Es erscheint der Rahmen mit den Einträgen im Baum und der leeren Diagrammfläche. Im nächsten Schritt müssen nun die Entitäten angelegt werden.

Im ersten Schritt sollte die Definition der Primarykeys definiert werden. Dazu ruft man den Menüeintrag „PROJEKT|Eigenschaften“ auf.



**Abbildung 2 Projekteigenschaften (Integer vs. Max Numeric)**

In der Comboliste kann man entscheiden, ob man der Datentyp eines Primarykeys ein Integer oder ein Numeric/Number-Datentyp ist.

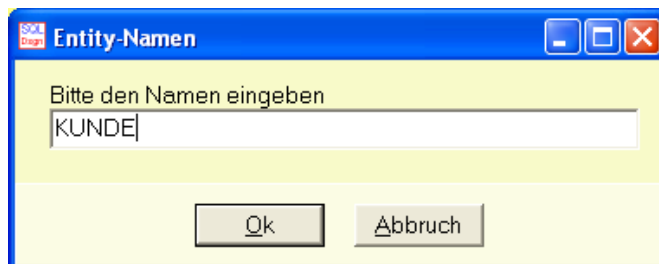
Hinweis:

- Fremdschlüsselattribute sollten nicht angelegt werden. Diese werden automatisch beim Erzeugen des logischen Modells eingetragen.
- Es gibt aber Ausnahmen, wenn alle „Primary Keys“ identisch sind.

#### 4.1.4 Entity anlegen

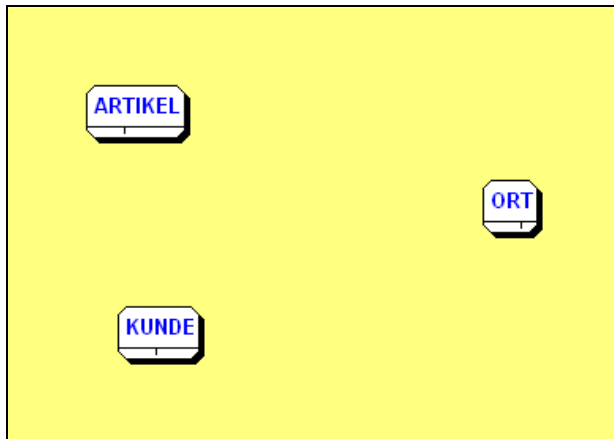
Mit dem Tastendruck „STRG+E“ werden die jeweiligen Entities angelegt.

Zur erst wird nach dem Namen gefragt, dieser darf nur aus Buchstaben, Zahlen und dem Underline („\_“) bestehen. Es darf kein Leerzeichen verwendet werden. Am Anfang muss auch ein Buchstabe stehen.



**Abbildung 3 Erzeugen eines Entities**

Nach der Eingabe der Namen, erscheinen die Entities in der Grafik



**Abbildung 4** Neue Entities

Mit einem Doppelklick kann das Entity „Kunde“ bearbeitet werden. Es erscheint die Abbildung 5. Hier sollten die Einträge ausgefüllt werden. Des Weiteren ist es ratsam, die nähere Beschreibung des Entity im Register „Beschreibung“ vorzunehmen. Im zweiten Register, siehe Abbildung 6, werden dann die Attribute eingetragen.

The screenshot shows a dialog box titled 'Einstellungen eines Entity: KUNDE'. It has five tabs: 'Allgemein' (selected), 'Attribute', 'Beschreibung', 'Farben', and 'Position'. Under the 'Allgemein' tab, there are two text input fields: 'Name' with the value 'KUNDE' and 'Kurzname' with the value 'kunde'. Below these is a large text area labeled 'Tabellen-Optionen' and 'Beschreibungen', which is currently empty. At the bottom right of the dialog are 'Ok' and 'Abbruch' buttons.

**Abbildung 5** Haupt-Eigenschaften eines Entity im konzeptionellen Modells

Das zweite Register verwaltet die Attribute. Mit den unteren Schaltern kann man diese bearbeiten, löschen und erzeugen. Die Schalter rechts an der Seite verschieben die Reihenfolge.

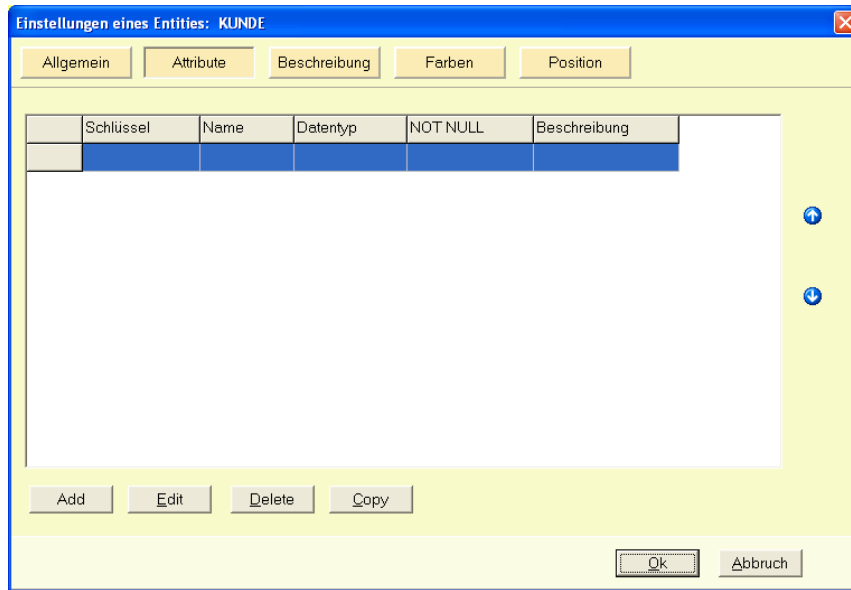


Abbildung 6 Bearbeiten, Erzeugen, Löschen von Attributen

#### 4.1.4.1 Attribute des Entity „Kunde“

- KNr
- KName

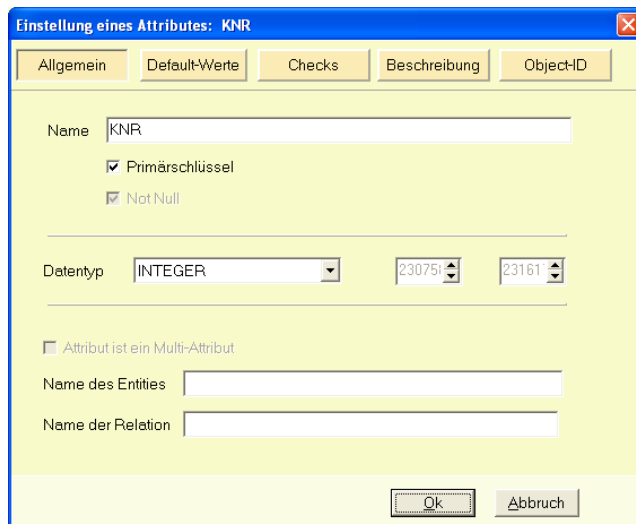


Abbildung 7 Kundennummer einfügen

Wenn man das Kontrollfeld „Primarykey“ anklickt, so wird automatisch beim ersten Mal der Datentyp „Integer“ bzw. Numerik/Number ausgewählt. Man diesen Datentyp aber danach auch wieder ändern.



	Schlüssel	Name	Datentyp	NOT NULL	Beschreibung
1	PK	KNR	INTEGER	NOT NULL	
2		NAME	CHAR	NOT NULL	

Abbildung 8 Attribute des Entities Kunde

Hinweis:

Die Beziehung zum Ort wird als Fremdschlüssel später hinzugefügt.

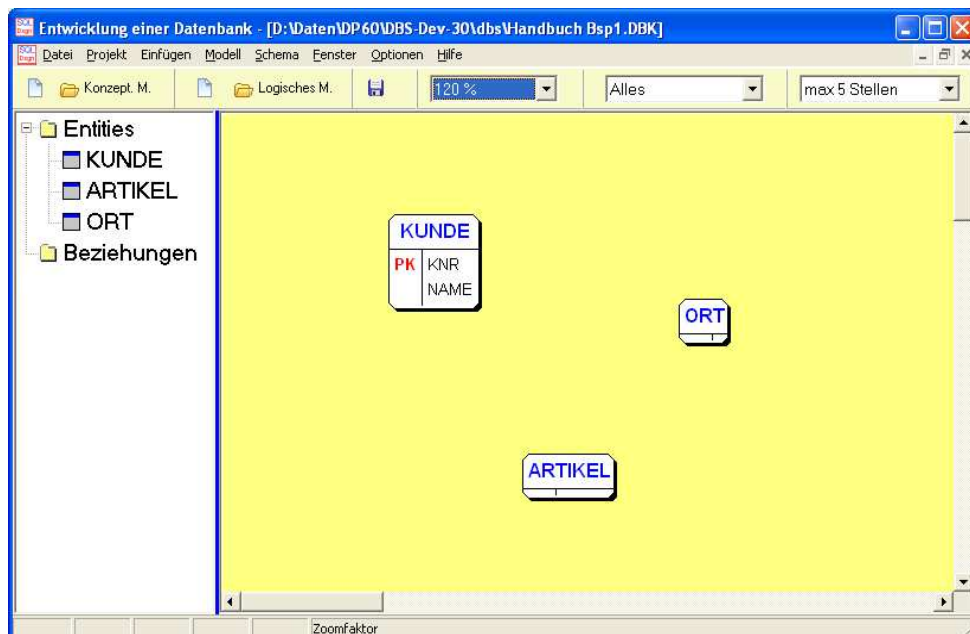


Abbildung 9 Entitäten mit Attributen des Kunden

#### 4.1.4.2 Attribute des Entity „Ort“

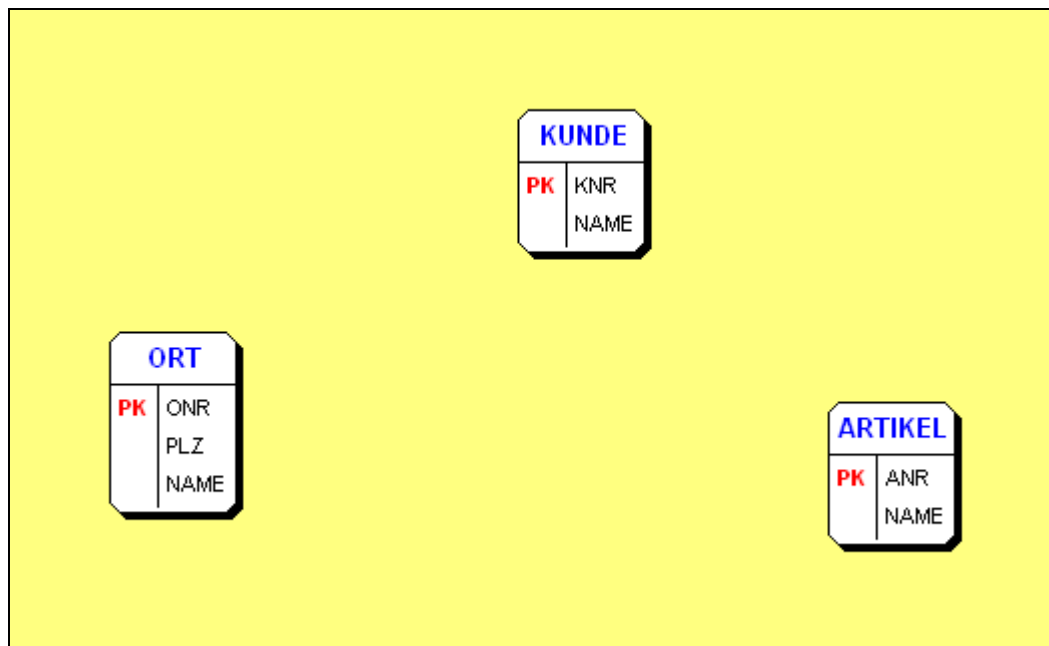
- ONr
- PLZ
- Name

Da die Postleitzahl ein Zeichenfeld sein muss, wird hier ein separater Schlüssel definiert.

#### 4.1.4.3 Attribute des Entity „Artikel“

- ANr
- Name
- Preis

Wenn alle Entitäten eingegeben wurden, hat das ER-Modell folgendes Aussehen:



**Abbildung 10 ER-Modell des ersten Beispiels**

Beim konzeptionellen Modell ist es noch nicht vorgesehen, dass ein Attribut nur eindeutige Werte – Unique – zugewiesen werden darf. Bei Bedarf muss man dieses im logischen Modell nachholen. Das Entity „Ort“ hat zwar das Attribut „PLZ“, dieses kann aber mehrfach vergeben werden.

#### 4.1.5 Beziehungen anlegen

Dazu ruft man aus dem Hauptmenü „Einfügen“ die Funktion „Beziehung“ auf. Es erscheint folgendes Fenster:

Abbildung 11 Erstellen der Beziehung Ort zu Kunde

In beiden Listen sind alle Entities eingetragen. In der linken Liste wählt man den Parent-Entity aus. Das wäre hier in diesem Beispiel „Ort“. In der rechten Liste wird die Beziehung eingetragen. Hier also „Kunde“.

Abbildung 12 Korrekte Einstellung für das erste Beispiel

Die Abbildung 13 zeigt die endgültigen Eingaben. Mit dem Schalter „Ok“ wird das Beziehungsfenster zur Definition der Beziehung angezeigt.

Abbildung 13 Definition der Beziehung Ort zu Kunde

Im ersten Register werden die Grundeinstellungen (Name, Beziehungstyp) eingetragen. Wichtig sind hier die Einträge „Parent-Verb“ und „Child-Verb“. Sie zeigen den Charakter der Beziehung an. Das Parent-Verb beschreibt die Beziehung durch Wörter wie „Kunde hat“ etc.

Im zweiten Register kann man weitere Beschreibungstexte eintragen

Die zweite Beziehung betrifft die Entitäten Kunde und Artikel. Hier ist es eine cm:n-Beziehung.

**Bearbeiten einer Beziehung**

Beziehungs-Name: KUNDE\_ARTIKEL  
 Relationsname wird Entity-Name !!!

Anzeigename: kunde\_artikel

Parent Verb: Ein Kunde hat eine oder mehrere Bestellungen (Artikel)

Child Verb: Ein Artikel wird von keinen, einen oder mehrere Kunden bestellt

Kardinalität:  
 Parent: CM (0, 1, oder viele) **ZU** Child: M (1 oder viele)

KUNDE: LParent\_Relation  
 ARTIKEL: LChild\_Relation

Ok Abbruch

Abbildung 14 Zweite Beziehung im ersten Beispiel

Nun hat man das fertige konzeptionelle Modell:

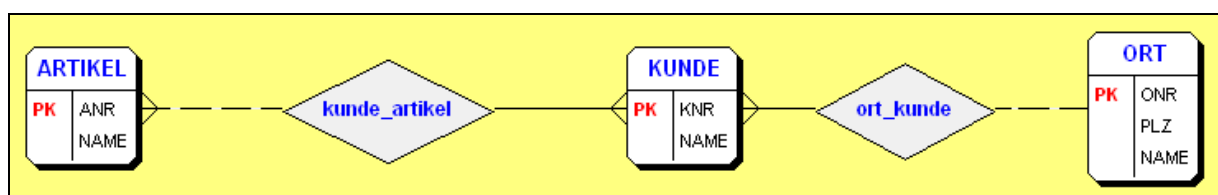


Abbildung 15 Konzeptionelles Modell des ersten Beispiels

#### 4.1.6 Weitere Schritte

Dieses Kapitel zeigt weitere mögliche Schritte im konzeptionellen Modell. Zum Einen kann man für verschiedene Attribute Defaultwerte und Checkbedingungen definieren.

#### 4.1.6.1 Default-Werte definieren

Mit einem Doppelklick auf ein Entity erhält man das normale Dialogfenster.

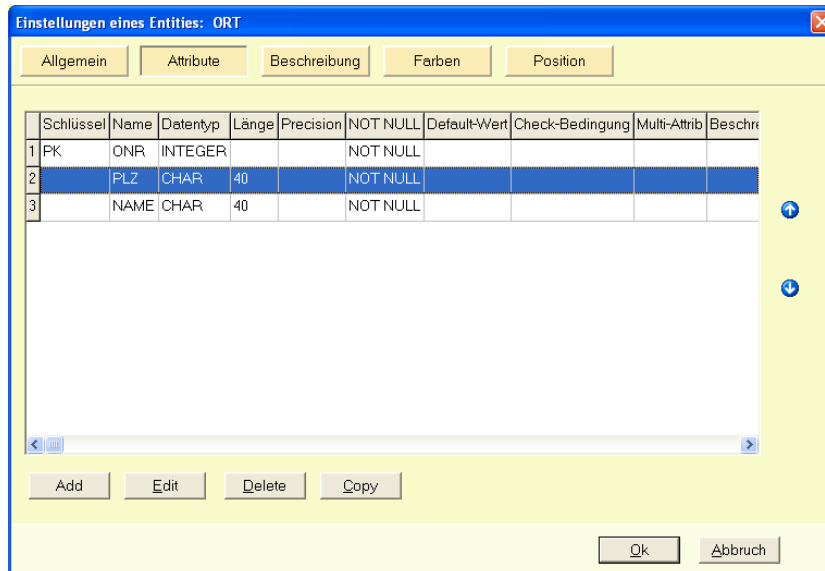


Abbildung 16 Attribute des Entity Ort

Mit einem Doppelklick auf das Attribut PLZ erscheint das Attributdialog. Hier kann man die Länge der Postleitzahl zum Einen etwas verkleinern, Länge auf 5 setzen, und zum Anderen die wichtigste Postleitzahl definieren. Dazu klickt man den Schalter „Default-Werte an.“

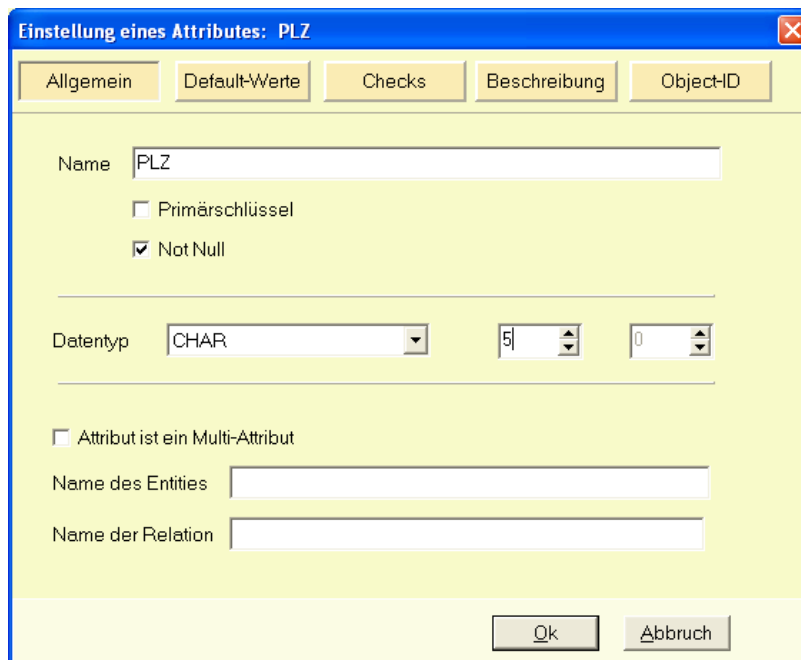


Abbildung 17 Postleitzahl auf Länge 5

**Einstellung eines Attributes: PLZ**

Allgemein | **Default-Werte** | Checks | Beschreibung | Object-ID

Default-Wert

☒ Default-Wert

Wert: 3855

Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung).

Ok Abbruch

**Abbildung 18 Postleitzahl als Default-Wert**

Für alle Werte in Wernigerode benötigt man nun nicht mehr die Eingabe der Postleitzahl.

#### 4.1.6.2 Unique-Attribute definieren

Für manche Attribute ist es sinnvoll, eindeutige Werte zu definieren (Beispiele Artikelname, Autonomie). Die Unique-Eigenschaft sagt, dass es nicht zwei identische Einträge für diese Spalte gibt. Dazu gibt es bei den Attribut-Registern im zweiten Register ein Eintrag. Man aktiviert das Kontrollfeld „Unique“ und trägt einen passenden Name in das Eingabefeld.

**Einstellung eines Attributes: NAME**

Allgemein | **Default / Unique** | Checks | Beschreibung | Object-ID

Default-Wert

☐ Default-Wert

Wert:

Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung).

Unique-Bedingungen

☒ Unique

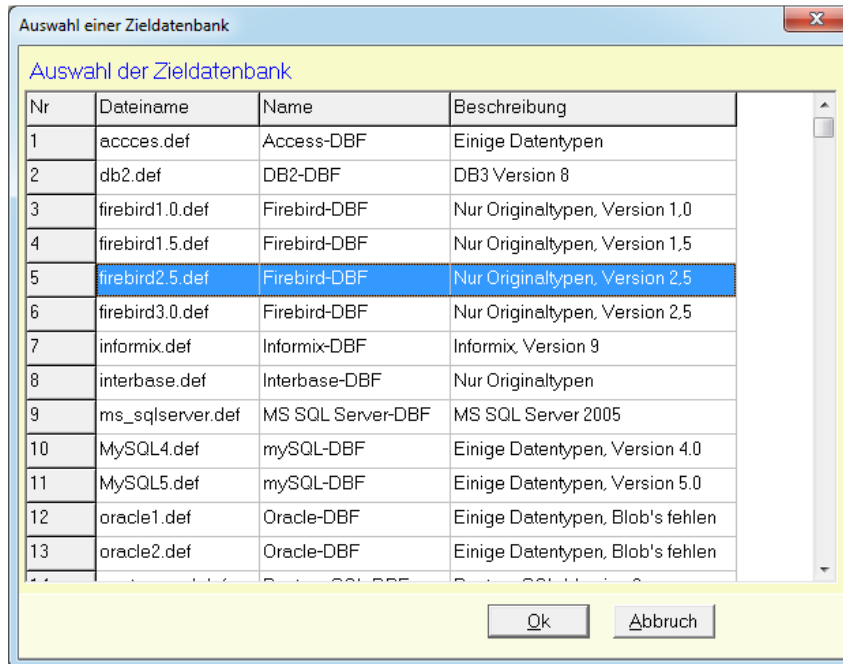
Check-C. Name: UNIQUE\_NAME

Ok Abbruch

**Abbildung 19 Unique-Constraint für den Ortsnamen einfügen**

#### 4.1.7 Umwandlung in ein logisches Modell

Mit dem Menü „Schema“ und dem Eintrag „Generieren logisches Modell“ wird ein neues Fenster erzeugt, in dem das logische Modell automatisch eingetragen wird. Vor diesem Schritt wird noch die Zieldatenbank abgefragt.



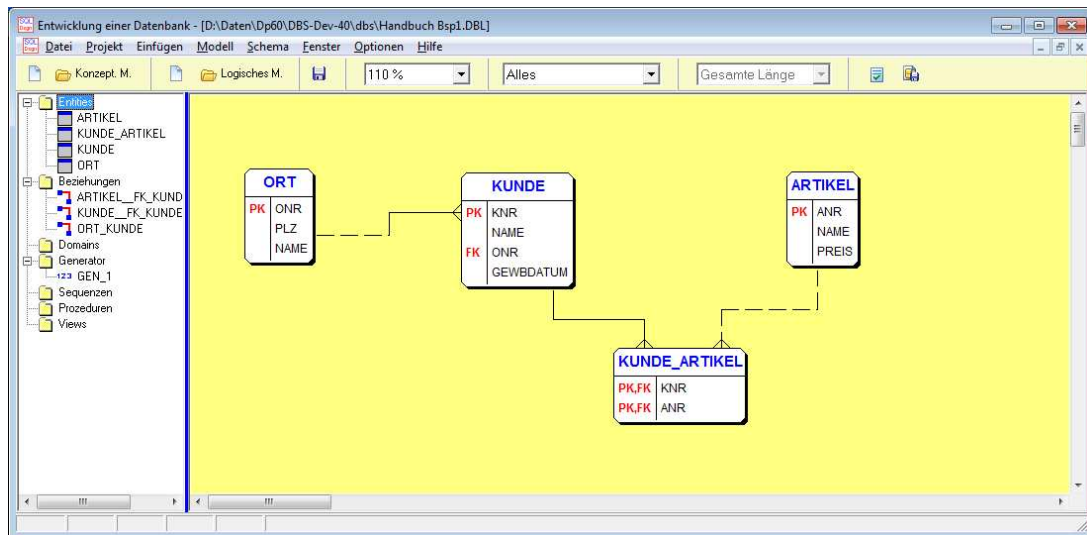
**Abbildung 20 Zieldatenbank Firebird**

Doppelklick oder der Schalter „Ok“ erzeugt dann das logische Modell in einem neuem Fenster.



**Abbildung 21 Portierungsmeldung**

Die nächste Abbildung zeigt das logische Modell:



**Abbildung 22 Das logische Modell des ersten Beispiels**

Die cm:m-Beziehung wurde durch eine zusätzliche Entität in ein hierarchisches Modell überführt. Des Weiteren wurde das Attribut „ORT\_ONR“ als Fremdschlüssel in das Entity „Kunde“ eingetragen.

#### 4.1.8 Weitere Schritte

##### 4.1.8.1 Check-Constraint definieren

Check-Bedingungen werden benutzt, um zusätzliche Überprüfungen VOR dem Eintragen in die Datenbanken vorzunehmen.

##### Beispiel:

Das Entity Kunde hat noch das Attribut „GebDatum“.



The screenshot shows the 'Einstellung eines Attributes: GEBDATUM' dialog box with the 'Allgemein' tab selected. The 'Name' field contains 'GEBDATUM'. Below it are three checkboxes: 'Add To PrimaryKey' (unchecked), 'Fremdschlüssel' (unchecked), and 'Not Null' (unchecked). The 'Datentyp-Definition' section has a 'Domain' dropdown set to 'keine Domain', a 'Datentyp' dropdown set to 'SMALLINT', and two numeric spinners both set to '0'. A 'Neue Domain' button is located below the 'Datentyp-Definition' section. At the bottom are 'Ok' and 'Abbruch' buttons.

**Abbildung 23** Attribut GebDatum

Die obige Abbildung zeigt die Standardeingaben eines Attributs.

The screenshot shows the same dialog box but with the 'Checks' tab selected. The 'Check-Bedingungen' section has a checked checkbox for 'Check Attribute'. Below it, the 'Check-C. Name' field contains 'CH\_GEBDATUM'. The 'Check-Bedingung' text area contains the SQL code: 

```
CHECK (  
  GEBDATUM < 1990  
)
```

. At the bottom are 'Ok' and 'Abbruch' buttons.

**Abbildung 24** Check-Bedingung

Das Attribut GebDatum signalisiert aber, dass man ein komplette Datum (Tag, Monat, Jahr) eingegeben hat. Entweder man ändert den Namen in „GebJahr“ oder man benutzt eine bessere Abfrage:

The screenshot shows the 'Einstellung eines Attributes: GEWB DATUM' dialog box with the 'Datentyp-Definition' tab selected. The 'Name' field contains 'GEWB DATUM'. Below it are three checkboxes: 'Primärschlüssel', 'Not Null', and 'Fremdschlüssel', all of which are unchecked. The 'Datentyp-Definition' section includes a 'Domain' dropdown set to 'keine Domain', a 'Datentyp' dropdown set to 'TIMESTAMP', and two spinners for '0'. Below this is a 'Formel' field and another 'Datentyp' dropdown set to 'kein Datentyp', with spinners for '1' and '0'. A 'Neue Domain' button is located to the right of the 'Domain' dropdown. The 'Autonumber' section has an 'aktiviert' checkbox (unchecked) and spinners for 'Start' (0) and 'Increment' (1). At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 25 Neue Definition des GebDatums

The screenshot shows the 'Einstellung eines Attributes: GEWB DATUM' dialog box with the 'Check-Bedingungen' tab selected. The 'Check Attribute' checkbox is checked. The 'Check-C. Name' field contains 'CH\_GEWB DATUM'. The 'Check-Bedingung' text area contains the following SQL code:

```

CHECK (
  datediff ( day, GebDatum,
    (
      select cast('NOW' as timestamp) from rdb$database
    )
  ) >= 6574.5
)

```

At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 26 Check-Bedingung

Alternativ kann man auch das aktuelle Jahr ermitteln und dann 16 bzw. 18 Jahre abziehen.

Für Firebird wäre das diese Syntax:

CAST('NOW' AS DATE)	liefert das aktuelle Datum
CAST('NOW' AS TIMESTAMP)	liefert das aktuelle Datum, Zeit
CAST('NOW' AS TIME)	liefert die aktuelle Zeit

EXTRACT(year FROM column)      Auslesen des Jahres

Bedingung GebDatum nur als Jahr:

```
CHECK (  
    GEBDATUM < EXTRACT(year FROM CAST('NOW' AS DATE))-17  
)
```

Bedingung GebDatum als Datum:

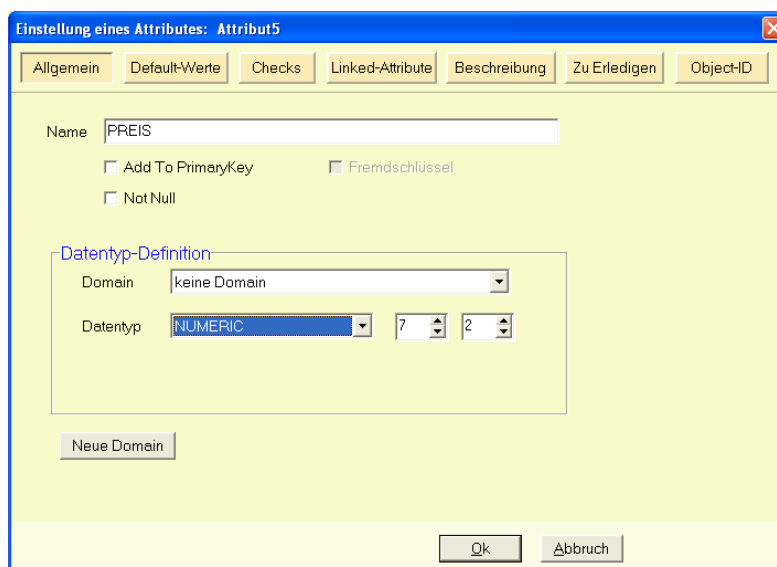
```
CHECK (  
    EXTRACT(year FROM GEBDATUM) < EXTRACT(year FROM CAST('NOW' AS  
DATE))-17  
)
```

oder mit Monaten etwas genauer

```
CHECK (  
    EXTRACT(year FROM GEBDATUM)*12 + EXTRACT(month FROM GEBDATUM)  
  
    <  
        EXTRACT(year FROM CAST('NOW' AS DATE))*12 +  
        EXTRACT(month FROM CAST('NOW' AS DATE))  
)
```

Weiteres Beispiel:

Man fügt das Attribut „Preis“ zum Entity „Artikel“. Dann sollte dieser nicht negativ sein.



**Abbildung 27 Erzeugen des Attributs Preis**

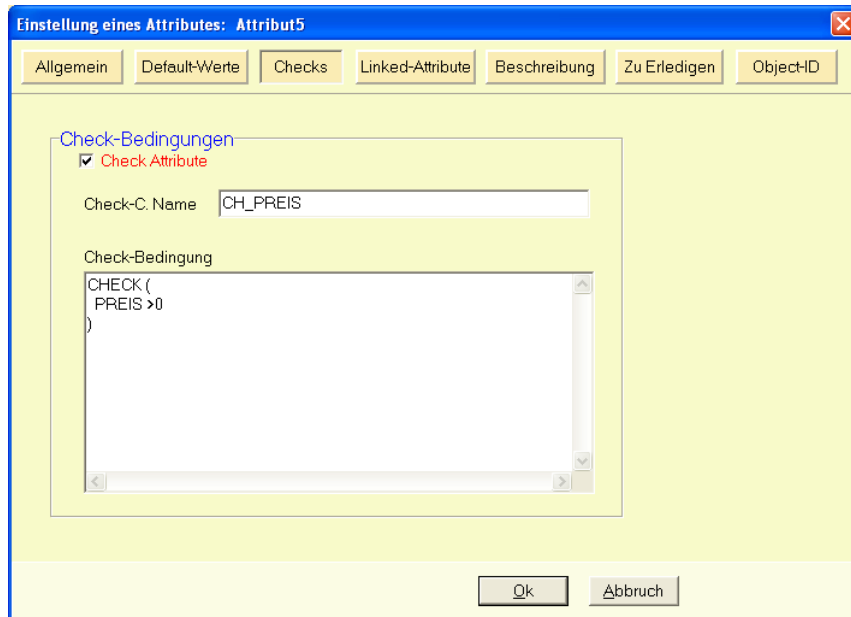


Abbildung 28 Check-Bedingung des Preises

#### 4.1.8.2 Generator oder Sequenz definieren

Einige Datenbanken verwenden Generatoren, andere Sequenzen. Access wiederum benutzt einen Autoincrement-Datentyp. Ziel eines Generators ist die automatische Erzeugung eindeutiger Schlüssel. Man könnte dies auch so definieren:

```
SELECT MAX(empno)
FROM employee;
```

Diese Nummer wird nun um eins erhöht und als neue empno verwendet. Dieses funktioniert im Single-User-Betrieb vollkommen. Nur im Netzwerk und im Multi-User-Betrieb können zwei oder drei neue Mitarbeiter die gleiche Nummer erhalten. Mit Hilfe eines Generators bzw. einer Sequenz wird eine Nummer nur einmal vergeben.

##### Nachteil:

Es kann zu Lücken kommen, wenn ein Abbruch der Transaktion, Rollback, vorgenommen wird.

##### Sequenz definieren:

Um eine Sequenz zu erzeugen, ruft man das Menü „Einfügen“ mit dem Eintrag „Sequenz“ auf.

The screenshot shows a dialog box titled "Bearbeiten von Sequenzen". It has three tabs: "Allgemein", "Beschreibung", and "Zu Erledigen". The "Allgemein" tab is selected. The fields are as follows:

- Sequenz-Name: seq
- Startwert: 0
- Schrittweite: 1 (checked "aktiviert")
- Minimum: 1 (unchecked "aktiviert")
- Maximum: MAX\_INT\_VALUE (unchecked "aktiviert")
- Zyklus: Kein Zyklus
- Cache: Kein Cache
- Order: Keine Order

At the bottom right are "Ok" and "Abbruch" buttons.

**Abbildung 29 Definition einer Sequenz**

In Firebird bzw. Interbase kann man neben dem Namen nur den Startwert definieren. Der erste richtige Wert ist dann um eins erhöht.

#### Beispielskripte:

Erzeugen einer Sequenz:

```
CREATE SEQUENCE SEQ;  
alter sequence seqtest restart with 0
```

#### Abfrage des aktuellen Wertes:

```
select gen_id( seq, 0 ) FROM RDB$DATABASE;
```

#### Sequenz definieren (z. B. Oracle):

Um einen Sequenz zu erzeugen, ruft man das Menü „Einfügen“ mit dem Eintrag „Sequenz“ auf.

The screenshot shows a dialog box titled "Bearbeiten von Sequenzen" with a close button in the top right corner. It has three tabs: "Allgemein", "Beschreibung", and "Zu Erledigen". The "Allgemein" tab is selected. The dialog contains the following fields and options:

- Sequenz-Name:** A text field containing "Seq\_1".
- Startwert:** A spin box containing "0".
- Schrittweite:** A spin box containing "1", followed by a checked checkbox labeled "aktiviert".
- Minimum:** A spin box containing "1", followed by an unchecked checkbox labeled "aktiviert".
- Maximum:** A text field containing "MAX\_INT\_VALUE", followed by an unchecked checkbox labeled "aktiviert".
- Zyklus:** A dropdown menu with "Kein Zyklus" selected.
- Cache:** A dropdown menu with "Kein Cache" selected.
- Order:** A dropdown menu with "Keine Order" selected.

At the bottom right, there are two buttons: "Ok" and "Abbruch".

**Abbildung 30 Definition einer Sequenz**

In Oracle hat man detailliertere Optionen. Zusätzlich definiert man die Schrittweite, das Minimum und Maximum. Die Einstellung Zyklus stellt sicher, dass man bei einem Erreichen des Maximalwerts wieder von vorne anfängt.

**Beispielskript:**

```
CREATE SEQUENZ SEQ_1
  INCREMENT BY 1
  START WITH 0
  NOMINVALUE
  NOMAXVALUE
  NOCYCLE
  NOCACHE
  NOORDER ;
```

Einige Datenbanken verwenden Sequenzen (Oracle, PostgreSQL, Informix). Die Definition ist aber nicht genormt. Deshalb können die obigen Teile separat definiert werden.

#### 4.1.8.3 Ergebnis des „Handbuch Bsp1.dbl“

Die untere Abbildung zeigt das vollständige und erweiterte ER-Modell.

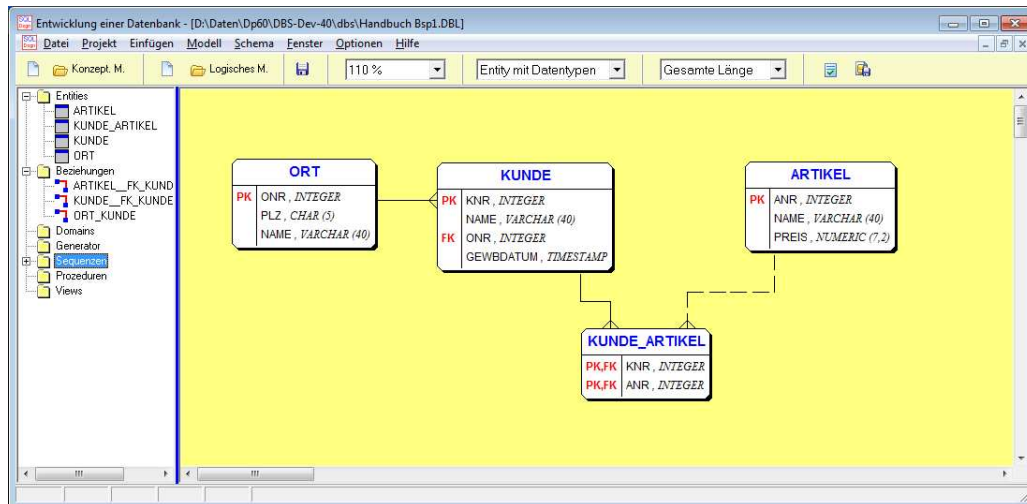


Abbildung 31 Logisches ER-Modell des ersten Beispiels

Mit dem Menü „Schema“, Eintrag „Generierung Datenbank-Skript“ kann man nun das SQL-Skript erstellen (F9).



Abbildung 32 Erzeugen eines SQL-Skriptes

Ergebnis mit SQL-Befehlen:

```
/*=====*/
/* Script generated with: MW Designer Version 4.74, Build 501 */
/* Project Filename: D:\Daten\Dp60\DBS-Dev-40\dbs\Handbuch Bspl.sql */
/* Project Name: Projekt HS-Harz */
/* Author: HS-Harz */
/* DBMS: Firebird-DBF */
/* Copyright: Copyright by Michael Wilhelm */
/* Generated on: 09.02.2016 16:49:41 */
/*=====*/

/*=====*/
/* Tables */
/*=====*/
CREATE TABLE ARTIKEL (
    ANR INTEGER NOT NULL,
    NAME VARCHAR(40) NOT NULL,
    PREIS NUMERIC(7,2) DEFAULT 0,

    CONSTRAINT PK_ANR PRIMARY KEY (ANR)
);

CREATE TABLE KUNDE_ARTIKEL (
    KNR INTEGER NOT NULL,
    ANR INTEGER NOT NULL,

    CONSTRAINT PK_KUNDE_ARTIKEL PRIMARY KEY (KNR, ANR )
);

CREATE TABLE KUNDE (
    KNR INTEGER NOT NULL,
    NAME VARCHAR(40) NOT NULL,
    ONR INTEGER NOT NULL,
    GEWBDATUM TIMESTAMP,

    CONSTRAINT PK_KNR PRIMARY KEY (KNR)
);

CREATE TABLE ORT (
    ONR INTEGER NOT NULL,
    PLZ CHAR(5) NOT NULL,
    NAME VARCHAR(40) NOT NULL UNIQUE,

    CONSTRAINT PK_ONR PRIMARY KEY (ONR)
);

/*=====*/
/* Foreign Keys */
/*=====*/
ALTER TABLE KUNDE_ARTIKEL ADD CONSTRAINT
    FK_ARTIKEL__FK_KUNDE_ARTIKEL FOREIGN KEY (ANR) REFERENCES ARTIKEL(ANR);

ALTER TABLE KUNDE_ARTIKEL ADD CONSTRAINT
    FK_KUNDE__FK_KUNDE_ARTIKEL FOREIGN KEY (KNR) REFERENCES KUNDE(KNR);

ALTER TABLE KUNDE ADD CONSTRAINT
    FK_ORT_KUNDE FOREIGN KEY (ONR) REFERENCES ORT(ONR);
```



```
/*=====*/
/*  Check-Constraints                                */
/*=====*/
ALTER TABLE ARTIKEL
    ADD CONSTRAINT CH_PREIS CHECK (
        PREIS >0
    );

ALTER TABLE KUNDE
    ADD CONSTRAINT CH_GEWBDATUM CHECK (
        datediff ( day, GebDatum,
            (
                select cast('NOW' as timestamp) from rdb$database
            )
        ) >=6574.5
    );

/*=====*/
/*  Sequenzen                                          */
/*=====*/
CREATE SEQUENCE SEQ
;

```

## 4.2 Entwicklung eines konzeptionelles ER-Modells

Das folgende Beispiel entwickelt eine einfache Firmendatenbank:, bei der Multiattribute im Entity „Abteilung“ definiert werden.

Dateiname: Handbuch Bsp2\_MultiAttribute.DBK

### 4.2.1 Entitäten

- Mitarbeiter
- Abteilung

### 4.2.2 Beziehungen:

Abteilung zu Mitarbeiter                      1:cm

### 4.2.3 Neues Projekt

Mit der Taste „STRG+M“ wird ein neues Projekt angelegt. Es erscheint der Rahmen mit den Einträgen im Baum und der leeren Diagrammfläche. Im nächsten Schritt müssen nun die

Entitäten angelegt werden. Vorab sollte die Definition der Primarykeys definiert werden. Dazu ruft man den Menüeintrag „PROJEKT|Eigenschaften“ auf (siehe erste Beispiel).

Hinweis:

Fremdschlüsselattribute sollten nicht angelegt werden. Diese werden automatisch beim Erzeugen des logischen Modells eingetragen.

#### 4.2.4 Entity anlegen

Mit dem Tastendruck „STRG+E“ werden die jeweiligen Entities angelegt.

Zur erst wird nach dem Namen gefragt, dieser darf nur aus Buchstaben, Zahlen und dem Underline („\_“) bestehen. Es darf kein Leerzeichen verwendet werden. Am Anfang muss auch ein Buchstabe stehen.

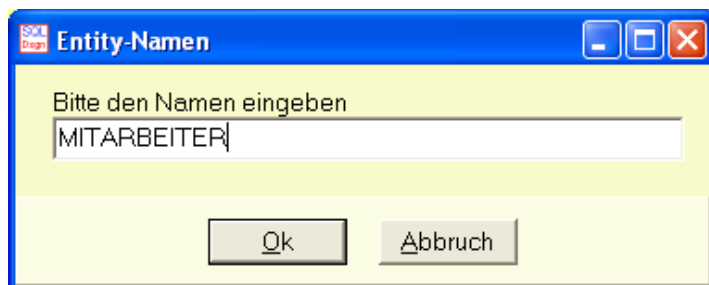


Abbildung 33 Erzeugen eines Entities

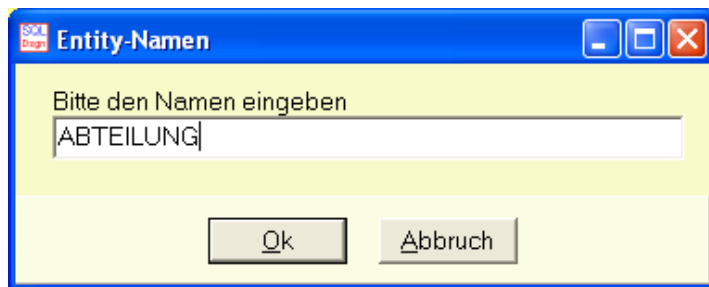


Abbildung 34 Erzeugen eines Entities

Nach der Eingabe der Namen, erscheinen die Entities in der Grafik



Abbildung 35 Neue Entities

## 4.2.4.1 Attribute des Entity „Mitarbeiter“

- MITNR INTEGER
- NAME VCHAR(50)
- VORNAME VCHAR(50)
- STRASSE VCHAR(50)
- PLZ CHAR(5)
- ORT VCHAR(50)

Mit einem Doppelklick kann das Entity „Mitarbeiter“ bearbeitet werden. Hier sollten die Einträge ausgefüllt werden. Des Weiteren ist es ratsam, die nähere Beschreibung des Entities im Register „Beschreibung“ vorzunehmen. Die fertigen Attribute sind in der unteren Abbildung dargestellt.

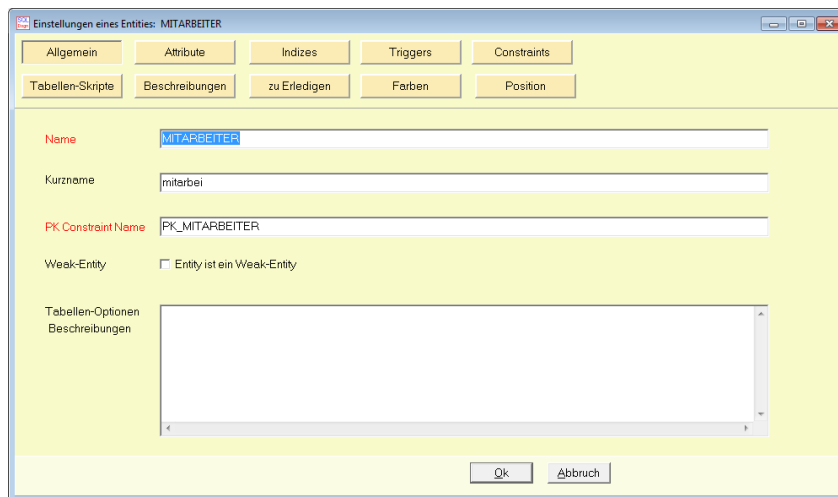


Abbildung 36 Startdialog für das 2. Beispiel

Nun wechseln in zweite Register (Attribute):

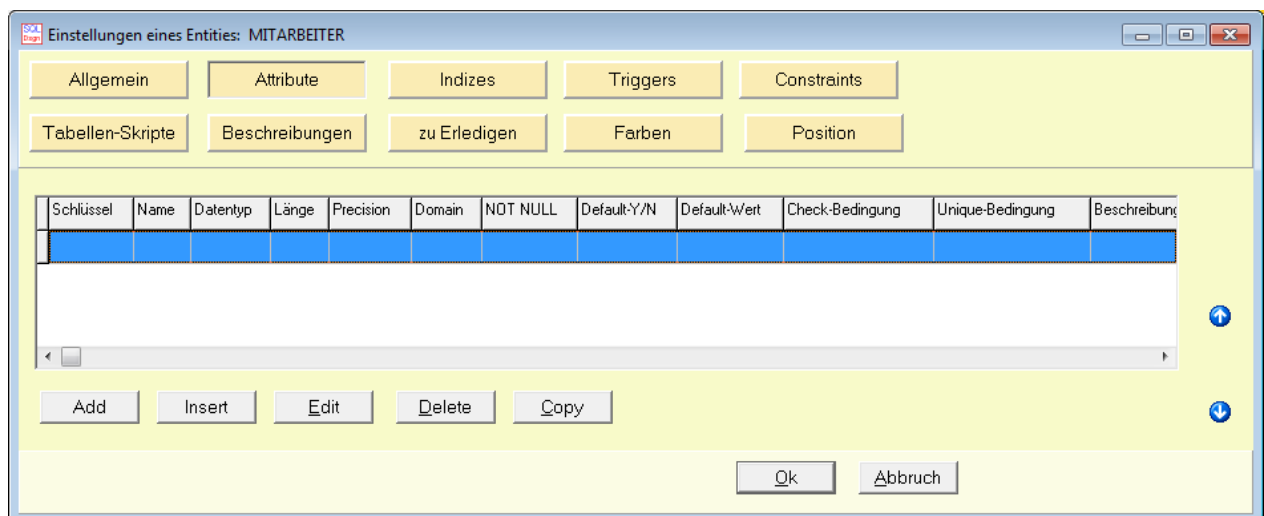


Abbildung 37 Attributdialog

Mitarbeiternummer:

Einstellung eines Attributes: Attribut1

Allgemein Default / Unique Checks Beschreibung Object-ID

Name MITNR

☒ Primärschlüssel

☒ Not Null

Datentyp INTEGER 1 0

☐ Attribut ist ein Multi-Attribut

Name des Entites

Name der Relation

Ok Abbruch

Abbildung 38 Einfügen des Attribut Mitarbeiter

Attribut Name:

Einstellung eines Attributes: NAME

Allgemein Default / Unique Checks Beschreibung Object-ID

Name NAME

☐ Primärschlüssel

☐ Not Null

Datentyp VCHAR 50 0

☐ Attribut ist ein Multi-Attribut

Name des Entites

Name der Relation

Ok Abbruch

Abbildung 39 Einfügen des Attribut Name

The screenshot shows a dialog box titled "Einstellung eines Attributes: NAME". It has five tabs: "Allgemein", "Default / Unique", "Checks", "Beschreibung", and "Object-ID". The "Default / Unique" tab is selected. Inside this tab, there are two sections. The first section, "Default-Wert", has a checked checkbox labeled "Default-Wert" and a text input field labeled "Wert". Below this, a red note states: "Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)". The second section, "Unique-Bedingungen", has an unchecked checkbox labeled "Unique" and a text input field labeled "Check-C. Name". At the bottom right, there are "Ok" and "Abbruch" buttons.

Abbildung 40 Einfügen des Defaultwerts für Name

Der Vorname und die Strasse werden mit dem Schalter „Copy“ kopiert. Dann werden auch die Default-Einstellungen übernommen.

The screenshot shows a dialog box titled "Einstellung eines Attributes: Attribut2". It has five tabs: "Allgemein", "Default / Unique", "Checks", "Beschreibung", and "Object-ID". The "Default / Unique" tab is selected. Inside this tab, the "Name" field contains "VORNAME". Below it are two unchecked checkboxes: "Primärschlüssel" and "Not Null". A horizontal line separates these from the "Datentyp" section, which has a dropdown menu set to "VCHAR", a spin box set to "50", and another spin box set to "0". Below this is another horizontal line, followed by an unchecked checkbox labeled "Attribut ist ein Multi-Attribut". At the bottom, there are two text input fields: "Name des Entities" and "Name der Relation". At the bottom right, there are "Ok" and "Abbruch" buttons.

Abbildung 41 Einfügen des Attribut Vorname

The dialog box is titled "Einstellung eines Attributes: VORNAME". It has five tabs: "Allgemein", "Default/Unique", "Checks", "Beschreibung", and "Object-ID". The "Default/Unique" tab is selected. It contains two sections: "Default-Wert" and "Unique-Bedingungen". In the "Default-Wert" section, the "Default-Wert" checkbox is checked, and there is a text input field labeled "Wert". Below it, a red note states: "Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)". In the "Unique-Bedingungen" section, the "Unique" checkbox is unchecked, and there is a text input field labeled "Check-C. Name". At the bottom, there are "Ok" and "Abbruch" buttons.

Abbildung 42 Einfügen des Defaultwerts Vorname

The dialog box is titled "Einstellung eines Attributes: STRASSE". It has seven tabs: "Allgemein", "Default/Unique", "Checks", "Linked-Attribute", "Beschreibung", "Zu Erledigen", and "Object-ID". The "Allgemein" tab is selected. It contains a "Name" field with "STRASSE" entered. Below it are three checkboxes: "Primärschlüssel", "Not Null", and "Fremdschlüssel", all of which are unchecked. There are two sections: "Datentyp-Definition" and "Autonumber". The "Datentyp-Definition" section has a "Domain" dropdown set to "keine Domain", a "Datentyp" dropdown set to "VARCHAR" with a length of 50 and a precision of 0, a "Formel" field, and another "Datentyp" dropdown set to "kein Datentyp" with a length of 1 and a precision of 0. A "Neue Domain" button is to the right. The "Autonumber" section has an "aktiviert" checkbox which is unchecked, a "Start" field set to 0, and an "Increment" field set to 1. At the bottom, there are "Ok" and "Abbruch" buttons.

Abbildung 43 Einfügen des Attribut Strasse

The screenshot shows a dialog box titled "Einstellung eines Attributes: Attribut2" with a close button (X) in the top right corner. It has five tabs: "Allgemein", "Default / Unique", "Checks", "Beschreibung", and "Object-ID". The "Default / Unique" tab is selected. Inside this tab, there are two sections. The first section, "Default-Wert", has a checked checkbox labeled "Default-Wert" and a text input field labeled "Wert". Below this, a red note states: "Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)". The second section, "Unique-Bedingungen", has an unchecked checkbox labeled "Unique" and a text input field labeled "Check-C. Name". At the bottom right, there are "Ok" and "Abbruch" buttons.

Abbildung 44 Defaultwert für Strasse

Die Postleitzahl muss vom Typ „char“ sein, da sonst die links stehende Null abgeschnitten wird.

The screenshot shows a dialog box titled "Einstellung eines Attributes: Attribut5" with a close button (X) in the top right corner. It has five tabs: "Allgemein", "Default / Unique", "Checks", "Beschreibung", and "Object-ID". The "Default / Unique" tab is selected. Inside this tab, there are several fields. The "Name" field contains "PLZ". Below it are two unchecked checkboxes: "Primärschlüssel" and "Not Null". A horizontal line separates these from the "Datentyp" section, which shows a dropdown menu set to "CHAR", and two spinners for length, set to "5" and "0". Another horizontal line follows. Below that is an unchecked checkbox labeled "Attribut ist ein Multi-Attribut". At the bottom are two text input fields: "Name des Entities" and "Name der Relation". At the bottom right, there are "Ok" and "Abbruch" buttons.

Abbildung 45 Einfügen des Attribut PLZ

The dialog box 'Einstellung eines Attributes: Attribut5' has five tabs: 'Allgemein', 'Default / Unique', 'Checks', 'Beschreibung', and 'ObjectID'. The 'Default / Unique' tab is active. It contains two sections: 'Default-Wert' and 'Unique-Bedingungen'. In the 'Default-Wert' section, the checkbox 'Default-Wert' is checked, and the 'Wert' field contains '00000'. Below this, a red note states: 'Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)'. In the 'Unique-Bedingungen' section, the 'Unique' checkbox is unchecked, and the 'Check-C. Name' field is empty. At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 46 Defaultwert für das Attribut PLZ

The dialog box 'Einstellung eines Attributes: ORT' has five tabs: 'Allgemein', 'Default / Unique', 'Checks', 'Beschreibung', and 'ObjectID'. The 'Allgemein' tab is active. It contains the following fields and options: 'Name' (text field with 'ORT'), 'Primärschlüssel' (unchecked checkbox), 'Not Null' (unchecked checkbox), 'Datentyp' (dropdown menu with 'VCHAR'), '50' (spin box), and '0' (spin box). Below these are three more fields: 'Attribut ist ein Multi-Attribut' (unchecked checkbox), 'Name des Entities' (text field), and 'Name der Relation' (text field). At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 47 Einfügen des Attribut Ort



	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Multi-Attrib	Be
1	PK	MITNR	INTEGER			NOT NULL				-		
2		VORNAME	VCHAR	50			x			-		
3		NAME	VCHAR	50			x			-		
4		STRASSE	VCHAR	50			x			-		
5		PLZ	CHAR	5			x	00000		-		
6		ORT	VCHAR	50			x			-		

Abbildung 48 Haupt-Eigenschaften des Entities Mitarbeiters

In der Spalte „Default-Y/N“ sieht man, welche Attribute einen Default-Wert haben.

#### 4.2.4.2 Attribute des Entity „Abteilung“

- NR                      INTEGER
- NAME                 VCHAR(50)
- STANDORTE         VCHAR(50)                      MultiAttribut !!!!!

Mit einem Doppelklick kann das Entity „Abteilung“ bearbeitet werden. Hier sollten die Einträge ausgefüllt werden. Des Weiteren ist es ratsam, die nähere Beschreibung des Entity im Register „Beschreibung“ vorzunehmen. Die fertigen Attribute sind in der unteren Abbildung dargestellt. Wichtig dabei ist, dass das Attribut „Standorte“ als Multiattribut definiert wird. Damit verstößt es gegen die erste Normalform. Im logischen Modell wird dieser Verstoß behoben.

Abbildung 49 Neues Attributs "Nr"

The dialog box 'Einstellung eines Attributes: NAME' has five tabs: 'Allgemein', 'Default / Unique' (selected), 'Checks', 'Beschreibung', and 'ObjectID'. The 'Default-Wert' section contains a checked checkbox 'Default-Wert', a text input field 'Wert', and a red warning message: 'Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)'. The 'Unique-Bedingungen' section contains an unchecked checkbox 'Unique' and a text input field 'Check-C. Name'. At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 50 Default-Wert für das Atribut "Nr"

The dialog box 'Einstellung eines Attributes: STANDORTE' has five tabs: 'Allgemein', 'Default / Unique' (selected), 'Checks', 'Beschreibung', and 'ObjectID'. The 'Name' field contains 'STANDORTE'. Below it are unchecked checkboxes for 'Primärschlüssel' and 'Not Null'. The 'Datentyp' section shows 'VCHAR' in a dropdown, with '50' and '0' in adjacent spinners. The 'Attribut ist ein Multi-Attribut' checkbox is checked. Below it, 'Name des Entities' is 'ABTEILUNG\_STANDORTE' and 'Name der Relation' is 'Rel\_ABTEILUNG\_STANDORTE'. At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 51 Neues Attributs "Standorte" mit Multi-Attribut

**Einstellung eines Attributes: Attribut3**

Algemein | **Default / Unique** | Checks | Beschreibung | Object-ID

**Default-Wert**

☒ **Default-Wert**

Wert

Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung).

**Unique-Bedingungen**

☐ **Unique**

Check-C. Name

Ok Abbruch

Abbildung 52 Check-Bedingung für das Attribut "Standorte"

**Einstellungen eines Entities: ABTEILUNG**

Algemein | **Attribute** | Beschreibung | Farben | Position

	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Multi-Attrib
1	PK	NR	INTEGER			NOT NULL				-	
2		NAME	VCHAR	50			x			-	
3		STANDORTE	VCHAR	50			x			-	ABTEILUNG_STANDORTE

Add Edit Delete Copy

Ok Abbruch

Abbildung 53 Haupt-Eigenschaften des Entities Abteilung

Das Attribut „Standorte“ wird als Multiattribut definiert. Dazu wird die Checkbox gesetzt und es müssen die beiden Namen (Entity und Relation) eingetragen werden.

**Einstellung eines Attributes: STANDORTE**

Tab: Allgemein | Default / Unique | Checks | Beschreibung | Object-ID

Name:

☐ Primärschlüssel

☐ Not Null

Datentyp:

☒ Attribut ist ein Multi-Attribut

Name des Entites:

Name der Relation:

Abbildung 54 Attribut als Multiattribut

Anzeige des aktuellen Stands:

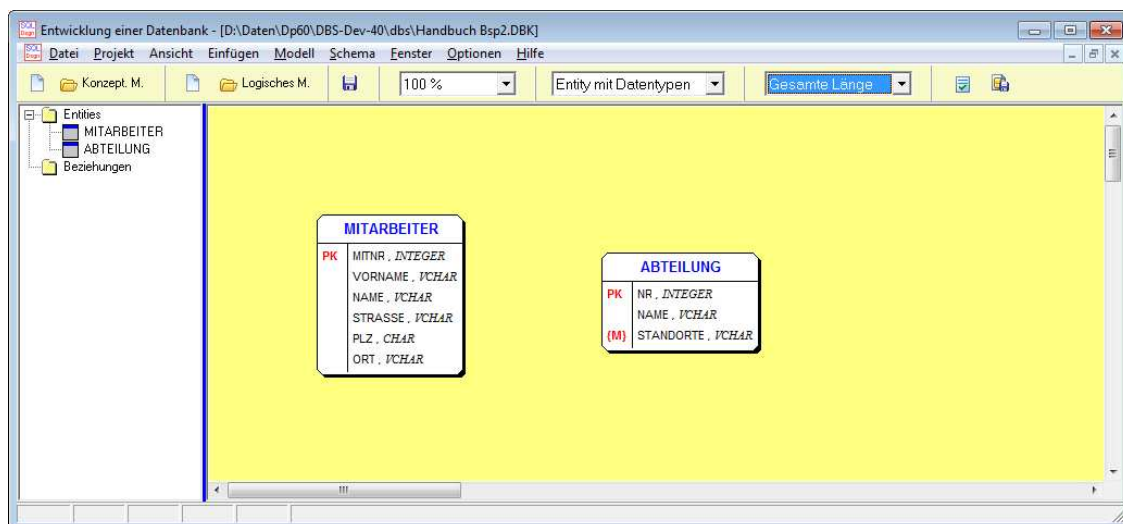


Abbildung 55 Anzeige der beiden Entities des 2. Beispiels

Nach diesen Vorbereitungen wird die Beziehung mittels STRG+R erstellt.

**Neue Beziehung**

Beziehung definieren

Parent Entity (c, 1, cm, m), PK:

Child Entity (c, 1, cm, m), FK:

Abbildung 56 Auswahl der Entities der Relation

Abbildung 57 Relation Mitarbeiter zu Abteilung

Nach der Eingabe der Relation sieht das konzeptionelle ER-Modell folgendermaßen aus:

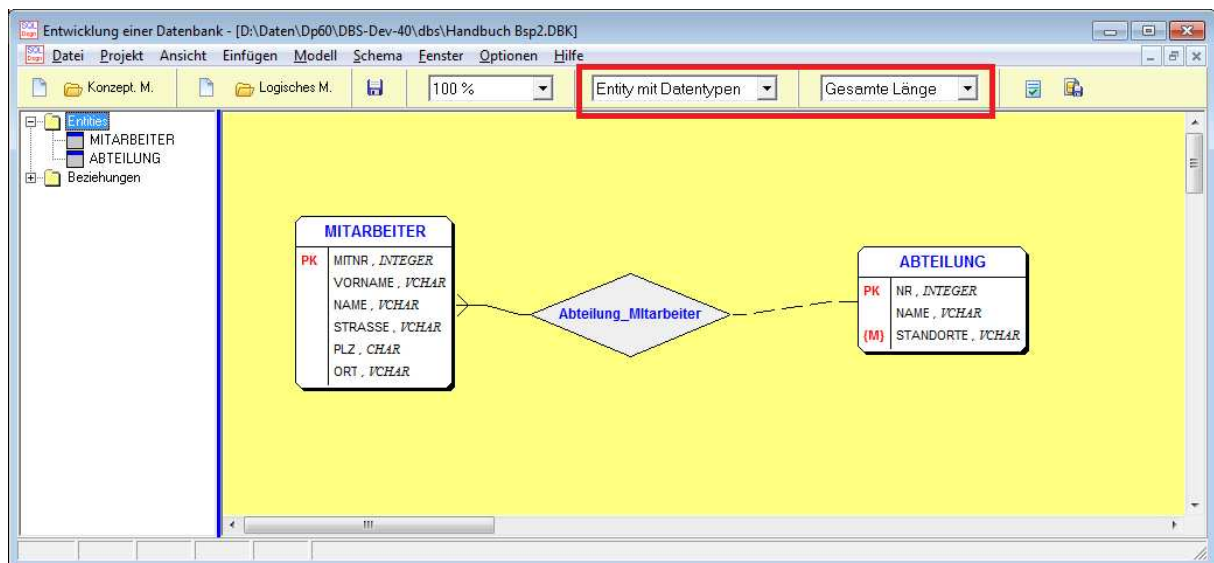


Abbildung 58 Standorte als Multi-Attribute

Mit den beiden aufklappbaren Listen kann man die Datentypen anzeigen lassen.

Interessant ist nun die Umsetzung in das logische Modell. Hier wird nun eine zusätzliche Entität in das Modell eingefügt. Mit der Taste „F9“ erzeugt man das logische Modell.

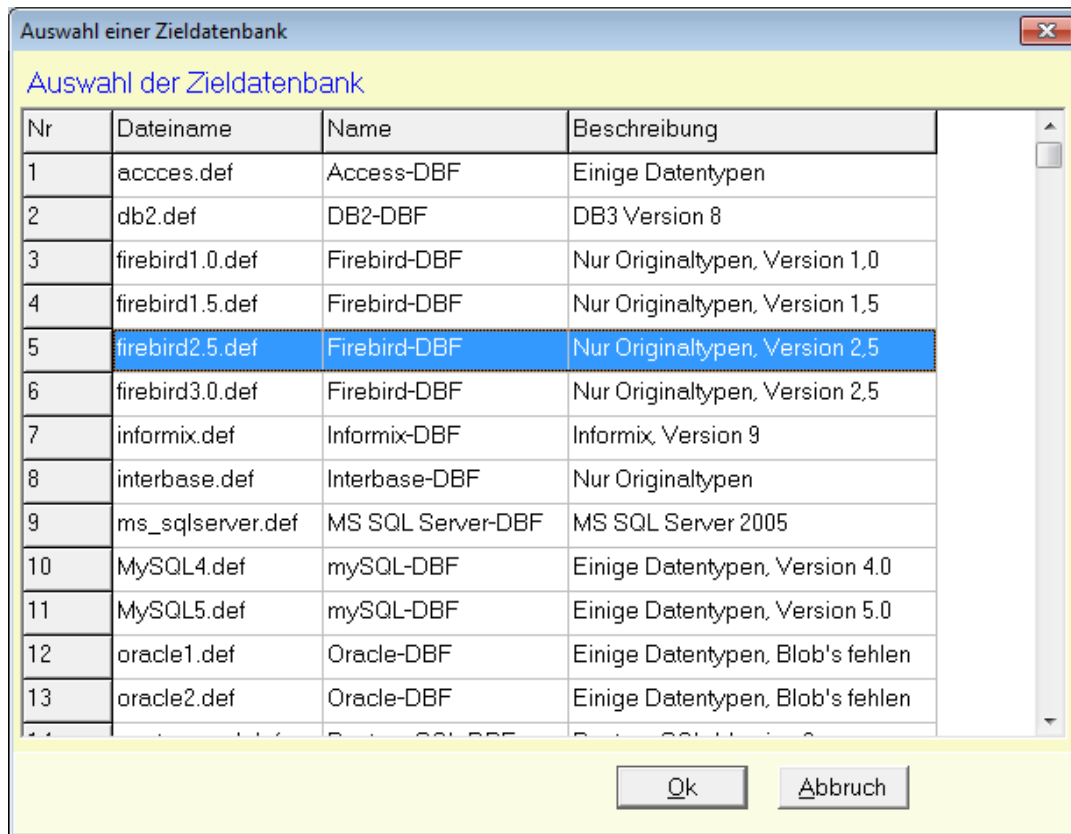


Abbildung 59 Auswahl der Datenbank für das logische Modell

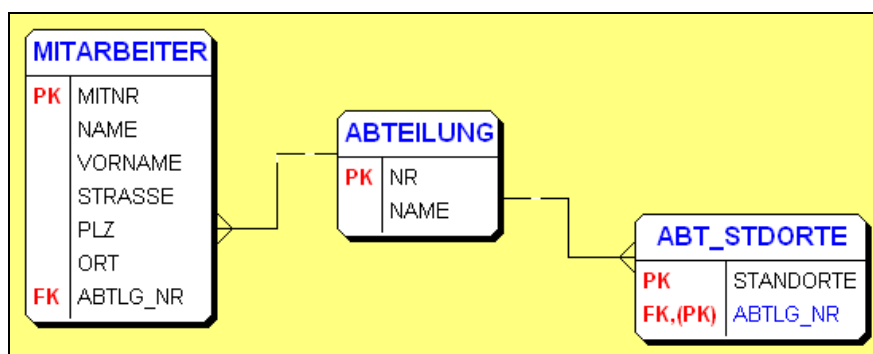


Abbildung 60 Multi-Attribute umgesetzt im logischen Modell

Die Standorte werden als „Weak-Entity“ eingetragen, das heißt, dass der Fremdschlüssel zusätzlich den Primärschlüssel definiert.

### 4.3 Entwicklung eines konzeptionelles ER-Modells

Das folgende Beispiel entwickelt eine einfache Kundendatenbank:, bei der die Relation ein Attribut besitzt.

Dateiname: Handbuch Bsp3 Kunden\_Bestellung\_Rel\_Attrib.DBK

#### 4.3.1 Entitäten

- Kunden
- Bestellung

#### 4.3.2 Beziehungen:

Kunden zu Bestellung                      m:cm

#### 4.3.3 Neues Projekt

Mit der Taste „STRG+M“ wird ein neues Projekt angelegt. Es erscheint der Rahmen mit den Einträgen im Baum und der leeren Diagrammfläche. Im nächsten Schritt müssen nun die Entitäten angelegt werden. Vorab sollte die Definition der Primarykeys definiert werden. Dazu ruft man den Menüeintrag „PROJEKT|Eigenschaften“ auf (siehe erste Beispiel).

##### Hinweis:

Fremdschlüsselattribute sollten nicht angelegt werden. Diese werden automatisch beim Erzeugen des logischen Modells eingetragen.

#### 4.3.4 Entity anlegen

Mit dem Tastendruck „STRG+E“ werden die jeweiligen Entities angelegt.

Nach der Eingabe der Namen, erscheinen die Entities in der Grafik



**Abbildung 61 Neue Entities**

##### 4.3.4.1 Attribute des Entity „Kunden“

- |            |           |
|------------|-----------|
| • KUNDENNR | INTEGER   |
| • NAME     | VCHAR(50) |
| • VORNAME  | VCHAR(50) |
| • STRASSE  | VCHAR(50) |
| • PLZ      | CHAR(5)   |
| • ORT      | VCHAR(50) |

Die untere Abbildung zeigt alle wichtigen Daten des Entity „Kunden“

	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Multi-Attrib	Beschrei
1	PK	KUNDENNR	INTEGER			NOT NULL				-		
2		NAME	VCHAR	50			x			-		
3		VORNAME	VCHAR	50			x			-		
4		STRASSE	VCHAR	50			x			-		
5		PLZ	CHAR	5		NOT NULL	x	00000		-		
6		ORT	VCHAR	50			x			-		

Abbildung 62 Haupt-Eigenschaften des Entities Kunden

#### 4.3.4.2 Attribute des Entity „Bestellung“

- ARTIKELNR                INTEGER
- MENGE                    INTEGER

Abbildung 63 Neues Attribut "Menge"

Numeric bzw. Numbers ist besser als Float (Nachkommastellen mit 0.1)



**Einstellung eines Attributes: MENGE**

Algemein | **Default/Unique** | Checks | Beschreibung | Object-ID

**Default-Wert**

☒ Default-Wert

Wert:

Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung).

**Unique-Bedingungen**

☐ Unique

Check-C. Name:

Ok Abbruch

Abbildung 64 Default-Wert für das Attribut "Menge"

Die untere Abbildung zeigt alle wichtigen Daten des Entity „Bestellung“

**Einstellungen eines Entitys: BESTELLUNG**

Algemein | **Attribute** | Beschreibung | Farben | Position

	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung
1	PK	ARTIKELNR	INTEGER			NOT NULL				-
2		MENGE	NUMERIC	7	2		x	0		-

Add Edit Delete Copy

Ok Abbruch

Abbildung 65 Haupt-Eigenschaften eines Entitys

Nach diesen Vorbereitungen wird die Beziehung mittels STRG+R erstellt.

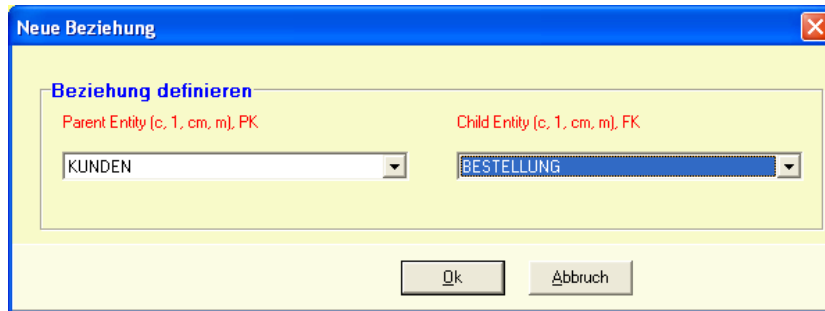


Abbildung 66 Auswahl der Entities der Relation

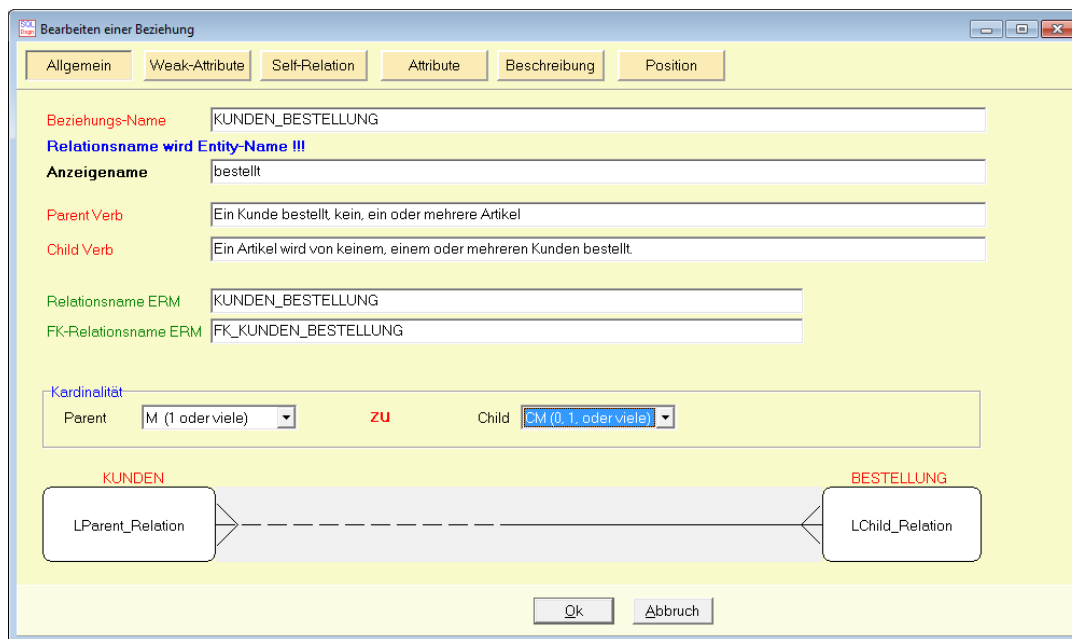


Abbildung 67 Relation Kunden zu Bestellung

Im zweiten Register wird nun das Attribut Datum eingetragen.

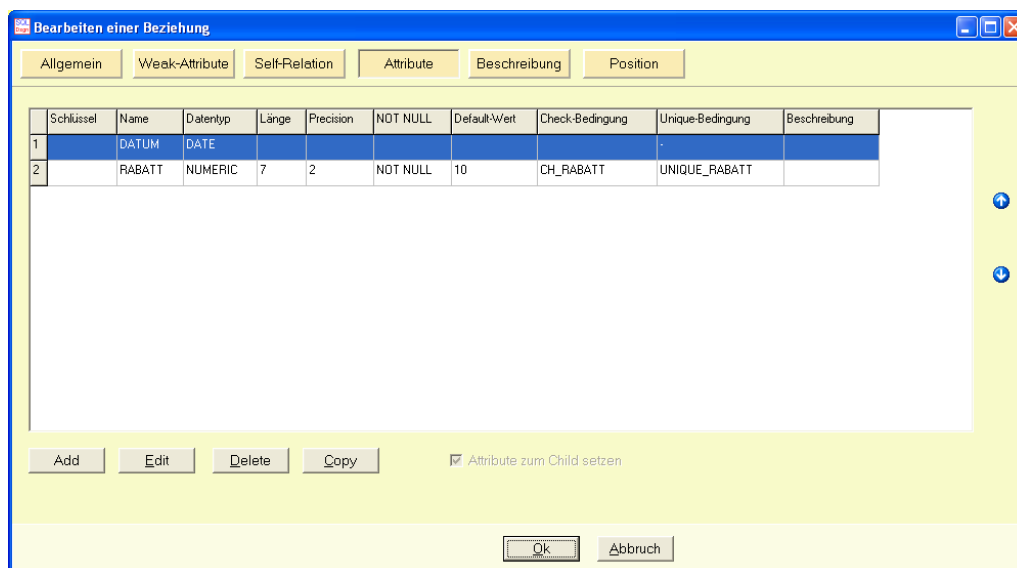


Abbildung 68 Attribut Datum und Rabatt in einer Relation

Im Datum wird das Bestelldatum gespeichert, das Rabattattribut speichert optionale Rabatte

Nach der Eingabe der Relation sieht das konzeptionelle ER-Modell folgendermaßen aus:

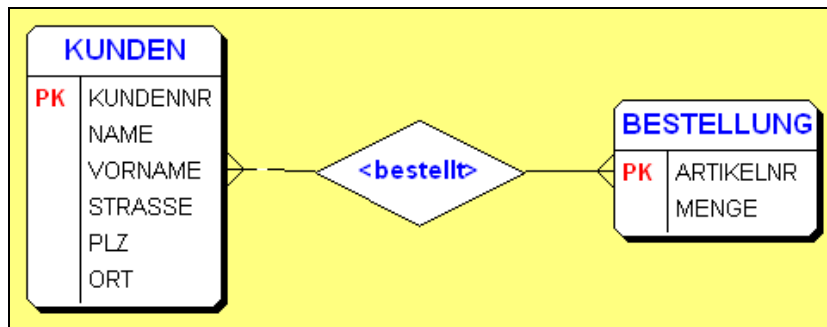


Abbildung 69 Standorte als Multi-Attribute

Im ER-Modell ist durch die Klammerung  $\diamond$  ersichtlich, dass diese Relation Attribute hat. Genauer erfährt man mit dem Mauscursor. Man geht mit dem Mauscursor über die Relation, das erscheint ein Hinweis, Hint, der die Beziehung beschreibt.

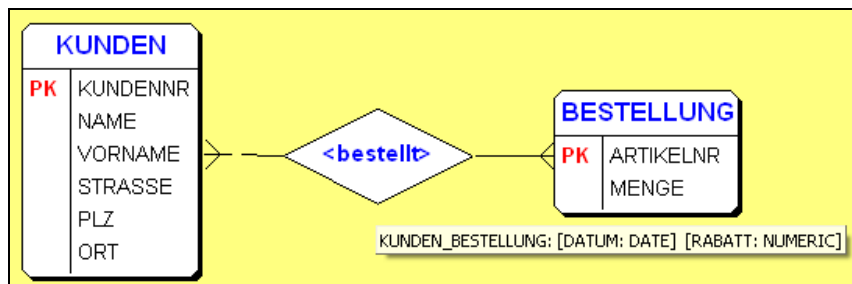


Abbildung 70 Hinweis, Hint, einer Relation

Interessant ist nun die Umsetzung in das logische Modell. Hier wird nun eine zusätzliche Entität in das Modell eingefügt.

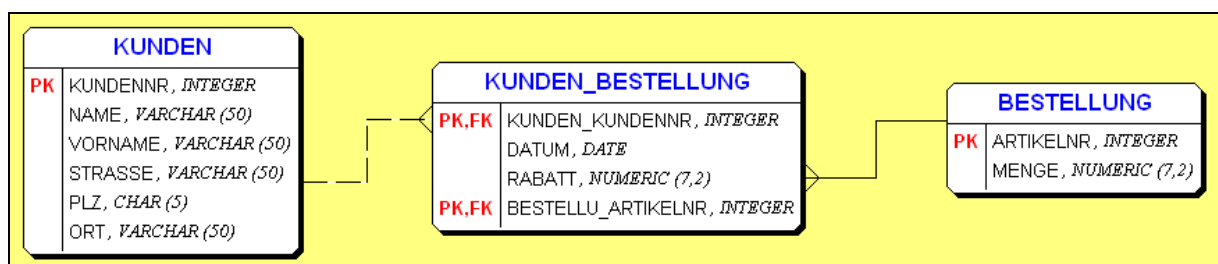


Abbildung 71 Relationen-Attribute umgesetzt im logischen Modell

Bei einer einfachen Beziehung werden die Attribute im „Child“ eingetragen.

#### 4.4 Entwicklung eines konzeptionelles ER-Modells

Das folgende Beispiel entwickelt eine einfache Lieferanten-Datenbank:, bei der eine ternäre Relation eingesetzt wird.

Dateiname: Handbuch Bsp4 Ternäre Beziehung.DBK

Eine Beziehung zwischen zwei Attributen ist die Regel. Manchmal benötigt man aber eine Beziehung, bei der drei oder mehrere Entities vertreten sind. Diese Sonderform ist nun im Designer als ternäre-Beziehung realisiert. Höherwertige Beziehung können nachprogrammiert werden. Ziel ist es, im konzeptionellen Modell schon diese Beziehung deutlich zu darzustellen, ohne die komplexere Darstellung im logischen Modell zu betrachten.

##### 4.4.1 Entitäten

- Lieferant
- Teil
- Projekt

##### 4.4.2 Beziehungen:

Alle Beziehungen als 1:cm

##### 4.4.3 Neues Projekt

Mit der Taste „STRG+M“ wird ein neues Projekt angelegt. Es erscheint der Rahmen mit den Einträgen im Baum und der leeren Diagrammfläche. Im nächsten Schritt müssen nun die Entitäten angelegt werden. Vorab sollte die Definition der Primarykeys definiert werden. Dazu ruft man den Menüeintrag „PROJEKT|Eigenschaften“ auf (siehe erste Beispiel).

##### Hinweis:

Fremdschlüsselattribute sollten nicht angelegt werden. Diese werden automatisch beim Erzeugen des logischen Modells eingetragen.

Ergebnis:

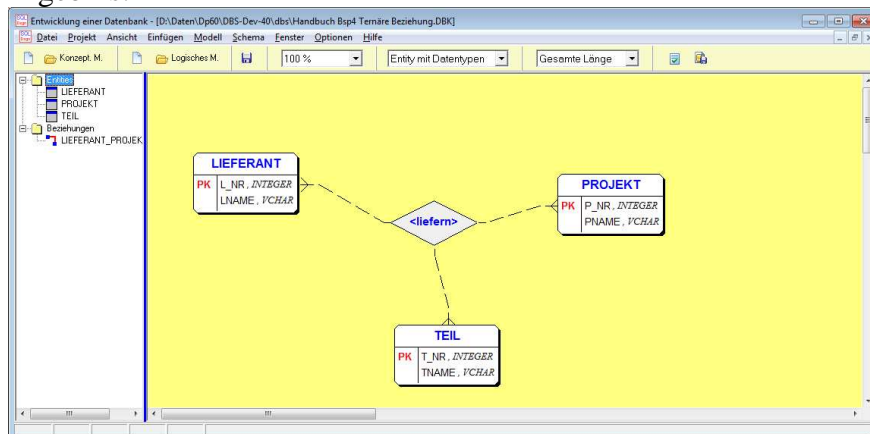


Abbildung 72 Ternäre Beziehung im konzeptionellen Modells

#### 4.4.4 Entity anlegen

Mit dem Tastendruck „STRG+E“ werden die jeweiligen Entities angelegt.

Nach der Eingabe der Namen, erscheinen die Entities in der Grafik

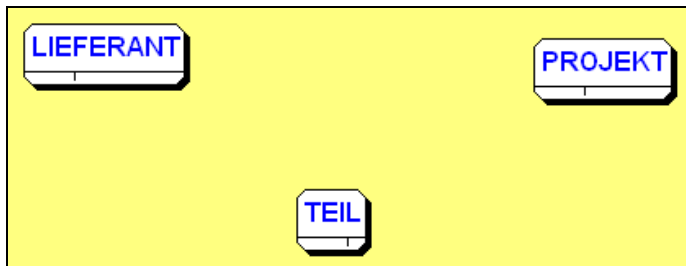


Abbildung 73 Neue Entities

##### 4.4.4.1 Attribute des Entity „Lieferant

- L\_NR                      INTEGER
- LNAME                  VCHAR(50)

Die untere Abbildung zeigt alle wichtigen Daten des Entity „Lieferant“

The screenshot shows the 'Einstellungen eines Entitys: LIEFERANT' dialog box. The 'Attribute' tab is selected. The table below lists the attributes of the 'Lieferant' entity.

	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Multi-Attrib
1	PK	L_NR	INTEGER			NOT NULL				-	
2		LNAME	VCHAR	50		NOT NULL	x			-	

Below the table are buttons for 'Add', 'Edit', 'Delete', and 'Copy'. At the bottom right are 'Ok' and 'Abbruch' buttons.

Abbildung 74 Haupt-Eigenschaften des Entities Lieferant

##### 4.4.4.2 Attribute des Entity „Teil“

- T\_NR                      INTEGER
- TNAME                  VCHAR(50)

Die untere Abbildung zeigt alle wichtigen Daten des Entity „Teil“

Ein Dialogfenster mit dem Titel 'Einstellungen eines Entities: TEIL'. Es enthält fünf Registerkarten: 'Allgemein', 'Attribute', 'Beschreibung', 'Farben' und 'Position'. Die 'Attribute'-Registerkarte ist aktiv. Darunter befindet sich eine Tabelle mit den Spalten: Schlüssel, Name, Datentyp, Länge, Precision, NOT NULL, Default-Y/N, Default-Wert, Check-Bedingung, Unique-Bedingung und Multi-Attrib. Die Tabelle enthält zwei Zeilen:

	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Multi-Attrib
1	PK	T_NR	INTEGER			NOT NULL				-	
2		TNAME	VCHAR	50		NOT NULL	x			-	

Unter der Tabelle befinden sich die Buttons 'Add', 'Edit', 'Delete' und 'Copy'. Am unteren Rand des Dialogs befinden sich die Buttons 'Ok' und 'Abbruch'.

Abbildung 75 Haupt-Eigenschaften des Entities Teil

#### 4.4.4.3 Attribute des Entity „Projekt“

- P\_NR                      INTEGER
- PNAME                  VCHAR(50)

Die untere Abbildung zeigt alle wichtigen Daten des Entity „Projekt“

Ein Dialogfenster mit dem Titel 'Einstellungen eines Entities: PROJEKT'. Es enthält fünf Registerkarten: 'Allgemein', 'Attribute', 'Beschreibung', 'Farben' und 'Position'. Die 'Attribute'-Registerkarte ist aktiv. Darunter befindet sich eine Tabelle mit den Spalten: Schlüssel, Name, Datentyp, Länge, Precision, NOT NULL, Default-Y/N, Default-Wert, Check-Bedingung, Unique-Bedingung, Multi-Attrib und Beschreib. Die Tabelle enthält zwei Zeilen:

	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Multi-Attrib	Beschreib
1	PK	P_NR	INTEGER			NOT NULL				-		
2		PNAME	VCHAR	50		NOT NULL	x			-		

Unter der Tabelle befinden sich die Buttons 'Add', 'Edit', 'Delete' und 'Copy'. Am unteren Rand des Dialogs befinden sich die Buttons 'Ok' und 'Abbruch'.

Abbildung 76 Haupt-Eigenschaften des Entities Teil

Nach diesen Vorbereitungen wird die ternäre Beziehung mittels STRG+T oder Menü „Einfügen“, Eintrag „Ternäre Beziehung“ erstellt.

**Neue Ternäre Beziehung**

**Beziehung definieren**

1. Parent Entity (c, 1, cm, m), PK  
LIEFERANT

2. Parent Entity (c, 1, cm, m), PK  
TEIL

3. Parent Entity (c, 1, cm, m), PK  
PROJEKT

OK Abbruch

Abbildung 77 Auswahl der Entities der ternären Relation

**Bearbeiten einer Beziehung**

Allgemein Attribute Beschreibung Position

Beziehungs-Name LIEFERANT\_PROJEKT\_TEIL

LEntityName

Anzeigename liefern

1. Verb LIEFERANT Ein Lieferant liefert kein, ein oder mehrere Teile

2. Verb PROJEKT Ein Projekt beinhaltet kein, ein oder mehrere Teile, Lieferanten

3. Verb TEIL Ein Teil wird von keinem, einem oder mehreren Lieferanten geliefert und wird in keinem, einem oder mehreren Projekten benutzt

Kardinalität

1. LIEFERANT CM (0, 1, oder viele)

2. PROJEKT CM (0, 1, oder viele)

3. TEIL CM (0, 1, oder viele)

OK Abbruch

Abbildung 78 Relation Kunden zu Bestellung

Beziehungsverben:

- Ein Lieferant liefert kein, ein oder mehrere Teile
- Ein Projekt beinhaltet kein, ein oder mehrere Teile, Lieferanten
- Ein Teil wird von keinem, einem oder mehreren Lieferanten geliefert und wird in keinem, einem oder mehreren Projekten benutzt

Im zweiten Register werden nun die beiden Attribute Datum und Menge eingetragen.

Bearbeiten einer Beziehung

Allgemein Attribute Beschreibung Position

	Schlüssel	Name	Datentyp	Länge	Precision	NOT NULL	Default-Wert	Check-Bedingung	Unique-Bedingung	Beschreibung
1	PK	DATUM	DATE			NOT NULL			-	
2		MENGE	INTEGER						-	

Ok Abbruch

Abbildung 79 Attribut Datum und Menge in einer ternären Relation

Nach der Eingabe der Relation sieht das konzeptionelle ER-Modell folgendermaßen aus:

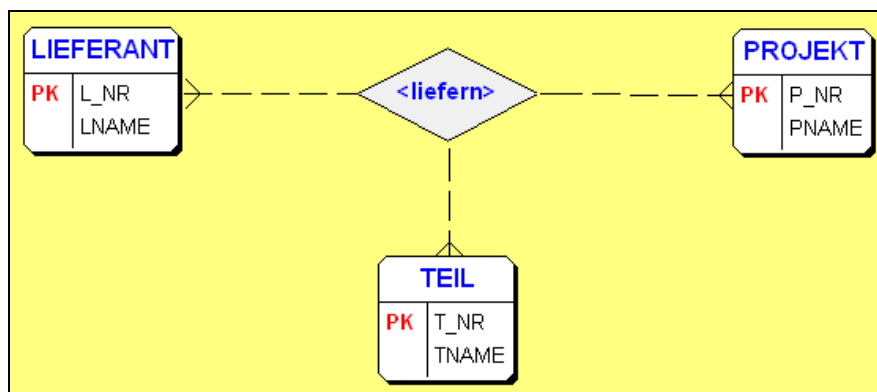


Abbildung 80 ER-Modell einer ternären Beziehung

Im ER-Modell ist durch die Klammerung  $\diamond$  ersichtlich, dass diese Relation Attribute hat. Genauer erfährt man mit dem Mauscursor. Man geht mit dem Mauscursor über die Relation, das erscheint ein Hinweis, Hint, der die Beziehung beschreibt.

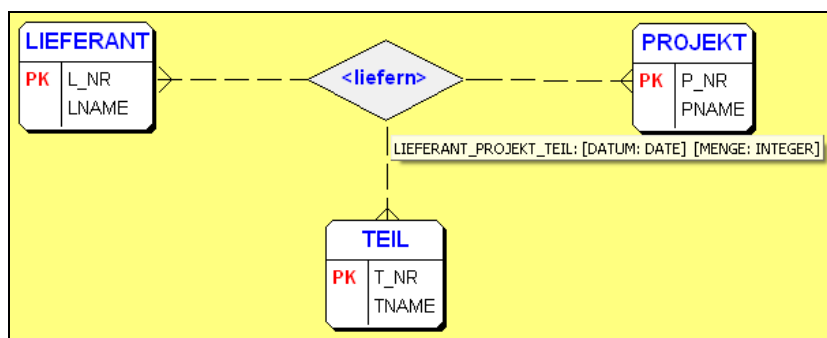


Abbildung 81 Hinweis, Hint, einer Relation

Interessant ist nun die Umsetzung in das logische Modell. Hier wird nun eine zusätzliche Entität in das Modell eingefügt.



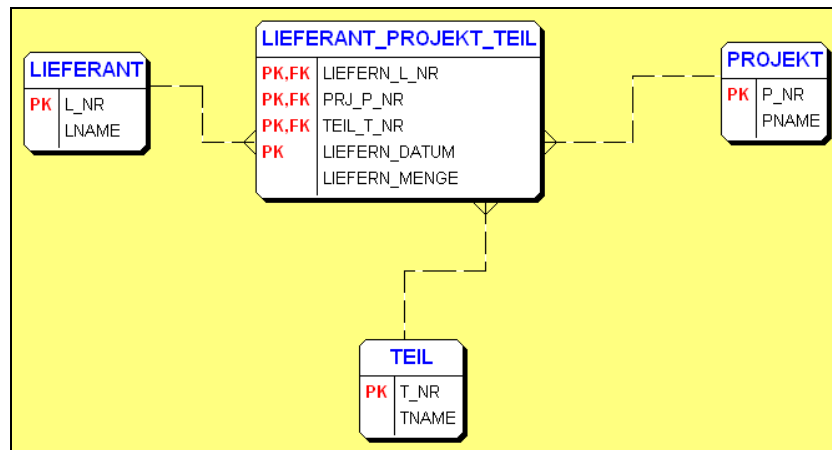


Abbildung 82 Relationen-Attribute umgesetzt im logischen Modell

Es wird eine zusätzliche Entität eingefügt, die aber alle Primärschlüssel als Fremdschlüssel einfügt. Zusätzlich werden die Attribute der ternären Relation in das neue Entity eingetragen.

## 4.5 Beispiel5, Entwicklung eines logischen ER-Modell

Dieses Beispiel zeigt die Entwicklung eines relativ einfachen logischen Modells. Die Vorgaben werden vorab aufgelistet, so dass man Übungshalber dieses Modell auch ohne diese Vorlage entwickeln kann.

### 4.5.1 Aufgabenstellung des fünften Beispiels

Entwickelt werden soll ein Modell zur Modellierung einer Studentendatenbank. Folgende Entities wurden definiert:

- Student
- Fachbereich
- Vorlesung

#### Teilaufgaben:

- Definieren der Attribute
- Definieren der Beziehungen (es sollten zwei unterschiedliche Arten sein)
- Einbau von Prüfbedingungen
- Generieren des SQL-Skriptes

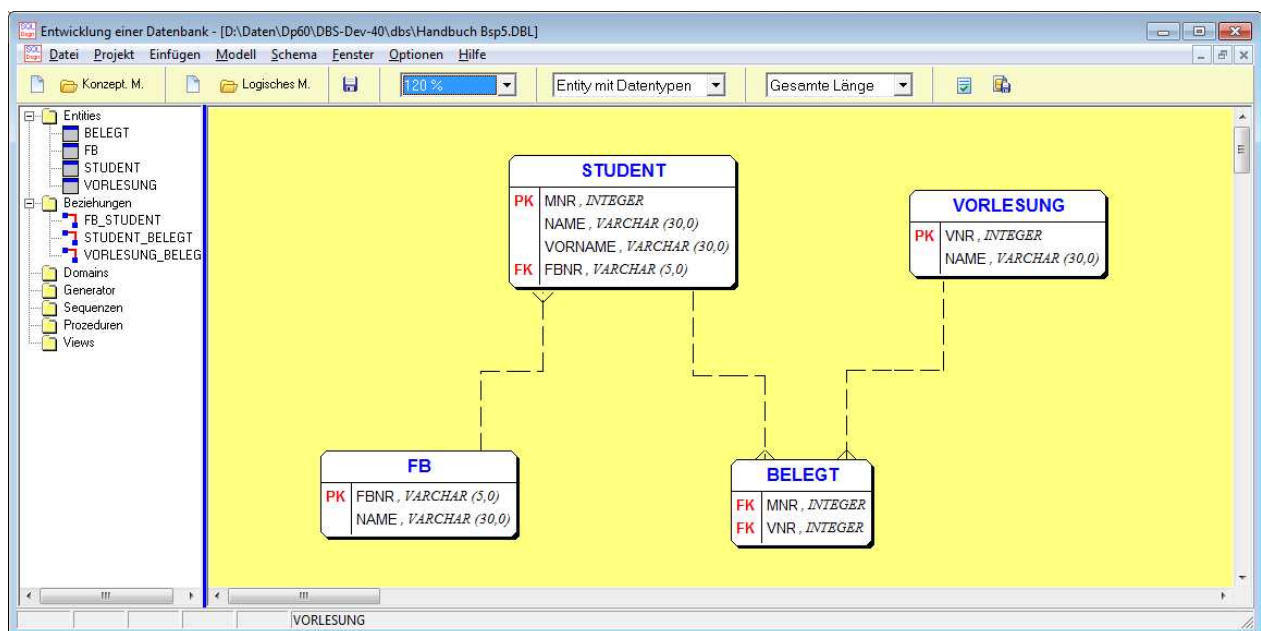


Abbildung 83 Beispiel Handbuch Bsp5.DBL

### 4.5.2 Neues Projekt

Mit der Taste „STRG+N“ wird ein neues Projekt angelegt. Es erscheint ein Dialogfenster, in dem man die Zieldatenbank auswählt (Abbildung 84).

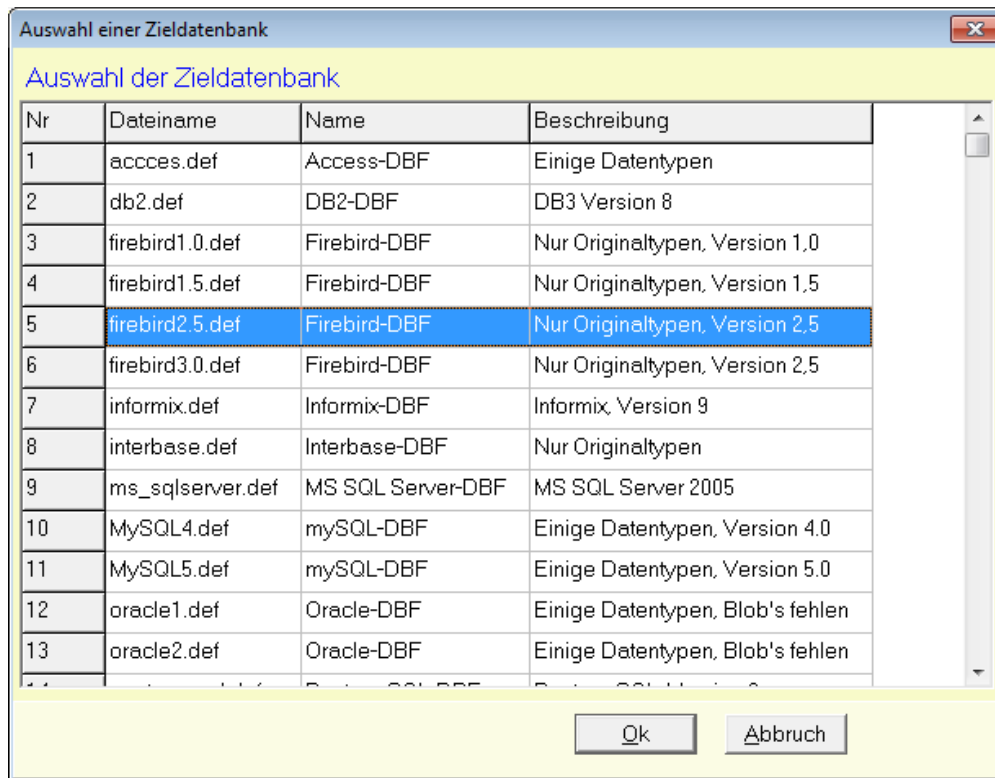


Abbildung 84 Auswahl der Datenbank

Im nächsten Schritt ruft man im Menü „Projekt“ den Eintrag „Eigenschaften“ auf. Im Dialogfenster kann man die notwendigen Einträge vornehmen. Besonders erwähnenswert ist die Auswahl der möglichen Darstellungen.

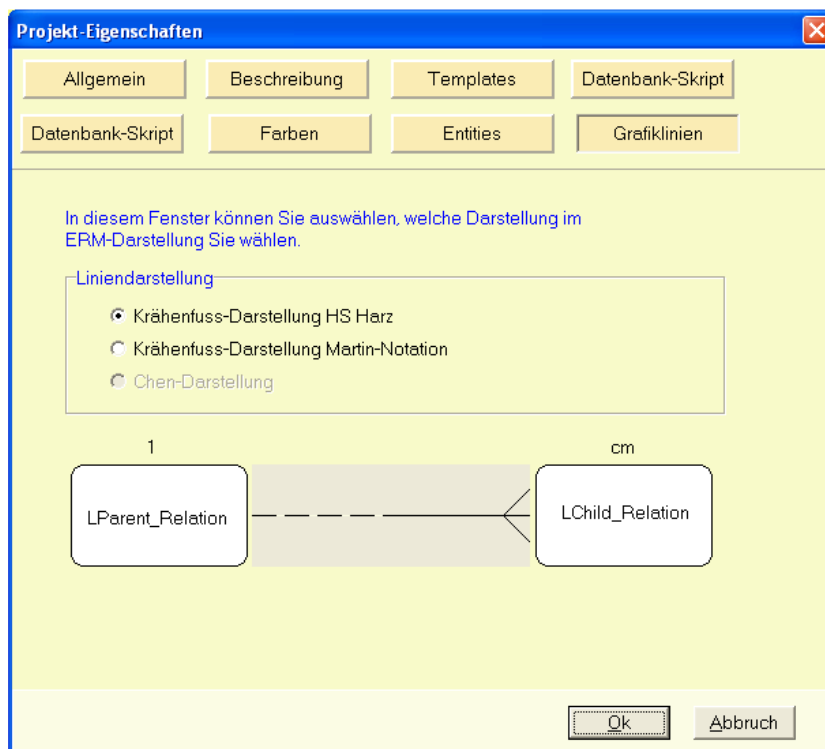


Abbildung 85 Auswahl der Darstellungen

Im ersten Schritt sollte die Definition der Primarykeys definiert werden. Dazu ruft man den Menüeintrag „PROJEKT|Eigenschaften“ auf.

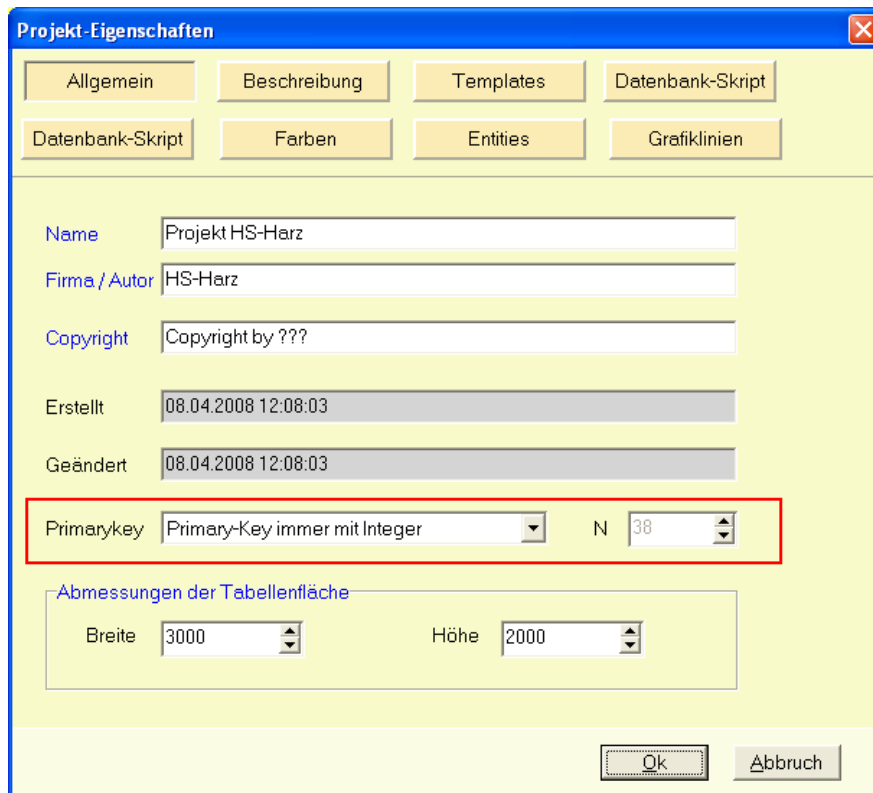


Abbildung 86 Projekteigenschaften (Integer vs. Max Numeric)

In der Comboliste kann man entscheiden, ob man der Datentyp eines Primarykeys ein Integer oder ein Numeric/Number-Datentyp ist.

#### 4.5.3 Entities anlegen

Mit dem Tastendruck „STRG+E“ werden die jeweiligen Entities angelegt.

##### Hinweis:

Dabei sollten die Fremdschlüssel mit anderen Entities noch nicht eingefügt werden!!!

Zur erst wird nach dem Namen gefragt, dieser darf nur aus Buchstaben, Zahlen und dem Underline („\_“) bestehen. Es darf kein Leerzeichen verwendet werden. Am Anfang muss auch ein Buchstabe stehen.

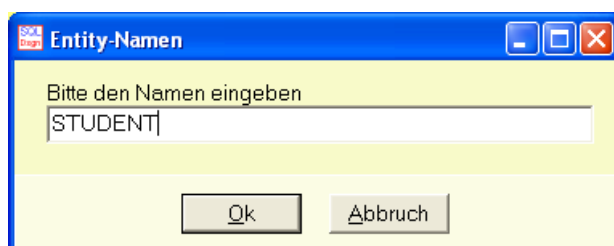


Abbildung 87 Erzeugen eines Entities

Nach der Eingabe der Namen, erscheint das Entity in der Grafik. Nun werden alle weiteren Entitäten eingetragen (siehe Abbildung 88).

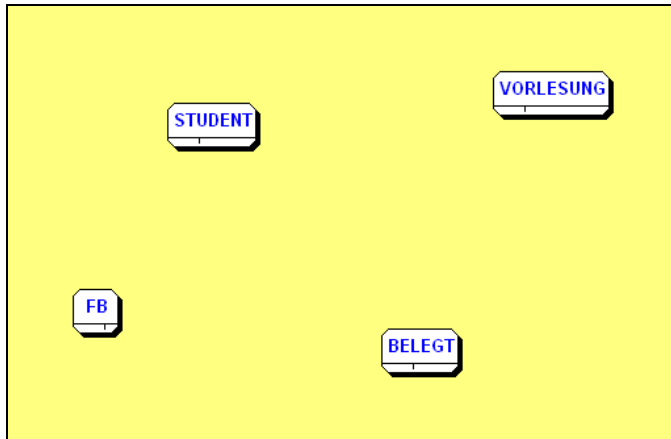


Abbildung 88 Entites des zweiten Beispiels

Mit einem Doppelklick werden nun die einzelnen Entitäten bearbeiten und die Attribute eingetragen (Abbildung 94). Hier sollten die Einträge ausgefüllt werden. Des Weiteren ist es ratsam, die nähere Beschreibung des Entities im Register „Beschreibung“ vorzunehmen. Im zweiten Register, siehe Abbildung 94, werden dann die Attribute eingetragen.

Das Screenshot zeigt das Dialogfeld 'Einstellung eines Attributes: Attribut1'. Das Dialogfeld hat eine gelbe Hintergrundfarbe und eine blaue Titelleiste. Es enthält mehrere Registerkarten: 'Allgemein', 'Default / Unique', 'Checks', 'Linked-Attribute', 'Beschreibung', 'Zu Erledigen' und 'Object-ID'. Die 'Allgemein'-Registerkarte ist aktiv. In der 'Allgemein'-Registerkarte sind folgende Felder und Kontrollen zu sehen:

- Ein Textfeld 'Name' mit dem Wert 'MNR'.
- Drei Kontrollkästchen: ☒ 'Primärschlüssel', ☒ 'Not Null' und ☐ 'Fremdschlüssel'.
- Ein Bereich 'Datentyp-Definition' mit einem Dropdown-Menü 'Domain' (aktuell 'keine Domain'), einem Dropdown-Menü 'Datentyp' (aktuell 'INTEGER') und zwei Spinboxen für die Werte '1' und '0'.
- Ein Textfeld 'Formel'.
- Ein Bereich 'Autonumber' mit einem Kontrollkästchen ☐ 'aktiviert', einem Textfeld 'Start' mit dem Wert '0' und einem Textfeld 'Increment' mit dem Wert '1'.
- Ein Button 'Neue Domain'.
- Am unteren Rand befinden sich die Buttons 'Ok' und 'Abbruch'.

Abbildung 89 Neues Attribut "Mnr"

Einstellung eines Attributes: Attribut2

Allgemein Default / Unique Checks Linked-Attribute Beschreibung Zu Erledigen Object-ID

Name NAME

☐ Primärschlüssel ☐ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain keine Domain Neue Domain

Datentyp VARCHAR 50 0

Formel

Datentyp kein Datentyp 1 0

**Autonumber**

☐ aktiviert Start 0 Increment 1

Ok Abbruch

Abbildung 90 Neues Attribut "Name"

Einstellung eines Attributes: Attribut2

Allgemein Default / Unique Checks Linked-Attribute Beschreibung Zu Erledigen Object-ID

**DefaultWert: (CHAR)**

☒ Defaultwert

Wert

Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung).

**Unique-Bedingungen**

☐ Unique

Check-C. Name

Ok Abbruch

Abbildung 91 Default-Wert für Name eintragen

Fachbereich hat eine Zeichenkette als Primarykey:

**Einstellung eines Attributes: Attribut1**

Name:

☒ Primärschlüssel
 ☒ Not Null
 ☐ Fremdschlüssel

**Datentyp-Definition**

Domain:

Datentyp:

Formel:

Datentyp:

**Autonumber**

☐ aktiviert
 Start: 
 Increment:

Abbildung 92 Neues Attribut "FB" als Primary Key" (Datentyp Zeichenkette)

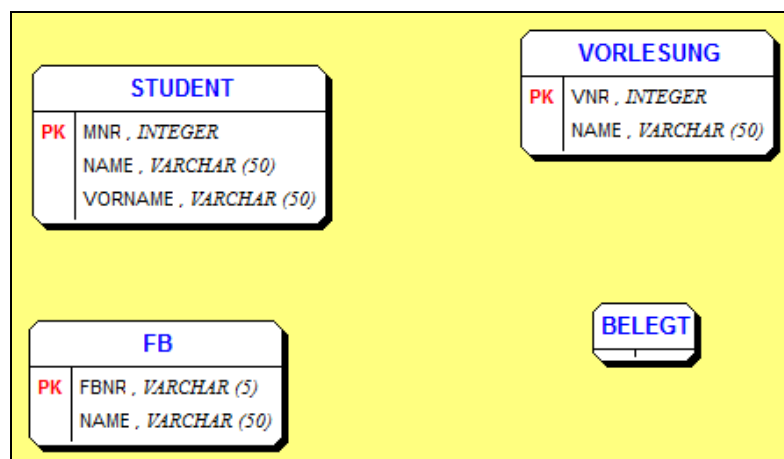


Abbildung 93 Entities ohne die Beziehungen

Die Attribute „MNR“ und „VNR“ werden in dem Entity „Belegt“ automatisch eingetragen.

	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Beschreibung
1	PK	MNR	INTEGER				NOT NULL				-	
2		NAME	VARCHAR	50				x			-	
3		VORNAME	VARCHAR	50				x			-	

Abbildung 94 Haupt-Eigenschaften des Entities Student

Das zweite Register verwaltet die Attribute. Mit den unteren Schaltern kann man diese bearbeiten, löschen und erzeugen. Die Schalter rechts an der Seite verschieben die Reihenfolge.

#### Bemerkungen:

- Die Beziehung „Student“ zu „FB“ ist eine normale 1:cm-Beziehung.
- Die Beziehung „Student“ zu „Vorlesung“ ist eine m:m-Beziehung, deshalb musste ein drittes Entity „belegt“ eingeführt werden.

Im nächsten Abschnitt werden die Beziehungen eingetragen.

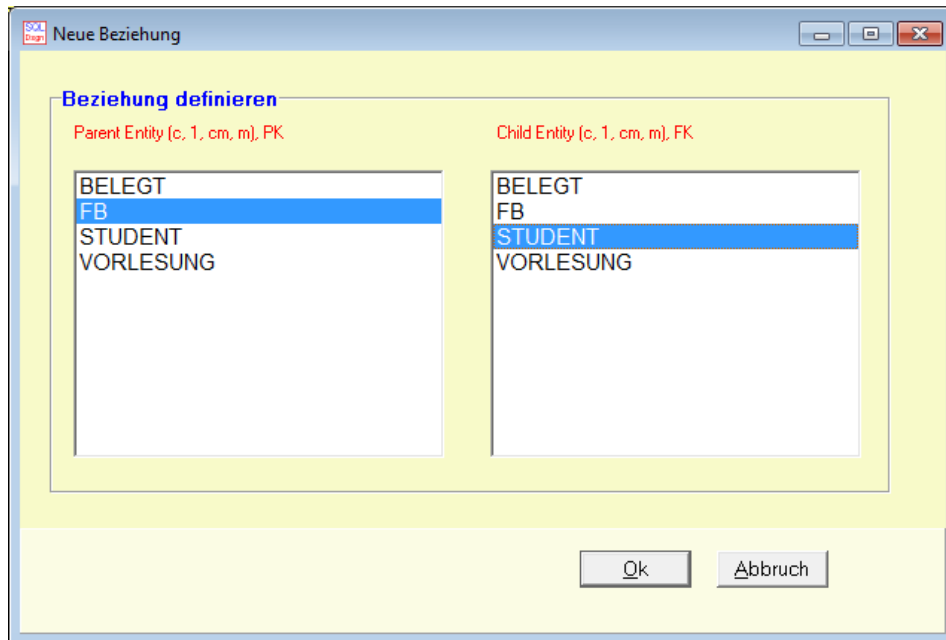
#### 4.5.4 Beziehungen anlegen

Dazu ruft man aus dem Hauptmenü „Einfügen“ die Funktion „Beziehung“ auf. Es erscheint folgendes Fenster:

Abbildung 95 Erstellen einer Beziehung



In beiden Listen sind alle Entities eingetragen. In der linken Liste wählt man den Parent-Entity aus. Das wäre hier in diesem Beispiel „FB“. In der rechten Liste wird die Beziehung eingetragen. Hier also „Student“.



**Abbildung 96 Korrekte Einstellung für das Beispiel**

Nach dem Betätigen des Schalters „Ok“ erscheint der Relationsdialog, indem die Beziehungsdaten eingetragen werden. Danach wird der Primarykey „FBNR“ automatisch als Fremdschlüssel in das Entity „Student“ eingefügt. Die Beziehungsverben sollten vervollständigt werden.

Die Abbildung 97 zeigt die endgültigen Eingaben. Mit dem Schalter „Ok“ wird die Beziehung eingefügt. Danach erscheint das eigentliche Fenster zur Definition bzw. zur Bearbeitung einer Beziehung.

Abbildung 97 Definition der Beziehung

Beziehungstexte:

- Ein Fachbereich hat keinen, einen oder mehrere Studenten
- Ein Student gehört zu einem Fachbereich

Im ersten Register werden die Grundeinstellungen (Name, Beziehungstyp) eingetragen. Wichtig sind hier die Einträge „Parent-Verb“ und „Child-Verb“. Sie zeigen den Charakter der Beziehung an. Das Parent-Verb beschreibt die Beziehung durch Wörter wie „Kunde hat“ etc.

Das zweite Register wird zurzeit nicht unterstützt.

Im dritten Register werden die Attribute verknüpft, mit denen die Beziehung definiert wird. Wie oben erwähnt, sind keine Fremdattribute im Child-Entity („Emp“) eingetragen. Wechselt man in das dritte Register, wird man einmal gefragt, ob alle Primärattribute an das Child übertragen werden sollen (**Fehler! Verweisquelle konnte nicht gefunden werden.**).

Sinnvoll ist es, diese Frage zu bejahen. Dann werden alle Primär-Attribute in das Child-Entity eingefügt. Diese Attribute erhalten dann die Eigenschaft „Fremdschlüssel“. Die Abbildung 100 zeigt die fertige Liste. Sie können aber auch die Attribute manuell verknüpfen. Dazu markiert man in der linken und in der rechten Tabelle jeweils das gewünschte Attribut. Mit dem Schalter „Connect“ werden diese in die obere Tabelle eingefügt. Mit dem Schalter „DisConnect“ wird die ausgewählte Verbindung wieder gelöst. Die Datentypen der beiden Attributen müssen natürlich exakt übereinstimmen und vorher in den Entities eingetragen worden sein.

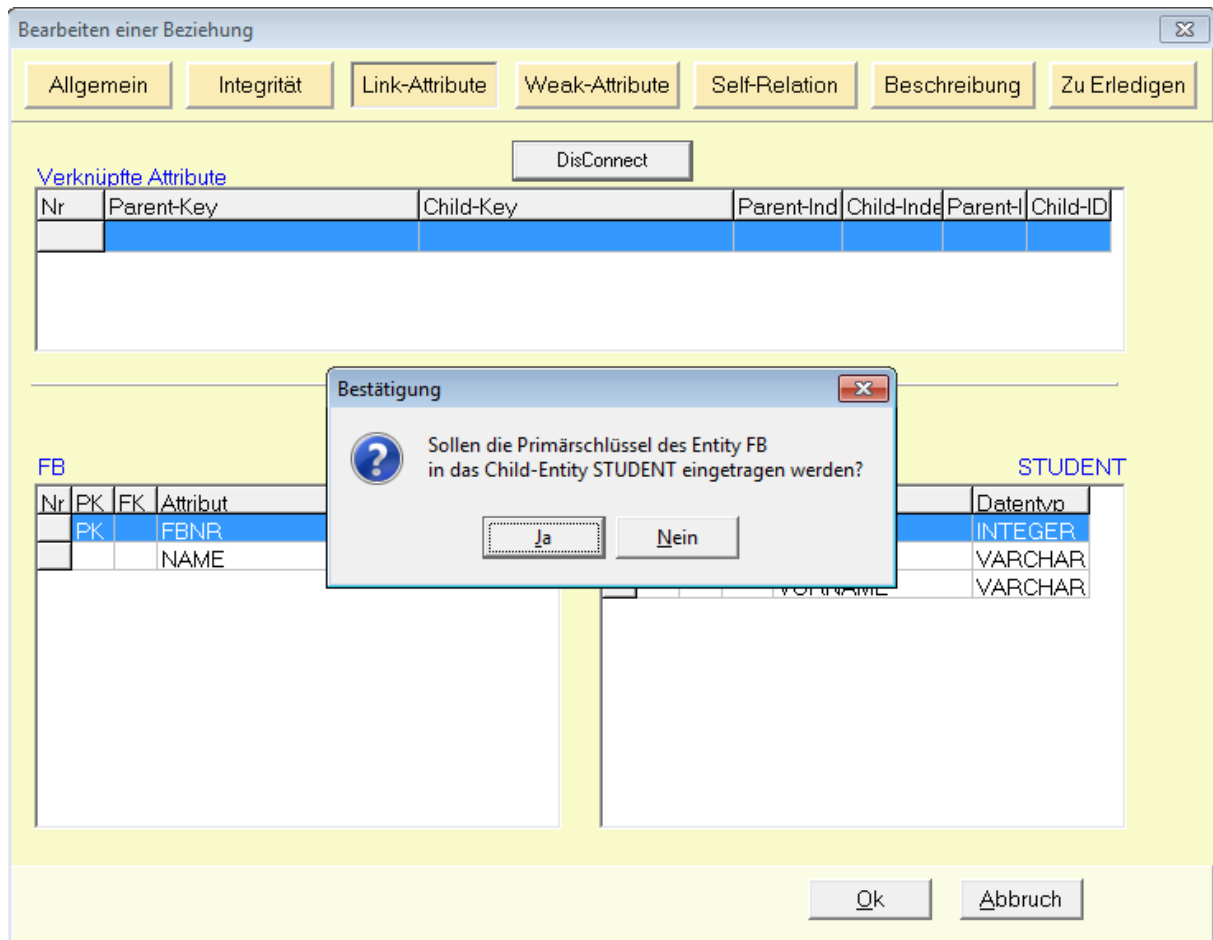


Abbildung 98 Schalter "Link-Attribute" gedrückt, Frage nach der Übertragung des Attribut

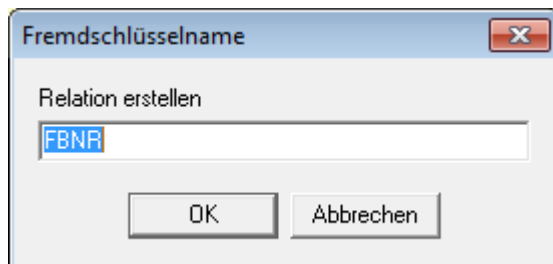


Abbildung 99 Eingabe des Foreignkey-Namens

Die untere Abbildung zeigt die Beziehung, nachdem die Primär-Attribute in das Child-Entity übertragen wurden.

Bearbeiten einer Beziehung

Allgemein   Integrität   Link-Attribute   Weak-Attribute   Self-Relation   Beschreibung   Zu Erledigen

DisConnect

Nr	Parent-Key	Child-Key	Parent-Ind	Child-Ind	Parent-ID	Child-ID
1	FBNR	FBNR	0	3	45	84

Connect

FB

Nr	PK	FK	Attribut	Datentyp
1	PK		FBNR	VARCHAR
			NAME	VARCHAR

STUDENT

Nr	PK	FK	Neu	Attribut	Datentyp
	PK			MNR	INTEGER
				NAME	VARCHAR
				VORNAME	VARCHAR
1		FK	Neu	FBNR	VARCHAR

Ok   Abbruch

Abbildung 100 Übertragen eines Fremdschlüssels

Das vierte Register „Weak-Attribute“ dient der Definition einer Beziehung einer Weak-Entity (siehe Kapitel 10.1, Seite 125).

Mit dem Betätigen des Schalters „Ok“ wird die Beziehung eingetragen und angezeigt. Mit der Taste F6 erhält man eine Liste aller Beziehungen.

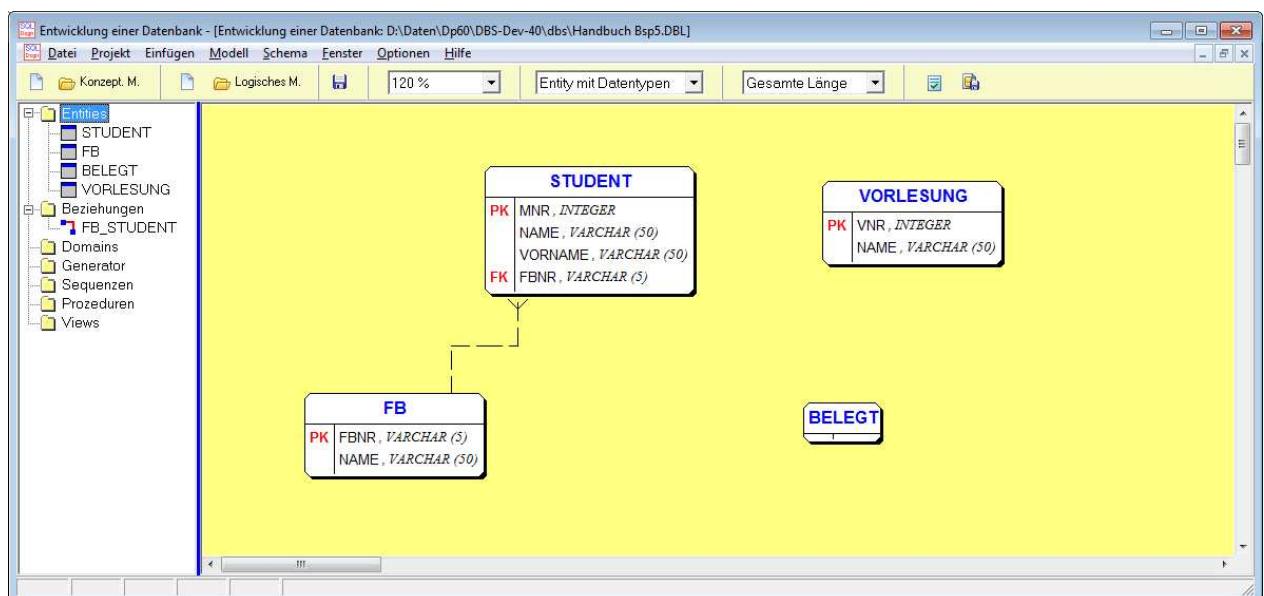


Abbildung 101 Erste Beziehung im fünften Beispiel

#### 4.5.5 Beziehung der Vorlesungen

Die Beziehung der Vorlesungen entspricht einer cm:cn-Beziehung. Dazu existiert das neue Entity „Belegt“. Es müssen nun zwei Relationen eingetragen werden.

- a) Student zu Belegt
- b) Vorlesungen zu Belegt

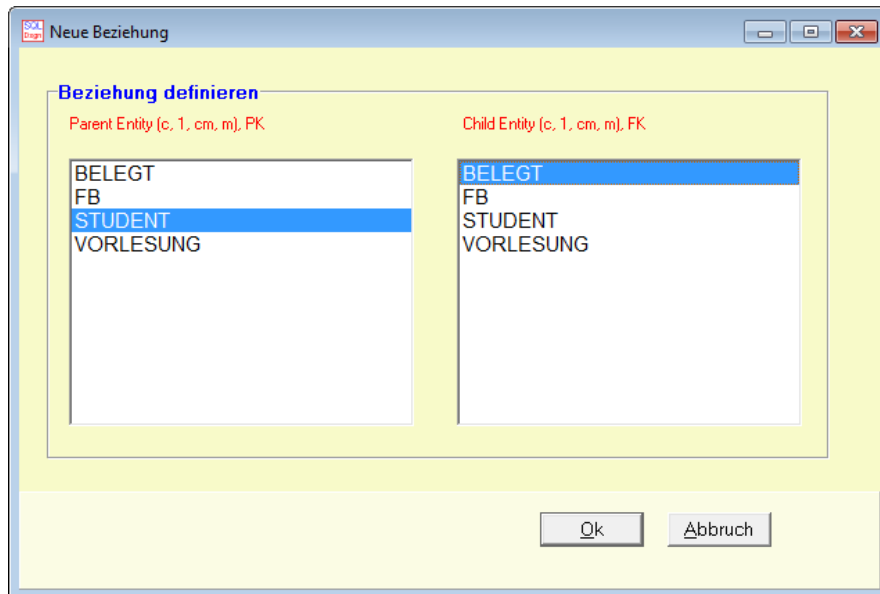


Abbildung 102 Beziehung Student zu Belegt

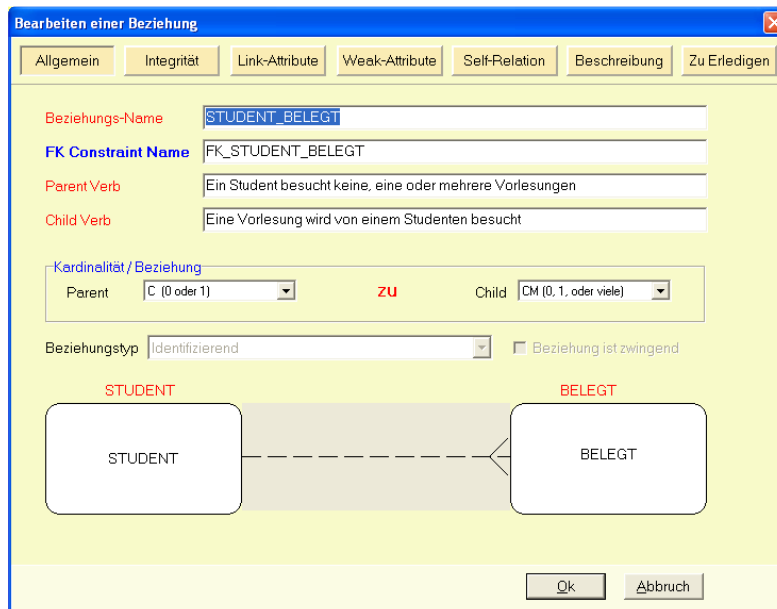


Abbildung 103 Beziehung „Student“ zu „Belegt Vorlesung“

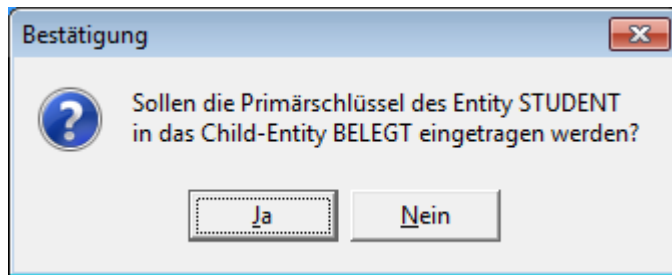


Abbildung 104 Automatische Übertragung des Primary-Keys als Foreign-Key

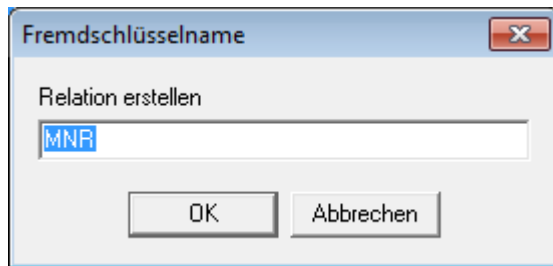


Abbildung 105 Name des Foreign-Keys

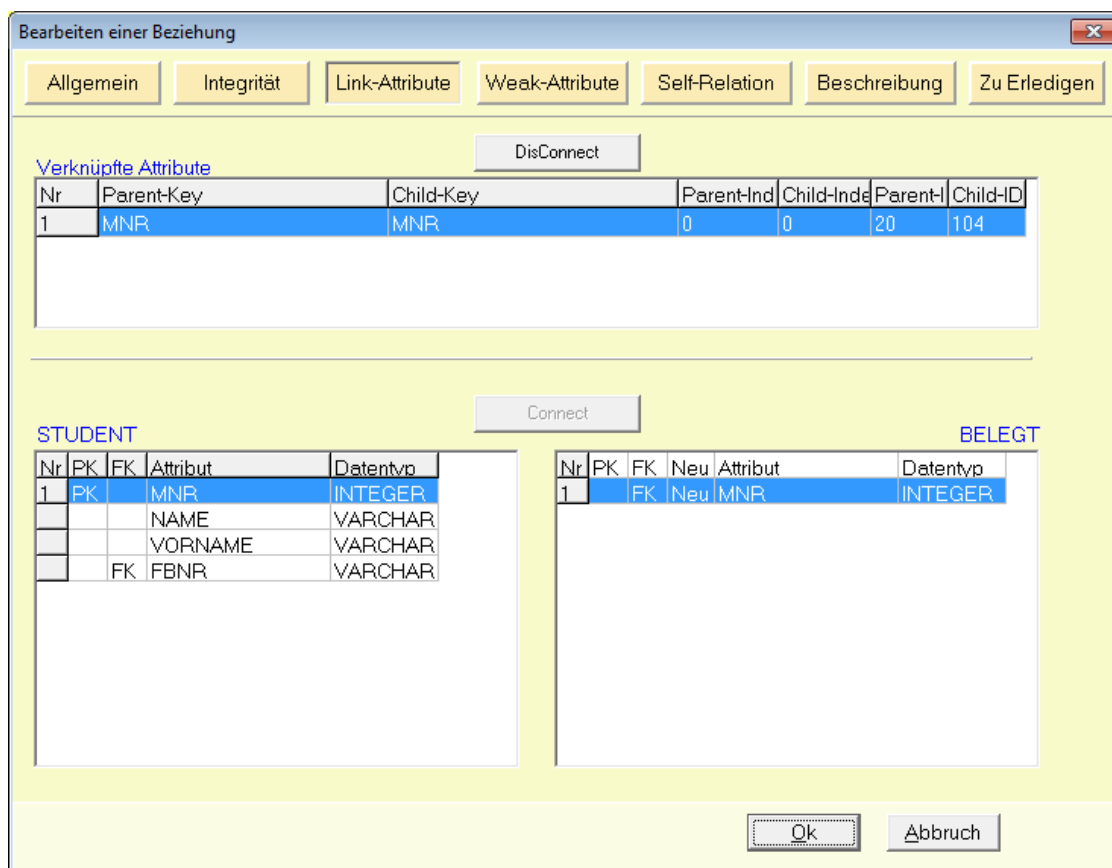


Abbildung 106 Anzeige der Eingaben der Beziehung

Ergebnis:

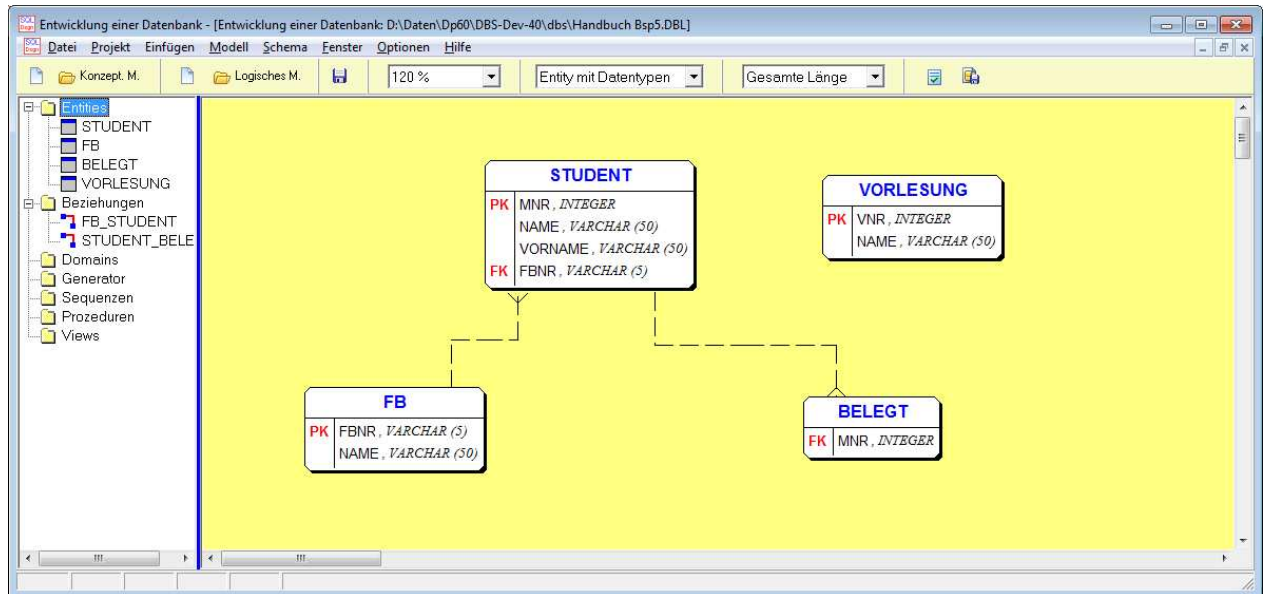


Abbildung 107 Aktuelle Fassung mit zwei Beziehungen

Das nächste Dialogfenster zeigt die korrekte Einstellung. Links das Parent-Entity, rechts das Child-Entity.

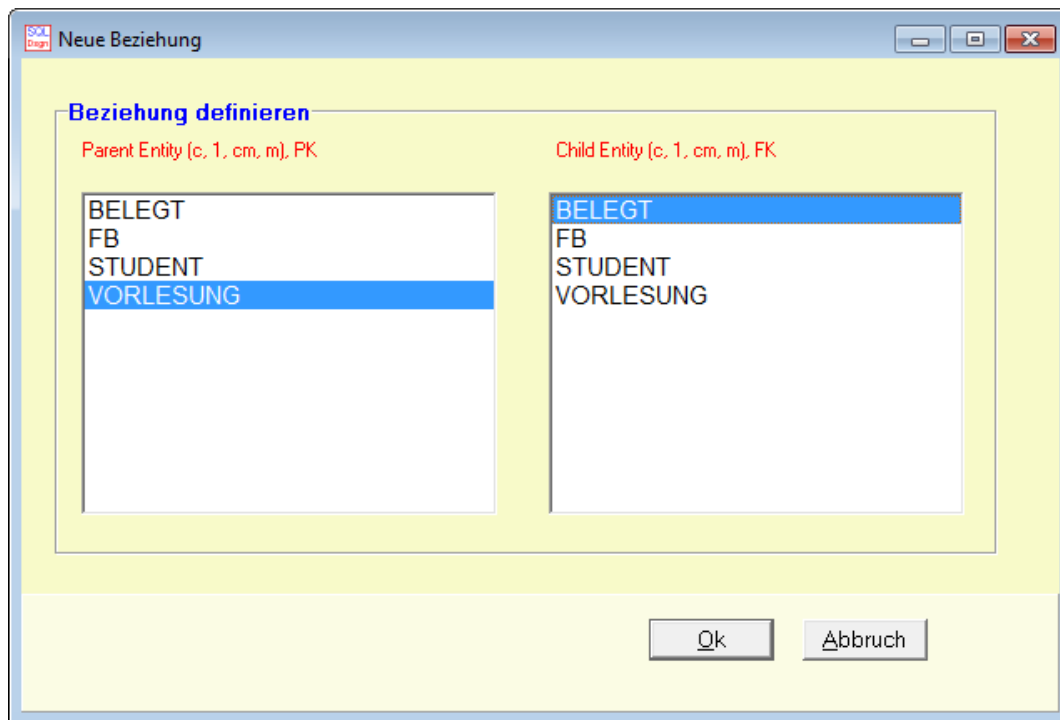


Abbildung 108 Aufruf Beziehung Vorlesung zu Belegt

Eingetragen werden dann die Beschreibungen und die Kardinalitäten.

**Bearbeiten einer Beziehung**

Allgemein | Integrität | Link-Attribute | Weak-Attribute | Self-Relation | Beschreibung | Zu Erledigen

Beziehungs-Name: VORLESUNG\_BELEGT  
 FK Constraint Name: FK\_VORLESUNG\_BELEGT  
 Parent Verb: Eine Vorlesung wird von keinem, einen oder mehreren Studenten besucht  
 Child Verb: Ein Belegt-Eintrag entspricht einer Vorlesung

Kardinalität / Beziehung:  
 Parent: C (0 oder 1) **ZU** Child: CM (0, 1, oder viele)

Beziehungstyp: Identifizierend ☒ Beziehung ist zwingend

VORLESUNG —————> BELEGT

Ok Abbruch

Abbildung 109 Beziehung „Vorlesung“ zu „Belegt“

**Bestätigung**

? Sollen die Primärschlüssel des Entity VORLESUNG in das Child-Entity BELEGT eingetragen werden?

Ja Nein

Abbildung 110 Automatische Übertragung des Primary-Keys als Foreign-Key

**Fremdschlüsselname**

Relation erstellen

VNR

OK Abbrechen

Abbildung 111 des Foreign-Keys



Bearbeiten einer Beziehung

Allgemein   Integrität   Link-Attribute   Weak-Attribute   Self-Relation   Beschreibung   Zu Erledigen

Verknüpfte Attribute

Disconnect

Nr	Parent-Key	Child-Key	Parent-Ind	Child-Inde	Parent-ID	Child-ID
1	VNR	VNR	0	1	33	113

Connect

VORLESUNG

Nr	PK	FK	Attribut	Datentyp
1	PK		VNR	INTEGER
			NAME	VARCHAR

BELEGT

Nr	PK	FK	Neu	Attribut	Datentyp
		FK		MNR	INTEGER
1		FK	Neu	VNR	INTEGER

Ok   Abbruch

Abbildung 112 Anzeige der Eingaben der Beziehung

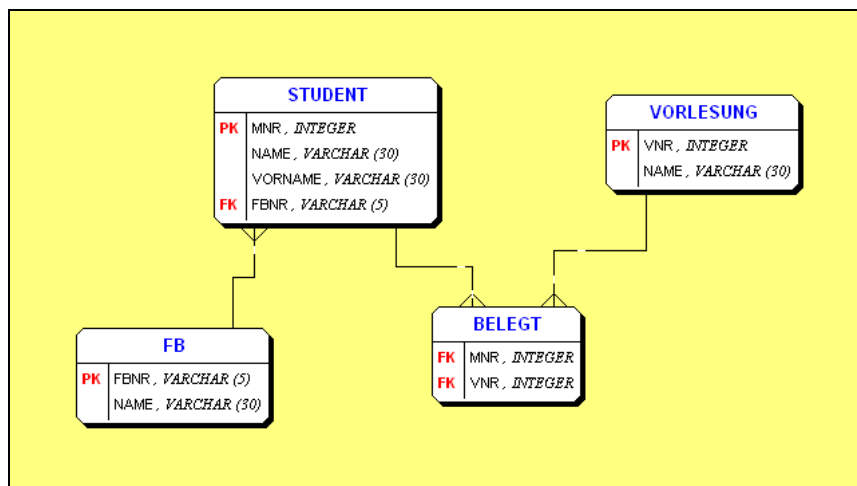
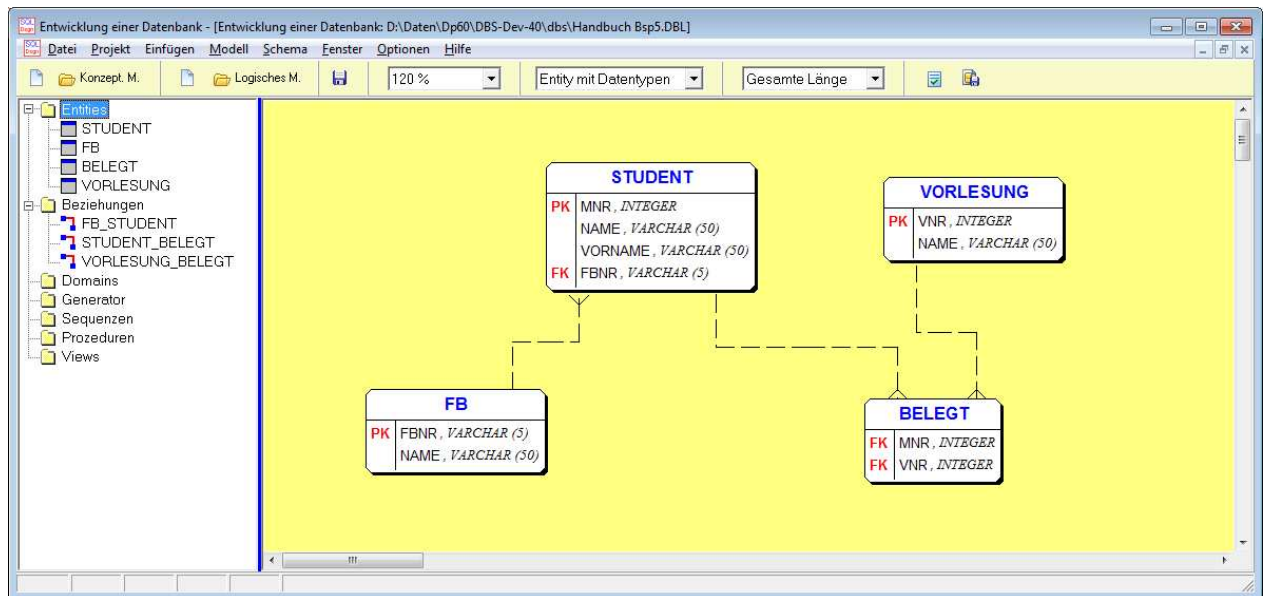


Abbildung 113 Aktuelle Fassung mit drei Beziehungen



**Abbildung 114** ER-Modell nach den Beziehungen der Hobbies

Im zusätzlichen Entity „Belegt“ wurden beide Attribute als Fremdschlüssel eingetragen. Sinnvoll ist es nun, diese auch als Primary-Key zu definieren.

#### 4.5.6 Weitere Schritte

Als weitere Schritte wären Prüfbedingungen möglich. Dazu zählen:

- Check-Constraint (Prüfbedingung des aktuellen Wertes)
- Unique-Constraint (Eindeutigkeit eines normalen Attributs)

#### 4.5.7 Zusätzliche SQL-Befehle

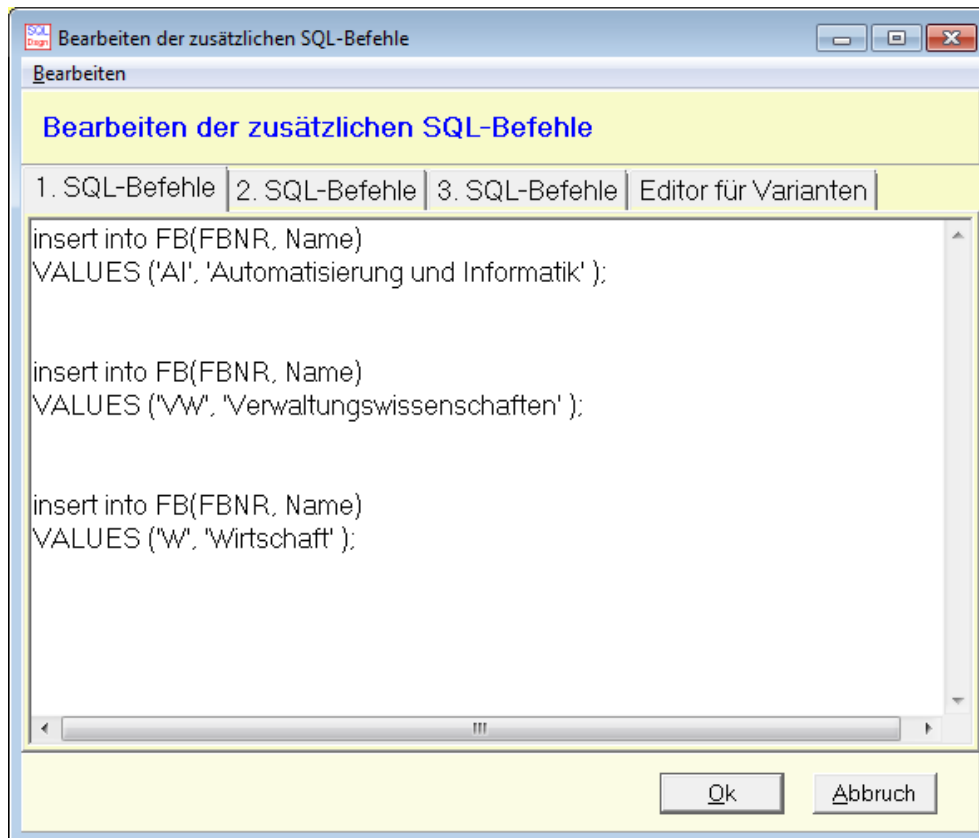
Mit dem Menüs „Modell“ und „zusätzliche SQL-Anweisungen“ können vordefinierte Eintragungen vorgenommen werden.

SQL-Anweisungen:

```
insert into FB(FBNR, Name)
VALUES ('AI', 'Automatisierung und Informatik' );
```

```
insert into FB(FBNR, Name)
VALUES ('VW', 'Verwaltungswissenschaften' );
```

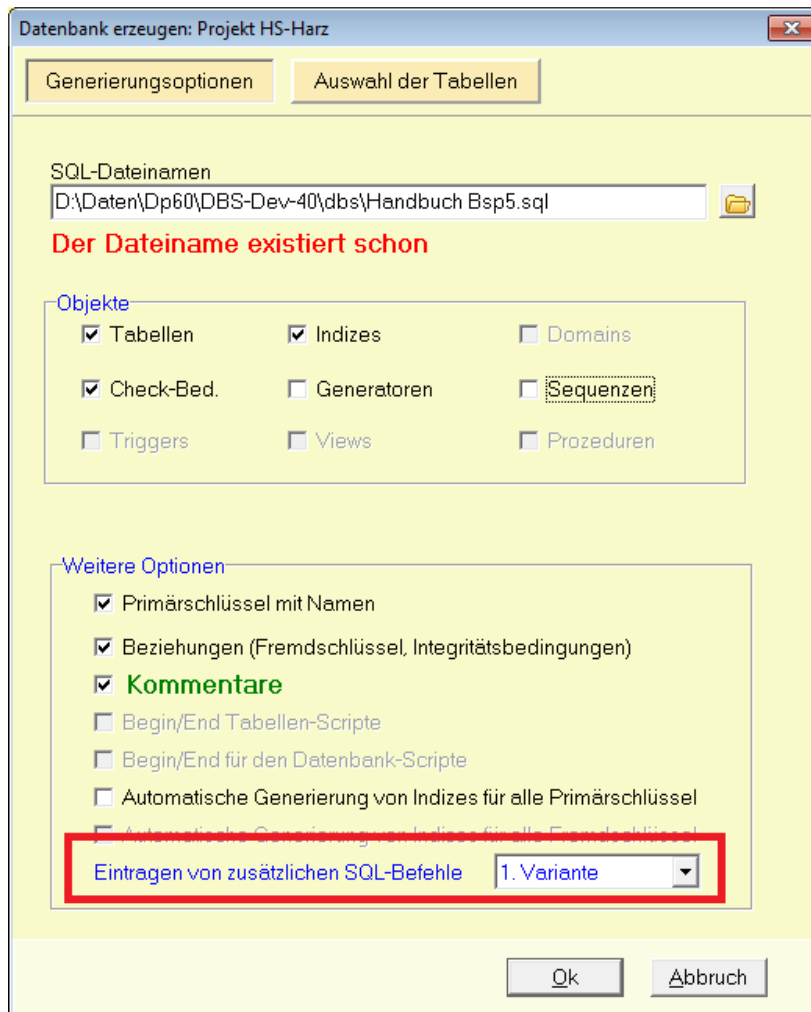
```
insert into FB(FBNR, Name)
VALUES ('W', 'Wirtschaft' );
```



**Abbildung 115 zusätzliche SQL-Anweisungen**

#### 4.5.8 Generierung einer Datenbank

Mit dem Menü „Schema“ und dem Eintrag „Generierung Datenbank“ wird die Erzeugung des SQL-Scriptes angestoßen. Als Kurztaste existiert die Taste F9. Es erscheint folgendes Fenster (siehe Abbildung 116). In der obersten Zeile wird der Dateiname eingetragen. Der nächste Teil definiert die Objekte, die generiert werden sollen. Im letzten Teil sind zusätzliche Optionen vorhanden.



**Abbildung 116** Erzeugung einer Datenbank

Mit dem Schalter „Ok“ wird das Script erzeugt und im Notepad dargestellt.

Ergebnis mit SQL-Befehlen:

```
/*=====*/
/* Script generated with: MW Designer Version 4.74, Build 501 */
/* Project Filename: D:\Daten\Dp60\DBS-Dev-40\dbs\Handbuch Bsp5.sql */
/* Project Name: 5. Handbuch-Beispiel */
/* Author: Michael Wilhelm */
/* DBMS: Firebird-DBF */
/* Copyright: by Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm */
/* Generated on: 09.02.2016 18:36:45 */
/*=====*/

/*=====*/
/* Tables */
/*=====*/

CREATE TABLE BELEGT (
    MNR INTEGER NOT NULL,
    VNR INTEGER NOT NULL
);

CREATE TABLE FB (
    FBNR VARCHAR(5) NOT NULL,
    NAME VARCHAR(50) DEFAULT '',

    CONSTRAINT PK_FB PRIMARY KEY (FBNR)
);

CREATE TABLE STUDENT (
    MNR INTEGER NOT NULL,
    NAME VARCHAR(50) DEFAULT '',
    VORNAME VARCHAR(50) DEFAULT '',
    FBNR VARCHAR(5) NOT NULL,

    CONSTRAINT PK_STUDENT PRIMARY KEY (MNR)
);

CREATE TABLE VORLESUNG (
    VNR INTEGER NOT NULL,
    NAME VARCHAR(50) DEFAULT '',

    CONSTRAINT PK_VORLESUNG PRIMARY KEY (VNR)
);

/*=====*/
/* Foreign Keys */
/*=====*/

ALTER TABLE STUDENT ADD CONSTRAINT
    FK_FB_STUDENT FOREIGN KEY (FBNR) REFERENCES FB(FBNR);

ALTER TABLE BELEGT ADD CONSTRAINT
    FK_STUDENT_BELEGT FOREIGN KEY (MNR) REFERENCES STUDENT(MNR);

ALTER TABLE BELEGT ADD CONSTRAINT
    FK_VORLESUNG_BELEGT FOREIGN KEY (VNR) REFERENCES VORLESUNG(VNR);

/* Zusaetzliche SQL-Befehle*/
```

```
insert into FB(FBNR, Name)
VALUES ('AI', 'Automatisierung und Informatik' );
```

```
insert into FB(FBNR, Name)
VALUES ('VW', 'Verwaltungswissenschaften' );
```

```
insert into FB(FBNR, Name)
VALUES ('W', 'Wirtschaft' );
```

## 5 FBConsole

Dieses Kapitel beschreibt das Erstellen, Füllen und Abfragen einer neuen Datenbank

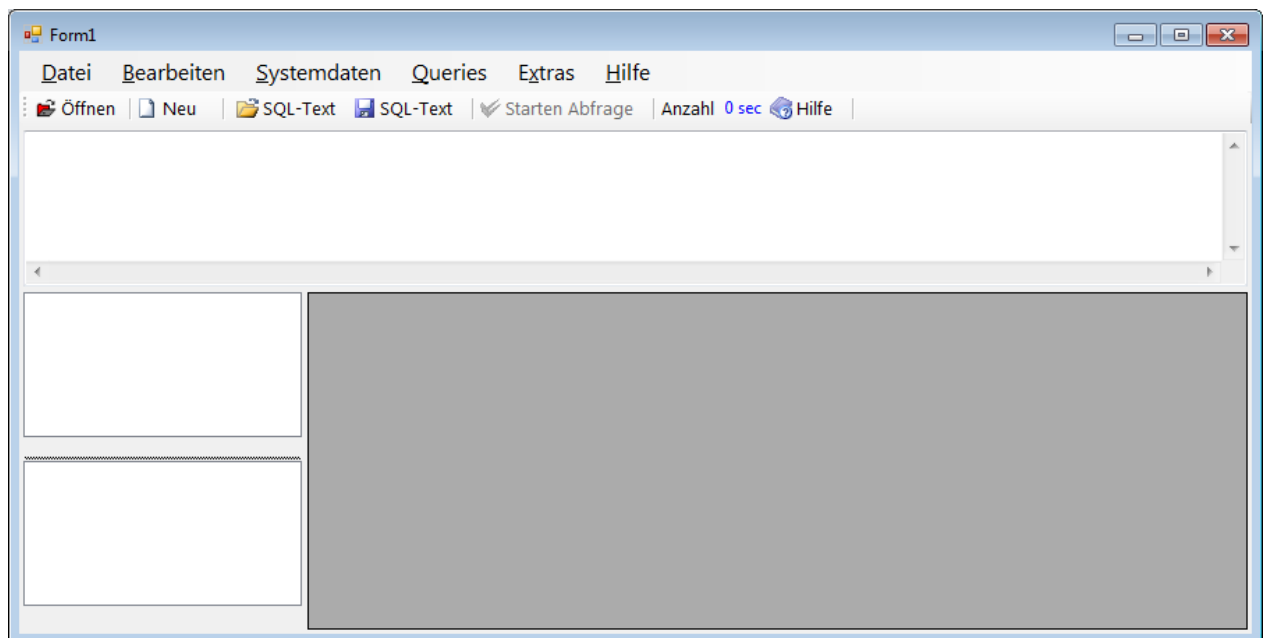


Abbildung 117 Hauptfenster der FBConsole

### Schalter:

- |                   |   |
|-------------------|---|
| • Öffnen          | Öffnen einer vorhandenen Datenbank.                         |
| • Neu             | Erstellen einer neuen Datenbank.                            |
| • SQL-Text        | Öffnen einer vorhandenen Textdatei mit einer SQL-Anweisung. |
| • SQL-Text        | Speichern der aktuellen SQL-Anweisung in eine Textdatei.    |
| • Starten Abfrage | Ausführen der Anweisung im Texteditor                       |

### Menü Datei:

- |                               |   |
|-------------------------------|---|
| • Öffnen einer Datenbank      | Öffnen einer vorhandenen Datenbank.                         |
| • Neue Datenbank              | Erstellen einer neuen Datenbank.                            |
| • Öffnen einer SQL-Datei      | Öffnen einer vorhandenen Textdatei mit einer SQL-Anweisung. |
| • Speichern eines SQL-Befehls | Speichern der aktuellen SQL-Anweisung in eine Textdatei.    |

**Menü Queries:**

- Ausführen der Abfrage      Ausführen der Anweisung im Texteditor.
- Export der Ergebnisse      Speichern des Ergebnisses der Abfrage in die Zwischenablage.
- Speichern der Ergebnisse      Speichern des Ergebnisses der Abfrage in eine Textdatei.
-

## 5.1 Neue Datenbank erstellen

### Ablauf:

- Erstellen der Skripte mittels eines Datenbank-Designers
- Erstellen einer neuen Datenbank: STRG+N

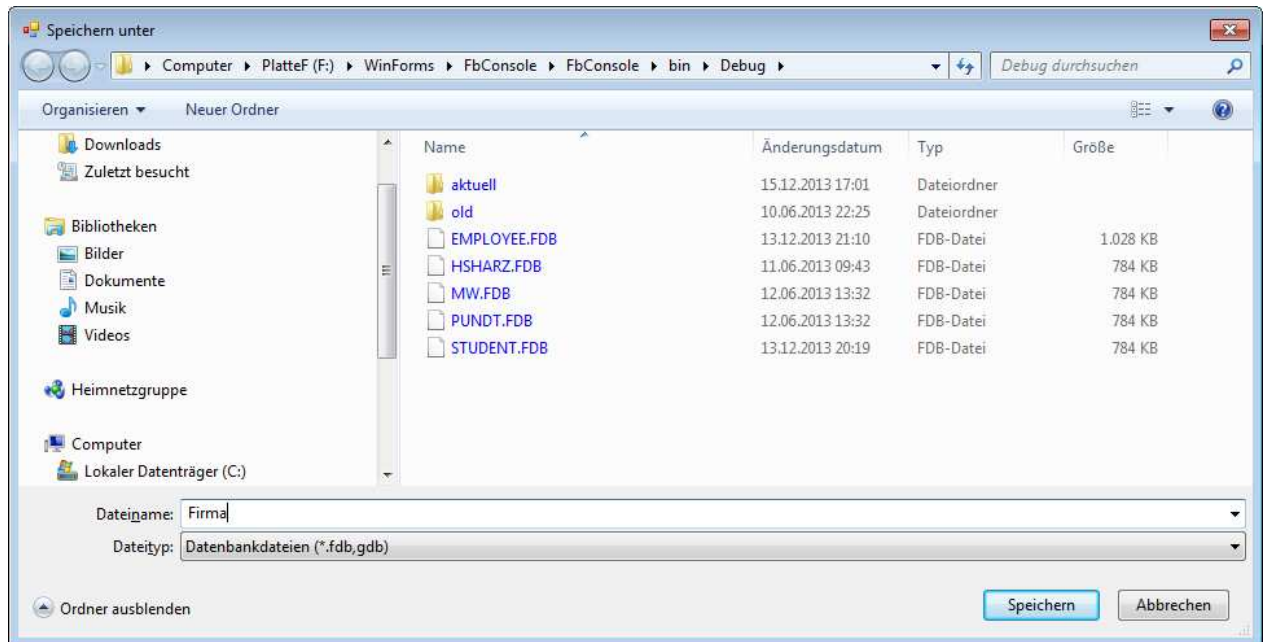
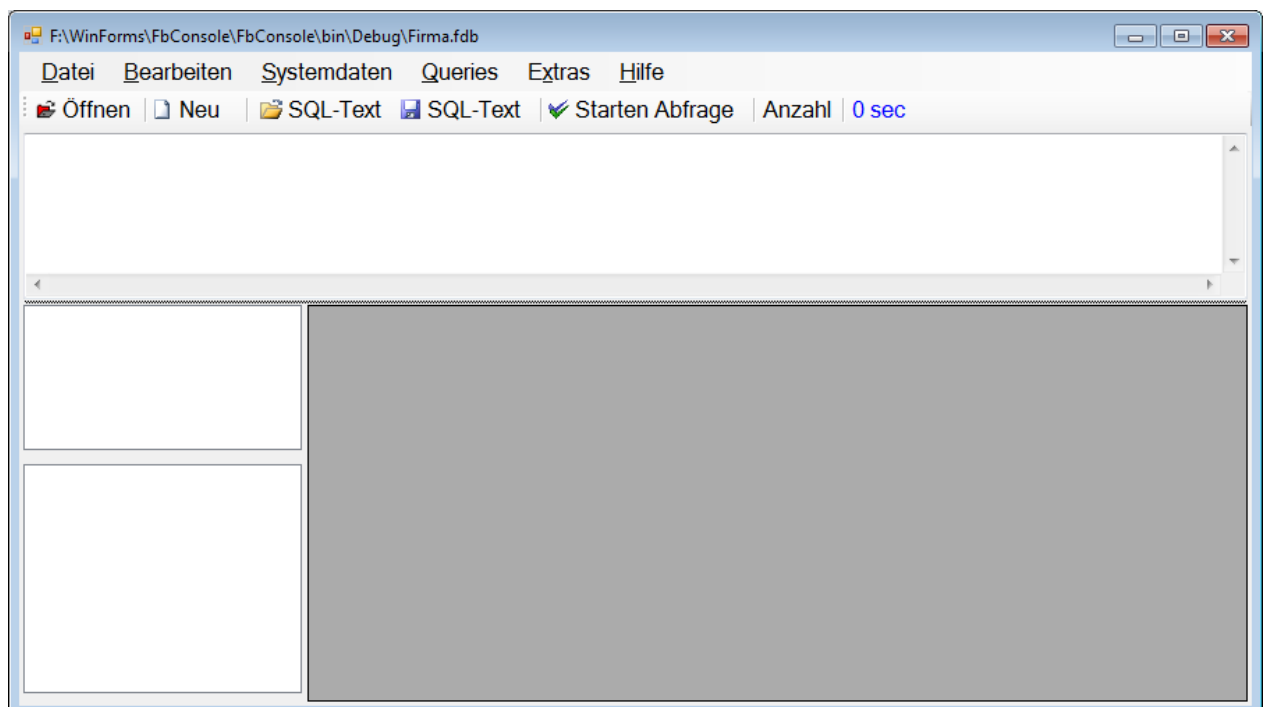


Abbildung 118 Erstellen der Datenbank "Firma.fdb"

### Ergebnis:



Einfügen der Skripte in den Editor:



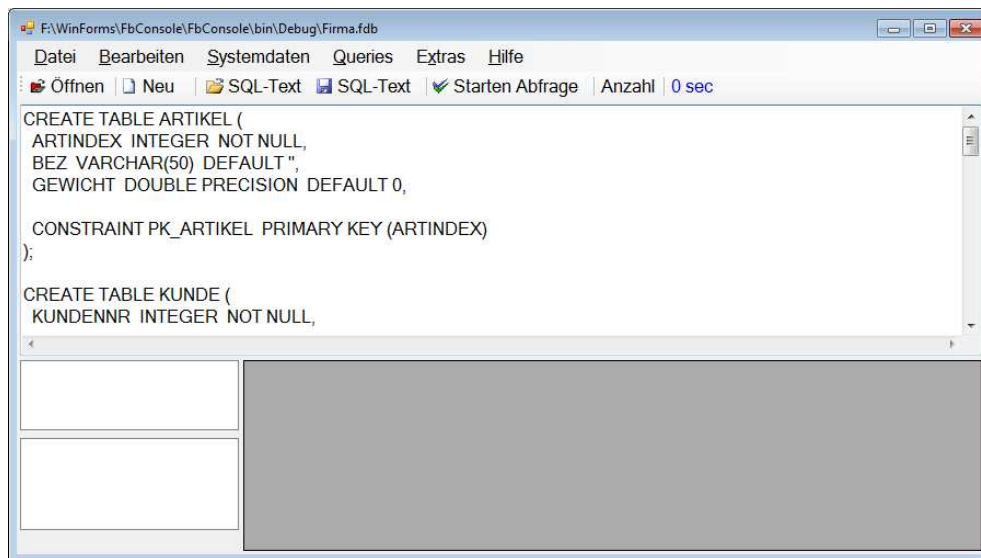


Abbildung 119 Eintragen der SQL-Skripte

Schalter „Starten Abfrage“

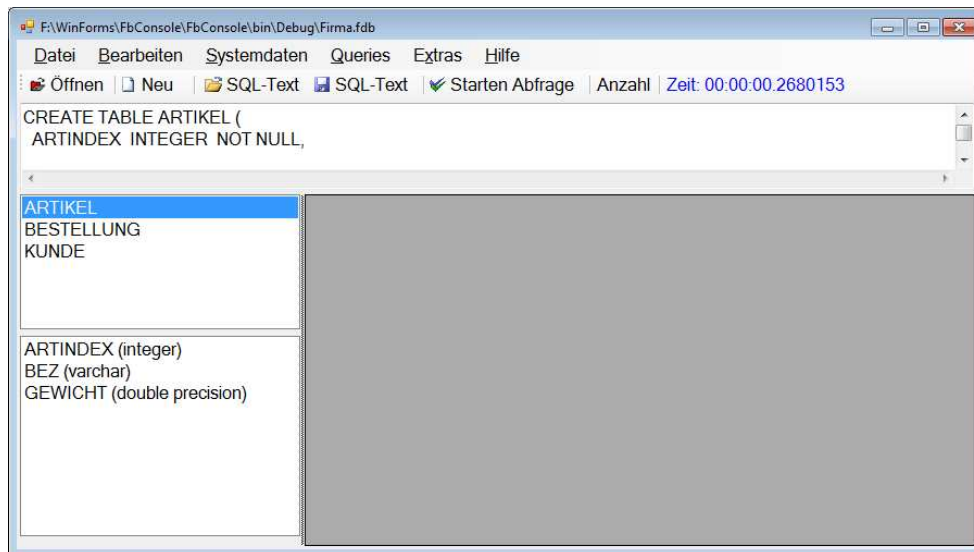


Abbildung 120 Anzeige der Datenbank-Tabellen

In der oberen Liste sind alle Entities eingetragen.

In der unteren Liste sind alle Spalten des aktuellen Entities eingetragen.

## 5.2 Daten eintragen

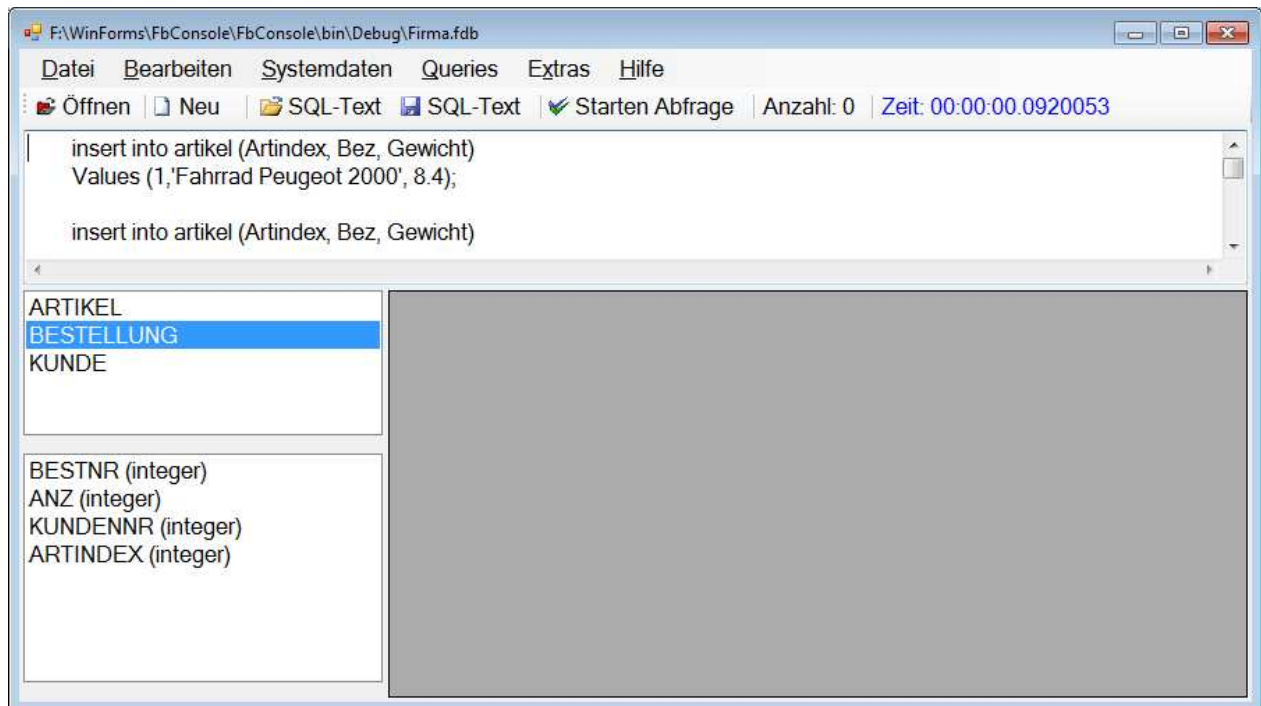


Abbildung 121 Daten mittels „Insert-Into“ eintragen

### 5.2.1 SQL-Befehle zum Eintragen:

```
insert into kunde (kundenr, name, vorname, strasse, plz,
ort)
Values (101,'Müller', 'Hans', 'Langer Weg 6', '01256',
'Weißenfels');
```

```
insert into kunde (kundenr, name, vorname, strasse, plz,
ort)
Values (102,'Schmidt', 'Andrea', 'Kastanienallee 42',
'39114', 'Magdeburg');
```

```
insert into kunde (kundenr, name, vorname, strasse, plz,
ort)
Values (103,'Schmidt', 'Andrea', 'Kastanienallee 42',
'39114', 'Magdeburg');
```

```
insert into kunde (kundenr, name, vorname, strasse, plz,
ort)
Values (104,'Meier', 'Antonia', 'Friedrichstraße 58',
'38855', 'Wernigerode');
```

```
insert into artikel (Artindex, Bez, Gewicht)
Values (1,'Fahrrad Peugeot 2000', 8.4);
```

```
insert into artikel (Artindex, Bez, Gewicht)
Values (2,'Tablet Omega', 0.6);
```

```
insert into artikel (Artindex, Bez, Gewicht)
Values (3,'Desktop HP 1200', 7.5);
```

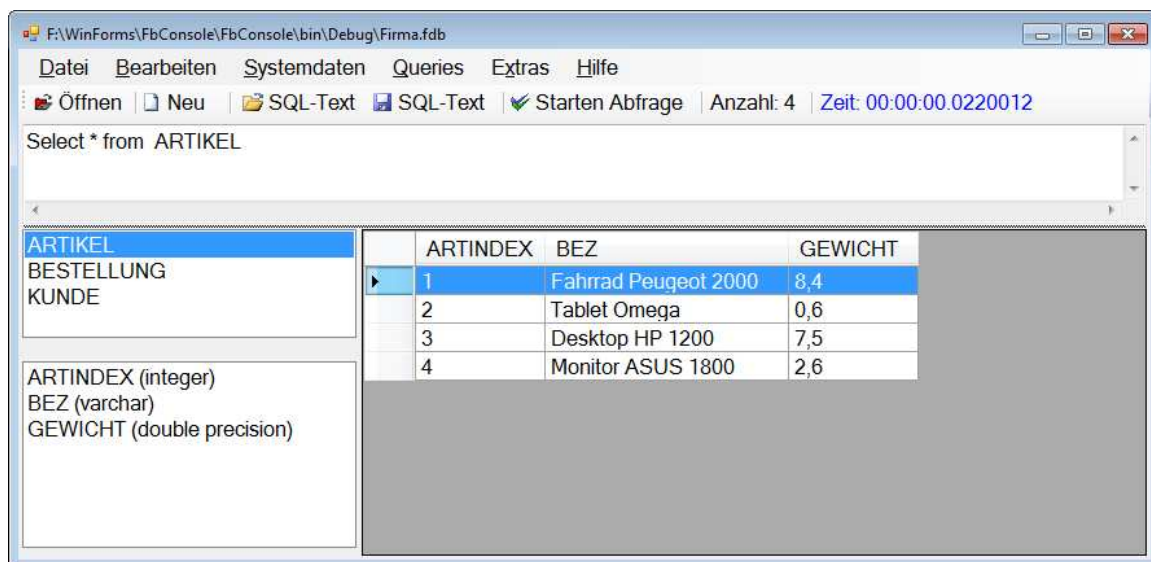
```
insert into artikel (Artindex, Bez, Gewicht)
Values (4,'Monitor ASUS 1800', 2.6);
```

```
insert into bestellung (bestnr, anz, kundenr, artindex)
Values (201, 2, 101, 1);
```

```
insert into bestellung (bestnr, anz, kundenr, artindex)
Values (202, 3, 104, 2);
```

```
insert into bestellung (bestnr, anz, kundenr, artindex)
Values (203, 1, 102, 4);
```

### 5.3 Testabfragen



The screenshot shows a database application window titled 'F:\WinForms\FbConsole\FbConsole\bin\Debug\Firma.fdb'. The menu bar includes 'Datei', 'Bearbeiten', 'Systemdaten', 'Queries', 'Extras', and 'Hilfe'. The toolbar contains icons for 'Öffnen', 'Neu', 'SQL-Text', 'SQL-Text', and 'Starten Abfrage', along with 'Anzahl: 4' and 'Zeit: 00:00:00.0220012'. The main query area displays 'Select \* from ARTIKEL'. On the left, a tree view shows the database structure with 'ARTIKEL' selected. Below the tree, the field definitions for 'ARTIKEL' are listed: 'ARTINDEX (integer)', 'BEZ (varchar)', and 'GEWICHT (double precision)'. The main display area shows a table with the following data:

	ARTINDEX	BEZ	GEWICHT
▶	1	Fahrrad Peugeot 2000	8,4
	2	Tablet Omega	0,6
	3	Desktop HP 1200	7,5
	4	Monitor ASUS 1800	2,6

Abbildung 122 Abfrage bezüglich der Artikel

The screenshot shows the FbConsole application with the query "Select \* from KUNDE" entered. The results table displays customer information.

KUNDENNR	NAME	VORNAME	STRASSE	PLZ	ORT
101	Müller	Hans	Langer We...	01256	Weißenfel...
102	Schmidt	Andrea	Kastanien...	39114	Magdebur...
103	Schmidt	Andrea	Kastanien...	39114	Magdebur...
104	Meier	Antonia	Friedrichst...	38855	Wernigero...

Left sidebar (Database Structure):

- ARTIKEL
- BESTELLUNG
- KUNDE**

Field details for KUNDE:

- KUNDENNR (integer)
- NAME (char)
- VORNAME (char)
- STRASSE (char)
- PLZ (char)
- ORT (char)

Abbildung 123 Abfrage bezüglich der Kunden

The screenshot shows the FbConsole application with the query "Select \* from BESTELLUNG" entered. The results table displays order information.

KUNDENNR	BESTNR	ANZ	ARTINDEX
101	201	2	1
104	202	3	2
102	203	1	4

Left sidebar (Database Structure):

- ARTIKEL
- BESTELLUNG**
- KUNDE

Field details for BESTELLUNG:

- BESTNR (integer)
- ANZ (integer)
- KUNDENNR (integer)
- ARTINDEX (integer)

Abbildung 124 Abfrage bezüglich der Bestellungen

The screenshot shows the FbConsole application with a complex query joining three tables. The results table displays a combined view of order, customer, and product information.

KUNDENNR	BESTNR	ANZ	ARTINDEX	NAME	VORNAME	BEZ
101	201	2	1	Müller	Hans	Fahrrad Peugeot 2...
104	202	3	2	Meier	Antonia	Tablet Omega
102	203	1	4	Schmidt	Andrea	Monitor ASUS 1800

Left sidebar (Database Structure):

- ARTIKEL
- BESTELLUNG
- KUNDE

Field details for ARTIKEL:

- ARTINDEX (integer)
- BEZ (varchar)
- GEWICHT (double p)

Abbildung 125 Abfrage bezüglich der Bestellungen

```

Select BESTELLUNG.* , kunde.NAME, kunde.VorNAME,  ARTIKEL.BEZ
from      BESTELLUNG, KUNDE,ARTIKEL

```

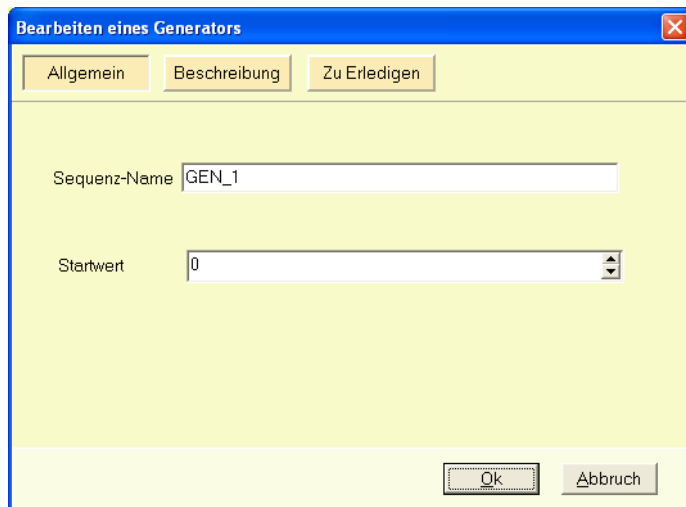
where (artikel.artindex=BESTELLUNG.artindex) and  
(kunde.KUNDENNR=bestellung.KUNDENNR)

## 6 Anhang

### 6.1 Generator

Generator definieren:

Um einen Generatoren zu erzeugen, ruft man das Menü „Einfügen“ mit dem Eintrag „Generator“ auf.



**Abbildung 126** Definition eines Generators

In Firebird bzw. Interbase kann man neben dem Namen nur den Startwert definieren. Der erste richtige Wert ist dann um eins erhöht.

Manuelles Skript:

```
CREATE GENERATOR GEN_KUNDE ;  
SET GENERATOR GEN_KUNDE TO 0 ;
```

### 6.2 Sequenz

```
CREATE SEQUENCE <name>;  
ALTER SEQUENCE <name> RESTART WITH <value>;  
DROP SEQUENCE <name>;
```

```
SELECT NEXT VALUE FOR <SequenceName> FROM RDB$DATABASE;
```

```
<intvar> = NEXT VALUE FOR <SequenceName>;
```

### 6.3 Daten-Eintrag

Möchte man größere Datenmengen per Programmcode eintragen, so wird pro Datensatz eine Verbindung erstellen. Dies beeinträchtigt massiv die Performanz. Als Abhilfe kann man mehrere SQL-Anweisungen blocken.

Anfangswert: `execute block as begin`

Endwert: `end`

**Beispiel:**

```
execute block as begin
    insert into student (Matrnr, Nachname, Vorname, Findex)
    Values (12345, 'Müller', 'Andreas', 1);

    insert into student (Matrnr, Nachname, Vorname, Findex)
    Values (12346, 'Schmidt', 'Petra', 2);

    insert into student (Matrnr, Nachname, Vorname, Findex)
    Values (12347, 'Meier', 'Ralf', 3);

    insert into student (Matrnr, Nachname, Vorname, Findex)
    Values (12348, 'Brandt', 'Ute', 1);
end
```

Sinnvoll sind fünfzig bis hunderter Blöcke.

## 7 SQL-Beispiele

### 7.1 Daten für die Tabelle „Vorlesung“

Daten:

```
INSERT INTO Vorlesung (vnr, name)
VALUES (1,'GDI1');
```

```
INSERT INTO Vorlesung (vnr, name)
VALUES (2,'GDI2');
```

```
INSERT INTO Vorlesung (vnr, name)
VALUES (3,'GDI3');
```

```
INSERT INTO Vorlesung (vnr, name)
VALUES (4,'BS');
```

```
INSERT INTO Vorlesung (vnr, name)
VALUES (15,'GUI');
```

```
INSERT INTO Vorlesung (vnr, name)
VALUES (21,'MFC');
```

```
INSERT INTO Vorlesung (vnr, name)
VALUES (7,'DBS');
```

### 7.2 Daten für die Tabelle „Student“

Hier wird als Fremdschlüssel die Fachbereichsnummer mit übergeben.

Daten:

```
INSERT INTO Student (mnr, name, vorname, fbnr)
VALUES (10001,'Müller', 'Hans', 'AI');
```

```
INSERT INTO Student (mnr, name, vorname, fbnr)
VALUES (10002,'Meyer', 'Ute', 'AI');
```

```
INSERT INTO Student (mnr, name, vorname, fbnr)
VALUES (10003,'Schulze', 'Robert', 'AI');
```

```
INSERT INTO Student (mnr, name, vorname, fbnr)
VALUES (10005,'Robertson', 'Bill', 'VW');
```

```
INSERT INTO Student (mnr, name, vorname, fbnr)
VALUES (10007,'Anderson', 'Andrea', 'VW');
```



```
INSERT INTO Student (mnr, name, vorname, fbnr)
VALUES (10015,'Steiner', 'Ralf', 'W');
```

```
INSERT INTO Student (mnr, name, vorname, fbnr)
VALUES (10017,'Steffenson', 'Gerd', 'W');
```

### **7.3 Daten für die Tabelle „Belegt“**

Daten:

```
INSERT INTO belegt (mnr, vnr)
VALUES (10001, 1);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10003, 1);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10001, 2);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10001, 4);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10005, 15);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10003, 2);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10007, 7);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10007, 2);
```

```
INSERT INTO belegt (mnr, vnr)
VALUES (10007, 15);
```

### **7.4 Update-Befehle**

Einfacher Update-Befehl:

```
UPDATE Student
SET Name='Hochstetter'
WHERE ( MNR = 10003);
```

Ändert des Namens eines Studenten mit der Matrikelnummer 10003

Kompexere Update-Befehle:**UPDATE Student**

```
SET Name='Hochstetter', FBNR='W'  
WHERE ( MNR = 10003);
```

Ändern des Namens eines Studenten mit der Matrikelnummer 10003 und der Wechsel zum Fachbereich W<sup>1</sup>.

**UPDATE Student**

```
SET Name='Kiesewetter', FBNR='VW'
```

Dieser Befehl setzt alle Namen auf „Kiesewetter“ und löscht zwei Fachbereiche !

## 7.4.1.1 Delete-Befehle

```
DELETE FROM Student  
WHERE MNR=10005;
```

Exmatrikulation des Studenten 10005

```
DELETE FROM Student
```

Löschen der gesamten Tabelle.

Hinweis:

Das Löschen geht nur, wenn keine weitere Beziehung vorhanden ist.

---

<sup>1</sup> Aus Sicht des Fachbereichs AI natürlich nicht zu unterstützen

## 8 Beispiele

### 8.1 *Beispiel6, Entwicklung eines logischen ER-Modell*

Dieses Beispiel zeigt die Entwicklung eines komplexen logischen Modells. Die Vorgaben werden vorab aufgelistet, so dass man übungshalber dieses Modell auch ohne diese Vorlage entwickeln kann.

#### 8.1.1 Aufgabenstellung des sechsten Beispiels

Entwickelt werden soll ein Modell zur Modellierung einer Firma. Folgende Entities wurden definiert:

- Emp
- Dept
- Kid
- Hobbies

#### Teilaufgaben:

- Definieren der Attribute
- Definieren der Beziehungen (es sollten drei unterschiedliche Arten sein)
- Einbau von Prüfbedingungen
- Generieren des SQL-Skriptes

#### 8.1.2 Neues Projekt

Mit der Taste „STRG+N“ wird ein neues Projekt angelegt. Es erscheint ein Dialogfenster, in dem man die Zieldatenbank auswählt (Abbildung 127).

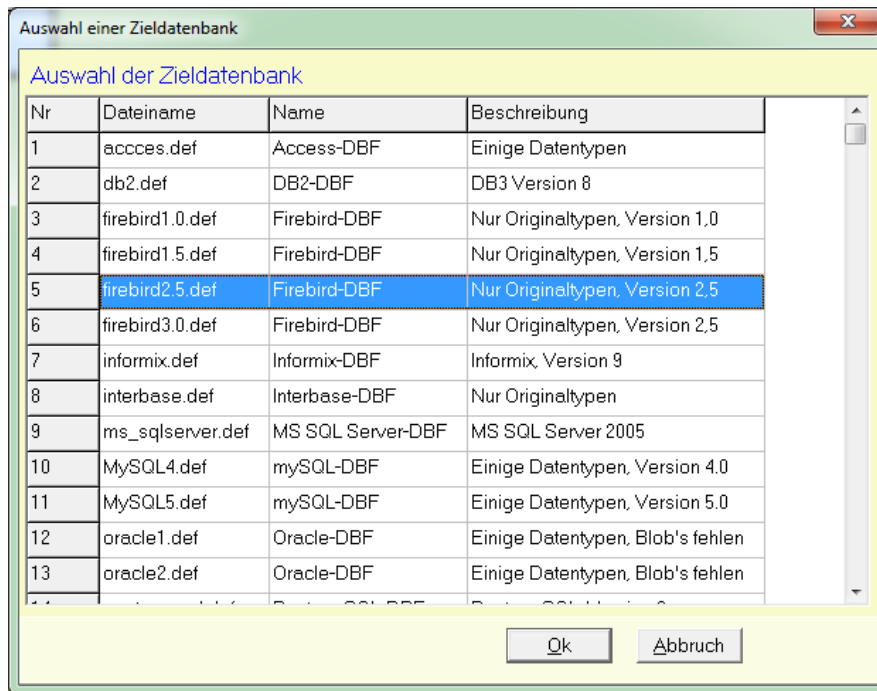


Abbildung 127 Auswahl der Datenbank

Im nächsten Schritt ruft man im Menü „Projekt“ den Eintrag „Eigenschaften“ auf. Im Dialogfenster kann man die notwendigen Einträge vornehmen. Besonders erwähnenswert ist die Auswahl der möglichen Darstellungen.

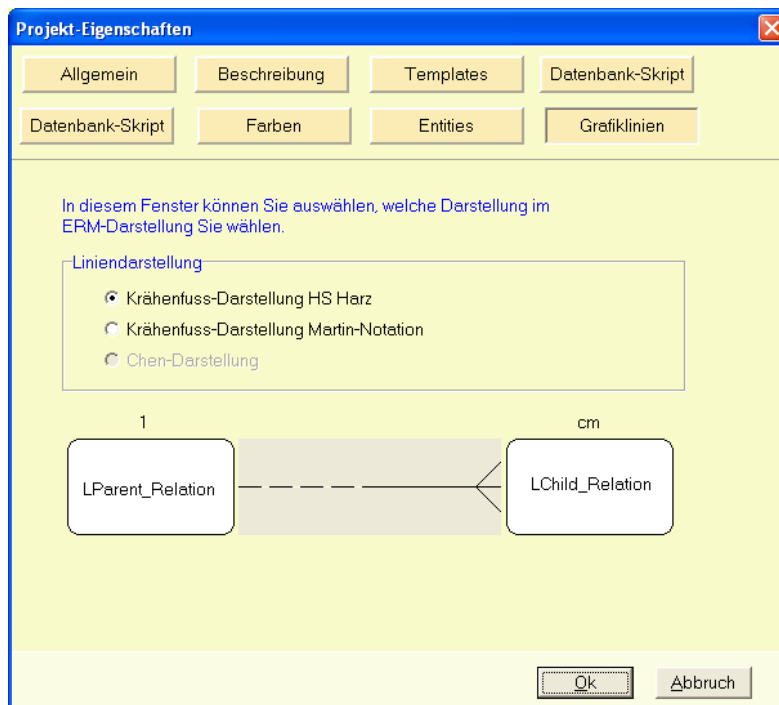


Abbildung 128 Auswahl der Darstellungen

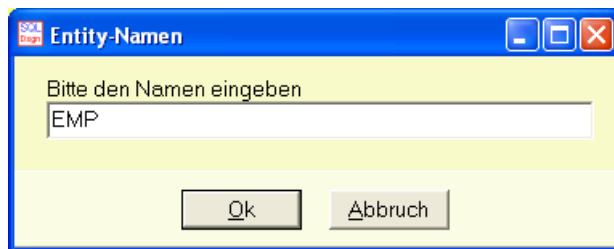
### 8.1.3 Entities anlegen

Mit dem Tastendruck „STRG+E“ werden die jeweiligen Entities angelegt.

Hinweis:

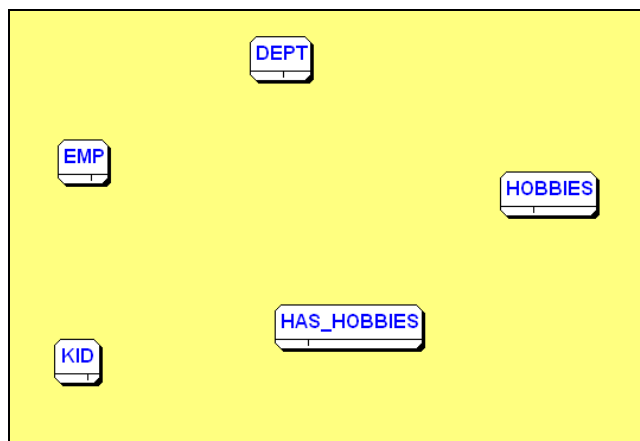
Dabei sollten die Fremdschlüssel mit anderen Entities noch nicht eingefügt werden!!!

Zur erst wird nach dem Namen gefragt, dieser darf nur aus Buchstaben, Zahlen und dem Underline („\_“) bestehen. Es darf kein Leerzeichen verwendet werden. Am Anfang muss auch ein Buchstabe stehen.



**Abbildung 129 Erzeugen eines Entities**

Nach der Eingabe des Names, erscheint das Entity in der Grafik. Nun werden alle weiteren Entitäten eingetragen (siehe Abbildung 130).



**Abbildung 130 Entites des sechsten Beispiels**

Mit einem Doppelklick werden nun die einzelnen Entitäten bearbeiten und die Attribute eingetragen (Abbildung 131). Hier sollten die Einträge ausgefüllt werden. Des Weiteren ist es ratsam, die nähere Beschreibung des Entities im Register „Beschreibung“ vorzunehmen. Im zweiten Register, Abbildung 132, werden dann die Attribute eingetragen.

**Einstellungen eines Entities: EMP**

Allgemein | Attribute | Indizes | Triggers | Constraints

Tabellen-Skripte | Beschreibungen | zu Erledigen | Farben | Position

Name: EMP

Kurzname: emp

PK Constraint Name:

Weak-Entity: ☐ Entity ist ein Weak-Entity

Tabellen-Optionen  
Beschreibungen:

Ok Abbruch

Abbildung 131 Haupt-Eigenschaften eines Entities

**Einstellungen eines Entities: EMP**

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

Add Edit Delete Copy

Ok Abbruch

Abbildung 132 Bearbeiten, Erzeugen, Löschen von Attributen

Das zweite Register verwaltet die Attribute. Mit den unteren Schaltern kann man diese bearbeiten, löschen und erzeugen. Die Schalter rechts an der Seite verschieben die Reihenfolge.

Am Schluss könnte das Projekt dann so aussehen:

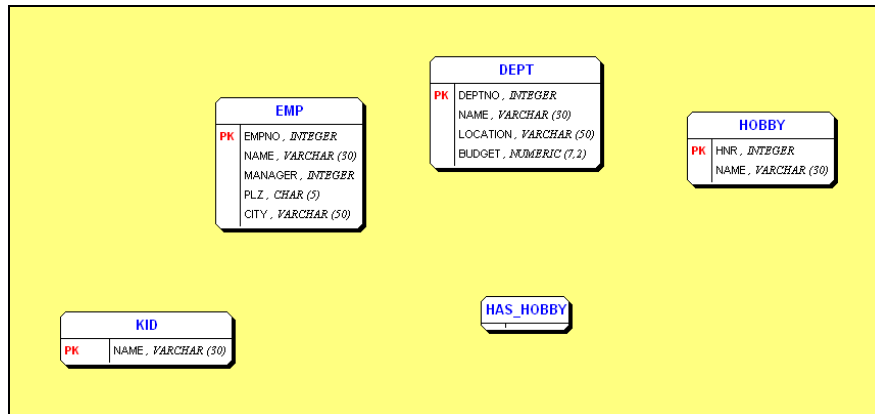


Abbildung 133 Fertige Entities des sechsten Beispiels

Bemerkungen:

- Die Beziehung „Emp“ zu „Dept“ ist eine normale 1:cm-Beziehung.
- Die Beziehung „Emp“ zu „Hobby“ ist eine m:m-Beziehung, deshalb musste ein drittes Entity „Has\_Hobby“ eingeführt werden.
- Das Entity „Kid“ ist ein schwaches Entity. Die Namen können nicht einzeln als Primärschlüssel dienen. Es sei denn, man führt eine Nummer ein oder beschränkt die Auswahl auf einheitliche Kindernamen. Dann darf man aber einige Bewerber nicht als Mitarbeiter einstellen. Die hier vorgestellte Lösung verwendet die Weak-Beziehung.
- Des Weiteren müssen die Attribute „Empno“, „Deptno“ und „HNR“ automatisch hochgezählt werden. Dies wird exemplarisch am Beispiel der „Empno“ vorgeführt.
- Mitarbeiter haben auch Vorgesetzte, so dass im Entity ein Attribut Manager existiert. Dies ist eine 1:cm-Beziehung mit dem Attribut „EmpNo“.

Im nächsten Abschnitt werden die Beziehungen eingetragen.

## 8.1.4 Beziehungen anlegen

Dazu ruft man aus dem Hauptmenü „Einfügen“ die Funktion „Beziehung“ auf. Es erscheint folgendes Fenster:

In beiden Listen sind alle Entities eingetragen. In der linken Liste wählt man den Parent-Entity aus. Das wäre hier in diesem Beispiel „Dept“. In der rechten Liste wird die Beziehung eingetragen. Hier also „Emp“.

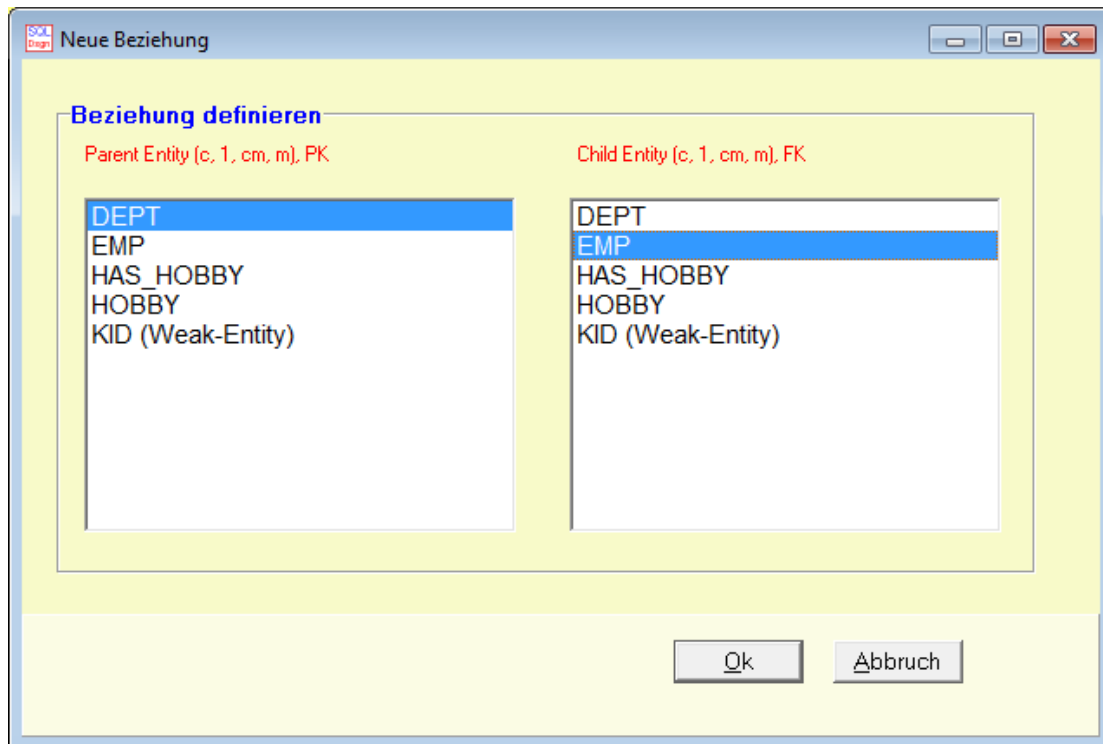


Abbildung 134 Korrekte Einstellung für das Beispiel

Nun muss die korrekte Beziehung eingetragen werden. Danach wird der Primarykey „Deptno“ automatisch als Fremdschlüssel in das Entity „Emp“ eingefügt. Die Beziehungsverben sollten vervollständigt werden.

Die Abbildung 135 zeigt die endgültigen Eingaben. Mit dem Schalter „Ok“ wird die Beziehung eingefügt. Danach erscheint das eigentliche Fenster zur Definition bzw. zur Bearbeitung einer Beziehung.

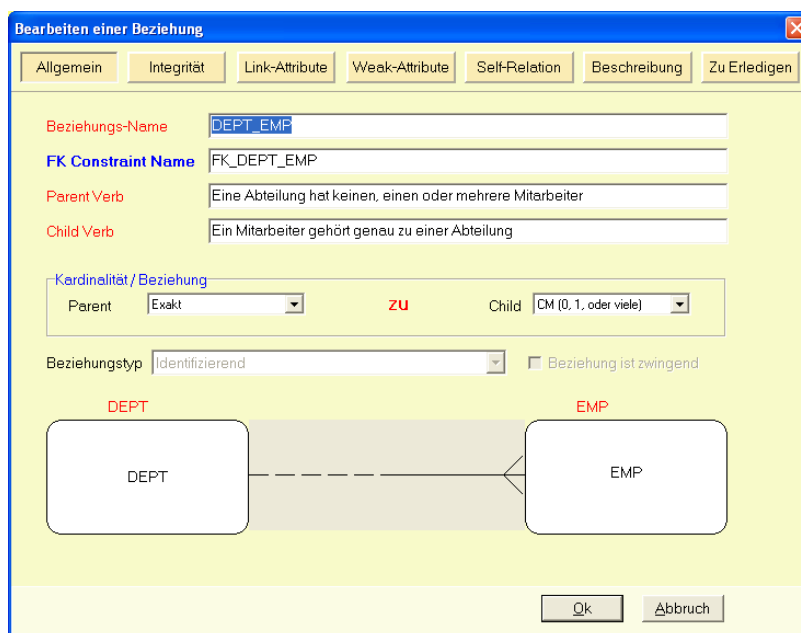


Abbildung 135 Definition der Beziehung

Beziehungstexte:



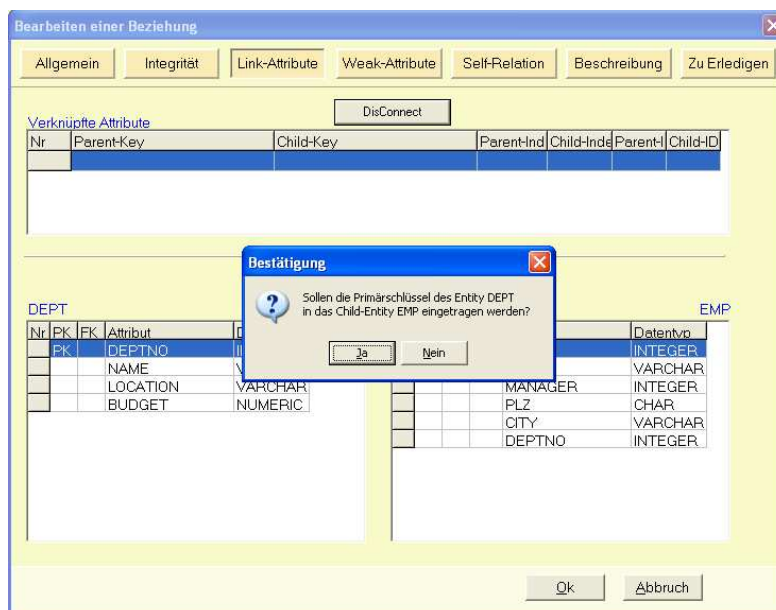
Eine Abteilung hat keinen, einen oder mehrere Mitarbeiter  
 Ein Mitarbeiter gehört genau zu einer Abteilung

Im ersten Register werden die Grundeinstellungen (Name, Beziehungstyp) eingetragen. Wichtig sind hier die Einträge „Parent-Verb“ und „Child-Verb“. Sie zeigen den Charakter der Beziehung an. Das Parent-Verb beschreibt die Beziehung durch Wörter wie „Kunde hat“ etc.

Das zweite Register wird zurzeit nicht unterstützt.

Im dritten Register werden die Attribute verknüpft, mit denen die Beziehung definiert wird. Wie oben erwähnt, sind keine Fremdattribute im Child-Entity („Emp“) eingetragen. Wechselt man in das dritte Register, wird man einmal gefragt, ob alle Primärattribute an das Child übertragen werden sollen (Abbildung 136).

Sinnvoll ist es, diese Frage zu bejahen. Dann werden alle Primär-Attribute in das Child-Entity eingefügt. Diese Attribute erhalten dann die Eigenschaft „Fremdschlüssel“. Die Abbildung 136 zeigt die fertige Liste. Sie können aber auch die Attribute manuell verknüpfen. Dazu markiert man in der linken und in der rechten Tabelle jeweils das gewünschte Attribut. Mit dem Schalter „Connect“ werden diese in die obere Tabelle eingefügt. Mit dem Schalter „DisConnect“ wird die ausgewählte Verbindung wieder gelöst. Die Datentypen der beiden Attributen müssen natürlich exakt übereinstimmen.



**Abbildung 136 Übertragen der Primär-Attribute**

Die untere Abbildung zeigt die Beziehung, nachdem die Primär-Attribute in das Child-Entity übertragen wurden.

**Bearbeiten einer Beziehung**

**Verknüpfte Attribute**

Nr	Parent-Key	Child-Key	Parent-Ind	Child-Inde	Parent-I	Child-ID
1	DEPTNO	DEPTNO	0	5	34	324

**DEPT**

Nr	PK	FK	Attribut	Datentyp
1	PK		DEPTNO	INTEGER
			NAME	VARCHAR
			LOCATION	VARCHAR
			BUDGET	NUMERIC

**EMP**

Nr	PK	FK	Neu	Attribut	Datentyp
	PK			EMPNO	INTEGER
				NAME	VARCHAR
				MANAGER	INTEGER
				PLZ	CHAR
				CITY	VARCHAR
1		FK	Neu	DEPTNO	INTEGER

Abbildung 137 Übertragen eines Fremdschlüssels

Das vierte Register „Weak-Attribute“ dient der Definition einer Beziehung einer Weak-Entity (siehe Kapitel 10.1, Seite 125).

Mit dem Betätigen des Schalters „Ok“ wird die Beziehung eingetragen und angezeigt. Mit der Taste F6 erhält man eine Liste aller Beziehungen.

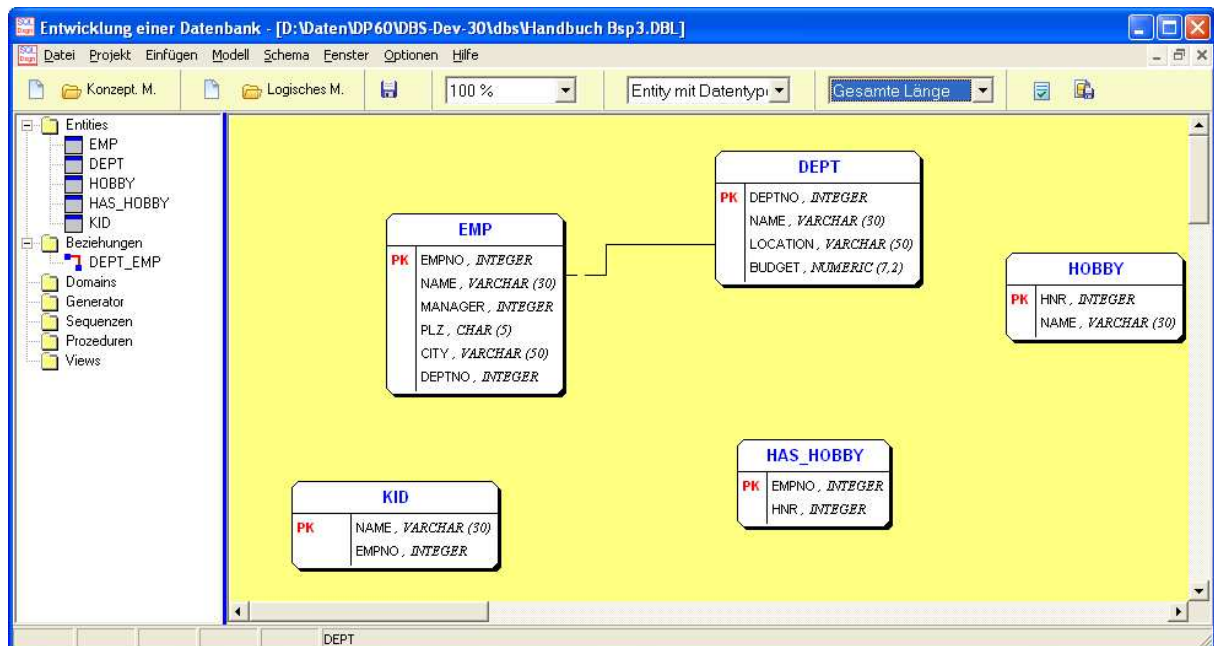


Abbildung 138 Erste Beziehung im sechsten Beispiel

### 8.1.5 Beziehung der Hobbies

Die Beziehung der Hobbies entspricht einer cm:cn-Beziehung. Dazu existiert das neue Entity „Has\_Hobbies“. Es müssen nun zwei Relationen eingetragen werden.

- a) Emp zu Has\_Hobbies
- b) Hobbies zu Has\_Hobbies

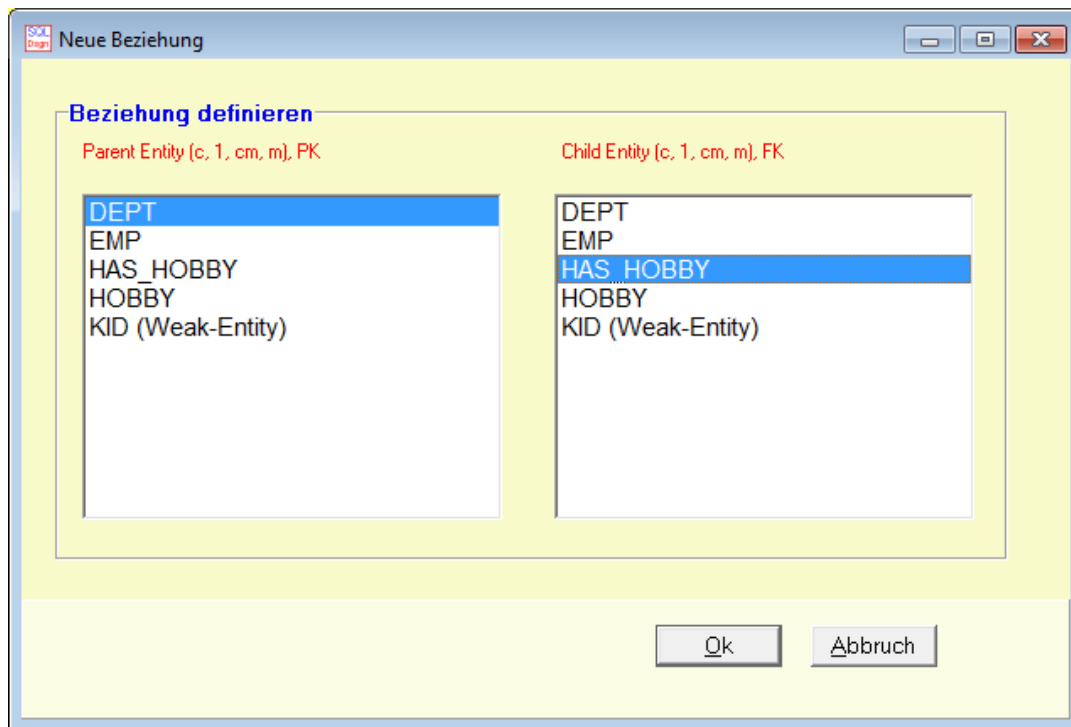


Abbildung 139 Beziehung Emp zu Has\_Hobby

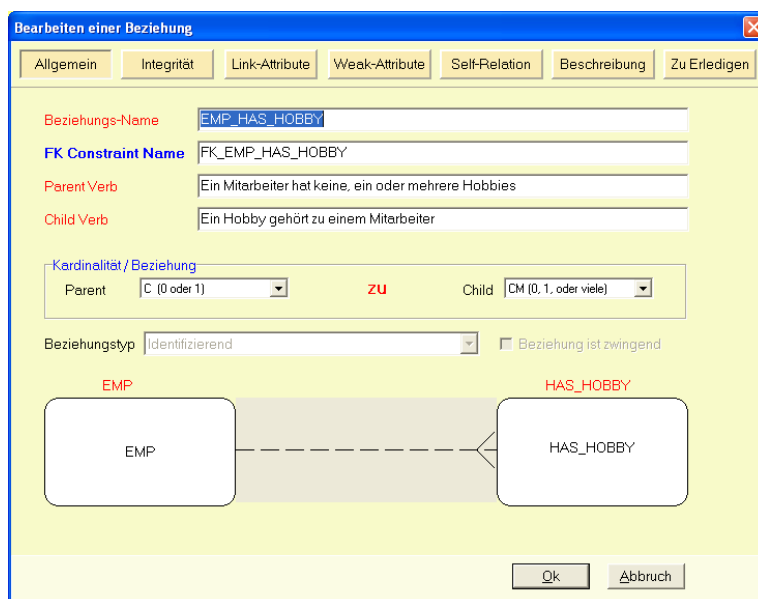


Abbildung 140 Beziehung „Emp“ zu „Has\_Hobby“

Das nächste Dialogfenster zeigt die korrekte Einstellung. Links das Parent-Entity, rechts das Child-Entity.

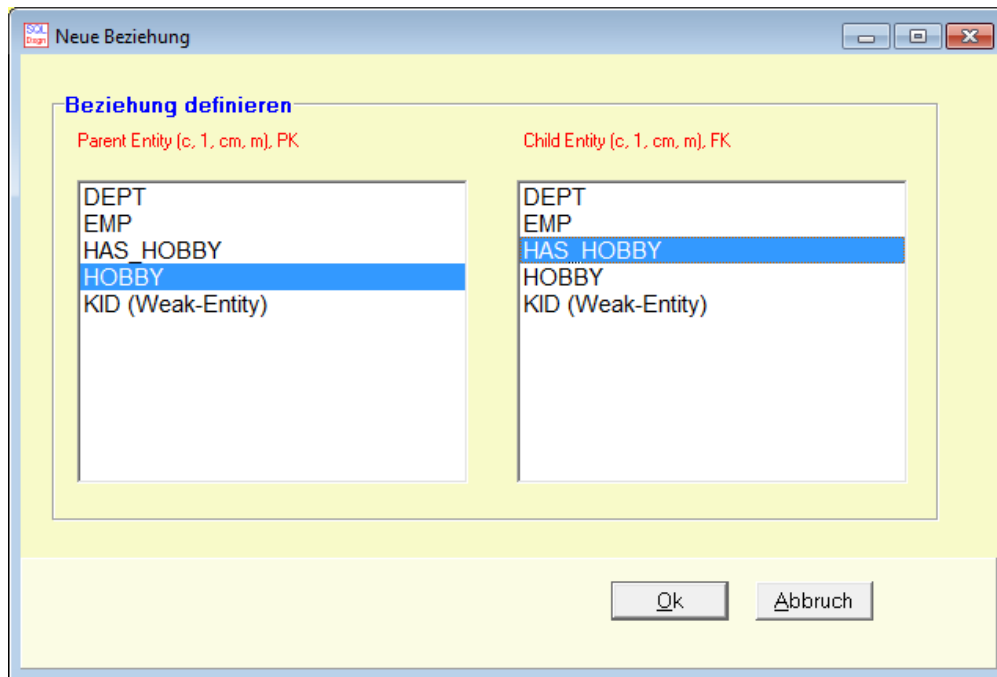


Abbildung 141 Aufruf Beziehung Emp zu Hobbies

Eingetragen werden dann die Beschreibungen und die Kardinalitäten.

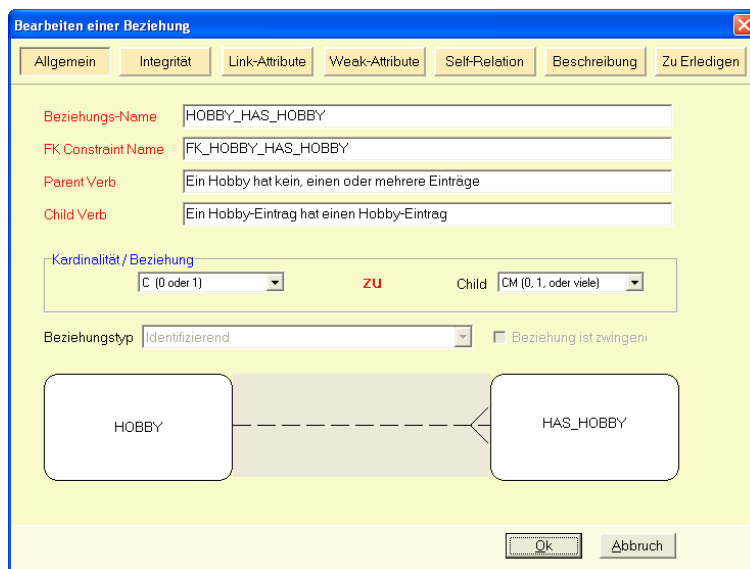


Abbildung 142 Eintrag der Beziehung "Emp" zu "Hobbies"

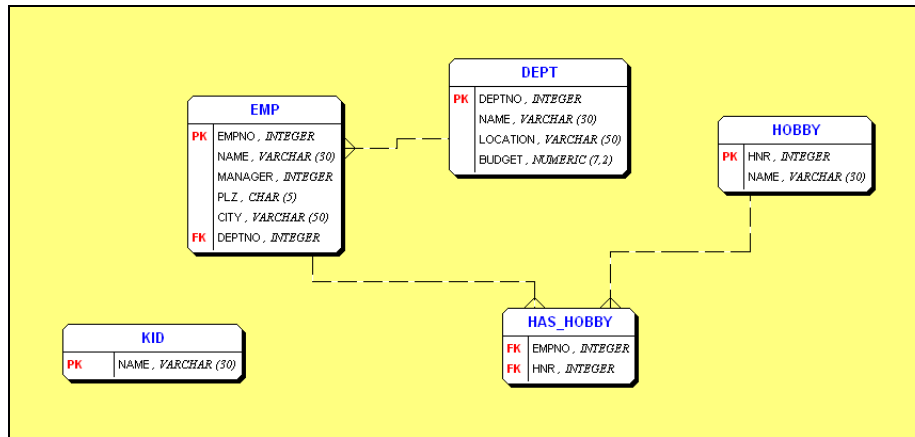


Abbildung 143 ER-Modell nach den Beziehungen der Hobbies

Im zusätzlichen Entity „Has\_Hobby“ wurden beide Fremdschlüssel eingetragen. Sinnvoll ist es nun, diese als Primary-Key zu definieren.

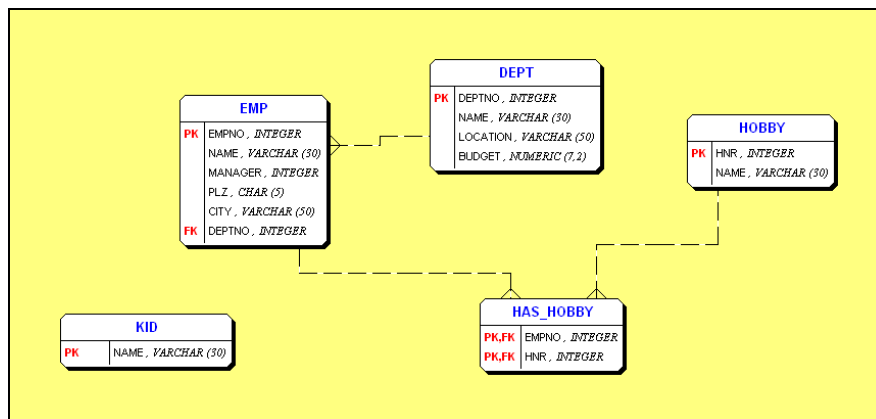


Abbildung 144 ER-Modell mit Primary-Key (Has\_Hobby)

### 8.1.6 Self-Relation

Das Entity „Emp“ hat eine spezielle Beziehung. Das Attribut „Manager“ verweist auf das Attribut „EmpNo“ im gleichen Entity. Das bedeutet zum Einen, dass Manager Mitarbeiter sein müssen, und zum Anderen dass in der Tabelle fast jeder Mitarbeiter einen Vorgesetzten hat.

Der Aufruf der Beziehung liefert:

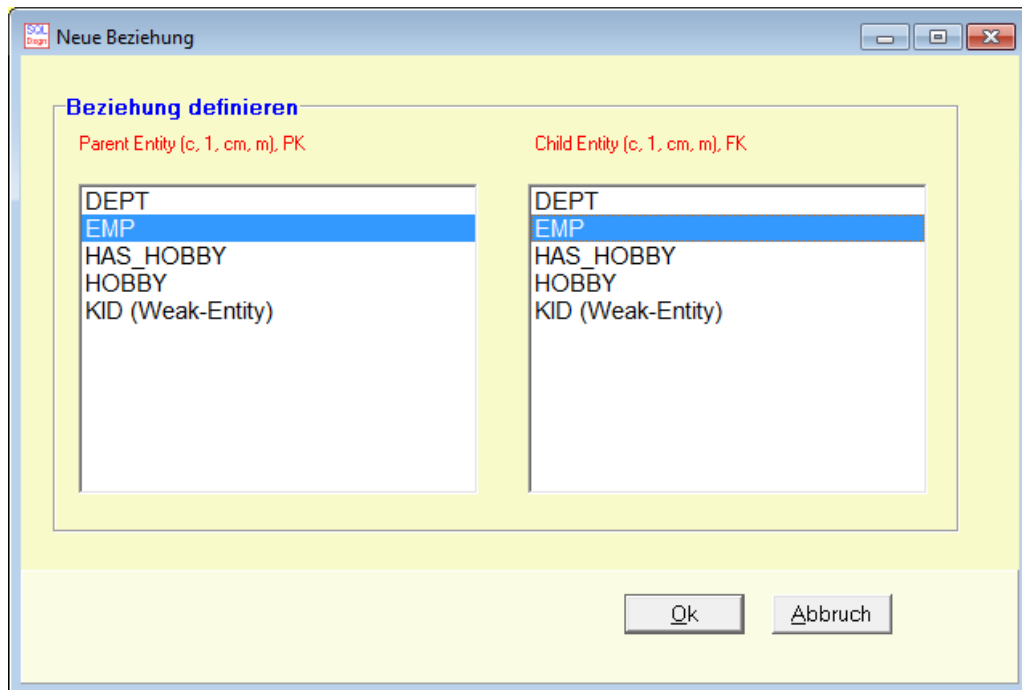


Abbildung 145 Erzeugen einer Self-Relation

#### Beziehungstexte:

- Ein Mitarbeiter kann kein oder einen Vorgesetzten haben
- Ein Vorgesetzter hat keinen, einen oder mehrere Mitarbeiter

Parent-Attribut: Emp

Child-Attribut: Manager

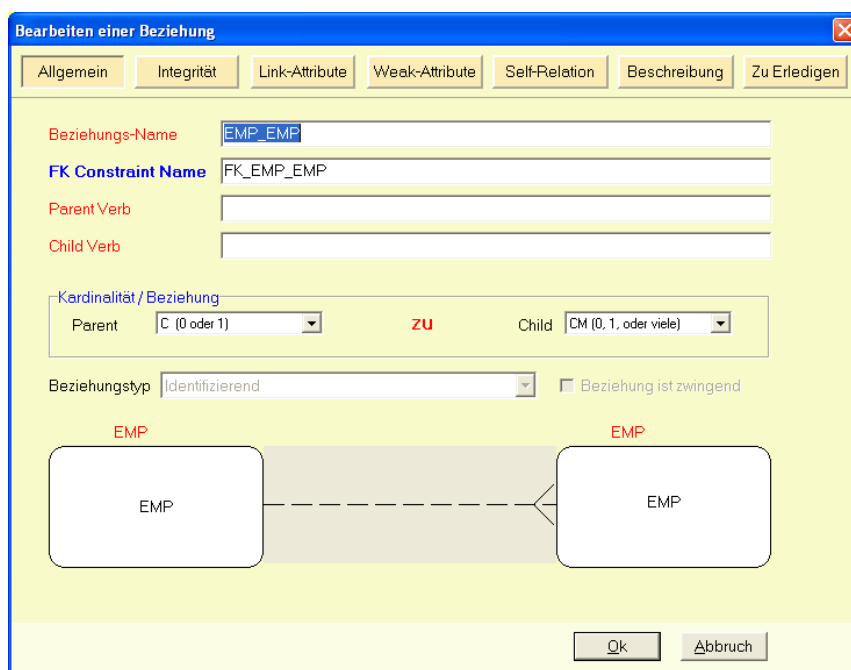


Abbildung 146 Self-Relation

Hinweis:

Das Fremdschlüssel-Attribut muss schon im Entity sein. Ansonsten würde es doppelt erscheinen.

- Aufruf des dritten Registers
- Abfrage **verneinen**
- Links anklicken Emp
- Rechts anklicken Manager
- Schalter „Connect“ betätigen

**Bearbeiten einer Beziehung**

Allgemein | Integrität | **Link-Attribute** | Weak-Attribute | Self-Relation | Beschreibung | Zu Erledigen

Disconnect

Nr	Parent-Key	Child-Key	Parent-Ind	Child-Ind	Parent-ID	Child-ID
1	EMPNO	MANAGER	0	2	9	18

Connect

**EMP**

Nr	PK	FK	Attribut	Datentyp
1	PK		EMPNO	INTEGER
			NAME	VARCHAR
		FK	MANAGER	INTEGER
			PLZ	CHAR
			CITY	VARCHAR
		FK	DEPTNO	INTEGER

**Manager**

Nr	PK	FK	Neu Attribut	Datentyp
	PK		EMPNO	INTEGER
			NAME	VARCHAR
1		FK	MANAGER	INTEGER
			PLZ	CHAR
			CITY	VARCHAR
		FK	DEPTNO	INTEGER

Ok Abbruch

Abbildung 147 EmpNo zu Manager definieren

Im fünften Register kann man die Darstellung der Self-Relation beeinflussen.

**Bearbeiten einer Beziehung**

Allgemein | Integrität | Link-Attribute | Weak-Attribute | **Self-Relation** | Beschreibung | Zu Erledigen

Dieses Register bestimmt die Darstellung der Self-Beziehung

Eine Self-Beziehung ist eine Relation mit dem gleichen Entity.  
Beispielsweise der Manager und der Mitarbeiter.

Text: Ein Mitarbeiter kann kein, oder einen Vorgesetzten haben  
Text: Ein Vorgesetzter hat keinen, einen oder mehrere Untergebene

Position der Relation

- ☒ Beziehung links oben
- ☐ Beziehung links unten
- ☐ Beziehung rechts oben
- ☐ Beziehung rechts unten

Links / Oben Entity Rechts / Oben

1. Attributname  
2. Attributname  
3. Attributname

Links / Unten Rechts / Unten

Ok Abbruch

Abbildung 148 Einstellungen für die Darstellung der Self-Beziehung

In der ersten Position wird diese links oben, wie in Abbildung 149, gezeichnet. Bei der nächsten Option wird diese dann links unten dargestellt. Und so weiter.

Das nächste Bild zeigt das neue ER-Modell.

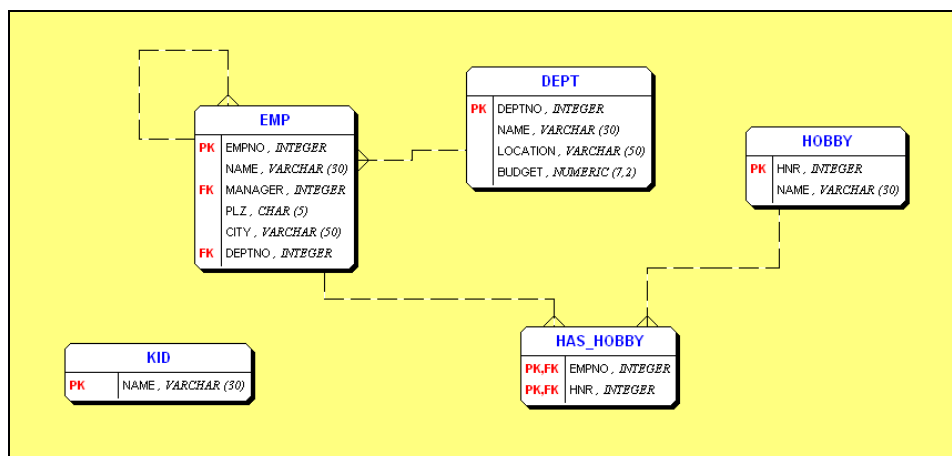


Abbildung 149 Darstellung der Self-Relation

Der letzte Punkt ist die Beziehung zwischen dem Entity „Emp“ und „Kid“.

### 8.1.7 Weak-Relation

Nach dem Aufruf der Beziehung werden folgende Eingaben getätigt:

- Eintragen der Beziehung (1:cm)
- Automatische Übertragung des Fremdschlüssels
- Im vierten Register die Eigenschaft „Weak-Entity“ aktivieren



**Bearbeiten einer Beziehung**

Allgemein   Integrität   Link-Attribute   Weak-Attribute   Self-Relation   Beschreibung   Zu Erledigen

Beziehungs-Name: EMP\_KID

FK Constraint Name: FK\_EMP\_KID

Parent Verb: Ein Mitarbeiter hat kein, ein oder mehrere Kinder

Child Verb: Ein Kind hat genau einen Mitarbeiter

Kardinalität / Beziehung: Parent: Exakt, Child: CM (0, 1, oder viele)

Beziehungstyp: Identifizierend ☐ Beziehung ist zwingend

EMP KID

Ok Abbruch

Abbildung 150 Beschreibung des Weak-Entities

Automatische Beziehung erstellen:

**Bearbeiten einer Beziehung**

Allgemein   Integrität   Link-Attribute   Weak-Attribute   Self-Relation   Beschreibung   Zu Erledigen

Verknüpfte Attribute

Disconnect

Nr	Parent-Key	Child-Key	Parent-Ind	Child-Ind	Parent-I	Child-I
1	EMPNO	EMPNO	0	1	9	214

Connect

EMP KID

Nr	PK	FK	Attribut	Datentyp
1	PK		EMPNO	INTEGER
			NAME	VARCHAR
		FK	MANAGER	INTEGER
			PLZ	CHAR
			CITY	VARCHAR
		FK	DEPTNO	INTEGER

Nr	PK	FK	Neu	Attribut	Datentyp
	PK			NAME	VARCHAR
1		FK		EMPNO	INTEGER

Ok Abbruch

Abbildung 151 Eintragen des Primarykeys als Fremdschlüssels

Aktivieren des Weak-Entity:

**Bearbeiten einer Beziehung**

Allgemein   Integrität   Link-Attribute   **Weak-Attribute**   Self-Relation   Beschreibung   Zu Erledigen

Dieses Register erlaubt die Einstellung eines Weak-Entity-Verbindung

Dies sind Objekte, die im Rahmen des logischen Schemas nicht selbst identifizierbar sind, sondern nur in Zusammenhang mit einem anderen Objekt, dem sie zugeordnet sind.

**Beispiel:**

- Entity: Mitarbeiter mit Schlüssel MNR
- Entity: Kind mit Schlüssel KName

Das Entity Kind muss zur Identifizierung den Fremdschlüssel des Mitarbeiter haben

Dann muss diese Beziehung so aussehen: Parent=Mitarbeiter Child=Kind (1:cm)

**Weak-Entity-Verbindung**

☐ Beziehung ist keine Weak-Entity-Beziehung

☒ Beziehung ist eine Weak-Entity-Beziehung

Ok   Abbruch

Abbildung 152 Aktivieren des Weak-Entity

Damit ist das ER-Modell fertig:

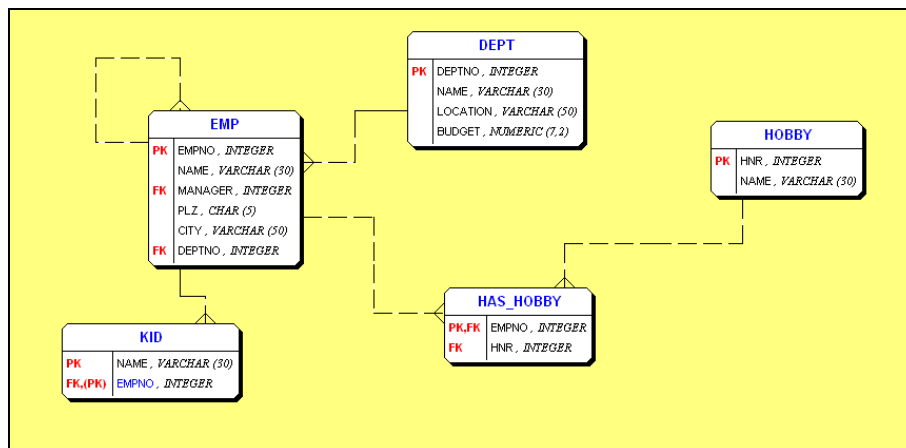


Abbildung 153 ER-Modell des sechsten Beispiels

Im Entity „Kid“ wurde zusätzlich im Attribut „Empno“ die Eigenschaft „PK“ eingetragen. Zur Unterscheidung gegenüber dem normalen Fall, siehe „Has\_Hobby“, wird die Bezeichnung „PK“ in Klammern gesetzt und das Attribut mit blauer Schriftfarbe gezeichnet. In der allgemeinen Notation wird ein Weak-Entity mit einem Rechteck mit doppelten Rand gezeichnet.

### 8.1.8 Weitere Schritte

Als weiteren Schritt wird nun eine Prüfbedingung für das Attribut „Budget“ im Entity „Dept“ eingetragen. Dazu klickt man doppelt auf das Entity in der Grafik oder man klickt im Baum auf den Eintrag „Dept“.

Im Dialogfenster ruft man den Editor für das Attribut „Budget“ auf.

The screenshot shows a dialog box titled 'Einstellung eines Attributes: BUDGET'. It has several tabs: 'Allgemein', 'Default / Unique', 'Checks', 'Linked-Attribute', 'Beschreibung', 'Zu Erledigen', and 'Object-ID'. The 'Default / Unique' tab is selected. Inside this tab, there is a section 'DefaultWert: (NUMERIC)' with a checked checkbox 'Defaultwert'. Below it is a text input field containing the value '0'. A red note below the input field states: 'Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)'. Below this is a section 'Unique-Bedingungen' with an unchecked checkbox 'Unique' and a text input field 'Check-C. Name' which is empty. At the bottom of the dialog are 'Ok' and 'Abbruch' buttons.

Abbildung 154 Defaultwert für Budget eingeben

Im Register „Default-Werte“ trägt man eine Ziffer 0 ein, oder einen beliebigen passenden Wert. Im dritten Register aktiviert man das Kontrollfeld „Check-Attribute“.

The screenshot shows the same dialog box, but now the 'Checks' tab is selected. In the 'Check-Bedingungen' section, the 'Check Attribute' checkbox is checked. The 'Check-C. Name' field contains the text 'CH\_BUDGET'. Below this is a text area for the 'Check-Bedingung' containing the code: 

```
CHECK (
  BUDGET > 0
)
```

 At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 155 Check-Constraint für das Attribut Budget

Dann wird automatisch ein passender Rahmen eingetragen. Für dieses Beispiel reicht es aus, wenn das Budget größer Null sein soll.

Als letzter Schritt wird eine Unique-Bedingung für den Abteilungsnamen eingeben.

The screenshot shows a dialog box titled "Einstellung eines Attributes: BUDGET". It has several tabs: "Allgemein", "Default / Unique", "Checks", "Linked-Attribute", "Beschreibung", "Zu Erledigen", and "Object-ID". The "Default / Unique" tab is active. In the "Defaultwert: (NUMERIC)" section, the "Defaultwert" checkbox is checked, and the "Wert" field contains "0". Below this, a red note states: "Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)". In the "Unique-Bedingungen" section, the "Unique" checkbox is checked, and the "Check-C. Name" field contains "UNIQUE\_BUDGET". This section is highlighted with a red rectangle. At the bottom, there are "Ok" and "Abbruch" buttons.

**Abbildung 156 Unique-Bedingung für den Abteilungsnamen**

Nach diesen Eingaben sind alle Vorbereitungen abgeschlossen und man kann das SQL-Skript erzeugen.

#### 8.1.9 Generierung einer Datenbank

Mit dem Menü „Schema“ und dem Eintrag „Generierung Datenbank“ wird die Erzeugung des SQL-Skriptes angestoßen. Als Kurztaste existiert die Taste F9. Es erscheint folgendes Fenster (siehe Abbildung 157). In der obersten Zeile wird der Dateiname eingetragen. Der nächste Teil definiert die Objekte, die generiert werden sollen. Im letzten Teil sind zusätzliche Optionen vorhanden.

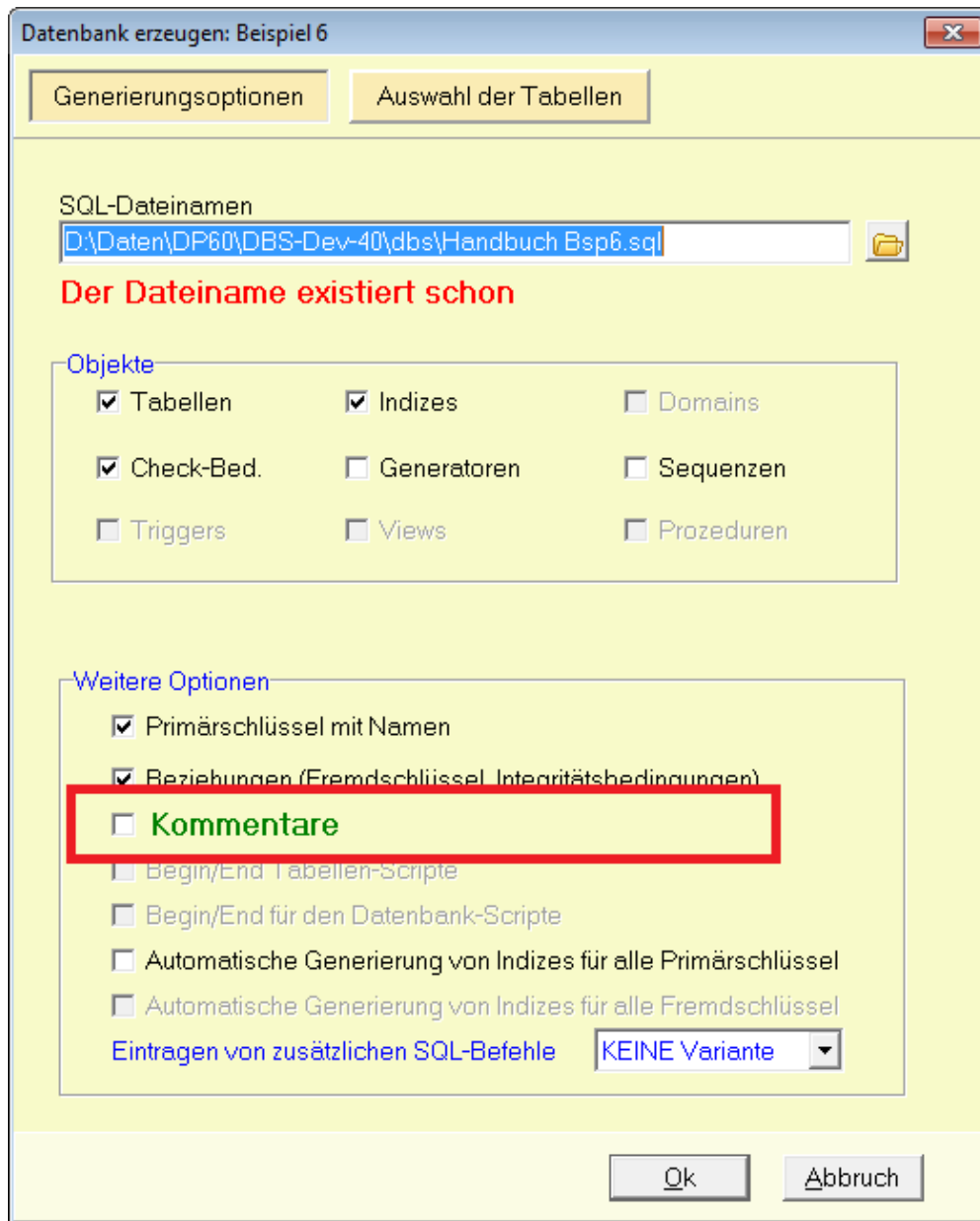


Abbildung 157 Erzeugung einer Datenbank, ohne Kommentare

Mit dem Schalter „Ok“ wird das Script erzeugt und im Notepad dargestellt.

#### Ergebnis mit SQL-Befehlen:

```
CREATE TABLE DEPT (
  DEPTNO  INTEGER NOT NULL,
  NAME    VARCHAR(30) NOT NULL UNIQUE,
  LOCATION VARCHAR(50),
  BUDGET  NUMERIC DEFAULT 0,

  CONSTRAINT PK_DEPTNO PRIMARY KEY (DEPTNO)
);
```

```
CREATE TABLE EMP (
  EMPNO  INTEGER NOT NULL,
  NAME    VARCHAR(30) NOT NULL,
  MANAGER INTEGER NOT NULL,
  PLZ    CHAR(5),
```

```
CITY VARCHAR(50),
DEPTNO INTEGER NOT NULL,

CONSTRAINT PK_EMPNO PRIMARY KEY (EMPNO)
);

CREATE TABLE HAS_HOBBY (
  EMPNO INTEGER NOT NULL,
  HNR INTEGER NOT NULL,

  CONSTRAINT PK_HOBBY_EMPNO PRIMARY KEY (EMPNO)
);

CREATE TABLE HOBBY (
  HNR INTEGER NOT NULL,
  NAME VARCHAR(30) NOT NULL,

  CONSTRAINT PK_HNR PRIMARY KEY (HNR)
);

/*
  Tabelle ist ein Weak-Entity
*/
CREATE TABLE KID (
  NAME VARCHAR(30) NOT NULL,
  EMPNO INTEGER NOT NULL,

  CONSTRAINT PK_NAME PRIMARY KEY (NAME, EMPNO )
);

ALTER TABLE EMP ADD CONSTRAINT
  FK_DEPT_EMP FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO);

ALTER TABLE EMP ADD CONSTRAINT
  FK_EMP_EMP FOREIGN KEY (MANAGER) REFERENCES EMP(EMPNO);

ALTER TABLE HAS_HOBBY ADD CONSTRAINT
  FK_EMP_HAS_HOBBY FOREIGN KEY (EMPNO) REFERENCES EMP(EMPNO);

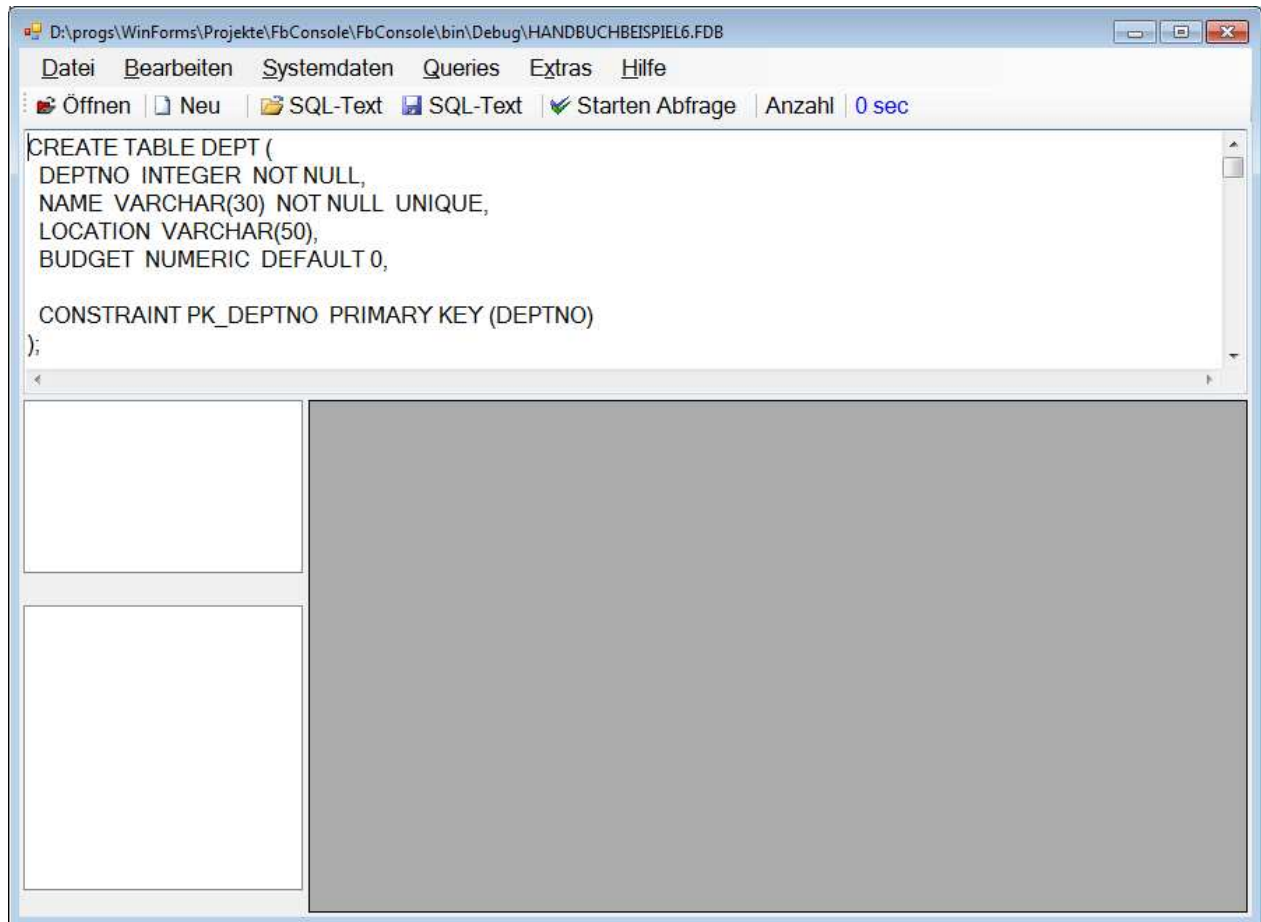
ALTER TABLE KID ADD CONSTRAINT
  FK_EMP_KID FOREIGN KEY (EMPNO) REFERENCES EMP(EMPNO);

ALTER TABLE HAS_HOBBY ADD CONSTRAINT
  FK_HOBBY_HAS_HOBBY FOREIGN KEY (HNR) REFERENCES HOBBY(HNR);

ALTER TABLE DEPT
  ADD CONSTRAINT CH_BUDGET CHECK (
    BUDGET >=0
  );
```

## 8.1.10 Erstellen der Datenbank

- Aufruf des Programms FBConsole
- Taste Strg+N
- Kopieren der SQL-Befehle mittels Strg+V
- Taste F5: Einfügen der Befehle



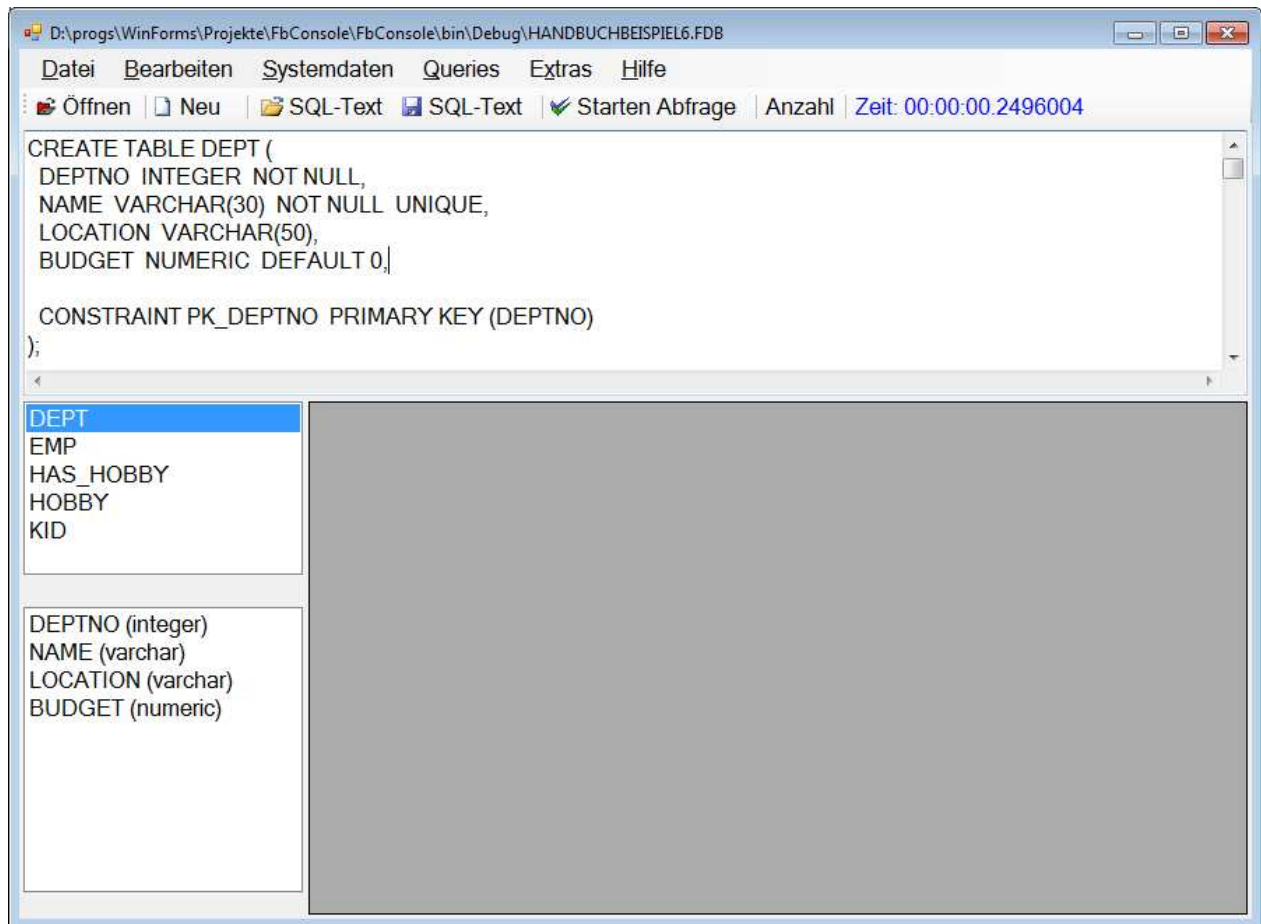


Abbildung 158 Anzeige der Tabellen, linke Liste

Im ersten Schritt müssen nun die Abteilungen eingetragen werden (warum?)

Antwort:

Die Abhängigkeiten der Entities in den Relationen bestimmt die Reihenfolge des Eintragens bzw. des Löschens.

#### 8.1.11 Daten für die Tabelle „Dept“

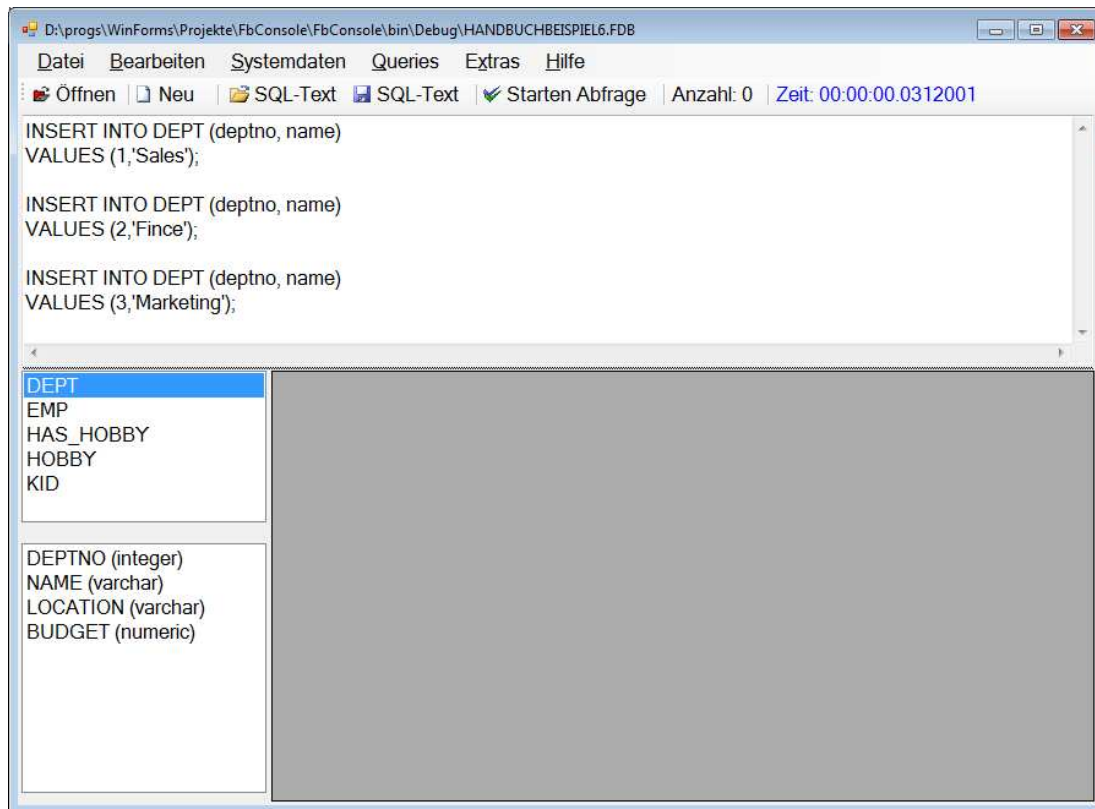
```
INSERT INTO DEPT (deptno, name)  
VALUES (1,'Sales');
```

```
INSERT INTO DEPT (deptno, name)  
VALUES (2,'Fince');
```

```
INSERT INTO DEPT (deptno, name)  
VALUES (3,'Marketing');
```

Die Insert-Befehle werden nun in den SQL-Editor eingetragen und mit F5 in die Datenbank eingefügt.





Eine Überprüfung zeigt die korrekte Eingabe.

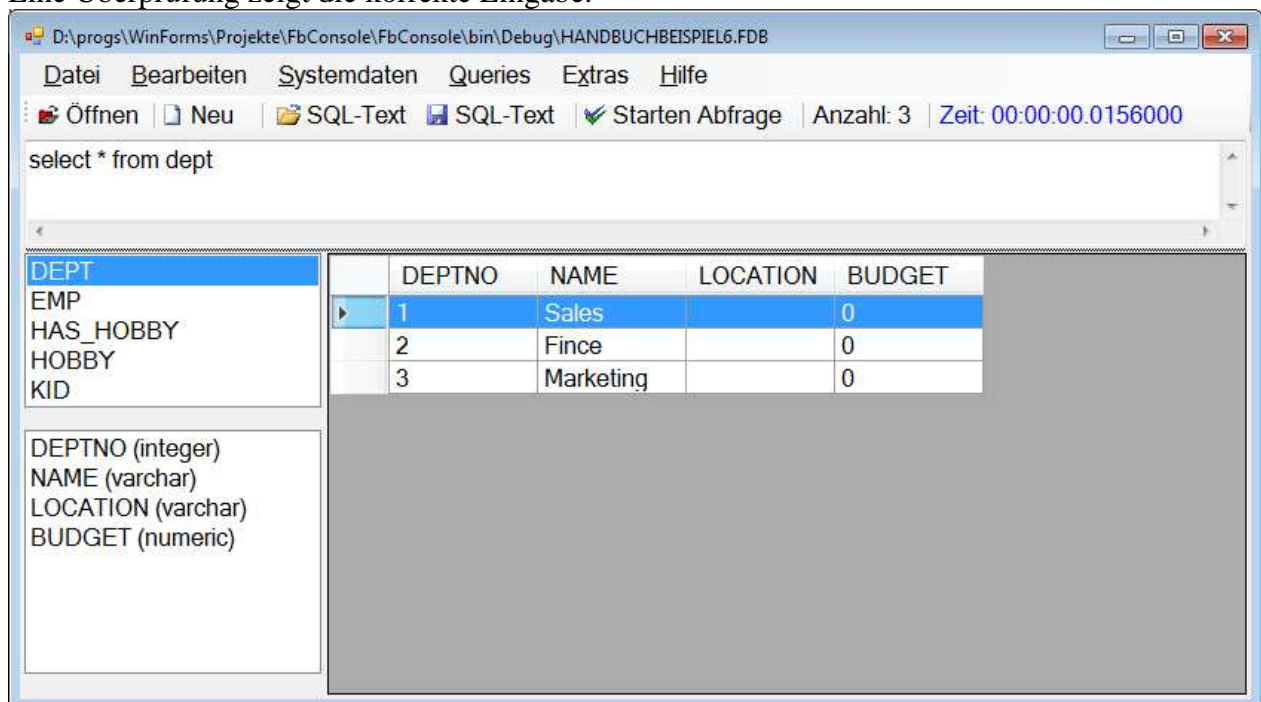
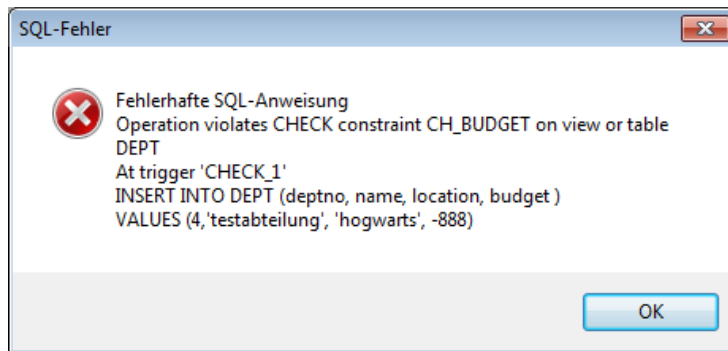


Abbildung 159 Abfrage der Daten für die Abteilung

Hinweis:

INSERT INTO DEPT (deptno, name, location, budget )  
VALUES (4,'testabteilung', 'hogwarts', -888);

Die obige Anweisung wird von der Datenbank nicht akzeptiert, da das Budget kleiner als Null ist.



**Abbildung 160** Test der Check-Bedingung Budget

Bei den Daten für die Mitarbeiter müssen die Manager mit eingegeben werden, da eine Fremdschlüsselbeziehung existiert. Deshalb müssen als erstes die Manager eingegeben werden. Des Weiteren brauchen diese natürlich auch „Vorgesetzte“. Der einfachste Weg ist es, für Manager ohne Chef, die eigene Nummer zu einzutragen. Oder den Constraint bei diesen Mitarbeitern auszuschalten.

Alternativ, kann man die Fremdschlüssel-Beziehung auch in einem „Before-Insert“-Trigger prüfen.

#### 8.1.12 Daten für die Tabelle „Emp“

##### Daten:

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (11,'müller', 11, '38855', 'Wernigerode',1);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (5,'schmidt', 5,'39114', 'Magdeburg',2);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (21,'schulze', 21, '06012', 'Halle (Saale)',3);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (1,'meier', 11, '38855', 'Wernigerode',1);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (2,'bernstein', 5, '39115', 'Magdeburg',1);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (4,'meyer', 21, '38855', 'Wernigerode',3);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (7,'schubert', 5, '39125', 'Magdeburg',2);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (44,'koch', 5, '38855', 'Wernigerode',3);
```

```
INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (27,'schulze', 21, '39125', 'Magdeburg',2);
```

## **8.2 Generator für die Mitarbeiternummer EMPNO**

Siehe Kapitel 10.7, Seite 130

## 9 Funktionen

Dieses Kapitel beschreibt kurz die wichtigsten Funktionen.

### 9.1 Menüfunktionen

#### 9.1.1 Menü Datei

Menüfunktionen	Beschreibung
Neues Projekt	Öffnet ein neues logisches Projekt(STRG+N)
Neues Projekt	Öffnet ein neues konzeptionelles Projekt(STRG+M)
Öffnen	Öffnet ein vorhandenes Projekt(STRG+O), logisches Modell
Öffnen	Öffnet ein vorhandenes Projekt(STRG+O), konzeptionelles Modell (STRG+K)
Projekt schließen	Schließt das aktuelle Projekt
Speichern	Speichert das Projekt. (STRG+S)
Speichern unter	Speichert das Projekt unter einem anderen Namen
Export als Bild	Das ER-Modell wird als Bild in die Zwischenablage kopiert
Export der Daten nach Winword	Alle Daten werden in eine neue WinWord-Datei eingetragen
Alte Dateien	Ein Untermenü zeigt die zuletzt bearbeiteten Dateien an.

#### 9.1.2 Menü Projekt

Menüfunktionen	Beschreibung
Eigenschaften	Zeigt ein Dialogfenster, in dem man diverse Einträge vornehmen kann.

The 'Projekt-Eigenschaften' dialog box is shown with the 'Allgemein' tab selected. It contains the following fields and controls:

- Name:** Text field with 'Beispiel 3'.
- Firma / Autor:** Text field with 'Michael Wilhelm'.
- Copyright:** Text field with 'Copyright by Michael Wilhelm'.
- Erstellt:** Date/time field with '08.05.2007 20:07:09'.
- Geändert:** Date/time field with '08.05.2007 20:07:09'.
- Primarykey:** A dropdown menu showing 'Primary-Key immer mit Integer' and a numeric field 'N' with the value '38'.
- Abmessungen der Tabellenfläche:** A sub-section containing 'Breite' (3000) and 'Höhe' (2000) with spinners.
- Buttons:** 'Ok' and 'Abbruch' at the bottom right.

**Abbildung 161 Projekt-Eigenschaften**

In diesem Dialogfenster werden alle Schriften und Farben der Entities definiert. Die Farben können nachher individuell pro Entity geändert werden. Im Register „Grafiklinien2“ hat man die Möglichkeit, die Darstellung der Beziehungen zu definieren.

The 'Projekt-Eigenschaften' dialog box is shown with the 'Grafiklinien' tab selected. It contains the following elements:

- Text:** 'In diesem Fenster können Sie auswählen, welche Darstellung im ERM-Darstellung Sie wählen.'
- Linien-Darstellung:** A group box containing three radio buttons:
  - ☒ Krähenfuss-Darstellung HS Harz
  - ☐ Krähenfuss-Darstellung Martin-Notation
  - ☐ Chen-Darstellung
- Diagram:** A graphical representation of a relationship. On the left, a box labeled 'LParent\_Relation' is connected by a dashed line to a box labeled 'LChild\_Relation' on the right. The line has a '1' near the parent box and a 'cm' near the child box. The child box has a crow's foot notation symbol (a line ending in a T-shape).
- Buttons:** 'Ok' and 'Abbruch' at the bottom right.

**Abbildung 162 Grafiklinien**

## 9.1.3 Menü Einfügen

Menüfunktionen	Beschreibung
Entity	Fügt ein neues Entity in das Projekt ein (STRG+E)
Beziehung	Fügt eine neue Beziehung in das Projekt ein (STRG+R)
Ternäre Beziehung	Fügt eine neue ternäre Beziehung in das Projekt ein (STRG+T)
Generator	Fügt einen Generator für ein Attribut in das Projekt
Sequenz	Fügt eine Sequenz für ein Attribut in das Projekt

## 9.1.4 Menü Modell

Menüfunktionen	Beschreibung
Eigenschaften	Zeigt ein Dialogfenster mit dem wichtigsten Eigenschaften des logischen Modells
Neues Entity	Fügt ein neues Entity in das Projekt ein (STRG+E)
Entitie-Liste	Es wird eine Liste aller Entities angezeigt, in der man Entities bearbeiten, erzeugen und löschen kann.
Neue Beziehung	Fügt eine neue Beziehung in das Projekt ein (STRG+R)
Liste aller Beziehungen	Es wird eine Liste aller Beziehungen angezeigt, in der man diese bearbeiten und löschen kann.
Neuen Generator	Fügt einen neuen Generator in das Projekt ein
Liste aller Generatoren	Es wird eine Liste aller Generatoren angezeigt, in der man diese bearbeiten und löschen kann.
Neue Sequenz	Fügt eine neue Sequenz in das Projekt ein
Liste aller Sequenzen	Es wird eine Liste aller Sequenzen angezeigt, in der man diese bearbeiten und löschen kann.

## 9.1.5 Menü Schema

Menüfunktionen	Beschreibung
Check Modell	Diese Funktion testet die Datenbank. Ist zurzeit noch nicht implementiert.
Generierung der Datenbank	Diese Funktion generiert ein SQL-Script. Im oberen Teil wird der Dateiname festgelegt, danach die Komponenten und am Schluss die zusätzlichen Optionen.



Abbildung 163 Generierung des SQL-Skriptes

Im zweiten Register können einzelne Tabellen ausgewählt werden.

#### 9.1.6 Menü Optionen

Menüfunktionen	Beschreibung
Erzeugen eines Datenbankmodells	Mit dieser Funktion erstellt man die Definitionen einer beliebigen relationalen Datenbank. Alle Datentypen sind frei wählbar.
Taschenrechner	Stellt den Windows-Taschenrechner zur Verfügung.
Dateiliste löschen	Löscht die Liste der alten Dateien.

## 9.2 Funktionen des Objekt-Baumes

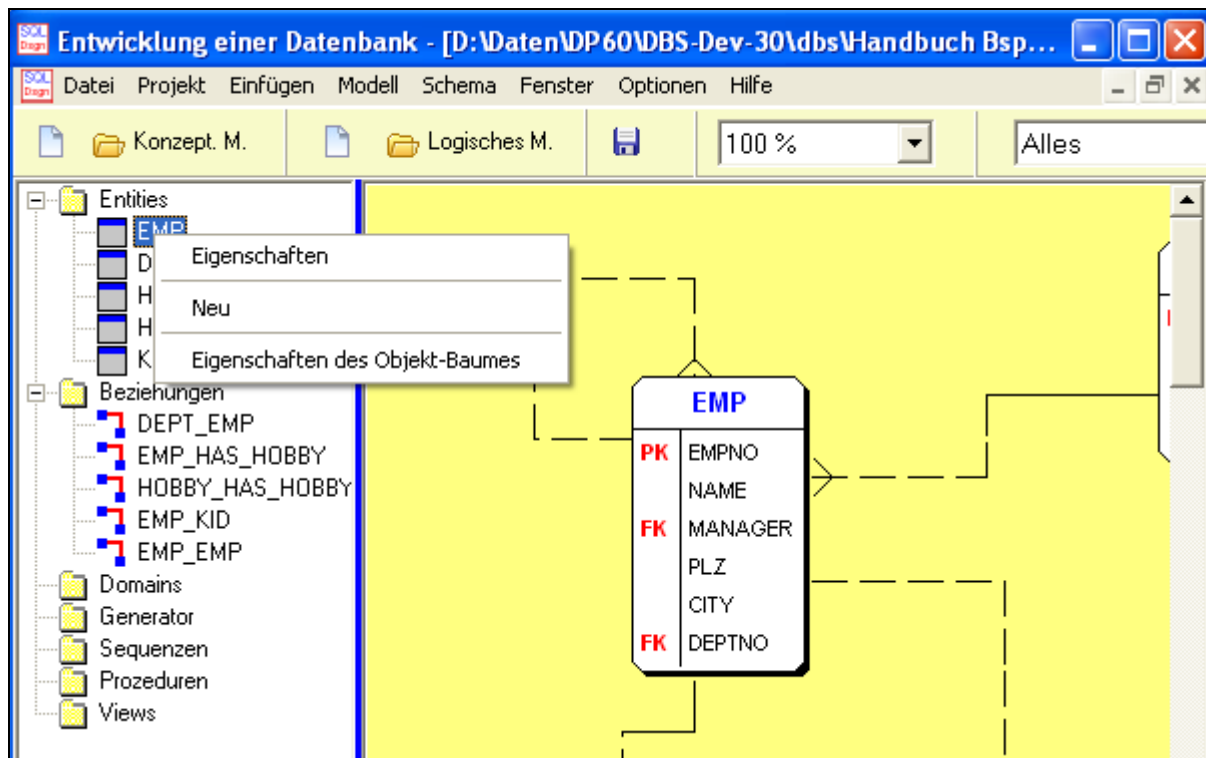


Abbildung 164 Funktionen des Objekt-Baumes

Um die Funktionen zu benutzen muss man zum Einen den Eintrag anklicken. Mit der rechten Maustaste erhält man Zugriff auf drei Einträge (Eigenschaften, Neu, Eigenschaften des Objektbaumes). Der Eintrag „Eigenschaften“ wird nur angezeigt, wenn man auf ein Objekt angeklickt hat. Die Funktion erlaubt das Erzeugen neuer Entities bzw. Relationen. Je nachdem, welches Element man angeklickt hat.

Mit dem letzten Punkt kann man die Schrift und Farbe des Baumes verändern.

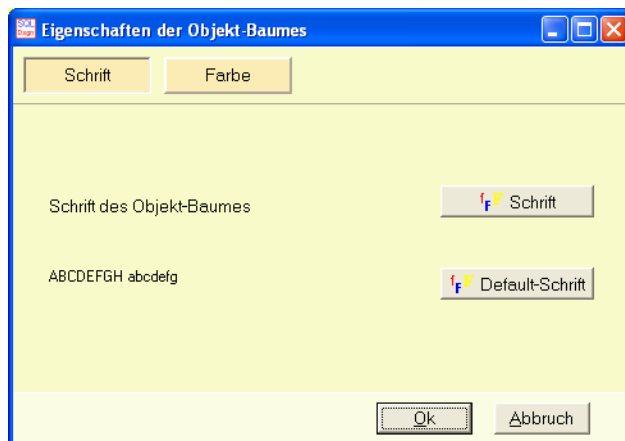


Abbildung 165 Eigenschaftsdialog des Baumes



### 9.3 Funktionen der Grafik

Die Grafik liefert zwei Funktionen. Zum Einen die Einstellung der Farbe und zum Anderen den Aufruf von Funktionen von Entities.

#### 9.3.1 Funktionen des Grafik-Panels

Ein Doppelklick auf eine freie Stelle liefert die Abbildung 166. Hier kann man die Hintergrundfarbe und die Größe einstellen.

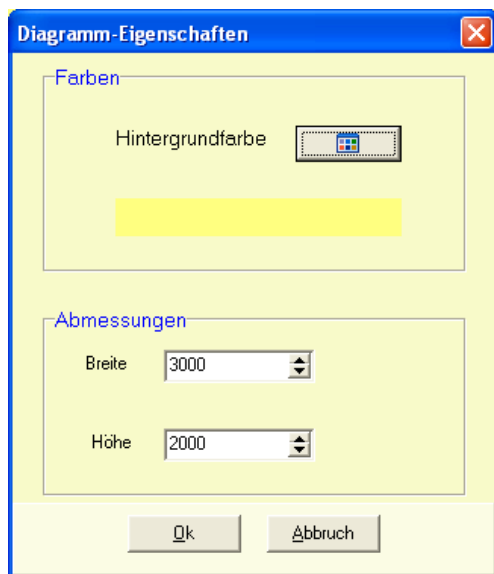


Abbildung 166 Eigenschaften des Grafikpanels

Mit der rechten Maustaste erscheint folgendes Menü. Hier kann man Entities und Beziehungen einfügen. Oder man wählt die zwei Beispiele, die vorgegeben sind.

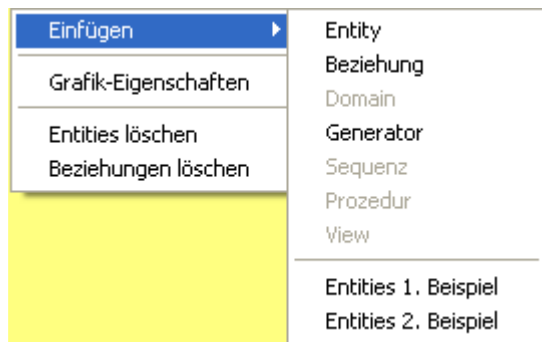


Abbildung 167 Popup-Funktionen des Grafik-Panels

### 9.3.2 Entity-Funktionen

Bewegt man den Cursor über die Entities und betätigt die rechte Maustaste, so erhält man die Funktionen (Eigenschaften und Farbe). Mit dem Eintrag „Eigenschaften“ wird das Dialogfenster Abbildung 168 angezeigt.

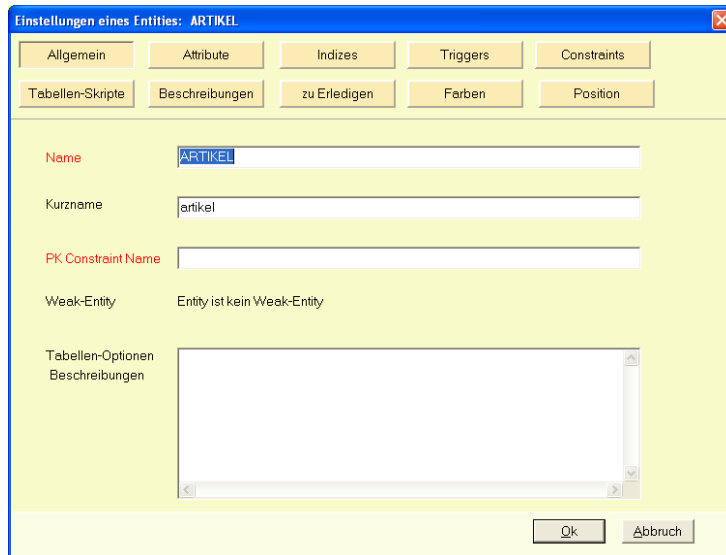


Abbildung 168 Eigenschaften eines Entities

In diesem Dialogfenster funktionieren folgende Register nicht:

- Indizes
- Triggers
- Constraints
- Tabellenscripte

Im Register „Farben“ kann man die verschiedenen Farben auswählen (siehe Abbildung 169).

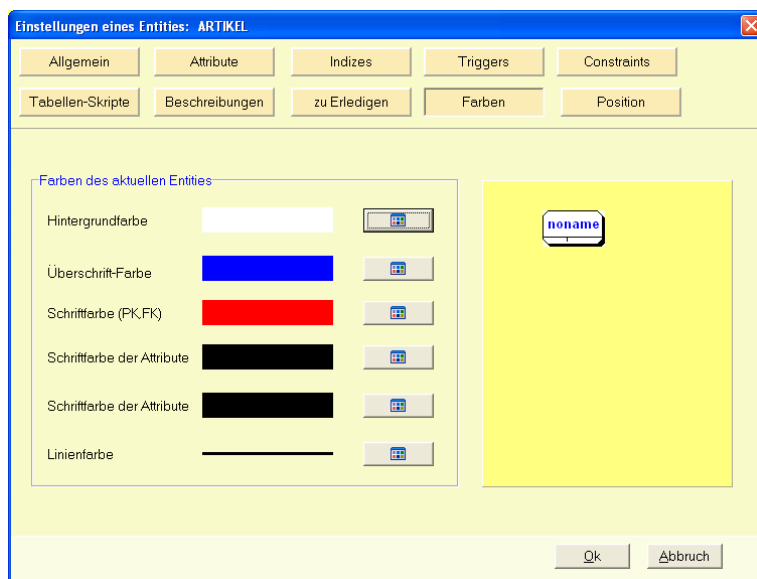


Abbildung 169 Register Farben in den Entity-Einstellungen

Hinweis:

Im Register „Attribute“ werden in einer Tabelle die vorhandenen angezeigt. Mit einem Doppelklick können diese bearbeitet werden.

## 9.4 Funktionen der Attribute

Aus dem Einstellungs-Dialogfenster „Entity“ kann man die einzelnen Attribute bearbeiten. Die Abbildung 170 zeigt das Fenster zur Bearbeitung der Attribute. Es werden zurzeit noch keine Domains unterstützt, so dass man alle Datentypen direkt eingeben muss. Das Register „Linked-Attributes“ hat noch keine Funktion. Im zweiten Register kann der Default-Wert eingetragen werden.

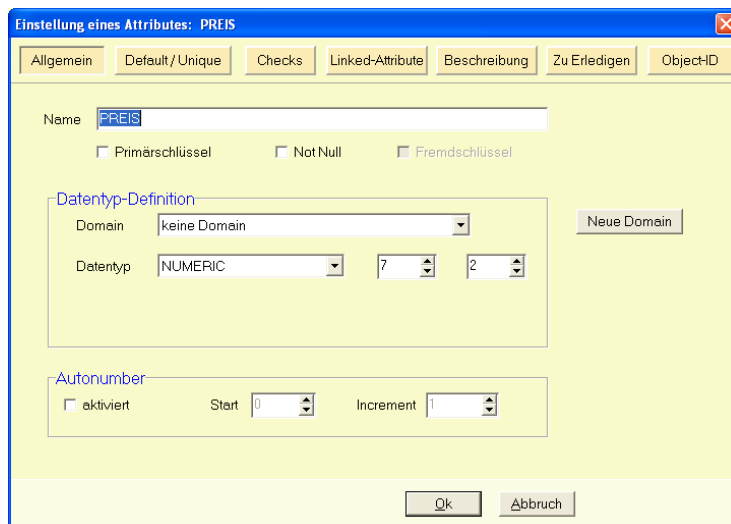
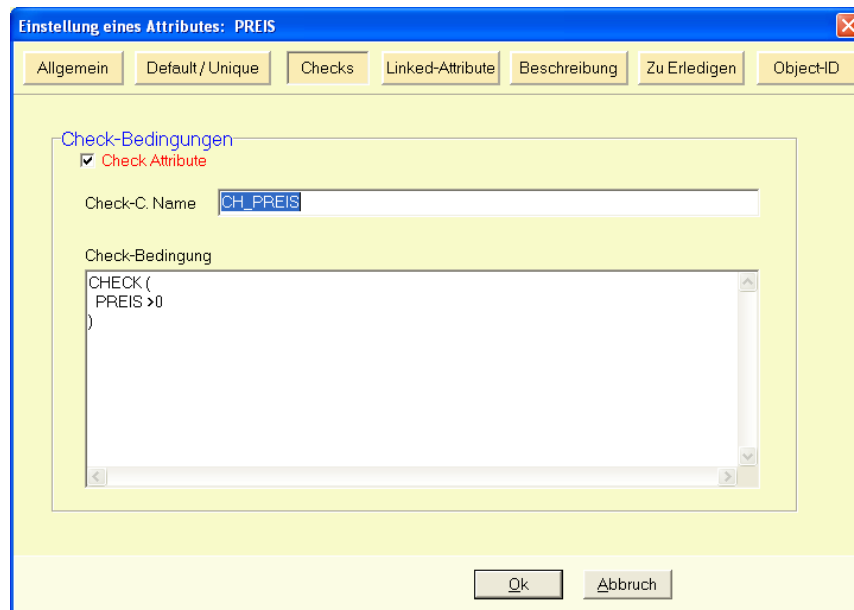


Abbildung 170 Einstellungen der Attribute. Primarykey, logische Modell

Klickt man das Kontrollfeld „Add to Primarykey“ an, so wird beim ersten Aufruf der Datentyp „Integer“ ausgewählt (siehe auch Abbildung 2, Seite 14, Projekteigenschaften (Integer vs. Max Numeric). In den weiteren Aufrufen bleibt der Datentyp erhalten.

Im zweiten Register setzt man einen optionalen Defaultwert und einen Uniquewert. Die Unique-Bedingung definiert ähnlich einem Primarykey, dass im Wertebereich eines Attributs keine doppelten Tupel vorkommen. Sinnvoll zum Beispiel bei Abteilungs- oder Artikelnamen.

Das dritte Register setzt eine Prüfbedingung für das Attribut. Verletzt ein neuer Wert diese Bedingung, so werden alle Attribute einer Insert- oder Update-Anweisung nicht in die Datenbank eingetragen.



**Abbildung 171 Check-Bedingung**

Standardmäßig wird eine numerische Abfrage eingefügt. Die Zeichenmanipulations- oder Datumsfunktionen sind sehr speziell.

## 10 Spezielle Funktionen

### 10.1 Computed By

Mit Hilfe von „Computed By“ kann man berechnende Attribute in das Entity einfügen. Die Eingangsattribute müssen alle im Entity vorhanden sein.

Beispiel	Eingangsparameter	Ausgangsparameter
Monats- und Jahresgehalt	Gehalt	12 * Gehalt
Netto- und Bruttogehalt	Preis	1.19 * Preis
Gewässerstationen	Von und Bis	Länge = Bis-Von

#### 10.1.1 Syntax von Computed by

Der erste Begriff zeigt den Namen, dann folgt der Begriff „Computed By“, am Schluss folgt die Formel:

```
PREIS COMPUTED BY ( PREISCENT*0.01 ) ,
```

```
CREATE TABLE ARTIKEL (
  ANR INTEGER NOT NULL,
  NAME VARCHAR(40) NOT NULL UNIQUE,
  PREISCENT INTEGER DEFAULT 0,
  PREIS_DOUBLE DOUBLE PRECISION,
  PREIS_NUMERIC NUMERIC(7,2),
  PREIS COMPUTED BY ( PREISCENT*0.01 ) ,

  CONSTRAINT PK_ANR PRIMARY KEY (ANR)
);
```

#### Beispieldaten für „Handbuch\_bsp7“:

```
INSERT INTO Artikel (ANR, NAME, PREISCENT, PREIS_DOUBLE, PREIS_NUMERIC)
VALUES (1,'Buch', 1245, 12.45, 12.45);
```

```
INSERT INTO Artikel (ANR, NAME, PREISCENT, PREIS_DOUBLE, PREIS_NUMERIC)
VALUES (2,'CD', 1210, 12.10, 12.10);
```

```
INSERT INTO Artikel (ANR, NAME, PREISCENT, PREIS_DOUBLE, PREIS_NUMERIC)
VALUES (3,'Computer', 14570, 145.70, 145.70);
```

Man sollte einen expliziten Datentyp immer mit angeben.

#### Definition mit Datentyp:

```
PREIS NUMERIC(7,2) COMPUTED BY ( PREISCENT*0.01 ) ,
```

## 10.2 Weak-Attribute

Siehe Kapitel 8.1.7, Seite 104

## 10.3 Self-Relation

Siehe Kapitel 8.1.6, Seite 101

## 10.4 Ternäre-Relation

Siehe Kapitel 4.4, Seite 52

## 10.5 Erstellung einer Datenbank-Typdatei

Dieses Kapitel zeigt die Definition einer neuen Typdatei für Datenbanken. Mit dieser kann man eine beliebige relationale Datenbank definieren. Dazu existieren Grundtypen (Integer, Char, Float, etc.), mit denen die echten Typen verbunden werden. Weitere Punkt sind die Definition der Syntax-Darstellung und die Generierung eines automatischen Counters.

### 10.5.1 Aufruf der Funktion

Aufruf im Menü Optionen

Eintrag: Erzeugen eines Datenbank-Modells

Erzeugen einer Datenbank-Definitions-Datei

Name Attribute Reservierte Wörter Domain Sequenz / Generator / Autonumber

Dateiname

Ziel DBS

Kommentar

Maximale Länge eines Namens

Hochkomma

Namensanfang  Namensende

Blockkommentar Anfang  Blockkommentar Ende

Zeilenkommentar

Blocktrenner  Absatztrenner

Modell laden Speichern Abbruch

Abbildung 172 Dialogfenster zum Erstellen einer Datenbank-Definition

### 10.5.2 SQL-Syntax

#### Hochkomma:

Das Hochkomma können die einfachen und doppelten fungieren. Meistens werden die einfachen benutzt.

#### Namensanfang / Namensende:

Bei einige Datenbanken werden die Namen der Entites und Relationen in eckige Klanmmern eingegeben (MS SQL-Server, MS Access, NexusDB2). Mit diesem Trick kann man auch Leerzeichen in Namen verwenden.

#### Blockkommentar:

Die meistens verwenden die normalen Blockkommentare /\* \*/. Einige verwenden aber nur Zeilenkommentare (DB2, MaxDB, mySQL).

#### Zeilenkommentar:

Die meistens verwenden nur die normalen Blockkommentare /\* \*/. Einige verwenden aber auch Zeilenkommentare (DB2, dBase, dbIsam, FoxPro, Informix, MaxDB, Access, MS SQLK-Server, mySQL, NexusDB, Oracle, Paradox, PostgreSQL, SyBase). Dazu verwenden sie das Zeichen „-“.

#### Blocktrenner /Absatztrenner:

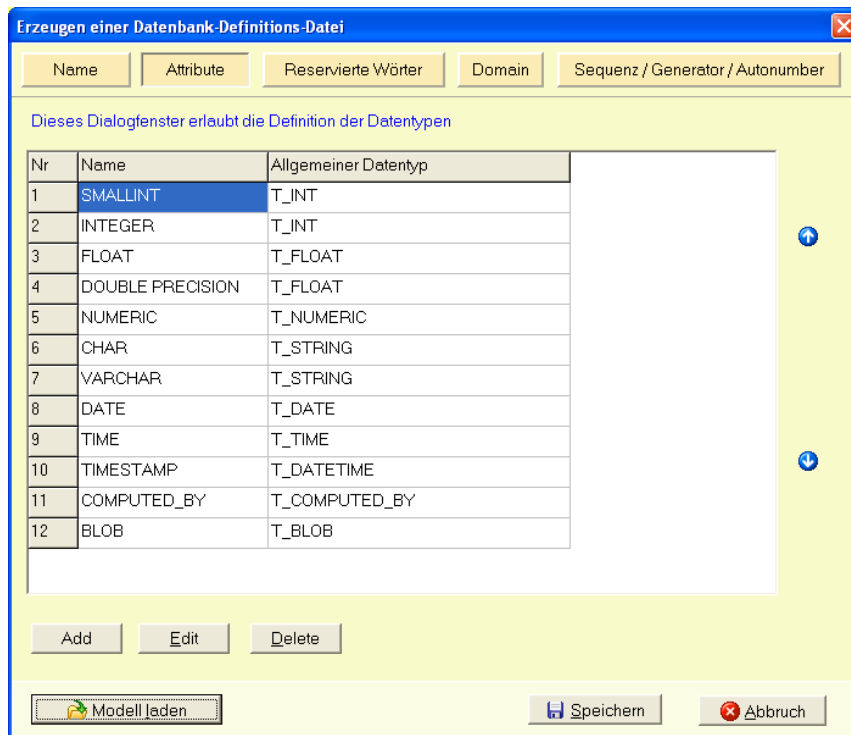
Der normalen Trenner ist das Semikolon. Einige wenige Datenbank verwenden hier folgende Definition:

#13#10GO#13#10

Dazu gehören: SyBase, MS SQL-Server, MaxDB

Mit den Einstellungen im ersten Register sollten alle Eventualitäten einstellbar sein.

Im ersten Register definiert man die allgemeinen Daten inklusive des Syntax. Im wichtigen zweiten Register definiert man die einzelnen Attribute mit ihren Grundtypen. Diese Typen dienen der Auswahl, welche grafischen Elemente bei der Auswahl aktiviert werden müssen. Das dritte Register zeigt die Begriffe, die in der Datenbank intern verwendet werden. Diese dürfen natürlich nicht als Namen für Entities oder Attribute dienen.



**Abbildung 173 Datenbank-Definitionsdialog**

Nicht jede Datenbank unterstützt Domains, also vordefinierte Typen mit optionalen Prüfbedingungen. Der Designer erlaubt aber trotz negativer Aussage, die Definition und Benutzung von Domains. Unterstützt die Datenbank keine Domains, werden die Domain-Typen durch die Definitionen ersetzt.

Ein etwas anderer Fall besteht bei der Definition der „Counter“. Die Datenbanken „Interbase“ und Firebird unterstützen den einfachen Generator. Oracle dagegen verwendet die Sequenz.

Man muss nun selber entscheiden, welche Definition bei der Zieldatenbank passender ist. Gegebenenfalls muss man das SQL-Skript anpassen.

#### Hinweis:

Die aktuelle Version 4,0 des Designers unterstützt noch nicht Domains und Trigger. Die Verhaltensweise wird aber dem obigen Text entsprechen!

Das letzte Register beinhaltet die Definition eines Counters. Auch hier ist die übliche Vielfalt zu bewundern. Access verwendet einen einfachen Zähler als Datentyp. Interbase und Firebird benutzen einen Generator. Oracle, Informix und Postgre verwenden die Sequenz. Hier aber auch in abgestufter Variante. Deshalb kann man alle Elemente einzeln aktivieren. Bei Problemen kann man ja das erstellte Skript immer noch abändern. Eine E-Mail wäre an den Entwickler wäre wünschenswert.

Sybase und MS-SQL verwenden für Integer und Numeric-Datentypen einen internen Counter. Dieser hat einen Start- und Incrementwert. In unteren Beispiel wird der Startwert auf zwei gesetzt und der Incrementwert ist drei (IDENTITY).



Beispielsyntax für MS SQL-Server:

```
CREATE TABLE [KUNDEN] (
    [KUNDENNR] INTEGER IDENTITY(2,3) NOT NULL,
    [NAME] CHAR(50),
    [VORNAME] CHAR(50),
    [STRASSE] CHAR(50),
    [PLZ] DECIMAL(5),
    [ORT] CHAR(50),

    PRIMARY KEY ([KUNDENNR])
)
GO
```

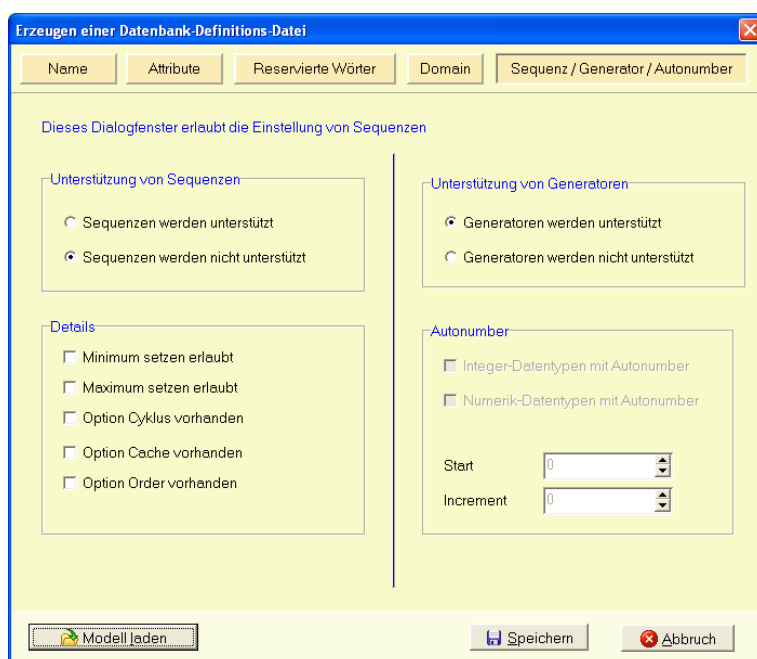


Abbildung 174 Sequenz, Generator definieren

## 10.6 Generator / Sequenz

Häufig benötigt man einen ganzzahligen aufsteigenden Schlüssel für ein Feld. Nun kann man mit folgendem Code diesen bestimmen (Pseudocode und PL-SQL):

```
MaxNr = SELECT max(MatNr)
        FROM Student
```

```
INSERT INTO Student (MatNr, Name)
VALUES (MaxNr + 1, "meier");
```

Diese Variante funktioniert leider nicht sicher in einem Netzwerk! Darum existieren Funktionen, die garantieren, dass ein Schlüssel immer nur **einmalig** vergeben wird.

### 10.6.1 Sequenz in der Datenbank Firebird

**Eintragen einer Abteilung ohne Sequenz:**

```
INSERT INTO dept (deptno, Name, location, budget)
VALUES ( 1, 'Marketing', 'Wernigerode', 1000 );
```

**Erzeugen einer Sequenz „seq“:**

```
CREATE SEQUENCE seq;
ALTER SEQUENCE seq RESTART WITH 0
```

**Anlegen einer Abteilung mit einer Sequenz:**

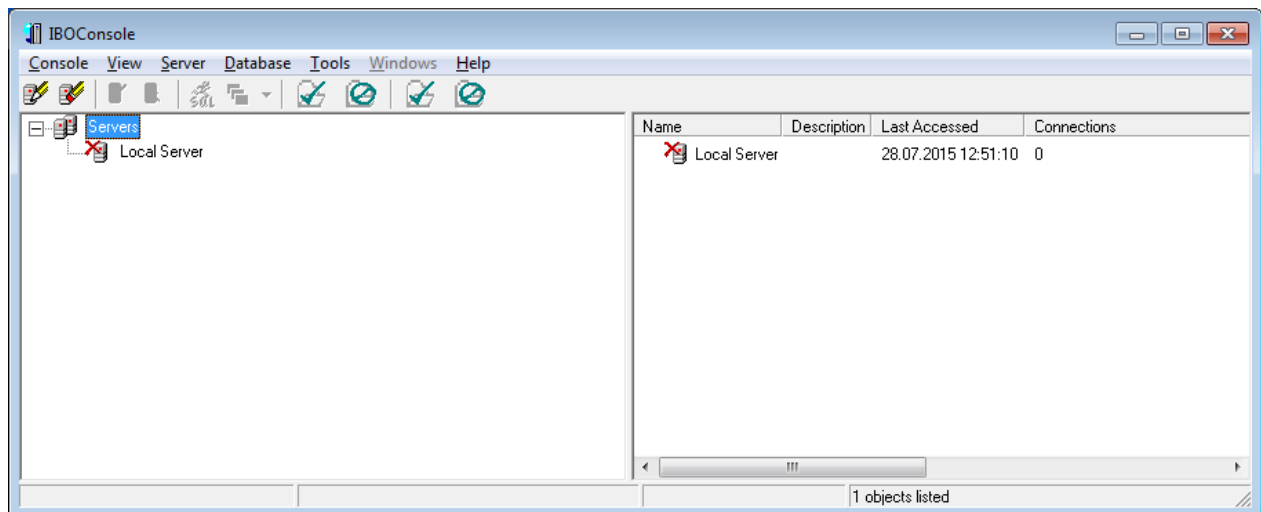
```
INSERT INTO dept (deptno, Name, location, budget)
VALUES (GEN_ID(seq,1), 'Finanzen', 'Wernigerode', 11000 );
```

**Abfrage des aktuellen Wertes:**

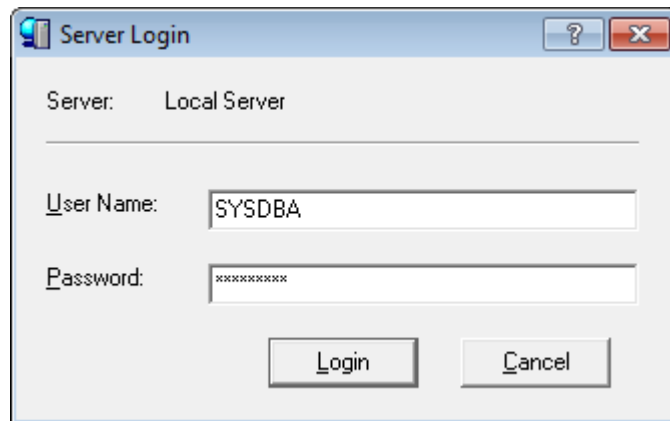
```
select gen_id( seq, 0 ) FROM RDB$DATABASE;
```

## 10.7 Sequenz für die Abteilungsnummer Deptno

Starten der IBOConsole:



Doppel-Klick auf “Local-Server”



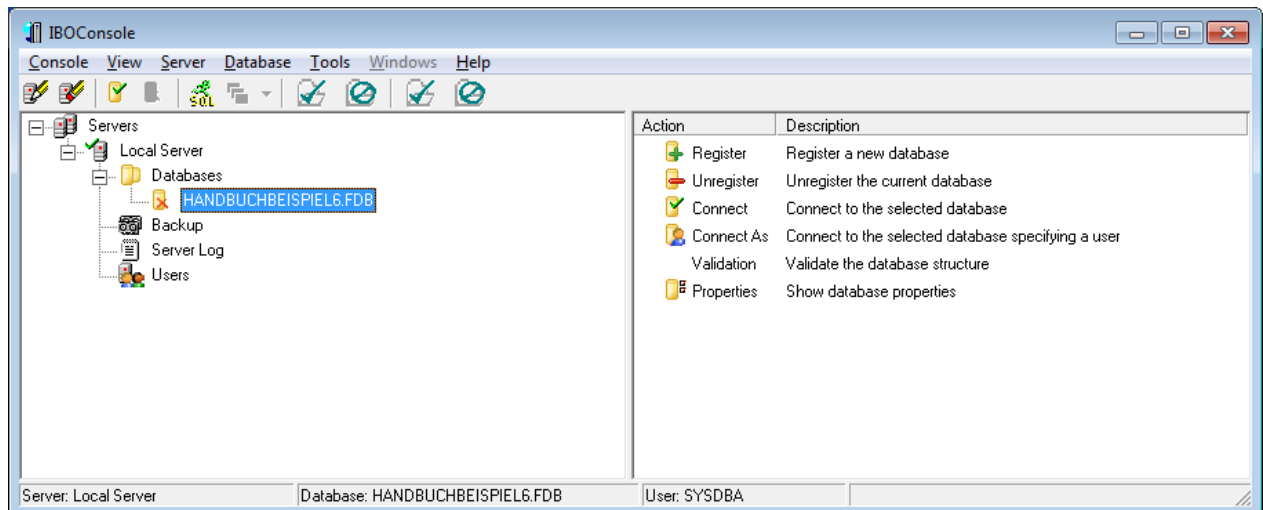
Eingabe:

- SYSDBA
- masterkey

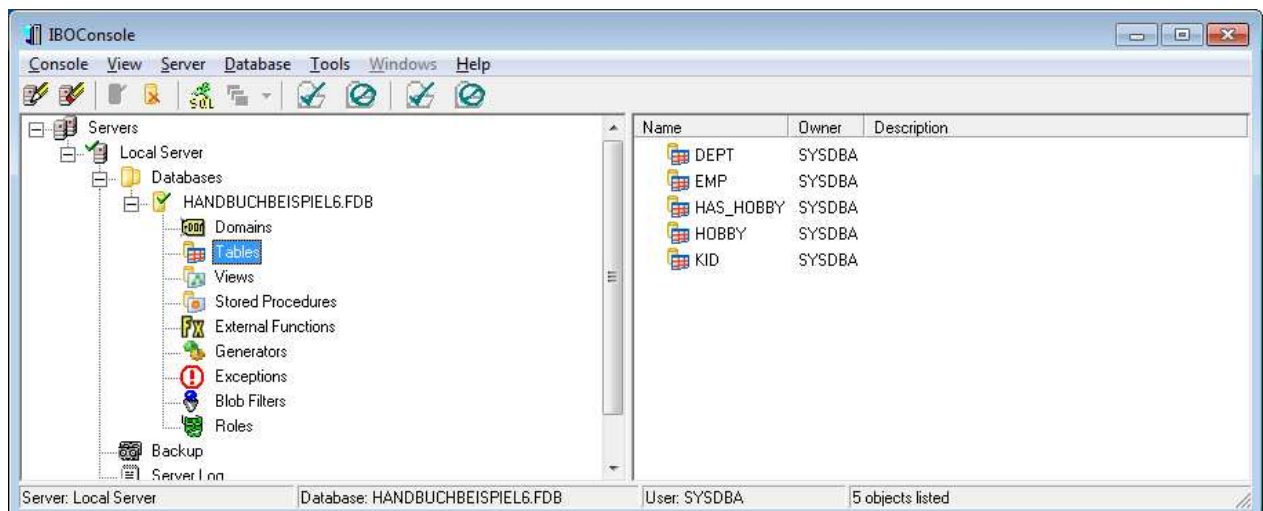
Menü Database, Eintrag „Register“

Eintragen des Pfades und Dateinamens

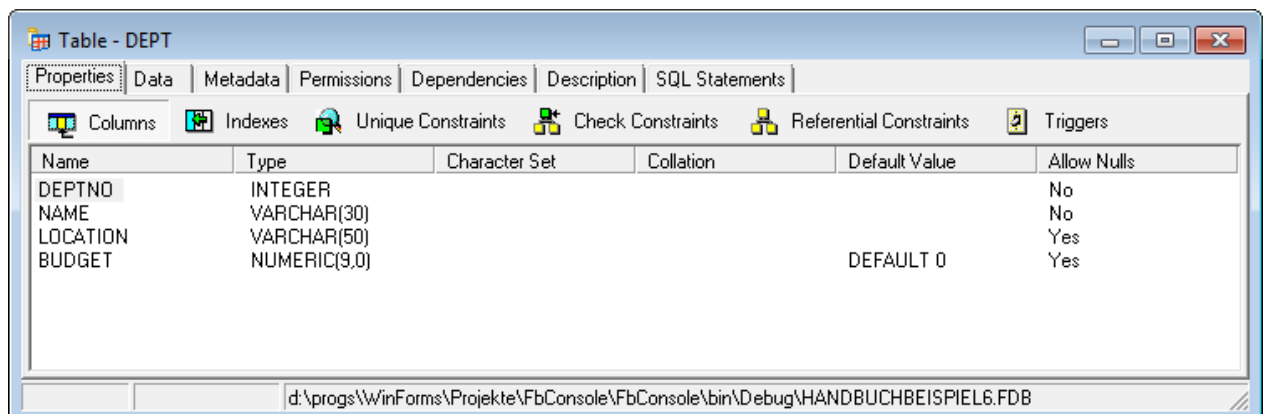




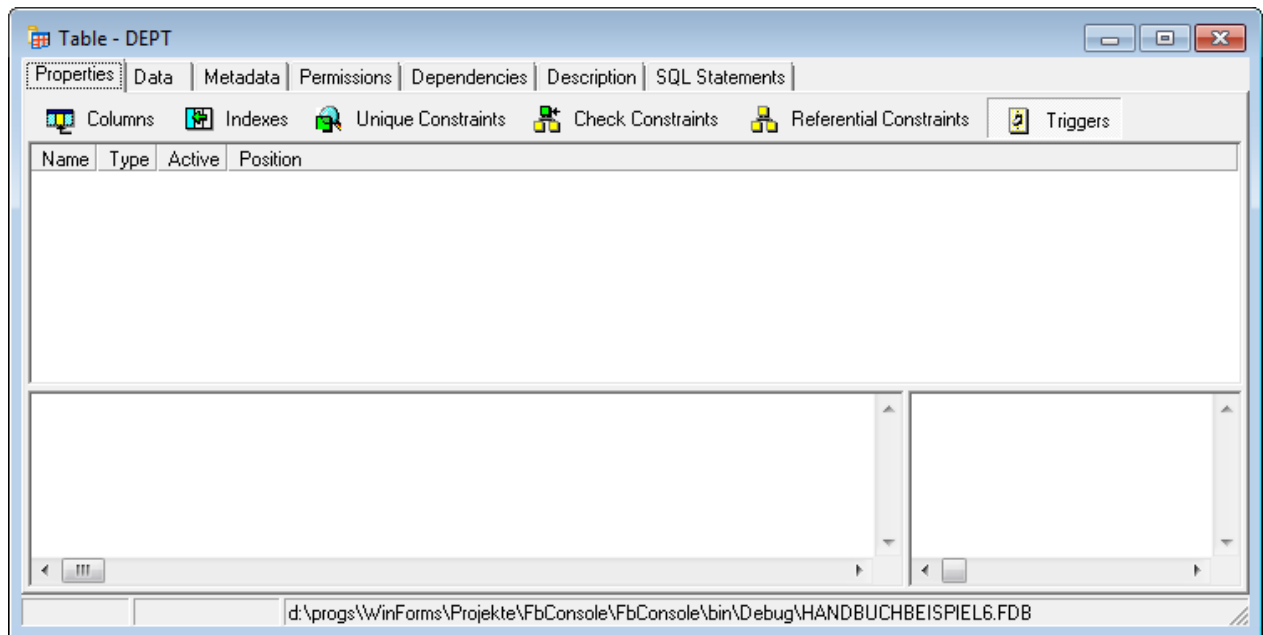
- Doppelklick auf den Eintrag „HANDBUCHBEISPIEL6.FDB“
- Klick auf Tables



Klick auf Tabelle „Dept“



Anklicken des Registers „Triggers“:



Mit der rechten Maustaste den Eintrag „new“ wählen.

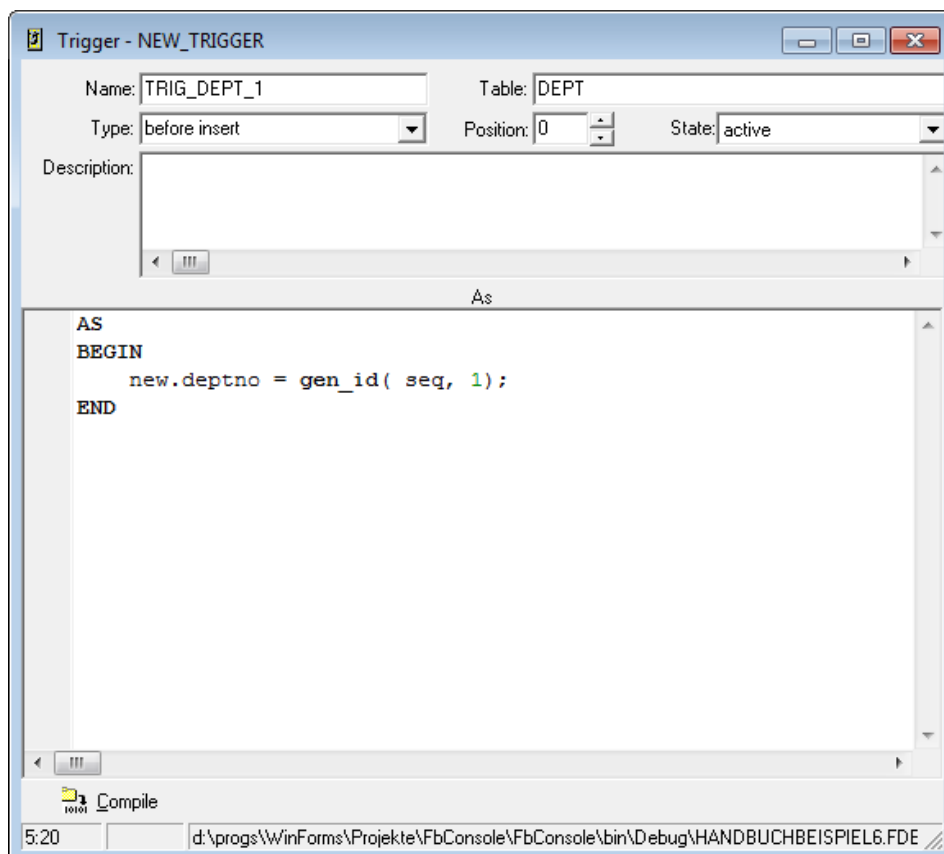


Abbildung 175 Sequenz im Trigger

Als Type wird natürlich „Insert Before“ ausgewählt. Die Position ist hier Null, da es der erste Trigger für diese Tabelle ist. Eingegeben wird der untere Quellcode:

AS

```
BEGIN
    new.deptno = gen_id( seq, 1);
END
```

Der Begriff „new“ beinhaltet den Zugriff auf den neuen Mitarbeiter. Die Sequenz wird über die Funktion „Gen\_id“ mit den beiden Parameter ausgeführt. Die eins zeigt den Sprung um jeweils eine Position.

Danach wird der Quellcode mit dem Schalter „Compile“ übersetzt (links unten).

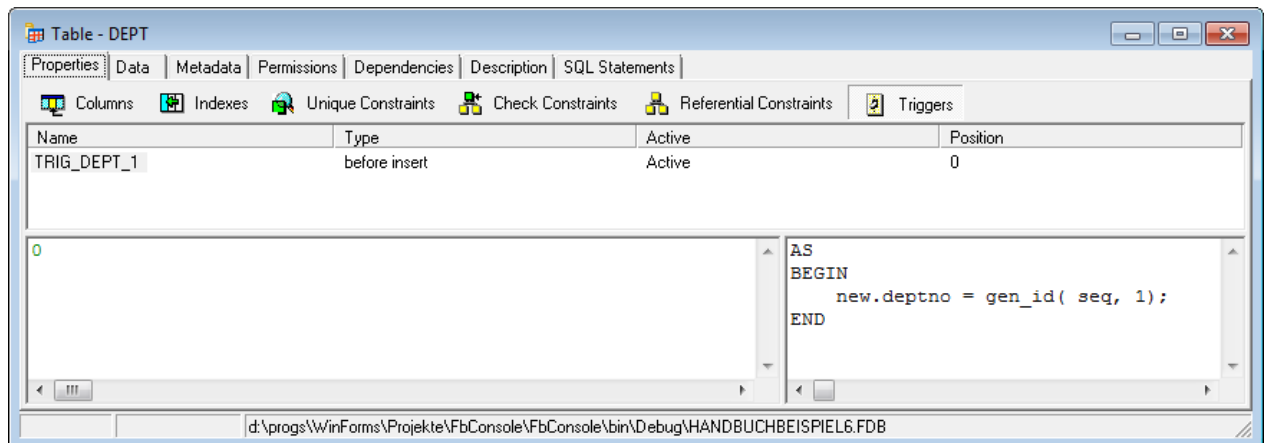
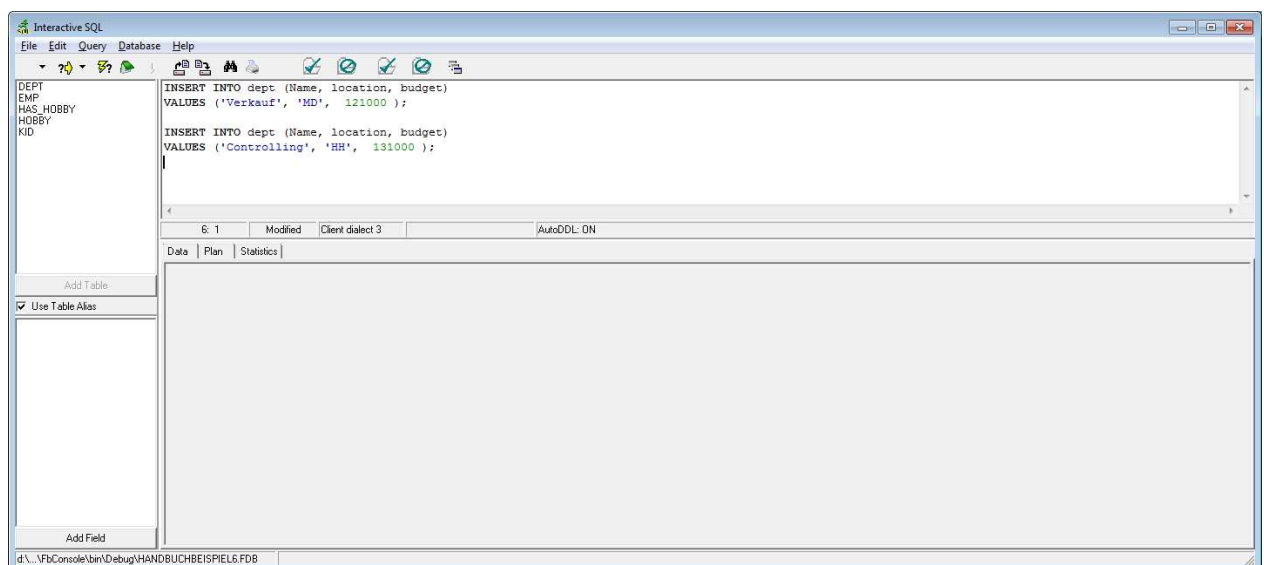


Abbildung 176 Fertiger Trigger

Es folgt der Test mit Insert-Befehlen:

```
INSERT INTO dept (Name, location, budget)
VALUES ('Verkauf', 'MD', 121000 );
```

```
INSERT INTO dept (Name, location, budget)
VALUES ('Controlling', 'HH', 131000 );
```



Ergebnis:

DEPTNO	NAME	LOCATION	BUDGET
1	Marketing	Wernigerode	1000
2	Finanz	Wernigerode	11000
3	Verkauf	MD	121000
4	Controlling	HH	131000

**Abbildung 177** Ergebnis eines Insert-Befehls

## 10.7.1 Erzeugen eines Triggers mittels SQL-Code:

```

set term !! ;
CREATE TRIGGER Dept_BI FOR Dept
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
    if (NEW.Deptno is NULL) then NEW.Deptno = GEN_ID(seq, 1);
END!!
set term ; !!

```

```

INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (2222, 'Bischoff', 11, '39147', 'Erleben',1);

```

```

INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES (3333,'Kardinal', 11, '38844', 'Halberstadt',1);

```

Ergebnis:

```

INSERT INTO EMP ( name, manager, plz, city, deptno)
VALUES ('Bischoff', 11, '39147', 'Erleben',1);

```

```

INSERT INTO EMP (empno, name, manager, plz, city, deptno)
VALUES ('Kardinal', 11, '38844', 'Halberstadt',1);

```

### 10.7.2 Sequenz in Oracle

#### Syntax:

```
CREATE SEQUENCE sequence  
[INCREMENT BY n]  
[START WITH n]  
[{MAXVALUE n | NOMAXVALUE}]  
[{MINVALUE n | NOMINVALUE}]  
[{CYCLE | NOCYCLE}]  
[{CACHE n | NOCACHE}];
```

#### Beispiel:

Anlegen eines Studenten mittels Sequenz

```
CREATE SEQUENCE stud_matrnr  
INCREMENT BY 1  
START WITH 100  
MAXVALUE 100000  
NOCACHE  
NOCYCLE;
```



## 11 Literatur

- [1] Ramez Elmasri, Shamkant B. Navathe: Grundlagen von Datenbanksystemen, ISBN: 3-8273-7153-8
- [2]/Codd-70/ Codd, E. F.:  
A Relational Model of Data for Large Shared Data Banks, Communications of the ACM, Vol 13, No. 6, p. 377-387, 1970
- [3]/Chen-76/ Chen, P. P.:  
The Entity-Relationship Model-Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, p. 9-36, 1976
- [4]/Date-85/Date, C. J.:  
An Introduction to Database System, Vol. 1, Reading Massachusetts, 1985
- [5]/Jack-89/ Jackson, G. A.:  
Entwurf relationaler Datenbanken, Hanser Verlag München, Wien, 1989
- [6]/Voss-91/ Vossen, G., Witt, K.-U.:  
Entwicklungstendenzen bei Datenbanksystemen, Oldenbourg Verlag München, Wien, 1991
- [7]/Wede-81/ Wedekind, H.:  
Datenbanksysteme I, Bibliographisches Institut Mannheim, 1981
- [8]/Trau-91/Trautloft, R., Lindner, U.:  
Datenbanken - Entwurf und Anwendung, Verlag Technik GmbH Berlin, 1991
- [9]/Schi-96/ Schicker, E.:  
Datenbanken und SQL, B. G. Teubner Stuttgart, 1996
- [10] /Sau-95/ Sauer, H.:  
Relationale Datenbanken, Addison-Wesley, 1995
- [11] /Mei-97/ Meier, A., Wüst, Th.:  
Objektorientierte Datenbanken, dpunkt Verlag, 1997
- [12] /Abb-99/ Abbey, Corey, Abramson:  
Oracle 8i für Einsteiger, Hanser Verlag 1999
- [13] /Pon-99/ Ponndorf, St., Matthäus, W.-G.:  
Oracle 8i und Java, Addison-Wesley, 1999
- [14] /Saa-99/ Saake, G., Heuer, A.:  
Datenbanken – Implementierungstechniken, MITP Verlag GmbH, 1999

- 
- [15] /Käh-99/ Kähler, W.-M.:  
Relationales und objektrelationales SQL, Vieweg & Sohn Verlagsgesellschaft mbH, 1999
- [16] /Cul-99/ McCullough-Dieter, C.:  
Oracle8i für Dummies, MITP-Verlag GmbH Bonn, 1999
- [17] /Stü-00/ Stürner, G.:  
Oracle 8i – Der objekt-relationale Datenbank Server, Verlag dbms publishing, 2000
- [18] /Kof-01/ Kofler, M.:  
MySQL, Addison-Wesley, 2001
- [19] /Ric-01/ Riccardi, G.:  
Datenbanksysteme mit Internet und Java-Applikationen, Addison-Wesley, 2001
- [20] /Hoh-02/ Hohenstein, U., Pleßer, V.:  
Oracle 9i, Effiziente Anwendungsentwicklung mit objektrelationalen Konzepten, dpunkt.verlag, 2002
- [21] /Kuh-01/ Kuhlmann, G., Müllmerstadt, F.:  
SQL, Der Schlüssel zu relationalen Datenbanken, Rowohlt Taschenbuch Verlag, 2001

## 12 Indexverzeichnis

### A

Attribute .....	123
Check-Bedingung .....	123
Defaultwert .....	123
Primarykey .....	123
Unique .....	123

### B

Beispiel1	
Beziehung anlegen .....	18
Check-Constraints .....	24
Default-Wert .....	21
Entity anlegen .....	14
Generator .....	28
Neues Projekt .....	13
Sequenz .....	28
Umwandlung eines Modells .....	23
Unique-Werte .....	22
Weitere Schritte .....	24
Beispiel2	
Delete-Befehl .....	90
Entity anlegen .....	34
Insert-Daten Tabelle Belegt .....	89
Insert-Daten Tabelle Student .....	88
Insert-Daten Tabelle Vorlesung .....	88
Neues Projekt .....	33
Update-Befehl .....	89
Beispiel3	
Entity anlegen .....	47, 53
Neues Projekt .....	47
Beispiel4	
Neues Projekt .....	52
Beispiel5 .....	58
Attribute anlegen .....	64
Aufgabenstellung .....	58
Beziehung anlegen .....	64
Beziehung der Vorlesungen .....	69
Entities anlegen .....	60
Generierung einer Datenbank .....	75
Neues Projekt .....	58
Beispiel6 .....	91
Attribute anlegen .....	94
Aufgabenstellung .....	91
Beziehung anlegen .....	95
Beziehung der Hobbies .....	98
Entities anlegen .....	92
Generierung einer Datenbank .....	108
Insert-Daten Tabelle Dept .....	112
Insert-Daten Tabelle Emp .....	114
Neues Projekt .....	91
Self-Relation .....	101
Weak-Beziehung .....	95
Weak-Relation .....	104
Beispiele .....	13

### E

Eigenschaften des Programms .....	9
Eigenschaften in Kurzform .....	9

Einführung .....	9
------------------	---

## F

FBConsole.....	78
Daten eintragen .....	82
Insert into .....	82
Neue Datenbank.....	80
Testabfragen.....	83
Firebird	
Sequenz.....	130
Funktionen der Grafik .....	121
Funktionen des Objekt-Baumes .....	120

## G

Generator.....	86
Größe Datenmengen.....	87

## H

Handbuch Bsp1.DBK.....	13
Handbuch Bsp2_MultiAttribute.DBK.....	33
Handbuch Bsp3 Kunden_Bestellung_Rel_Attrib.DBK.....	46
Handbuch Bsp4 Ternäre Beziehung.DBK.....	52
Hint einer Relation .....	51, 56

## I

Inhalt der ZIP-Datei .....	12
Insert into .....	87
Installation.....	11

## K

Konzeptionelles Modell .....	11, 13
------------------------------	--------

## L

Logisches Modell .....	11, 13
------------------------	--------

## M

Menü Datei.....	116
Menü Einfügen.....	118
Menü Modell.....	118
Menü Optionen.....	119
Menü Projekt.....	116
Menü Schema.....	118
Menüfunktionen .....	116
Multi-Attribut.....	43

## N

Neue Datenbank .....	80
----------------------	----

## O

Oracle	
Sequenz.....	136

**R**

Registry .....	11
----------------	----

**S**

Schwaches Entity .....	95
Self-Relation .....	101
Sequenz .....	29, 86, 129
Spezielle Funktionen .....	125
Computed By .....	125
Datenbank-Typdatei .....	126
Self-Relation .....	126
Ternäre-Relation .....	126
Weak-Attribute .....	126

**W**

Weak-Beziehung .....	95
Weak-Relation .....	104