

**„AccessDBConsole“**

**mit der Datenbank**

**„Access“**

**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>**  
**Link: <http://www.miwilhelm.de/scripte/download/dbs/Datenbank-Access.zip>**  
**Link: <http://www.miwilhelm.de/scripte/download/dbs/Datenbank-Access.7z>**

## Inhaltsverzeichnis

1	Zip-Datei .....	5
1.1	Installation .....	5
1.2	Inhalt der ZIP-Datei .....	5
2	Datenbank-Designer.....	6
1.1	Eigenschaften des Programms .....	6
1.2	Die Eigenschaften in Kurzform: .....	6
1.3	Ablauf bei der Erstellung eines logischen Modells .....	7
2.1	Beispiel eines logischen Modells .....	8
2.1.1	Wichtige Datentypen in Access .....	8
2.1.2	Entity Artikel .....	9
2.1.3	Entity Kunde: .....	14
2.1.4	Entity Bestellung: .....	18
2.1.5	Relationen: .....	21
2.1.6	Beziehung: Artikel zu Bestellung .....	24
2.2	Erstellen der SQL-Skripte .....	27
2.3	Ergebnis: .....	28
3	AccessDBConsole.....	30
3.1	Neue Datenbank erstellen .....	31
3.2	Daten eintragen .....	33
3.2.1	SQL-Befehle zum Eintragen: .....	33
3.3	Testabfragen .....	34
4	Anhang .....	36
4.1	AutoWert .....	36
4.1.1	Quellcode des zweiten Beispiels .....	38
4.1.2	Quellcode der SQL-Befehle: .....	38
4.2	SQL-Einfügedaten .....	39
4.3	Ja/Nein .....	41
5	Funktionen .....	45
5.1	String-Manipulation .....	45
5.2	Numerische Funktion .....	46
5.3	Datum / ZeitFunktionen .....	47
5.4	Allgemeine Funktionen .....	50
5.5	Group Funktionen .....	50
6	Indexverzeichnis .....	51

## Abbildungsverzeichnis

Abbildung 1 Überblick des Programms	6
Abbildung 2 Auswahl der Datenbank	8
Abbildung 3 Neues Entity "Artikel"	9
Abbildung 4 PK-Constraint „PK_Artikel“ einfügen	9
Abbildung 5 Zweites Register mit dem Attributen	9
Abbildung 6 Primary-Key eintragen	10
Abbildung 7 Bezeichnung einfügen	10
Abbildung 8 Default-Wert eintragen (ohne Hochkommas)	11
Abbildung 9 Attribut Gewicht	11
Abbildung 10 Defaultwert	12
Abbildung 11 Check-Bedingung	12
Abbildung 12 Vollständiges Entity Artikel	13
Abbildung 13 Darstellung im Hauptfenster	13
Abbildung 14 Attribut Kunde	14
Abbildung 15 Attribut Nachname	14
Abbildung 16 Attribut Nachname mit dem Defaultwert	15
Abbildung 17 Attribut "PLZ"	16
Abbildung 18 Attribut "PLZ" mit dem Defaultwert	16
Abbildung 19 Entity Kunde	17
Abbildung 20 Testdatenbank mit zwei Entities	17
Abbildung 21 Primarykey "Bestnr"	18
Abbildung 22 Attribut "Anz"	18
Abbildung 23 Attribut "Anz" mit dem Defaultwert	19
Abbildung 24 Attribut "Anz" mit der Check-Bedingung	19
Abbildung 25 Attribut "Kundenr" für die Beziehung	20
Abbildung 26 Attribut "ArtIndex" für die Beziehung	20
Abbildung 27 Entity Bestellung	21
Abbildung 28 Aktueller Stand der drei Entities	21
Abbildung 29 Relation definieren	22
Abbildung 30 Hauptfenster der Relationen	22
Abbildung 31 Frage nach dem Eintragen	22
Abbildung 32 Dialog für das Eingeben der Beziehung	23
Abbildung 33 Kundenr wird eingefügt	23
Abbildung 34 Hauptfenster mit der ersten Relation	24
Abbildung 35 Zweite Relation	24
Abbildung 36 Zweite Relation erstellt	25
Abbildung 37 Hauptfenster mit der ersten und zweiten Relation	26
Abbildung 38 Erzeugen der SQL-Skripte	27
Abbildung 39 Liste der Tabellen	28
Abbildung 40 Hauptfenster der AccessConsole	30
Abbildung 41 Erstellen der Datenbank "Beispiel1.mdb"	31
Abbildung 42 Neue datenbank mit leerer Entity-Liste	31
Abbildung 43 Eintragen der SQL-Skripte	32
Abbildung 44 Anzeige der Datenbank-Tabellen	32
Abbildung 45 Daten mittels „Insert-Into“ eintragen	33
Abbildung 46 Abfrage bezüglich der Artikel	34
Abbildung 47 Abfrage bezüglich der Kunden	34
Abbildung 48 Abfrage bezüglich der Bestellung	35
Abbildung 49 Abfrage bezüglich der Bestellungen	35
Abbildung 50 Das Entity Artikel mit einem Counter als Primary-Key	36
Abbildung 51 Das Entity Bestellung mit einem Counter als Primary-Key	37
Abbildung 52 Das Entity Kunde mit einem Counter als Primary-Key	37
Abbildung 53 Die Entities des zweiten Beispiels	38
Abbildung 54 IstPremiumKunde	41

---

Abbildung 55	Kunden-Entity mit einem Logical-Attribut	41
Abbildung 56	Anzeige des dritten Beispiels	42
Abbildung 57	Abfrage "SELECT * FROM KUNDE"	43
Abbildung 58	Anzeige der Tabelle "Kunde" nach dem Setzen „IstPremiumKunde“	44

# 1 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.

## 1.1 Installation

- Entzippen der Datei in einen beliebigen Ordner

## 1.2 Inhalt der ZIP-Datei

### Ordner Designer

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

### Ordner Access

- Datenbank\_Employee.pdf
- Tabelle der Firebird-Beispieldatenbank
  - Employee.mdb
  - Firma.mdb
- Access-Datenbank-Konsole, nur Programm
  - AccessDBDesigner.exe

## 2 Datenbank-Designer

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 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 grafisch dargestellt.

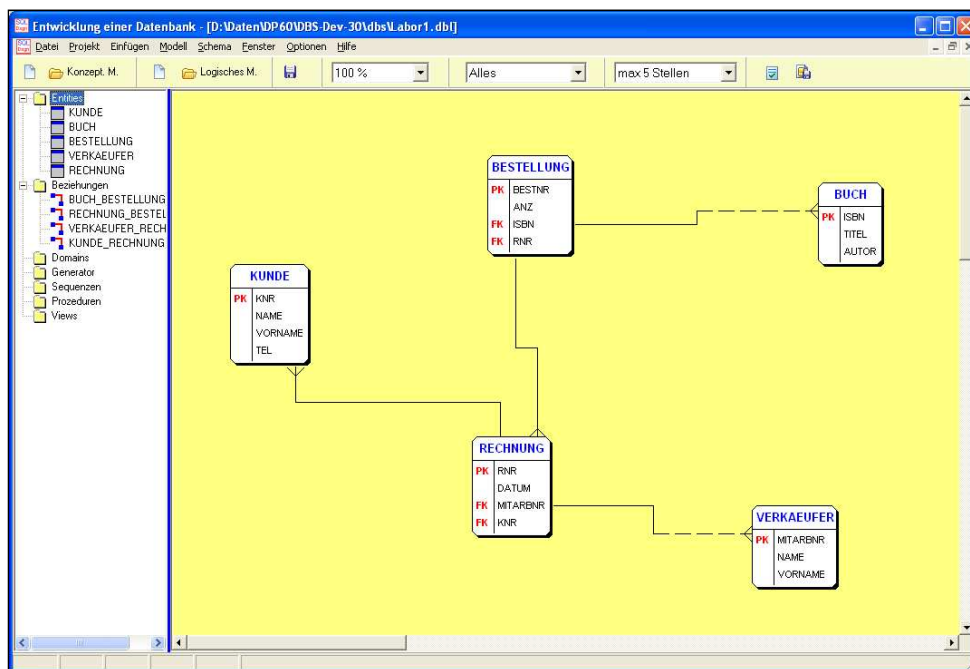


Abbildung 1 Überblick des Programms

### 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-Scriptes
- Vielfältige Farb- und Schriftwahl
- Anzeige der Relationen mit Krähenfuss oder Punktnotation
- Export der Daten nach Winword
- Export der ERM-Grafik in die Zwischenablage
- 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

#### Konzeptionellen 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 nun auch 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

#### Logische Modell:

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

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

- **Erzeugen eines neuen Fensters (STRG+N)**, nur wenn es kein konzeptionelles Modell gab.
  - Datenbank: Firebird, Version 2,5
- 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)**

## 2.1 Beispiel eines logischen Modells

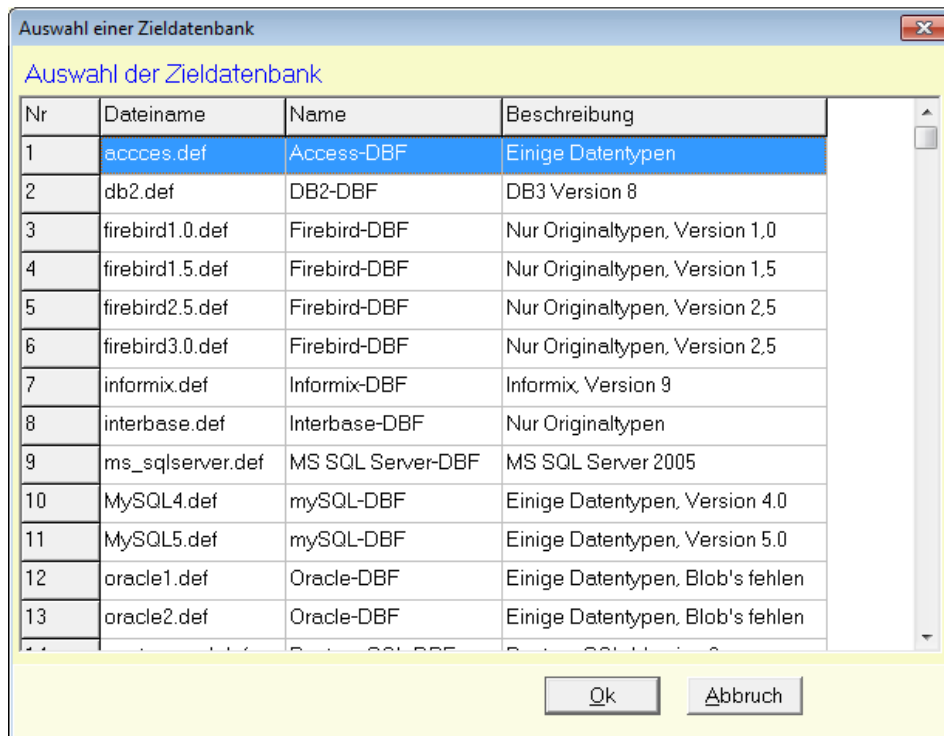


Abbildung 2 Auswahl der Datenbank

### Hinweis:

**Bitte wählen Sie für dieses Programmpaket immer die Version „Access“ aus.**

### 2.1.1 Wichtige Datentypen in Access

Datentyp	Bereich
BIT	Yes/No
Logical	Yes/No
TinyInt	0 bis 255
Money	– 922.337.203.685.477,5808 -> 922.337.203.685.477,5807
DateTime	Datum- und Uhrzeit
UNIQUEIDENTIFIER	eindeutige Identifikationsnummer
Real	4 Byte, entspricht einem Float bzw. Single
Float	8 Byte, entspricht einem Double
Smallint	-32787 bis +32767
Integer	-2.147.483.648 und 2.147.483.647
Decimal/Numeric	17 bytes
Text	0 bis 2,14 GB
Character	0 bis 255 Zeichen
VarChar	0 bis 8000 Zeichen
Counter	Autoincrement-Zähler



## 2.1.2 Entity Artikel

Neues Entity „Artikel“ mit Strg+E

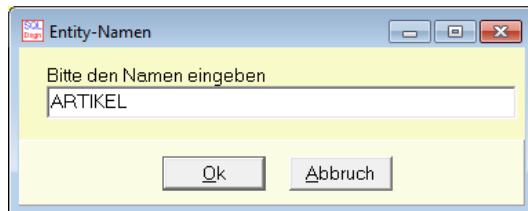


Abbildung 3 Neues Entity "Artikel"

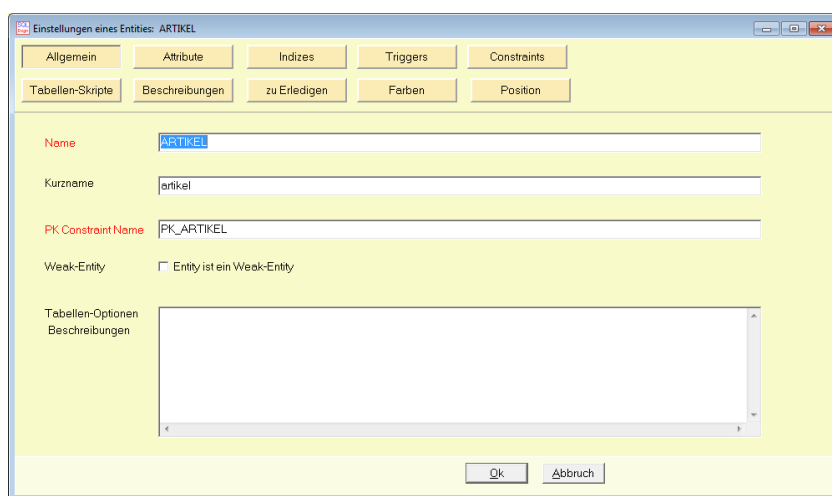


Abbildung 4 PK-Constraint „PK\_Artikel“ einfügen

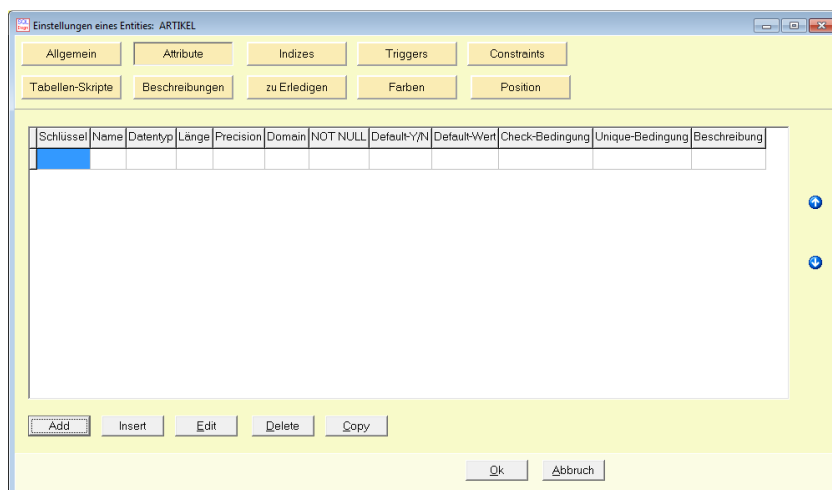


Abbildung 5 Zweites Register mit dem Attributen

**Attribute einfügen mit dem Schalter „Add“:**

Einstellung eines Attributes: Attribut1

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

Name: ARTINDEX

☒ Primärschlüssel ☒ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain: keine Domain [Neue Domain]

Datentyp: INTEGER [1] [0]

Formel: [ ]

Datentyp: kein Datentyp [1] [0]

**Autonumber**

☐ aktiviert Start: 0 Increment: 1

Ok Abbruch

**Abbildung 6 Primary-Key eintragen**

Einstellung eines Attributes: Attribut2

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

Name: BEZ

☐ 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 7 Bezeichnung einfügen**

The screenshot shows the 'Einstellung eines Attributes: Attribut2' dialog box with the 'Default/Unique' tab selected. The 'DefaultWert: (VARCHAR)' section has the 'Defaultwert' checkbox checked, and the 'Wert' field is empty. A red note below states: 'Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)'. The 'Unique-Bedingungen' section has the 'Unique' checkbox unchecked, and the 'Check-C. Name' field is empty. At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 8 Default-Wert eintragen (ohne Hochkommas)

### 3. Attribut: Gewicht

The screenshot shows the 'Einstellung eines Attributes: Attribut3' dialog box with the 'Default/Unique' tab selected. The 'Name' field contains 'GEWICHT'. Below it are checkboxes for 'Primärschlüssel', 'Not Null', and 'Fremdschlüssel', all of which are unchecked. The 'Datentyp-Definition' section has a 'Domain' dropdown set to 'keine Domain', a 'Datentyp' dropdown set to 'FLOAT', and two spinners for '1' and '0'. A 'Formel' field is empty. Below this is another 'Datentyp' dropdown set to 'kein Datentyp' with '1' and '0' spinners. A 'Neue Domain' button is to the right. The 'Autonumber' section has the 'aktiviert' checkbox unchecked, and 'Start' and 'Increment' spinners set to '0' and '1' respectively. At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 9 Attribut Gewicht

The screenshot shows a dialog box titled 'Einstellung eines Attributes: Attribut3'. 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 are two sections. The first section is 'DefaultWert: (FLOAT)' and contains a checked checkbox labeled 'Defaultwert', a text input field with the value '0', and a red warning text: 'Datum, Zeit, Boolean, Currency werden als String gespeichert (Ohne Prüfung)'. The second section is 'Unique-Bedingungen' and contains an unchecked checkbox labeled 'Unique' and a text input field labeled 'Check-C. Name' which is currently empty. At the bottom of the dialog are 'Ok' and 'Abbruch' buttons.

**Abbildung 10 Defaultwert**

The screenshot shows the same dialog box 'Einstellung eines Attributes: Attribut3', but with the 'Checks' tab selected. The 'Check-Bedingungen' section is active, showing a checked checkbox labeled 'Check.Attribute'. Below it, the 'Check-C. Name' text input field contains the value 'CH\_GEWICHT'. Further down, there is a text area labeled 'Check-Bedingung' containing the SQL check constraint code: 

```
CHECK (
GEWICHT >= 0
)
```

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

**Abbildung 11 Check-Bedingung****Ergebnis:**

Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung
1 PK	ARTINDEX	INTEGER				NOT NULL				-
2	BEZ	VARCHAR	50				x			-
3	GEWICHT	FLOAT					x	0	CH_GEWICHT	-

Abbildung 12 Vollständiges Entity Artikel



Abbildung 13 Darstellung im Hauptfenster

### 2.1.3 Entity Kunde:

The screenshot shows the 'Einstellung eines Attributes: Attribut1' dialog box. The 'Allgemein' tab is selected. The 'Name' field contains 'KUNDENNR'. The 'Primärschlüssel' checkbox is checked, while 'Not Null' and 'Fremdschlüssel' are unchecked. The 'Datentyp-Definition' section shows 'Domain' set to 'keine Domain', 'Datentyp' set to 'INTEGER', and 'Formel' set to an empty field. The 'Autonumber' section shows 'aktiviert' unchecked, 'Start' set to 0, and 'Increment' set to 1. The 'Ok' and 'Abbruch' buttons are at the bottom.

Abbildung 14 Attribute Kunde

The screenshot shows the 'Einstellung eines Attributes: Attribut2' dialog box. The 'Allgemein' tab is selected. The 'Name' field contains 'NACHNAME'. The 'Primärschlüssel', 'Not Null', and 'Fremdschlüssel' checkboxes are all unchecked. The 'Datentyp-Definition' section shows 'Domain' set to 'keine Domain', 'Datentyp' set to 'VARCHAR' with a length of 50, and 'Formel' set to an empty field. The 'Autonumber' section shows 'aktiviert' unchecked, 'Start' set to 0, and 'Increment' set to 1. The 'Ok' and 'Abbruch' buttons are at the bottom.

Abbildung 15 Attribut Nachname

The screenshot shows a dialog box titled 'Einstellung eines Attributes: Attribut2'. It has several tabs: 'Allgemein', 'Default / Unique' (selected), 'Checks', 'Linked-Attribute', 'Beschreibung', 'Zu Erledigen', and 'ObjectID'. The 'Default / Unique' tab contains two sections. The first section, 'DefaultWert: (CHAR)', has a checked checkbox labeled 'Defaultwert' 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 of the dialog are 'Ok' and 'Abbruch' buttons.

**Abbildung 16** Attribut Nachname mit dem Defaultwert

Die nächsten Attribute können mit dem Schalter „Copy“ kopiert werden.

- VORNAME
- STRASSE
- ORT

Anklicken des Attributs „Nachname“

Schalter „Copy“ anklicken

Die Postleitzahl muss als Character gewählt werden, da Integer die führende Null entfernt.

Einstellung eines Attributes: Attribut6

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

Name

☐ Primärschlüssel ☐ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain

Datentyp

Formel

Datentyp

**Autonumber**

☐ aktiviert Start  Increment

Abbildung 17 Attribut "PLZ"

Einstellung eines Attributes: Attribut6

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

Abbildung 18 Attribut "PLZ" mit dem Defaultwert



	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Beschreib
1	PK	KUNDENNR	INTEGER				NOT NULL				-	
2		NACHNAME	VARCHAR	50				x			-	
3		VORNAME	VARCHAR	50				x			-	
4		STRASSE	VARCHAR	50				x			-	
5		ORT	VARCHAR	50				x			-	
6		PLZ	CHAR	5				x			-	

Abbildung 19 Entity Kunde

Name hat den Defaultwert "  
 Vorname hat den Defaultwert "  
 Strasse hat den Defaultwert "  
 Ort hat den Defaultwert "

Aktueller Stand der Datenbank:

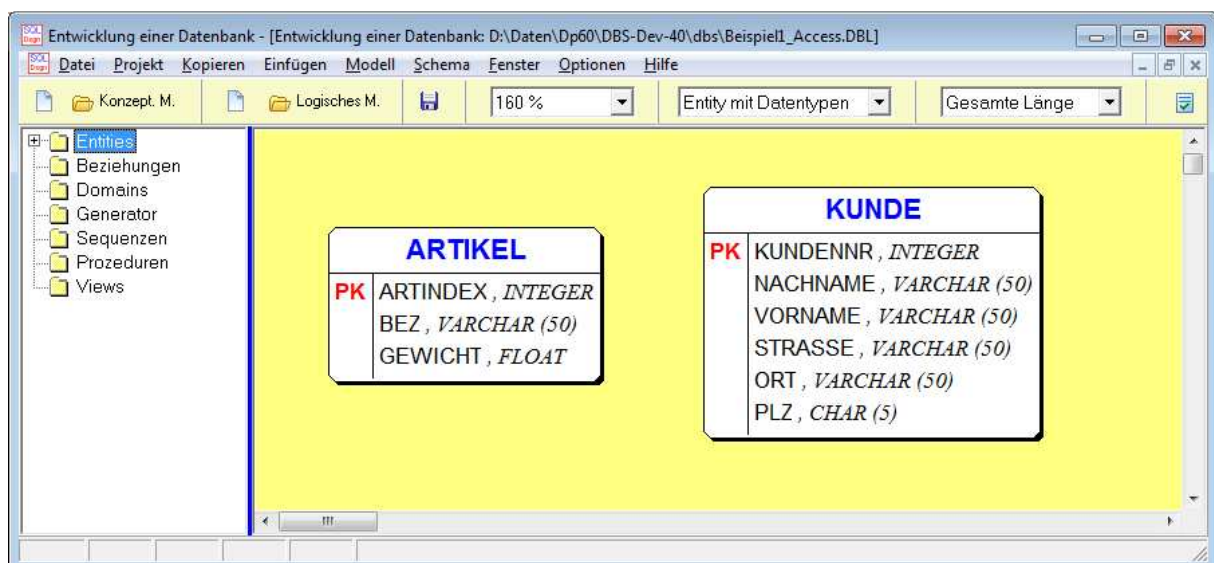


Abbildung 20 Testdatenbank mit zwei Entities

### 2.1.4 Entity Bestellung:

Einstellung eines Attributes: Attribut1

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

Name: BESTNR

☒ Primärschlüssel ☒ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain: keine Domain Neue Domain

Datentyp: INTEGER 1 0

Formel:

Datentyp: kein Datentyp 1 0

**Autonumber**

☐ aktiviert Start: 0 Increment: 1

Ok Abbruch

Abbildung 21 Primarykey "Bestnr"

Einstellung eines Attributes: Attribut2

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

Name: ANZ

☐ Primärschlüssel ☐ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain: keine Domain Neue Domain

Datentyp: INTEGER 1 0

Formel:

Datentyp: kein Datentyp 1 0

**Autonumber**

☐ aktiviert Start: 0 Increment: 1

Ok Abbruch

Abbildung 22 Attribut "Anz"

The screenshot shows a dialog box titled 'Einstellung eines Attributes: Attribut2'. It has several tabs: 'Allgemein', 'Default / Unique' (selected), 'Checks', 'Linked-Attribute', 'Beschreibung', 'Zu Erledigen', and 'Object-ID'. In the 'Default / Unique' tab, there is a section 'DefaultWert: (INTEGER)' with a checked checkbox 'Defaultwert'. Below it is a text input field labeled 'Wert' containing the number '0'. A red note below the 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 labeled 'Check-C. Name' which is empty. At the bottom are 'Ok' and 'Abbruch' buttons.

Abbildung 23 Attribut "Anz" mit dem Defaultwert

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

```
CHECK(  
  ANZ >= 0  
)
```

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

Abbildung 24 Attribut "Anz" mit der Check-Bedingung

Für die Beziehung wird nun noch die Kundennr eingefügt.

Einstellung eines Attributes: Attribut3

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

Name: KUNDENNR

☐ Primärschlüssel ☐ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain: keine Domain Neue Domain

Datentyp: INTEGER 1 0

Formel:

Datentyp: kein Datentyp 1 0

**Autonumber**

☐ aktiviert Start: 0 Increment: 1

Ok Abbruch

**Abbildung 25 Attribut "Kundenr" für die Beziehung**

Einstellung eines Attributes: Attribut4

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

Name: ARTINDEX

☐ Primärschlüssel ☐ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain: keine Domain Neue Domain

Datentyp: INTEGER 1 0

Formel:

Datentyp: kein Datentyp 1 0

**Autonumber**

☐ aktiviert Start: 0 Increment: 1

Ok Abbruch

**Abbildung 26 Attribut "ArtIndex" für die Beziehung**

Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Beschreibung
1 PK	BESTNR	INTEGER				NOT NULL					
2	ANZ	INTEGER					x	0	CH_ANZ		
3	KUNDENNR	INTEGER									
4	ARTINDEX	INTEGER									

Abbildung 27 Entity Bestellung

Anz hat den Defaultwert 0

Anz hat die Check-Bedingung  $anz \geq 0$

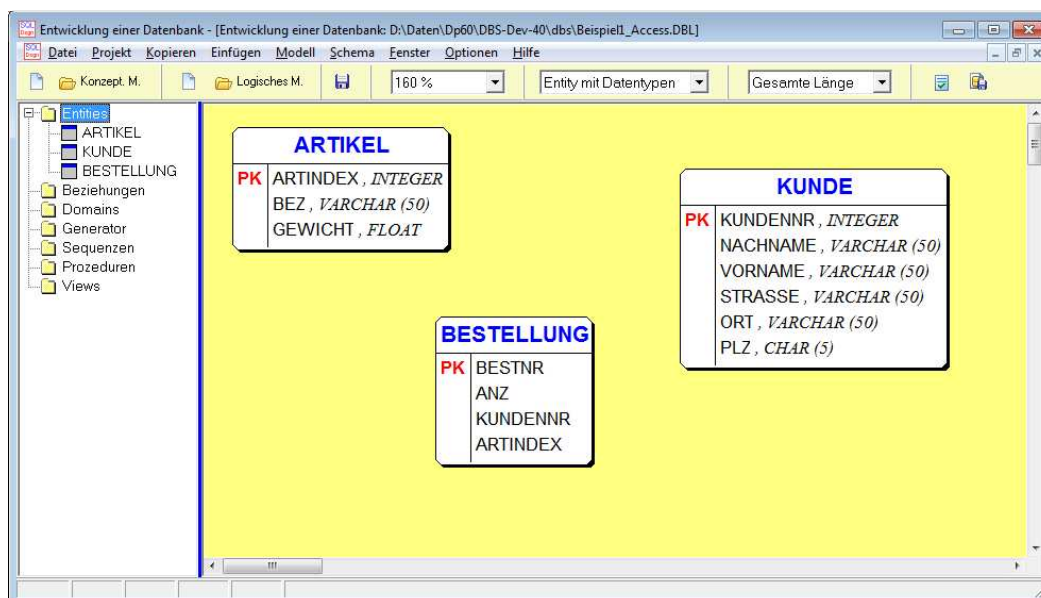


Abbildung 28 Aktueller Stand der drei Entities

### 2.1.5 Relationen:

1. Beziehung: Kunde zu Bestellung

Taste: Strg+R

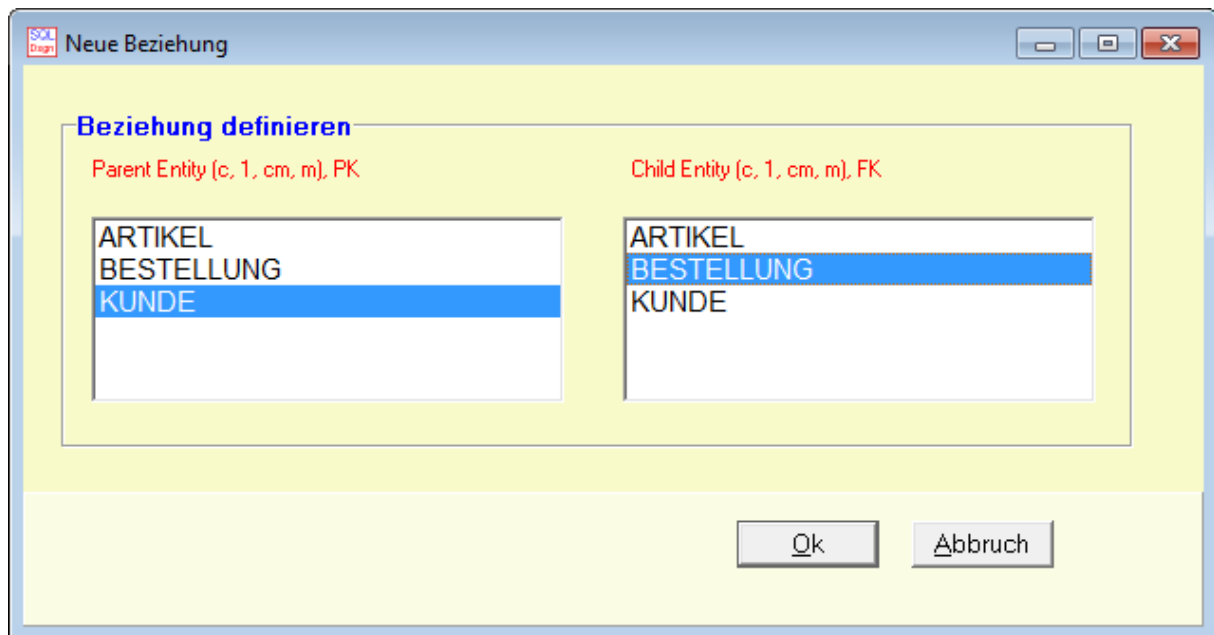


Abbildung 29 Relation definieren

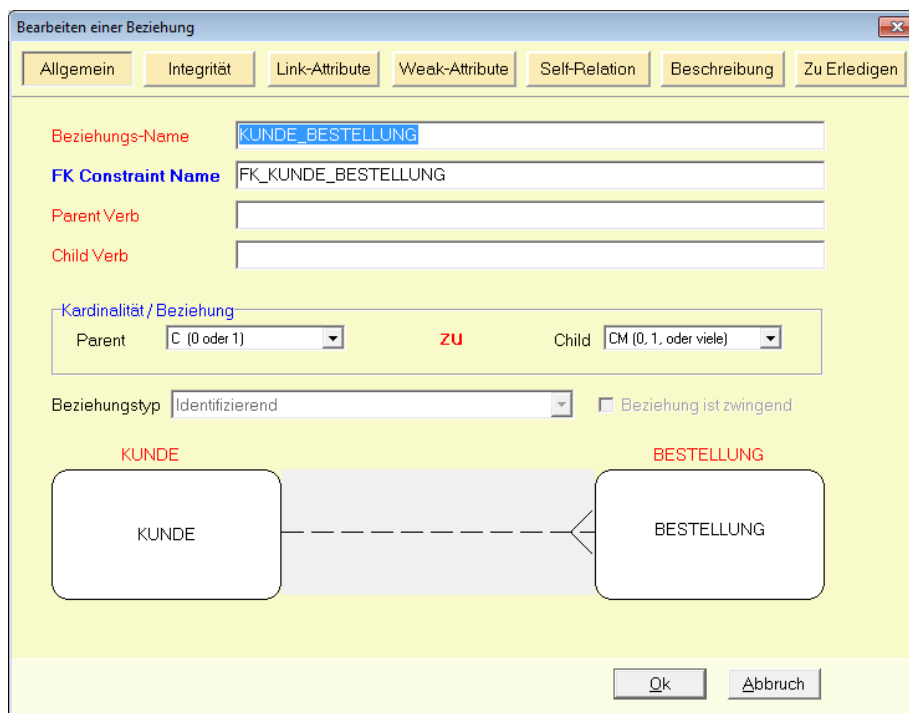


Abbildung 30 Hauptfenster der Relationen

Nun umschalten zum Register „Link-Attribute“

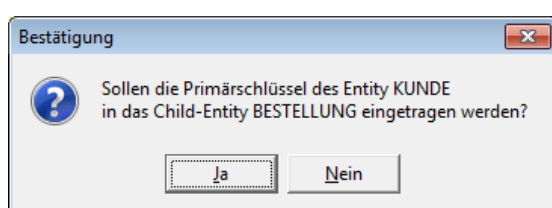


Abbildung 31 Frage nach dem Eintragen

Bitte hier mit Nein antworten.

- Nun in der linken Tabelle das Attribut „Kundennr“ anklicken.
- Nun in der rechten Tabelle das Attribut „Kundennr“ anklicken.
- Danach den Schalter „Connect“ betätigen.
- Damit wird die Verbindung eingetragen.

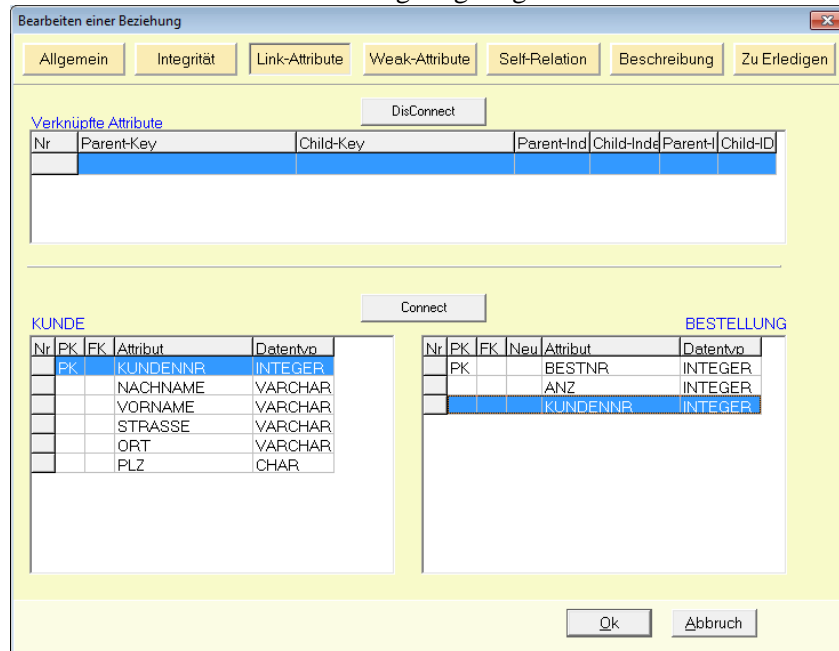


Abbildung 32 Dialog für das Eingeben der Beziehung

nach dem Schalter „Connect“

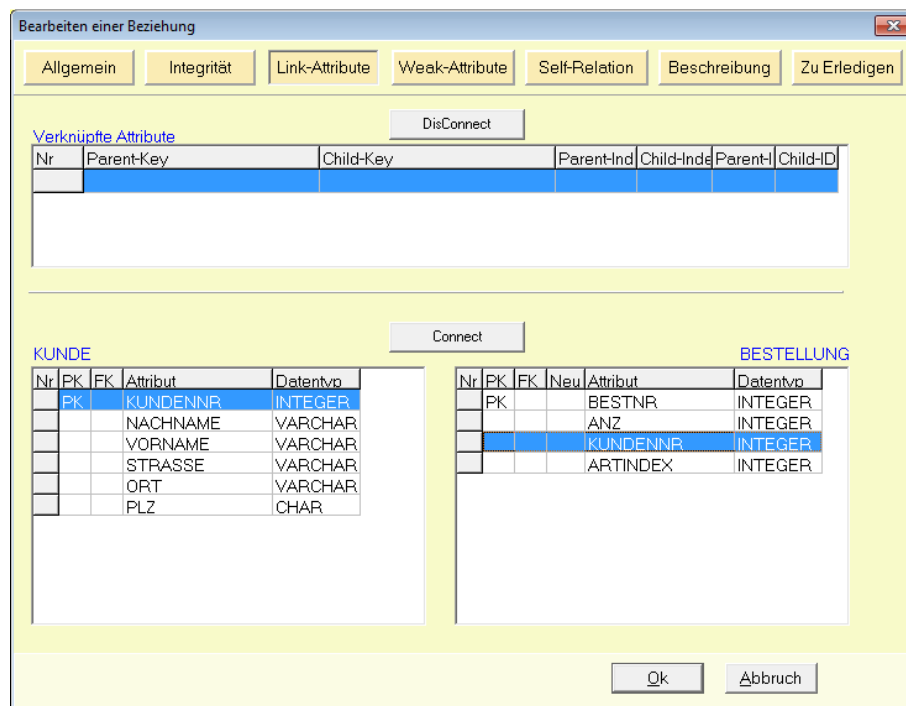


Abbildung 33 Kundenr wird eingefügt

Ergebnis im Hauptfenster:

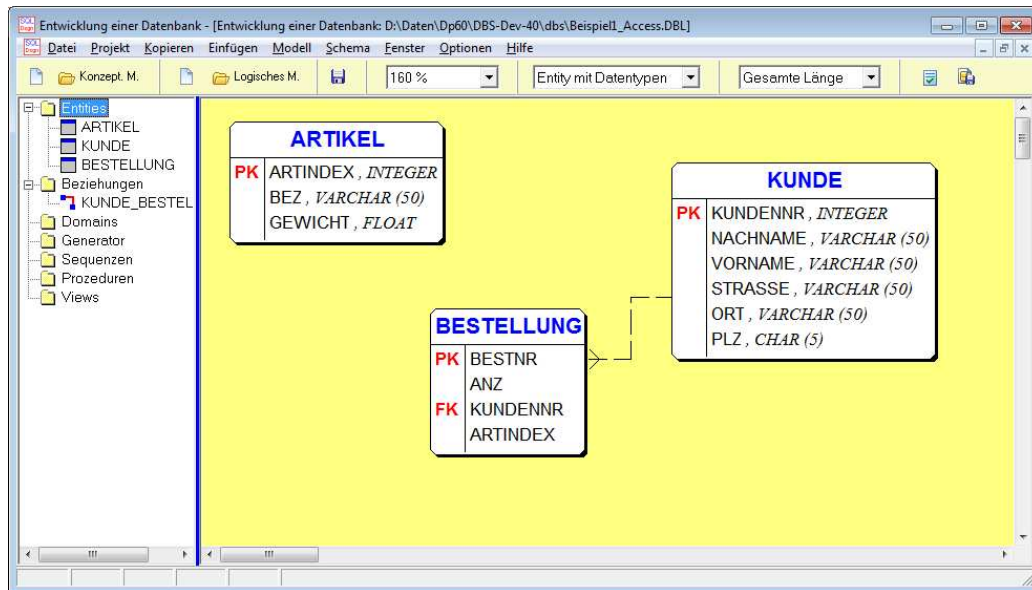


Abbildung 34 Hauptfenster mit der ersten Relation

### 2.1.6 Beziehung: Artikel zu Bestellung

Es kann pro Eintrag nur ein Artikel bestellt werden, aber mit beliebiger Anzahl!

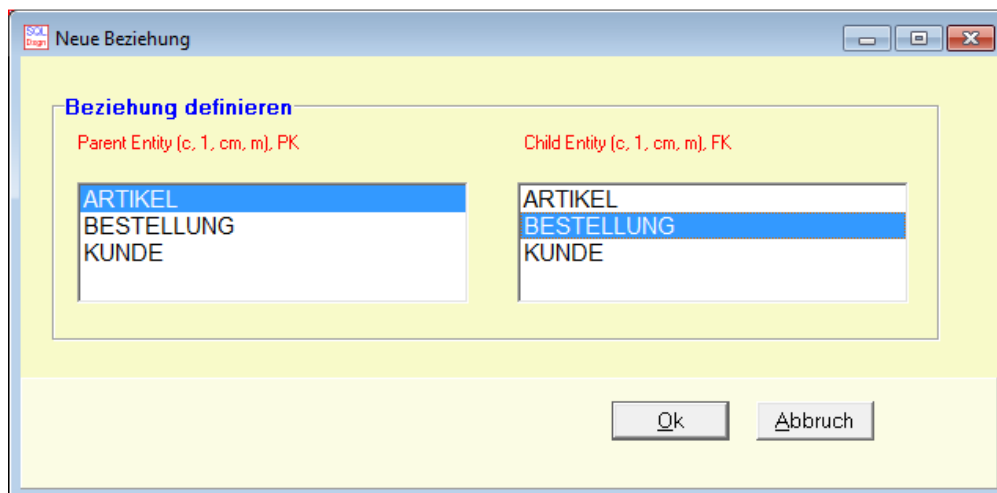


Abbildung 35 Zweite Relation



Bearbeiten einer Beziehung

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

DisConnect

Verknüpfte Attribute

Nr	Parent-Key	Child-Key	Parent-Ind	Child-Ind	Parent-ID	Child-ID

Connect

ARTIKEL

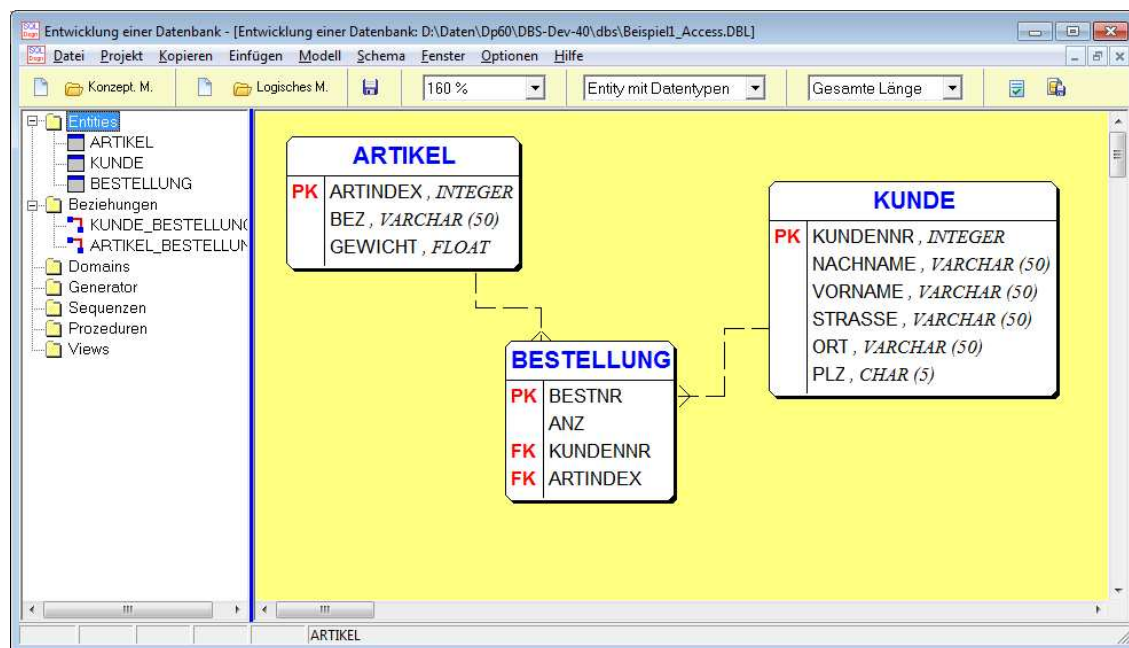
Nr	PK	FK	Attribut	Datentyp
	PK		ARTINDEX	INTEGER
			BEZ	VARCHAR
			GEWICHT	FLOAT

BESTELLUNG

Nr	PK	FK	Neu Attribut	Datentyp
	PK		BESTNR	INTEGER
			ANZ	INTEGER
	FK		KUNDENNR	INTEGER
			ARTINDEX	INTEGER

Ok Abbruch

Abbildung 36 Zweite Relation erstellt

**Ergebnis:****Abbildung 37** Hauptfenster mit der ersten und zweiten Relation

Die Datenbank ist nun fertig und die Create-Skripte können erstellt werden.

## 2.2 Erstellen der SQL-Skripte

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 untere Abbildung 38). 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.

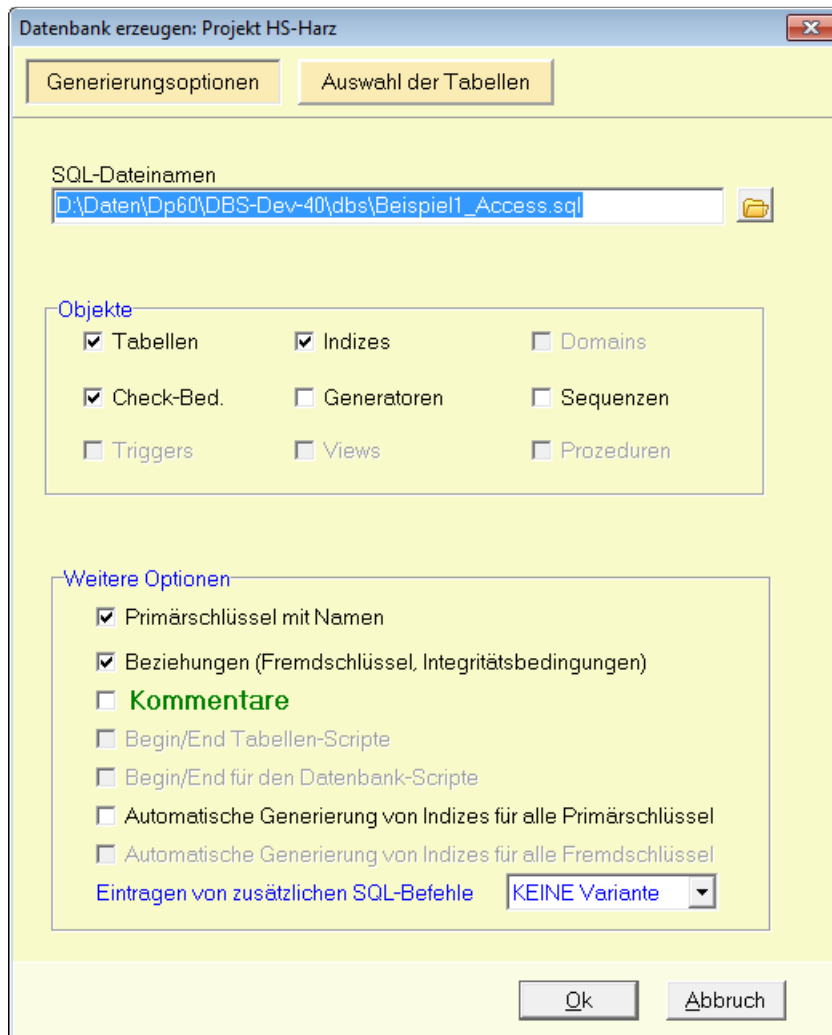


Abbildung 38 Erzeugen der SQL-Skripte

Im zweiten Register kann man nun einzelne Tabellen ausschließen.

Mit dem Schalter „Ok“ werden die SQL-Anweisungen erstellt.

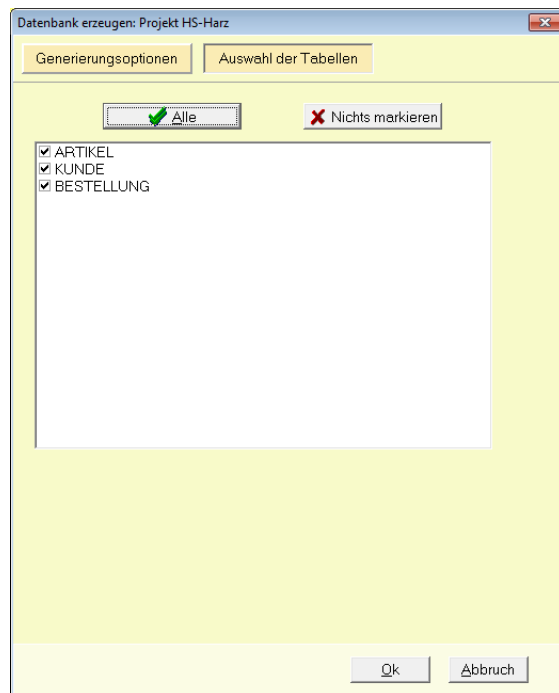


Abbildung 39 Liste der Tabellen

## 2.3 Ergebnis:

```
CREATE TABLE ARTIKEL (  
    ARTINDEX INTEGER NOT NULL,  
    BEZ VARCHAR(50) DEFAULT '',  
    GEWICHT FLOAT DEFAULT 0,  
  
    CONSTRAINT PK_ARTIKEL PRIMARY KEY (ARTINDEX)  
);  
  
CREATE TABLE KUNDE (  
    KUNDENNR INTEGER NOT NULL,  
    NACHNAME VARCHAR(50) DEFAULT '',  
    VORNAME VARCHAR(50) DEFAULT '',  
    STRASSE VARCHAR(50) DEFAULT '',  
    ORT VARCHAR(50) DEFAULT '',  
    PLZ CHAR(5) DEFAULT '',  
  
    CONSTRAINT PK_KUNDE PRIMARY KEY (KUNDENNR)  
);  
  
CREATE TABLE BESTELLUNG (  
    BESTNR INTEGER NOT NULL,  
    ANZ INTEGER DEFAULT 0,  
    KUNDENNR INTEGER NOT NULL,  
    ARTINDEX INTEGER NOT NULL,  
  
    CONSTRAINT PK_BESTELLUNG PRIMARY KEY (BESTNR)  
);
```

**ALTER TABLE BESTELLUNG ADD CONSTRAINT**

```
FK_KUNDE_BESTELLUNG FOREIGN KEY (KUNDENNR) REFERENCES  
KUNDE(KUNDENNR);
```

**ALTER TABLE BESTELLUNG ADD CONSTRAINT**

```
FK_ARTIKEL_BESTELLUNG FOREIGN KEY (ARTINDEX) REFERENCES  
ARTIKEL(ARTINDEX);
```

**ALTER TABLE ARTIKEL**

```
ADD CONSTRAINT CH_GEWICHT CHECK (  
    GEWICHT >=0  
);
```

**ALTER TABLE BESTELLUNG**

```
ADD CONSTRAINT CH_ANZ CHECK (  
    ANZ >=0  
);
```

### 3 AccessDBConsole

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

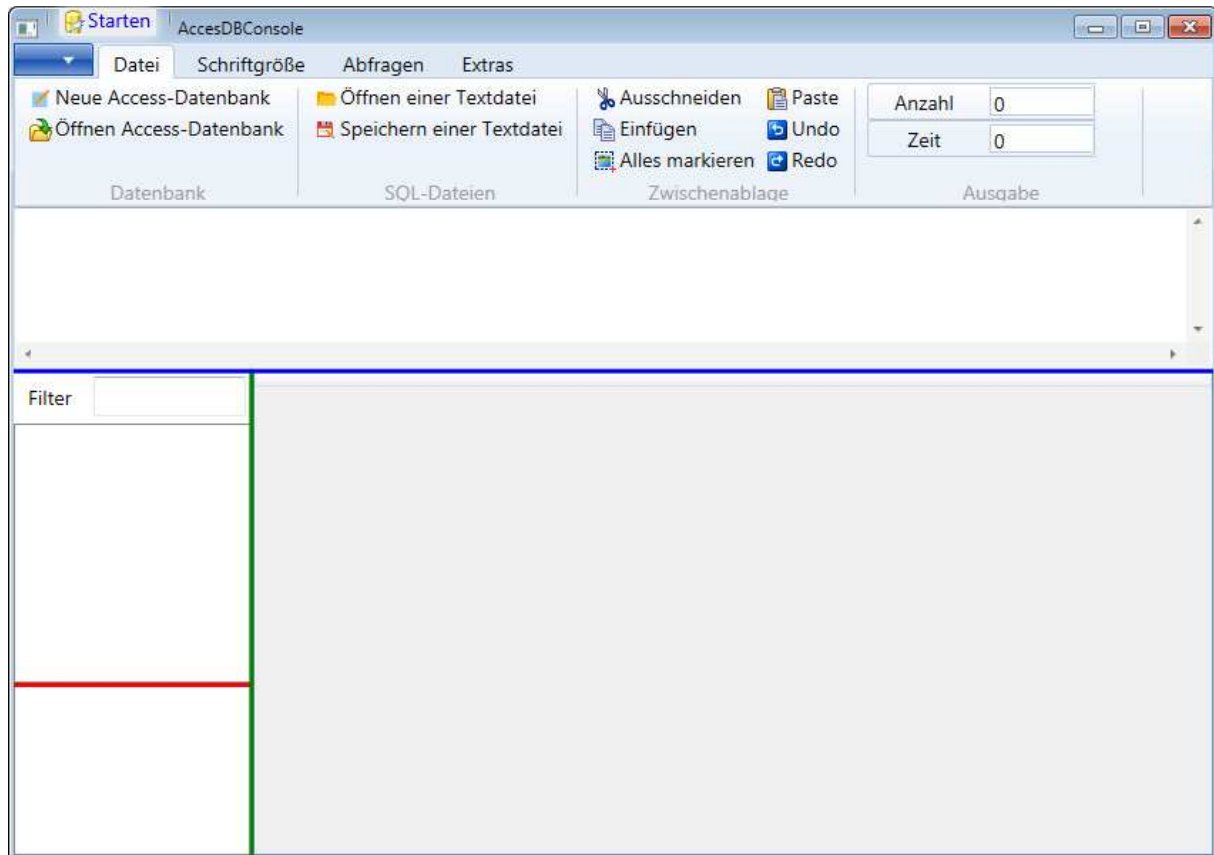


Abbildung 40 Hauptfenster der AccessConsole

#### 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.     |
| •                          |  |

### 3.1 Neue Datenbank erstellen

#### Ablauf:

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

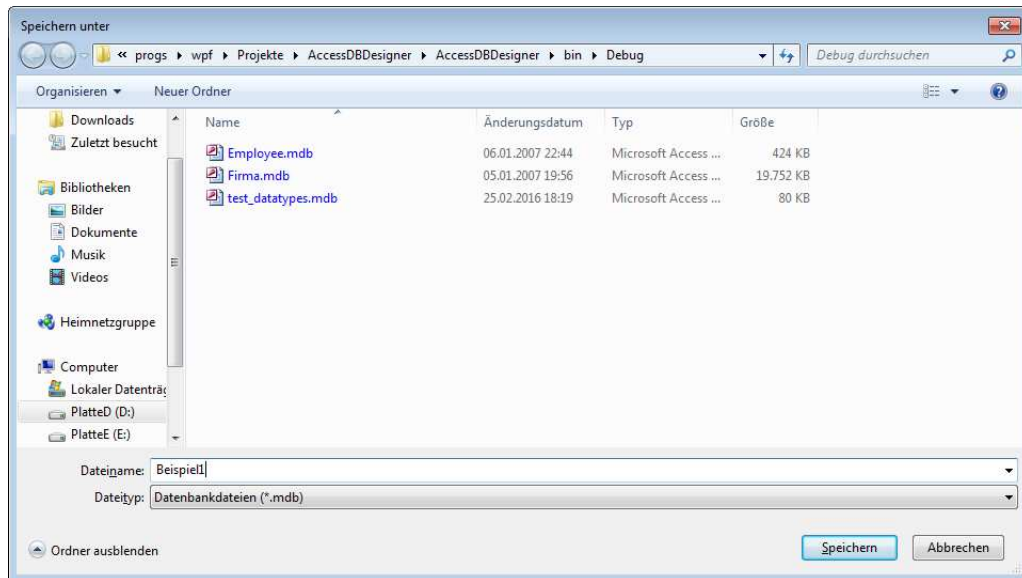


Abbildung 41 Erstellen der Datenbank "Beispiel1.mdb"

#### Ergebnis:

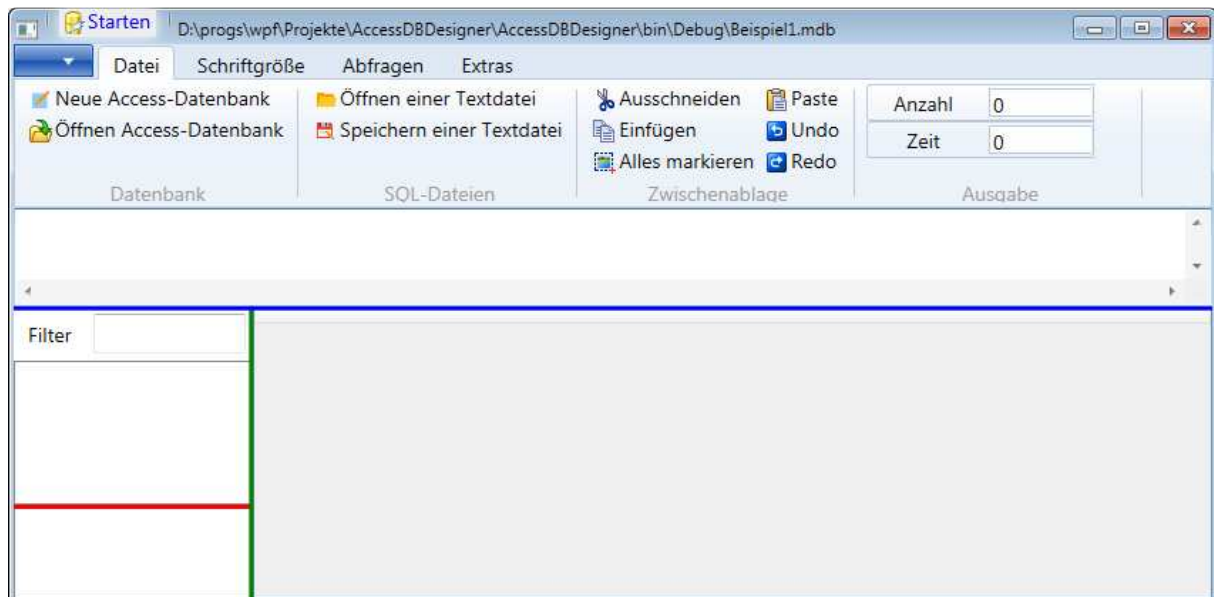


Abbildung 42 Neue datenbank mit leerer Entitiy-Liste

Einfügen der Skripte in den Editor:

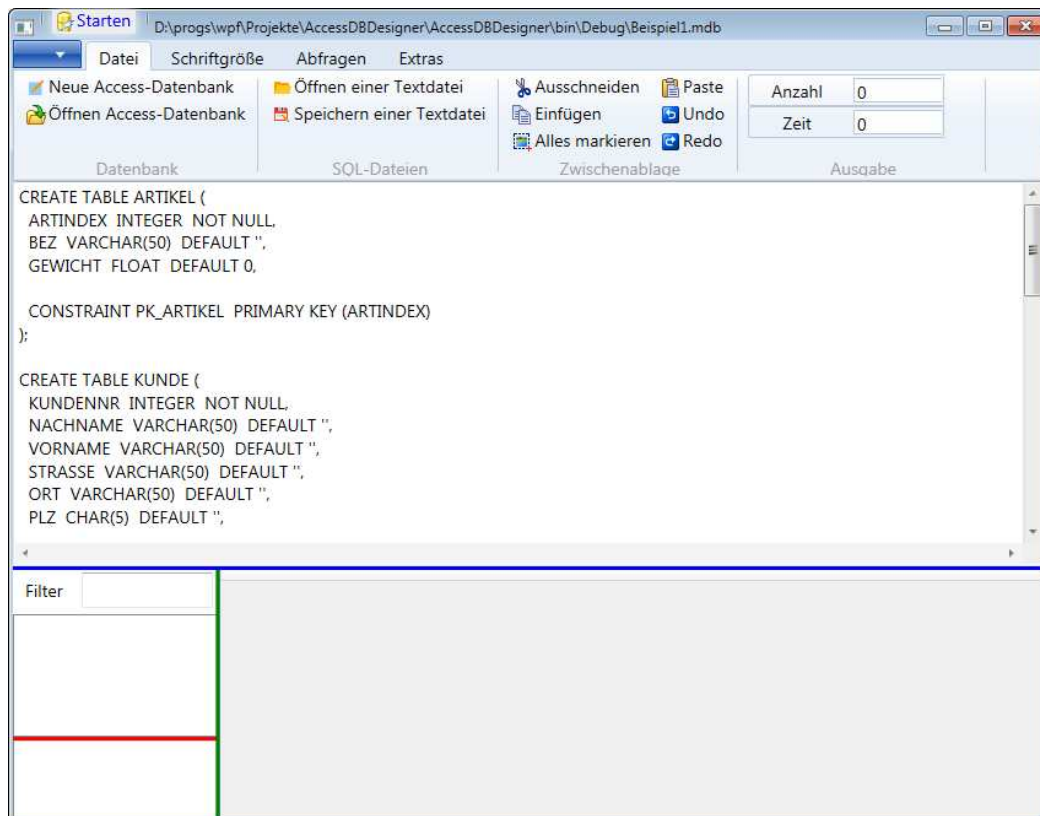


Abbildung 43 Eintragen der SQL-Skripte

Schalter „Starten Abfrage“

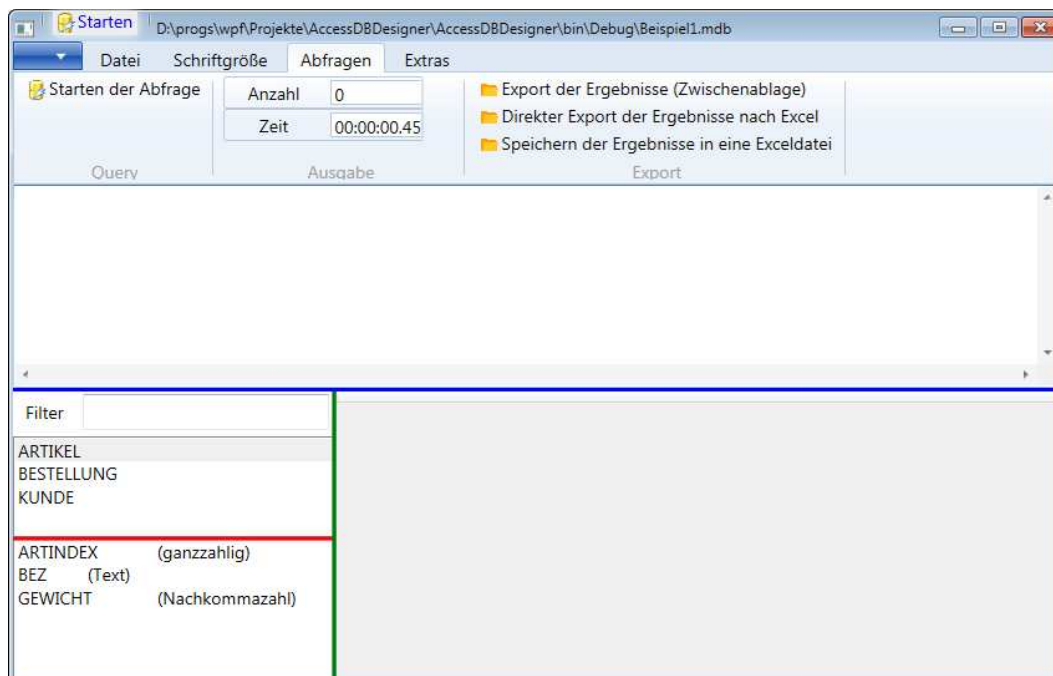


Abbildung 44 Anzeige der Datenbank-Tabellen

In der oberen Liste sind alle Entities eingetragen.

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



## 3.2 Daten eintragen

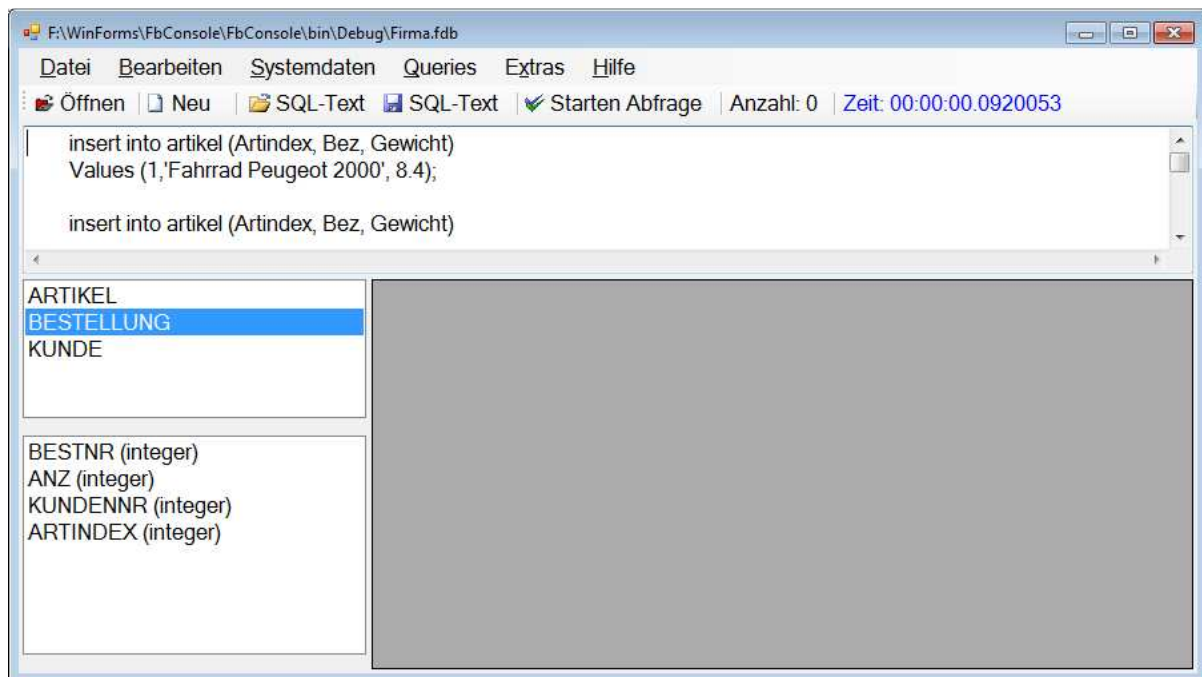


Abbildung 45 Daten mittels „Insert-Into“ eintragen

### 3.2.1 SQL-Befehle zum Eintragen:

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

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

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

```
insert into kunde (kundennr, nachname, 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, kundennr, artindex)
Values (201, 2, 101, 1);

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

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

```

### 3.3 Testabfragen

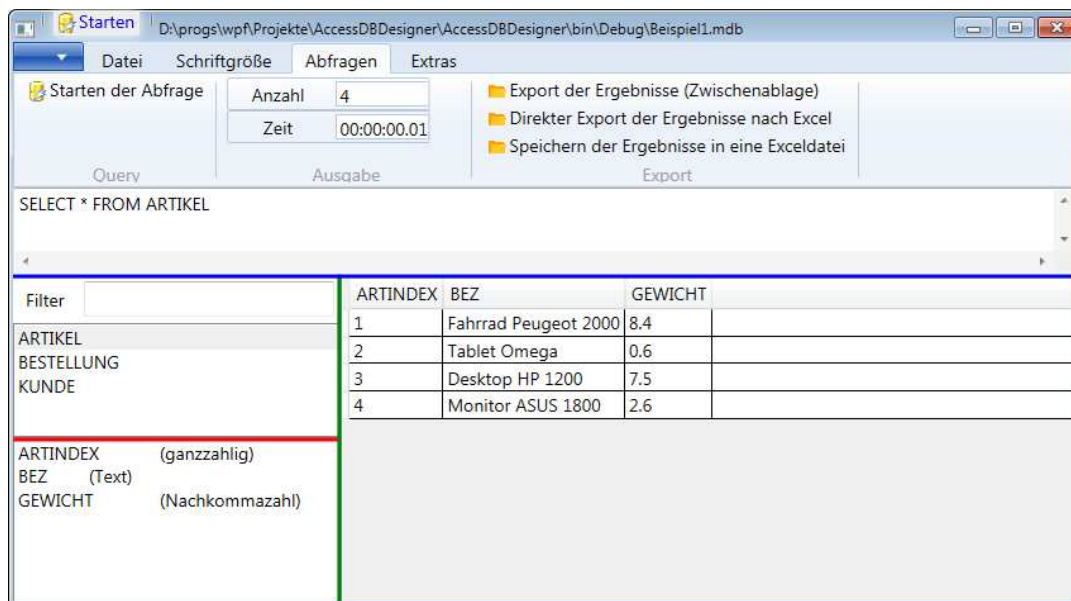


Abbildung 46 Abfrage bezüglich der Artikel

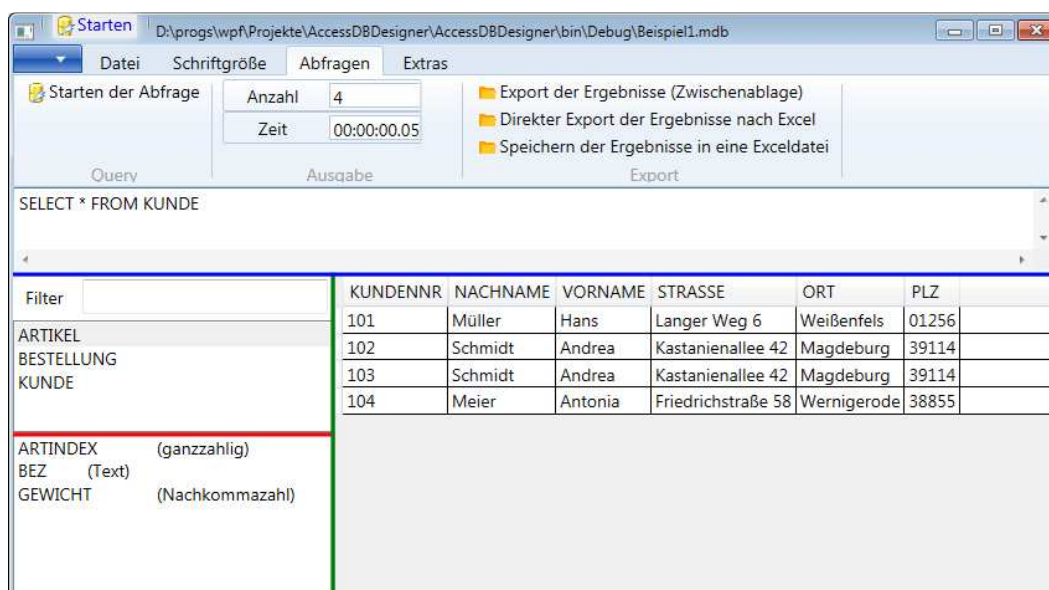


Abbildung 47 Abfrage bezüglich der Kunden

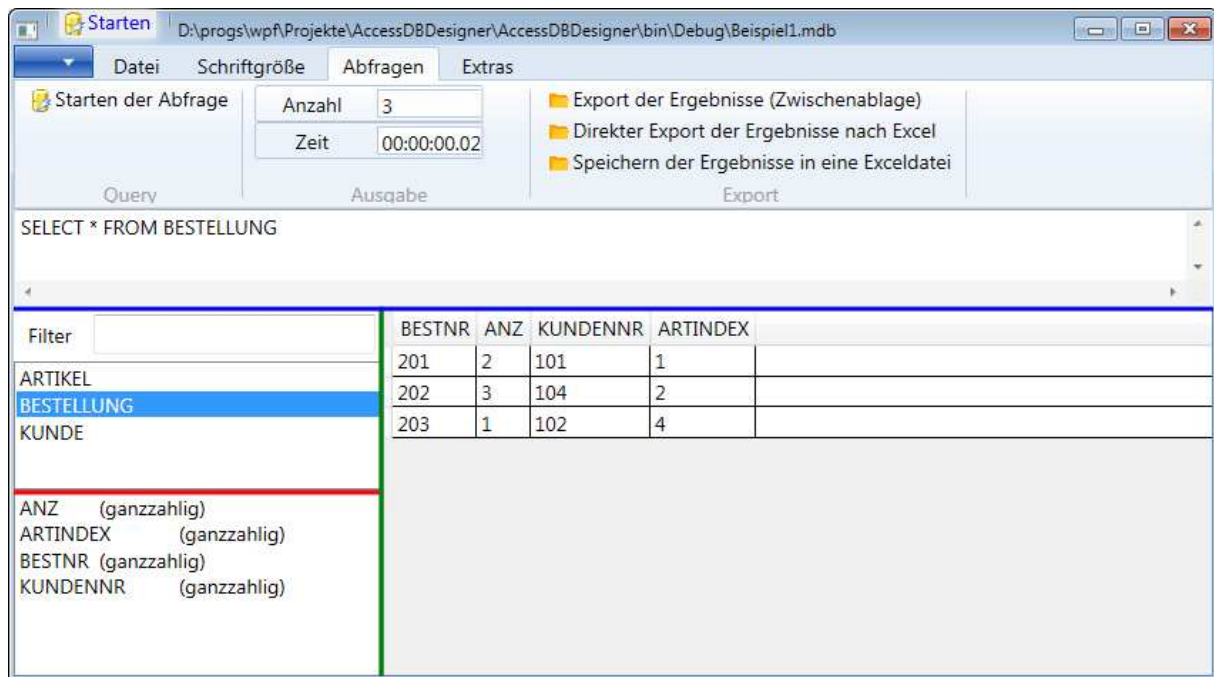


Abbildung 48 Abfrage bezüglich der Bestellung

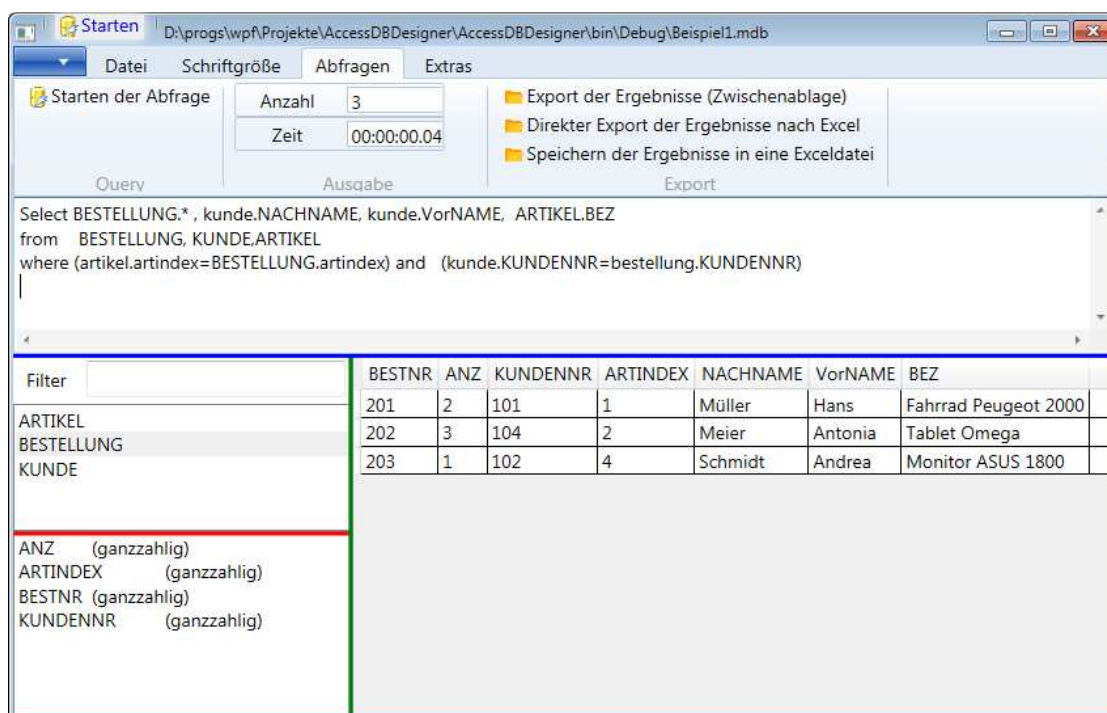


Abbildung 49 Abfrage bezüglich der Bestellungen

```
Select BESTELLUNG.* , kunde.NACHNAME, kunde.VorNAME, ARTIKEL.BEZ
from BESTELLUNG, KUNDE,ARTIKEL
where (artikel.artindex=BESTELLUNG.artindex) and
      (kunde.KUNDENNR=bestellung.KUNDENNR)
```

## 4 Anhang

### 4.1 AutoWert

Wenn man Datensätze von Entities einträgt, so müssen alle PrimaryKeys einen eindeutigen Wert haben. Normale relationale Datenbanken haben dafür eine Sequenz. Diese ist dann Netzwerkweit eindeutig abrufbar. Damit ist also sichergestellt, dass auch im Multi-User-Betrieb nur die Primary-Nummer immer nur einmal existiert.

Unter Access ist das nicht möglich (Transact-SQL)

<https://msdn.microsoft.com/de-de/library/ff878091.aspx>

Die schlechtere Alternative ist der AutoIncrement-Wert (Counter). Man trägt einen Datensatz ein, und die Access-Datenbank trägt den Primary-Key automatisch ein. Leider erhält man diesen Wert **nicht** zurück!

Ganz so schlimm ist es nicht, dass eine Access-Datenbank nur einen User haben kann. Man kann mit dem Max-Befehl den Primary-Key bekommen. *Das würde im Netzwerk absolut nicht funktionieren!*

Das erste Beispiel wird nun mit Counter verändert.

	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Beschre
1	PK	ARTINDEX	COUNTER				NOT NULL				-	
2		BEZ	VARCHAR	50				x			-	
3		GEWICHT	FLOAT					x	0	CH_GEWICHT	-	

Abbildung 50 Das Entity Artikel mit einem Counter als Primary-Key

Einstellungen eines Entities: BESTELLUNG

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

	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Besc
1	PK	BESTNR	COUNTER				NOT NULL				-	
2		ANZ	INTEGER					x	0	CH_ANZ	-	
3	FK	KUNDENNR	INTEGER				NOT NULL				-	
4	FK	ARTINDEX	INTEGER				NOT NULL				-	

Add Insert Edit Delete Copy

Ok Abbruch

Abbildung 51 Das Entity Bestellung mit einem Counter als Primary-Key

Einstellungen eines Entities: KUNDE

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

	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung	Besc
1	PK	KUNDENNR	COUNTER				NOT NULL				-	
2		NACHNAME	VARCHAR	50				x			-	
3		VORNAME	VARCHAR	50				x			-	
4		STRASSE	VARCHAR	50				x			-	
5		ORT	VARCHAR	50				x			-	
6		PLZ	CHAR	5				x			-	

Add Insert Edit Delete Copy

Ok Abbruch

Abbildung 52 Das Entity Kunde mit einem Counter als Primary-Key

Anzeige der kompletten Datenbank:

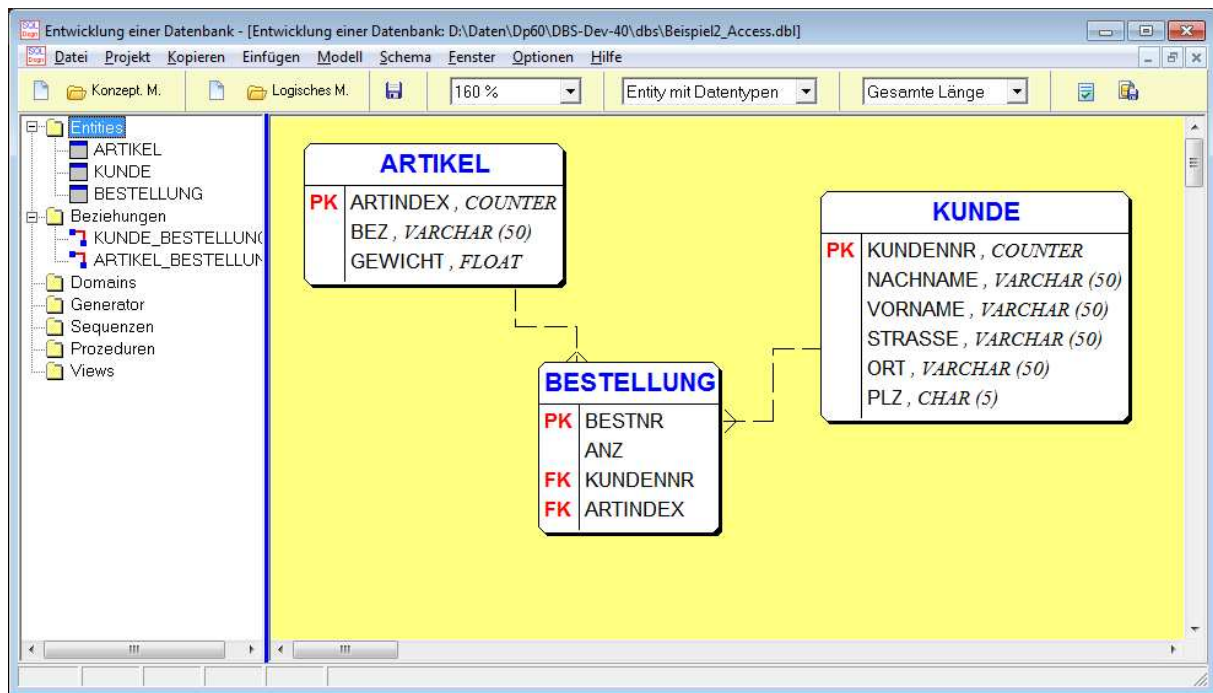


Abbildung 53 Die Entities des zweiten Beispiels

#### 4.1.1 Quellcode des zweiten Beispiels

#### 4.1.2 Quellcode der SQL-Befehle:

```
CREATE TABLE ARTIKEL (
    ARTINDEX COUNTER,
    BEZ VARCHAR(50) DEFAULT "",
    GEWICHT FLOAT DEFAULT 0,

    CONSTRAINT PK_ARTIKEL PRIMARY KEY (ARTINDEX)
);
```

```
CREATE TABLE KUNDE (
    KUNDENNR COUNTER,
    NACHNAME VARCHAR(50) DEFAULT "",
    VORNAME VARCHAR(50) DEFAULT "",
    STRASSE VARCHAR(50) DEFAULT "",
    ORT VARCHAR(50) DEFAULT "",
    PLZ CHAR(5) DEFAULT "",
    ISTPREMIUMKUNDE LOGICAL DEFAULT 0,

    CONSTRAINT PK_KUNDE PRIMARY KEY (KUNDENNR)
);
```

```
CREATE TABLE BESTELLUNG (
    BESTNR COUNTER,
    ANZ INTEGER DEFAULT 0,
    KUNDENNR INTEGER NOT NULL,
    ARTINDEX INTEGER NOT NULL,
```

```
CONSTRAINT PK_BESTELLUNG PRIMARY KEY (BESTNR)
);
```

```
ALTER TABLE BESTELLUNG ADD CONSTRAINT
FK_KUNDE_BESTELLUNG FOREIGN KEY (KUNDENNR) REFERENCES
KUNDE(KUNDENNR);
```

```
ALTER TABLE BESTELLUNG ADD CONSTRAINT
FK_ARTIKEL_BESTELLUNG FOREIGN KEY (ARTINDEX) REFERENCES
ARTIKEL(ARTINDEX);
```

```
ALTER TABLE ARTIKEL
ADD CONSTRAINT CH_GEWICHT CHECK (
GEWICHT >=0
);
```

```
ALTER TABLE BESTELLUNG
ADD CONSTRAINT CH_ANZ CHECK (
ANZ >=0
);
```

## 4.2 SQL-Einfügedaten

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

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

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

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

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

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

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

```
insert into artikel (Bez, Gewicht)
```

```
Values ('Monitor ASUS 1800', 2.6);
```

```
insert into bestellung (anz, kundennummer, artindex)
```

```
Values (2, 101, 1);
```

```
insert into bestellung (anz, kundennummer, artindex)
```

```
Values (3, 104, 2);
```

```
insert into bestellung (anz, kundennummer, artindex)
```

```
Values (1, 102, 4);
```

Die Primary-Keys für die Kundendatenbank wurde beibehalten. Die Primary-Keys für die beiden anderen Tabellen werden von der Access-Datenbank übernommen.

### **Zweite Variante:**

```
insert into kunde (nachname, vorname, strasse, plz, ort)
```

```
Values ('Müller', 'Hans', 'Langer Weg 6', '01256', 'Weißenfels');
```

```
insert into kunde (nachname, vorname, strasse, plz, ort)
```

```
Values ('Schmidt', 'Andrea', 'Kastanienallee 42', '39114',  
'Magdeburg');
```

```
insert into kunde (nachname, vorname, strasse, plz, ort)
```

```
Values ('Schmidt', 'Andrea', 'Kastanienallee 42', '39114',  
'Magdeburg');
```

```
insert into kunde (nachname, vorname, strasse, plz, ort)
```

```
Values ('Meier', 'Antonia', 'Friedrichstraße 58', '38855',  
'Wernigerode');
```

```
insert into artikel (Bez, Gewicht)
```

```
Values ('Fahrrad Peugeot 2000', 8.4);
```

```
insert into artikel (Bez, Gewicht)
```

```
Values ('Tablet Omega', 0.6);
```

```
insert into artikel (Bez, Gewicht)
```

```
Values ('Desktop HP 1200', 7.5);
```

```
insert into artikel (Bez, Gewicht)
```

```
Values ('Monitor ASUS 1800', 2.6);
```

```
insert into bestellung (anz, kundennummer, artindex)
```

```
Values (2, 1, 1);
```

```
insert into bestellung (anz, kundennummer, artindex)
```

```
Values (3, 2, 2);
```

```
insert into bestellung (anz, kundennummer, artindex)
```

```
Values (1, 3, 4);
```



### 4.3 Ja/Nein

Für bool'sche Werte verwendet man einen logischen Datentyp. Normalerweise erfährt man im Internet, dass der Datentyp Ja/Nein heißt. Das ist aber ein Pseudonym. In der SQL-Welt heißt der Datentyp „Logical“ bzw. „YesNo“.

Folgende Werte kann man diesen Datentyp zuweisen:

- 0 Falsch
- -1 Wahr

Das dritte Datenbank-Beispiel mit dem Attribut „IstPremiumKunde“.

**Einstellung eines Attributes: Attribut7**

Algemein | Default / Unique | Checks | Linked-Attribute | Beschreibung | Zu Erledigen | ObjectID

Name:

☐ Primärschlüssel ☐ Not Null ☐ Fremdschlüssel

**Datentyp-Definition**

Domain:

Datentyp:

Formel:

Datentyp:

**Autonumber**

☐ aktiviert Start:  Increment:

Abbildung 54 IstPremiumKunde

**Einstellungen eines Entites: KUNDE**

Allgemein | Attribute | Indizes | Triggers | Constraints

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

	Schlüssel	Name	Datentyp	Länge	Precision	Domain	NOT NULL	Default-Y/N	Default-Wert	Check-Bedingung	Unique-Bedingung
1	<input checked="" type="checkbox"/>	KUNDENNR	COUNTER				NOT NULL				-
2		NACHNAME	VARCHAR	50				x			-
3		VORNAME	VARCHAR	50				x			-
4		STRASSE	VARCHAR	50				x			-
5		ORT	VARCHAR	50				x			-
6		PLZ	CHAR	5				x			-
7		ISTPREMIUMKUNDE	LOGICAL					x	0		-

Abbildung 55 Kunden-Entity mit einem Logical-Attribut

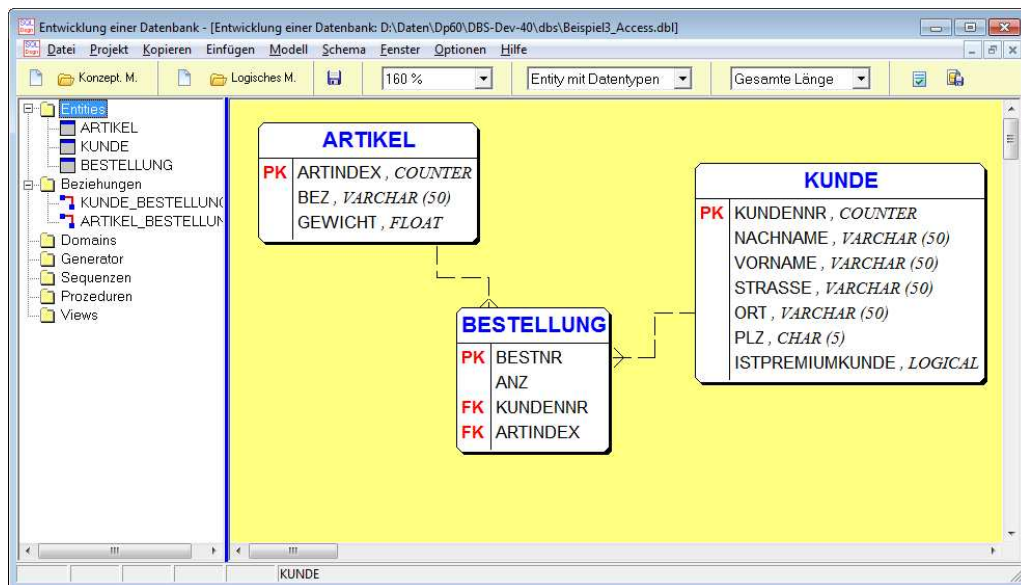


Abbildung 56 Anzeige des dritten Beispiels

SQL-Script:

```
CREATE TABLE KUNDE (
  KUNDENNR COUNTER,
  NACHNAME VARCHAR(50) DEFAULT "",
  VORNAME VARCHAR(50) DEFAULT "",
  STRASSE VARCHAR(50) DEFAULT "",
  ORT VARCHAR(50) DEFAULT "",
  PLZ CHAR(5) DEFAULT "",
  ISTPREMIUMKUNDE LOGICAL DEFAULT 0,

  CONSTRAINT PK_KUNDE PRIMARY KEY (KUNDENNR)
);
```

Quellcode des dritten Beispiels:

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

```
insert into kunde (nachname, vorname, strasse, plz, ort, ISTPREMIUMKUNDE)
Values ('Schmidt', 'Andrea', 'Kastanienallee 42', '39114', 'Magdeburg', -1);
```

```
insert into kunde (nachname, vorname, strasse, plz, ort, ISTPREMIUMKUNDE)
Values ('Schmidt', 'Andrea', 'Kastanienallee 42', '39114', 'Magdeburg', 0);
```

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

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

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

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

**insert into artikel (Bez, Gewicht)**

Values ('Monitor ASUS 1800', 2.6);

**insert into bestellung (anz, kundennummer, artindex)**

Values (2, 1, 1);

**insert into bestellung (anz, kundennummer, artindex)**

Values (3, 2, 2);

**insert into bestellung (anz, kundennummer, artindex)**

Values (1, 3, 4);

**Hinweis:**

- Wenn man mehrfach die Daten eingibt und wieder löscht, funktionieren nicht mehr die Beziehungen, da die Counter nicht zurückgesetzt wurden.
- Dann sollte man die Datenbank neu erstellen.

The screenshot shows the Access DB Designer interface. The query window displays 'SELECT \* FROM KUNDE'. Below the query, a table of results is shown with columns: KUNDENNR, NACHNAME, VORNAME, STRASSE, ORT, PLZ, and ISTPREMIUMKUNDE. The table contains four rows of customer data. The 'ISTPREMIUMKUNDE' column uses checkboxes to represent logical values.

KUNDENNR	NACHNAME	VORNAME	STRASSE	ORT	PLZ	ISTPREMIUMKUNDE
1	Müller	Hans	Langer Weg 6	Weißenfels	01256	<input type="checkbox"/>
2	Schmidt	Andrea	Kastanienallee 42	Magdeburg	39114	<input checked="" type="checkbox"/>
3	Schmidt	Andrea	Kastanienallee 42	Magdeburg	39114	<input type="checkbox"/>
4	Meier	Antonia	Friedrichstraße 58	Wernigerode	38855	<input type="checkbox"/>

Below the table, the field list for the 'KUNDE' table is visible:

- ISTPREMIUMKUNDE (boolean, bit)
- KUNDENNR (ganzzahlig)
- NACHNAME (Text)
- ORT (Text)
- PLZ (Text)
- STRASSE (Text)
- VORNAME (Text)

**Abbildung 57 Abfrage "SELECT \* FROM KUNDE"**

Die logischen Werte sind in der Tabelle mit Kontrollkästchen dargestellt.

Setzen eines logischen Wertes:

```
UPDATE kunde
SET ISTPREMIUMKUNDE=-1
WHERE KUNDENNR=4
```

Starten D:\progs\wpf\Projekte\AccessDBDesigner\AccessDBDesigner\bin\Debug\Beispiel3.mdb

Datei Schriftgröße Abfragen Extras

Starten der Abfrage Anzahl 4 Zeit 00:00:00.01

Export der Ergebnisse (Zwischenablage)  
Direkter Export der Ergebnisse nach Excel  
Speichern der Ergebnisse in eine Exceldatei

Query Ausgabe Export

SELECT \* FROM KUNDE

Filter	KUNDENNR	NACHNAME	VORNAME	STRASSE	ORT	PLZ	ISTPREMIUMKUNDE
ARTIKEL	1	Müller	Hans	Langer Weg 6	Weißenfels	01256	<input type="checkbox"/>
BESTELLUNG	2	Schmidt	Andrea	Kastanienallee 42	Magdeburg	39114	<input checked="" type="checkbox"/>
KUNDE	3	Schmidt	Andrea	Kastanienallee 42	Magdeburg	39114	<input type="checkbox"/>
	4	Meier	Antonia	Friedrichstraße 58	Wernigerode	38855	<input checked="" type="checkbox"/>

ISTPREMIUMKUNDE (boolean, bit)  
KUNDENNR (ganzzahlig)  
NACHNAME (Text)  
ORT (Text)  
PLZ (Text)  
STRASSE (Text)  
VORNAME (Text)

Abbildung 58 Anzeige der Tabelle "Kunde" nach dem Setzen „IstPremiumKunde“

## 5 Funktionen

Dieses Kapitel befasst sich mit SQL-Funktionen. Dazu gehören Funktionen zur Manipulation von Zeichenketten, Datumsfunktionen, Konvertierungsfunktionen und allgemeine Funktionen.

### 5.1 String-Manipulation

Die Schreibweise von Texten in Attributen ist nicht immer eindeutig. So kann ein Mitarbeiter die Namen in Kleinbuchstaben oder komplett in Großbuchstaben eingeben. Eine lexikalische Suche ist dann nicht mehr möglich. SQL bietet hier mit String-Funktionen Abhilfe.

Folgende Funktionen sind möglich:

Funktion	Zweck
Asc	Liefert den Ascii-Code: Asc("A") 65
Chr	Liefert das Zeichen Chr(65) A
LCase(column   Expression)	Konvertiert alphanumerische Zeichenwerte in Kleinbuchstaben
UCase(column   Expression)	Konvertiert alphanumerische Zeichenwerte in Großbuchstaben
Instr(column   Expression)	Sucht im Ausdruck InstrRev ( string_being_searched, string2 [, start [, compare] ] ) Instr ( 1, "Tech on the Net", "the" ) Result: 9
InstrRev	Sucht im Ausdruck vom letzten Zeichen Instr ( [start], string_being_searched, string2, [compare] ) InstrRev ( "alphabet", "a" ) Result: 5
Left	Extrahiert vom Anfang eines Teilstrings Left ( text, number_of_characters ) Left ( "Tech on the Net", 4 ) Result: "Tech"
Len	Berechnet die Länge des Strings/Ausdrucks
LTrim	Löscht die Leerzeichen am Anfang, nicht aber in der Mitte LTrim(" AB C ") ergibt "AB C "
Mid	Ausgabe eines Teilstrings Mid ( text, start_position, number_of_characters ) Mid ( "Tech on the Net", 1, 4 ) Result: "Tech"
Replace	Ersetzt im Ausdruck einen Teilstring Replace ( string1, find, replacement, [start, [count, [compare]]] ) Replace( "alphabet", "bet", "hydro" ) Result: "alphahydro"  Replace ( "alphabet", "a", "e" ) Result: "elphebet"  Replace( "alphabet", "a", "e", 2 ) Result: "lphebet"  Replace( "alphabet", "a", "e", 1, 1 ) Result: "elphabet"
Right	Extrahiert vom Ende eines Teilstrings Right ( text, number_of_characters ) Right ( "Tech on the Net", 3 ) Result: "Net"
RTrim	Löscht die Leerzeichen am Ende, nicht aber in der Mitte RTrim(" AB C ") ergibt " AB C"

Space	Erzeugt n-Leerzeichen
Str	Umwandlung einer Zahl in einen String. Nur notwendig, wenn man Zahlen und Texte verbinden will.
StrConv	Umwandlungen eines Strings: StrConv ( text, conversion, LCID ) vbUpperCase 1 HALLO vbLowerCase 2 hallo vbProperCase 3 Hallo vbUnicode 64 Unicode vbFromUnicode 128 Unicode
StrReverse	Umdrehen eines String-Ausdrucks
Trim	Löscht die Leerzeichen am Anfang und am Ende, nicht aber in der Mitte RTrim(" AB C ") ergibt "AB C"

## 5.2 Numerische Funktion

Funktion	Beschreibung
Abs(a)	absoluter Wert von a
Atn	Arcus Tangens
Avg	Average, Mittelwert
Cos	Cosinus
Count	Anzahl der Datensätze
Exp	liefert e hoch n ( $e=2,71828...$ ), $e^x$
Fix	Gibt nur den ganzzahligen Anzeil ohne zu runden aus
Format	Formatiert nach den bekannten Formeln Format (210.6, "#,##0.00") Result: '210.60'  Format (210.6, "Standard") Result: '210.60'  Format (0.981, "Percent") Result: '98.10%'  Format (1267.5, "Currency") Result: '\$1,267.50'
Int	Gibt nur den ganzzahligen Anzeil ohne zu runden aus. select fix(12.56) , int(12.56) , fix(-12.56) , int(-12.56) from job 12      12      -12 <b>-13</b>
Max	Bestimmt das Maximum eines numerischen Ausdrucks
Min	Bestimmt das Minimum eines numerischen Ausdrucks
Randomize	Initialisiert die Erzeugung einer Zufallszahl
Rnd	Liefert eine Zufallszahl. Sinnvoll bei Testdaten. select rnd from job
Round	Rundet einen numerischen Wert Round (210.67, 1) Result: 210.7  Round (210.67, 0) Result: 211  Round (210.67)

	<i>Result: 211</i>
Sgn	Vorzeichenfunktion -1 < 0 0 = 0 +1 > 0
Sqr	Berechnet die Wurzel (SquareRoot)
Sum	Berechnet die Summe eines Ausdrucks
Val	Umwandeln eines Strings in eine Zahl <i>select val('123.567')</i> <i>from job</i>

### 5.3 Datum / ZeitFunktionen

Ein Datum wird intern als binäre Bitfolge dargestellt. Meistens als Anzahl der Tage ab einem bestimmten Stichtag (z. B. 1.1.1970). Ein Datum vor diesem Tag ist dann nicht möglich. Jede Datenbank speichert die Datumswerte individuell. Deshalb existieren Funktionen, die die Tage, Monate und Jahre aus den binären Werten extrahieren.

Date()	Ausgabe des aktuellen Datums: Date()
DateDiff	Ausgabe der Differenz zweier Daten DateDiff ("yyyy", #15/10/1998#, #22/11/2003#) <i>Result: 5</i>  DateDiff ("m", #15/10/2003#, #22/11/2003#) <i>Result: 1</i>  DateDiff ("d", #15/10/2003#, #22/11/2003#) <i>Result: 38</i>
DdatePart	Gibt nur Teile eines Datums aus. DatePart (interval, date, [firstdayofweek], [firstweekofyear]) yyyy Year q Quarter m Month y Day of year d Day w Weekday ww Week h Hour n Minute s Second DatePart ("yyyy", #15/10/1998#) <i>Result: 1998</i>  DatePart ("m", #15/10/2003#) <i>Result: 10</i>  DatePart ("d", #15/10/2003#) <i>Result: 15</i>
DateSerial	Umwandeln von Tag, Monat und Jahr in ein Datum DateSerial (2004, 6, 30)

	<i>Result: '6/30/2004'</i>  DateSerial (2004-1, 6, 30) <i>Result: '6/30/2003'</i>  DateSerial (2004, 6-2, 14) <i>Result: '4/14/2004'</i>
DateValue	Konvertiert einen String in ein Datum DateValue ("June 30, 2004") <i>Result: 6/30/2004</i>  DateValue ("6/30/2004") <i>Result: 6/30/2004</i>  DateValue ("June 30") <i>Result: 6/30/2004</i>
Day	Liefert den aktuellen Tag eines Datums Day (#22/11/2003#) <i>Result: 22</i>
Format	Formatiert die Ausgabe eines Datums Format ( expression, [ format, [ firstdayofweek, [firstweekofyear] ] ] ) Format (#17/04/2004#, "Short Date") <i>Result: '17/04/2004'</i>  Format (#17/04/2004#, "Long Date") <i>Result: 'April 17, 2004'</i>  Format (#17/04/2004#, "yyyy/mm/dd") <i>Result: '2004/04/17'</i>
Hour	Liefert die aktuellen Stunde eines Datums Hour (#10:42:58 PM#) <i>Result: 22</i>
Minute	Liefert die aktuellen Minute eines Datums Minute (#10:42:58 PM#) <i>Result: 42</i>
Month	Liefert den aktuellen Monat eines Datums Month (#22/11/2003#) <i>Result: 11</i>
MonthName	Liefert den Monatsnamen eines Datums MonthName ( number, [abbreviate] ) Der zweite Parameter erlaubt eine Verkürzte Form der Eingabe. MonthName (3) <i>Result: 'March'</i>  MonthName (3, TRUE) <i>Result: 'Mar'</i>  MonthName (7, FALSE) <i>Result: 'July'</i>
Now	Liefert das aktuellen Datums select now() from job



Second	<p>Liefert die Sekunden des aktuellen Datums</p> <p>Second (#10:42:58 PM#) Result: 58</p> <p>Second (#10:14:13 AM#) Result: 13</p>
Time	<p>Liefert die Zeit des aktuellen Datums. Das Datum ist aber sehr seltsam!</p> <p>select Time() from job</p>
TimeSerial	<p>Umwandlung von drei Zahlen (Hour, Min, Sec) in eine Zeit</p> <p>TimeSerial (14, 6, 30) Result: 2:06:30 PM</p> <p>TimeSerial (20 - 8, 6, 30) Result: 12:06:30 PM</p>
TimeValue	<p>Konvertiert einen String in eine Zeit.</p> <p>TimeValue ("3:12:57 PM") Result: 3:12:57 PM</p> <p>TimeValue ("15:12:57") Result: 3:12:57 PM</p>
Weekday	<p>Anzeige des Wochentags</p> <p>Weekday ( date_value, [firstdayofweek] ) 1 = Sonntag</p> <p>Weekday (#22/11/2003#) Result: 7</p> <p>Weekday (#22/11/2003#, vbThursday) Result: 3</p> <p>Weekday (#22/11/2003#, 5) Result: 3</p> <p>Weekday (#01/01/1998#) Result: 5</p>
WeekdayName	<p>Ermittelt den Wochentag im Bereich 1 bis 7.</p> <p>WeekdayName ( number, [abbreviate], [firstdayofweek] )</p> <p>WeekdayName (3) Result: 'Tuesday'</p> <p>WeekdayName (3, TRUE) Result: 'Tue'</p> <p>WeekdayName (3, TRUE, vbMonday) Result: 'Wed'</p> <p>WeekdayName (3, TRUE, 2) Result: 'Wed'</p>
Year	<p>Liefert das Jahr eines Datums</p> <p>Year (#22/11/2003#) Result: 2003</p>

## 5.4 Allgemeine Funktionen

And	Logische Und
First	Liefert den ersten Wert <i>SELECT first(full_name)</i> <i>FROM employee</i>
Last	Liefert den letzten Wert <i>SELECT last(full_name)</i> <i>FROM employee</i>
Nz	Liefert keinen Nullwert, sondern einen Leerstring. Besser ist es aber, wenn man mit Defaultwerten arbeitet. Manchmal bekommt man aber auch Datenbanken von Kollegen ,-)
Or	Logisches Oder

## 5.5 Group Funktionen

Avg	Mittelwert
Count	Anzahl
Max	Maximum
Min	Minimum
Sum	Summe

## 6 Indexverzeichnis

### A

AccessDBConsole	30
Insert into	33
Testabfragen	34
Allgemeine Funktionen	50
Autowert	36

### B

Beispiel eines logisches Modell	8
---------------------------------	---

### D

Datumsfunktionen	47
DB-Designer	6

### E

Eigenschaften des DB-Designer-Programms	6
Eigenschaften in Kurzform	6
Erstellen der SQL-Skripte	27

### F

FBConsole	
Daten eintragen	33
Neue Datenbank	31
Funktionen	45

### G

Group Funktionen	50
------------------	----

### I

Inhalt der ZIP-Datei	5
----------------------	---

### J

Ja/Nein	41
---------	----

### L

Logical	41
Logisches Modell	7

### N

Neue Datenbank	31
----------------	----

### S

Sequenz	36
String-Manipulation	45