

Hägele

Relationale Datenbanken

ACCESS 2013

KFZ m -- geführt -- 1 Fahrzeughalter
KFZ m -- gefertigt -- 1 Hersteller

Kunde: Kfz-Nr., Straße, PLZ, Ort, Telefon

Datenbank-Management-System (DBMS):
Nutzer 1: Suchanfrage
Nutzer 2: Verwalten
Nutzer 3: Suchanfrage
Funktionen: Ändern, Löschen, Einfügen, Sortieren, Suchen
Datenbanken: Kundenliste, Produktliste

Relationale Datenbanken am Beispiel von ACCESS 2013 (berufliches Umfeld)



Mit Lösungs-CD

Inhaltsverzeichnis

Vorwort.....	3
1. Kapitel: Allgemeines zu Datenbanksystemen	4
1.1. Aufgaben einer Datenbank.....	5
1.2. Drei-Schichten-Architektur eines Datenbanksystems.....	6
1.3. Das Prinzip relationaler Datenbanken.....	8
1.4. Aufgaben zu Kapitel 1.....	9
2. Kapitel: Entwurf und Implementierung einer Datenbank	10
2.1. Phasen beim Erstellen einer neuen Datenbank.....	11
2.2. Entity-Relationship-Modell - Fallbeispiele.....	13
2.3. Normalformen 1 bis 3 für relationale Datenbanken.....	18
2.4. Anomalien von Datenbanken	20
2.5. Aufgaben zu Kapitel 2.....	21
3. Kapitel: Arbeiten mit Access 2013.....	22
3.1. Anlegen einer Datenbank in ACCESS2013.....	23
3.2. Objekte in ACCESS2013.....	24
3.3. Tabellen erstellen	25
3.4. Tabellen in Beziehung setzen	27
3.5. Abfragen erstellen	29
3.5.1. Kreuzabfragen erstellen.....	30
3.6. Formulare.....	33
3.6.1. Erstellen von Formulare in der Entwurfsansicht.....	33
3.6.2. Unterformulare – erstellen und anwenden.....	34
3.7. Steuerelemente in Formulare	37
3.7.1. Steuerelemente mit Makros	37
3.7.2. Steuerelemente mit VBA.....	38
3.8. Berichte.....	39
3.9. Datenaustausch zu Excel.....	42
3.10. Aufgaben zu Kapitel 3.....	44

4.	VBA in ACCESS	45
4.1.	Begriffe in ACCESS.....	45
4.2.	Variablen und Konstanten.....	48
4.3.	Zugriffe auf Datensätze mittels VBA.....	50
4.4.	Fehlersteuerung.....	50
5.	Kapitel: Tipps und Tricks	54
5.1.	Bedingte Formatierungen.....	55
5.2.	Echtzeit-Uhr in Ihrem Formular.....	56
6.	Kapitel: Sonstiges	58
6.1.	Datenbank pflegen und sichern.....	59
6.1.1.	Datenbank komprimieren.....	59
6.1.2.	Datenbank mit Kennwort verschlüsseln.....	61
6.2.	Speichern unter ACCDE-Datei.....	62
6.3.	SkyDrive.....	63
7.	Kapitel: Ausgewählte Arbeitsaufträge	64
7.1.	Arbeitsauftrag: Kundenrechnungen verwalten.....	65
7.2.	Arbeitsauftrag: CD-Liste verwalten.....	69
7.3.	Arbeitsauftrag: Bilder.- und Datenverwaltung.....	74
7.4.	Arbeitsauftrag: Datenbank für Cocktailbar.....	80
8.	Lösungen zu den einzelnen Kapiteln	84
9.	Stichwortverzeichnis:	86

Vorwort

In diesem Lernheft sollen Sie die allgemeinen Grundlagen relationaler Datenbanken kennenlernen. ACCESS2013 dient nun dazu, eine Datenbank dann auch tatsächlich zu implementieren. Hierbei lernen Sie die Datenbankobjekte, wie Abfrage, Berichte, Formulare, etc. in gewünschter Form zu erstellen und einzusetzen.

Das erste Kapitel beschäftigt sich mit dem Begriff ‚Datenbank‘ im Allgemeinen. Definition, Aufgaben und Aufbau werden dargestellt.

Im zweiten Kapitel werden relationale Datenbanken kleinschrittig entwickelt. Die Schritte hierzu werden allgemeingültig aufgezeigt.

Das dritte Kapitel beschreibt die Implementierung einer Datenbank in Access 2013. Praxisnah wird hier die Programmierumgebung nahegebracht, so dass der Leser selbst in der Lage ist, eine auf dem „Papier“ entwickelte Datenbank in ein MS-Access 2013-File umzusetzen.

Die Themen des vierten Kapitels sind allgemeingültig, wurden hier jedoch mittels Access 2013 aufgezeigt. Alle Arbeitsaufträge des fünften Kapitels befinden sich auf der Lösungs-CD. Sie stellen eine Sammlung aus Access 2013 - Ideen dar, welche auch leicht den eigenen Bedürfnissen angepasst werden. Kleinschrittig und verständlich werden die Lösungsschritte aufgezeigt.

1. Kapitel: Allgemeines zu Datenbanksystemen

1.1 Aufgaben einer Datenbank

1.2 Drei-Schichten-Architektur– ANSI-Architektur-Modell

1.3 Das Prinzip relationaler Datenbanken

1.4 Aufgaben zu Kapitel 1

1.1. Aufgaben einer Datenbank

Unter einer Datenbank besteht aus einer Datenmenge, welche zu verwalten ist und einem Datenbankmanagementsystem (DBMS), welches diesen Datenbestand verwaltet. Generell ist die Aufgabe von Datenbanksystemen das Speichern und Bereitstellen von (betrieblichen) Daten, die von verschiedenen Nutzern, z.B. eines Unternehmens benötigt werden.

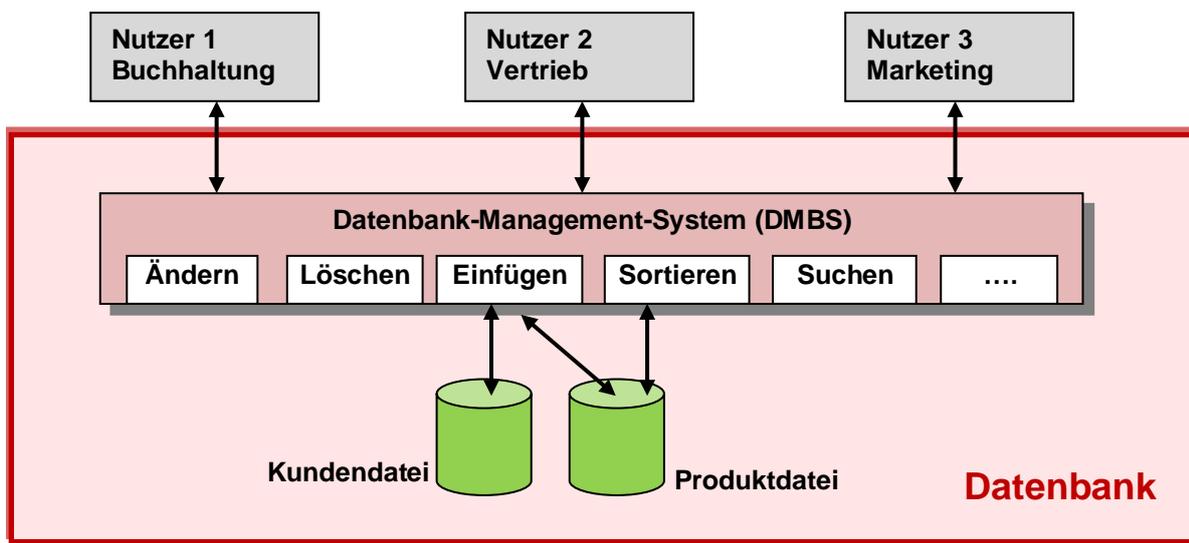


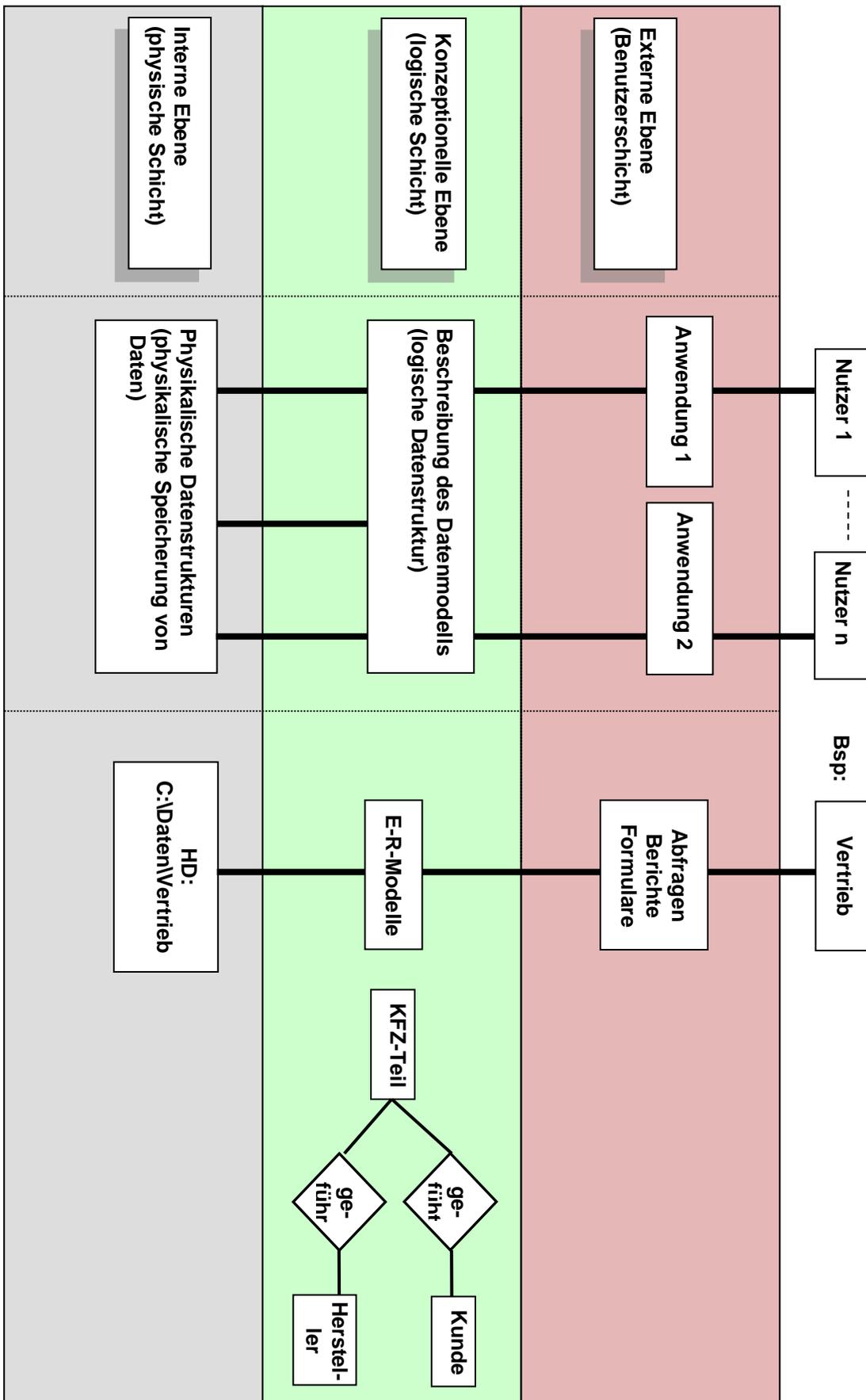
Abbildung: Zentrale Datenverwaltung durch das DBMS

Aufgaben DBMS

Wenn nun mehrere Nutzer gleichzeitig lesend oder schreibend auf Daten zugreifen wollen, muss es zwangsläufig zu Konflikten kommen. Genau hier greift das sogenannte Datenbank-Management-System (DBMS) ein:

- Durch die Verwendung einer zentralen Datenverwaltung, d.h. Kontrolle sollen mehrere **Anforderungen** erfüllt werden:
- Benutzerfreundlichkeit, d.h. eine grafische Bedienoberfläche für den Endbenutzer; eine leicht zu erlernende Benutzersprachen für professionellen Nutzer (Systementwickler, Programmierer).
- **Mehrbenutzerfähigkeit**, d.h. Datenbank für mehrere Benutzer gleichzeitig nutzbar.
- **Vermeidung einer Mehrfachspeicherung (Redundanz)** von Daten, d.h. Daten sollen nur einmal abgespeichert sein. Hierdurch wird die Datenpflege erleichtert.
- Alle Anwender arbeiten mit der gleichen Datenbasis, d.h. **übereinstimmende (konsistente)** Daten für alle.
- Die **Integrität** des Datenbestandes ist gewährleistet. Das bedeutet, der Datenbestand ist in sich schlüssig, d.h. die Daten müssen vollständig, korrekt und widerspruchsfrei sein. Beispiel: Zu jedem Auftrag gibt es einen Auftraggeber.
- Gewährleistung des **Datenschutzes** (Verschlüsselung, Kennwortvergabe) muss auf jeden Fall gegeben sein. Dazu gehört auch, welche Teilnehmer zu welchen Daten zugriffsberechtigt sind, und zwar lesend und/oder schreibend.
- **Datensicherheit**, d.h. der Datenbestand muss gegen jegliche Art von Hardware und Softwareausfällen geschützt werden.

1.2. Drei-Schichten-Architektur eines Datenbanksystems



Erläuterungen zum Drei-Schichten-Modell¹:

Die **Benutzerschicht** bildet die grafische Benutzeroberfläche (engl. **grafic user interface**, kurz **GUI**) für den Anwender dar. In ACCESS stehen hierfür Formulare, Berichte und Tabellen zur Verfügung.

Die **Logikschicht** beschreibt die Beziehungen der einzelnen Daten zueinander. Es gehört mit zur Aufgabe bei der Entwicklung einer Datenbank, unterschiedliche Daten zueinander in Beziehung zu setzen, diese in Modelle zu beschreiben (z.B. Entity-Relationship-Model). Speziell in einer relationaler Datenbank müssen zusammengehörende Daten in unterschiedlichen Tabellen ablegt und in Beziehung gesetzt werden.

Das Abspeichern der Daten auf z.B. einer Festplatte beschreibt die **physikalische Schicht**. Da diese Aufgabe des Betriebssystems ist und nicht zur Aufgabe eines Entwicklers einer Datenbank gehört, wird an dieser Stelle nicht näher darauf eingegangen.



Hinweis: Das ANSI-Architekturmodell zeigt nicht die Schritte der Datenbankentwicklung, sondern die verschiedenen Teile eines Datenbankkonzepts.

¹ Der Urheber des Drei-Schichten-Architektur-Modells (veröffentlicht 1975) ist das ‚American National Standards Institut (ANSI)‘. Daher findet sich in diesem Zusammenhang auch die Benennung ‚ANSI-Architektur-Modell‘.

1.3. Das Prinzip relationaler Datenbanken

Große Datenmengen bereiten Sie übersichtlich auf, indem Sie diese in verschiedenen Tabellen aufteilen. So wird eine Firma Kundendaten in eine Tabelle ablegen, die Aufträge eines Kunden in einer weiteren Tabelle.

Tabelle: Aufträge			
Auftrags-Nr	KNr	Auftrag	Auftragsdatum
100	2	Lackieren	8.4.2009
101	1	Bremsen prüfen	12.5.2009
102	4	Stoßdämpfer wechseln	13.7.2009
103	3	Ölwechsel	20.12.2009
...			

Tabelle: Kundendaten				
KNr	Nachname	Vorname	Adresse	Ort
1	Aumeister	Anton	Meranerstrasse 16	81455 Dinkelshofen
2	Kessidis	Karl	Blumenstrasse 5	71533 Bad Limburg
3	Müller	Martin	Nansenstrasse 22	86338 Mering
4	Hägele	Uwe	unbekannt	unbekannt
...				

Tabellen in Beziehung setzen

Nun müssen diese Tabellen in eine Beziehung (sprich ‚Relation‘) gebracht werden. Wenn Sie sich die hier zwei abgebildeten Tabellen näher betrachten, dann fällt Ihnen auf, dass die Spalte ‚KNr‘ Bestandteil beider ist. Nun ist es ein Leichtes, über die Kundennummer den erteilten Auftrag und später die Rechnung an die korrekte Adresse zu senden. Die Kundennummer stellt in diesem Beispiel der sogenannte ‚Primärschlüssel‘ (sprich ‚key‘) dar, mit dem die Tabellen verknüpft werden.



Die Bedeutung des Primärschlüssels

- Wird für ein Datenfeld (Attribut) die Eigenschaft des Primärschlüssels festgelegt, kann mit Hilfe dieses Datenfeldes der einzelne Datensatz eindeutig identifiziert werden.
- Jede Kundennummer kann nur einmal vorkommen.
- Die Datensätze sind standardmäßig nach dem Primärschlüsselfeld (hier: Kundennummer) sortiert.

Die Kunst beim Erstellen einer rationalen Datenbank liegt nun darin, die Datensätze übersichtlich in Tabellen zu verteilen und diese Tabellen in Beziehungen zu setzen.

Zur Erläuterung der Bestandteile und der Struktur von Tabellen in relationalen Datenbanken siehe folgend:

Knr	Nachname	Vorname	Strasse	PLZ	Ort
1	Aumeister	Anton	Meranerstrasse 16	81455	Dinkelshofen
2	Kessidis	Karl	Blumenstrasse 5	71533	Bad Limburg
3	Müller	Martin	Nansenstrasse 66	86338	Mering
4	Hägele	Uwe	Mozartstrasse 12	23441	Van Linden

Elemente einer Tabelle

- ✎ Eine Tabelle besteht aus **Zeilen** und **Spalten**.
- ✎ Eine Zeile (= **Datensatz** / **Tupel**) enthält die Daten eines Kunden. Alle Zeilen sind gleich aufgebaut.
- ✎ Die Identifizierung eines Datensatzes erfolgt über einen eindeutigen **Primärschlüssel** (hier: Kundennummer).
- ✎ Die Identifizierung einer Spalte erfolgt über einen eindeutigen **Spaltennamen** (= Attributname). Jede Spalte enthält einen bestimmten Typ von Attributen (Attributtyp, z.B. Zahlen, Text, Datum).
- ✎ Ein **Datenfeld** (= Attributfeld) ist der Schnittpunkt einer Zeile und einer Spalte.

1.4. Aufgaben zu Kapitel 1

- (1) Welche Bedeutung hat das Datenmanagementsystem innerhalb einer Datenbank?
- (2) Welche Aufgaben hat ein Datenmanagementsystem?
- (3) Wodurch kennzeichnet sich eine relationale Datenbank aus?
- (4) In welcher Schicht innerhalb der Drei-Schichten-Architektur bewegt sich ein Anwender einer Datenbank?

2. Kapitel: Entwurf und Implementierung einer Datenbank

2.1 Phasen beim Erstellen einer neuen Datenbank

2.2 Fallbeispiel: Datenbank für Reparatur-Werkstatt

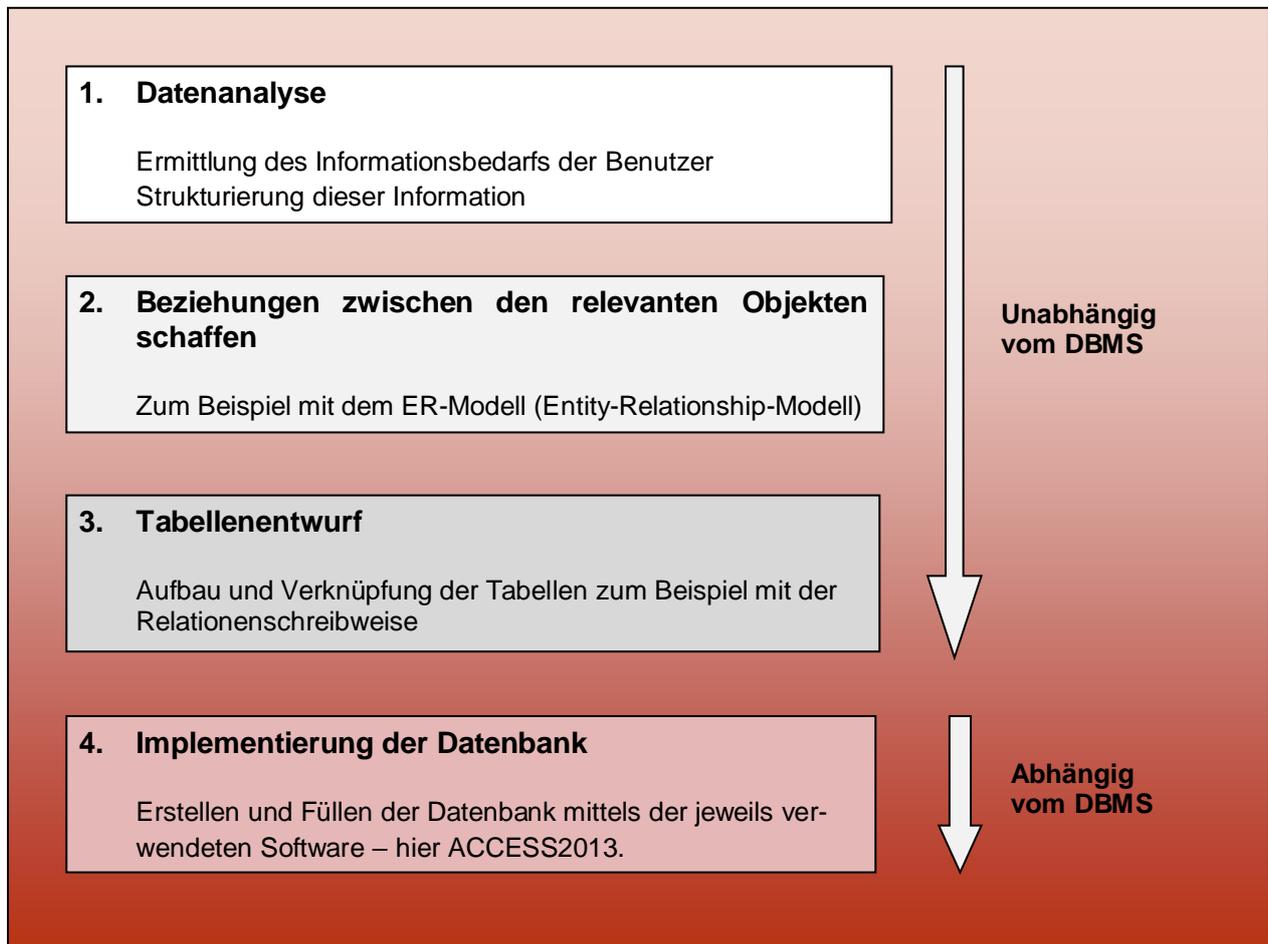
2.3 Entity-Relationship-Modell angewandt auf unterschiedliche Beziehungen – Fallbeispiele

2.4 Normalform

2.5 Anomalien in Datenbanken

2.6 Aufgaben zu Kapitel 2

2.1. Phasen beim Erstellen einer neuen Datenbank



Sammeln Sie zunächst alle relevanten Daten. Schon die Entscheidung, welche Daten relevant und nicht sind, gehört zur Datenanalyse.

Um eine Datenbank aufzubauen, muss ein möglichst exaktes Abbild dieser Daten definiert werden, ein Datenmodell. Dazu müssen alle relevanten Objekte und Beziehungen zwischen den Objekten in einem semantischen Datenmodell erfasst und beschrieben werden.

Danach werden die Beziehungen in den Tabellenentwurf übernommen, d.h. Primärschlüssel und Fremdschlüssel schaffen die Beziehungen zwischen den Tabellen untereinander.

Die Datenbank steht eigentlich schon auf dem Blatt Papier. Nun gilt es diese zu implementieren. In diesem Buch werden Sie es in ACCESS 2013 durchführen.

Fallbeispiel: Datenbank für Reparatur-Werkstatt erstellen

Um einen Überblick über die Aufträge und Kunden zu haben, wollen Sie eine benutzerfreundliche Datenbank erstellen. Entwickeln, die eine einfache Pflege der Daten für die Abwicklung von Kundenaufträgen erlaubt.

Schritt 1: Datenanalyse – Ermittlung des Informationsbedarfs

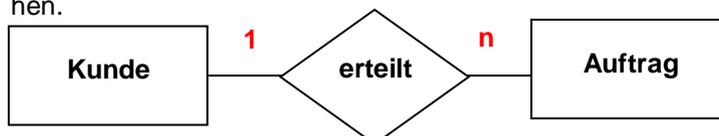
Auftragseingang und Benennung des Auftrags müssen bekannt sein genauso wie die genaue Kundenanschrift, um die Rechnung zu versenden.



Schritt 2: Erstellen des Entity-Relationship-Modells

ER-Modell

Sie haben für diesen Auftrag zwei Tabellen (d.h. zwei Identitätstypen) „Kunde“, „Auftrag“. Nun geht es darum, darzustellen, wie diese Identitätstypen zueinander in Beziehung stehen.



Dieses Modell liest sich wie folgt: Ein Kunde kann ein oder mehrere Aufträge vergeben. Aber zu einem Auftrag gehört immer nur ein Kunde. Man spricht: „Die Kardinalität zwischen Kunde und Auftrag ist 1 zu n“.

Schritt 3: Beschreibung der DB in Relationenschreibweise

Relationen - Schreibweise als Vorstufe zu Tabellen

Auftrag (Auftrags_Nr, KNr, Reparatur, Datum, Betrag)



Kunde (KNr, Nachname, Vorname, Strasse, Ort)

Legende: Primärschlüssel Fremdschlüssel

2.2. Entity-Relationship-Modell - Fallbeispiele

Begriffsbestimmung: Entitätstyp - Entität - Attribut

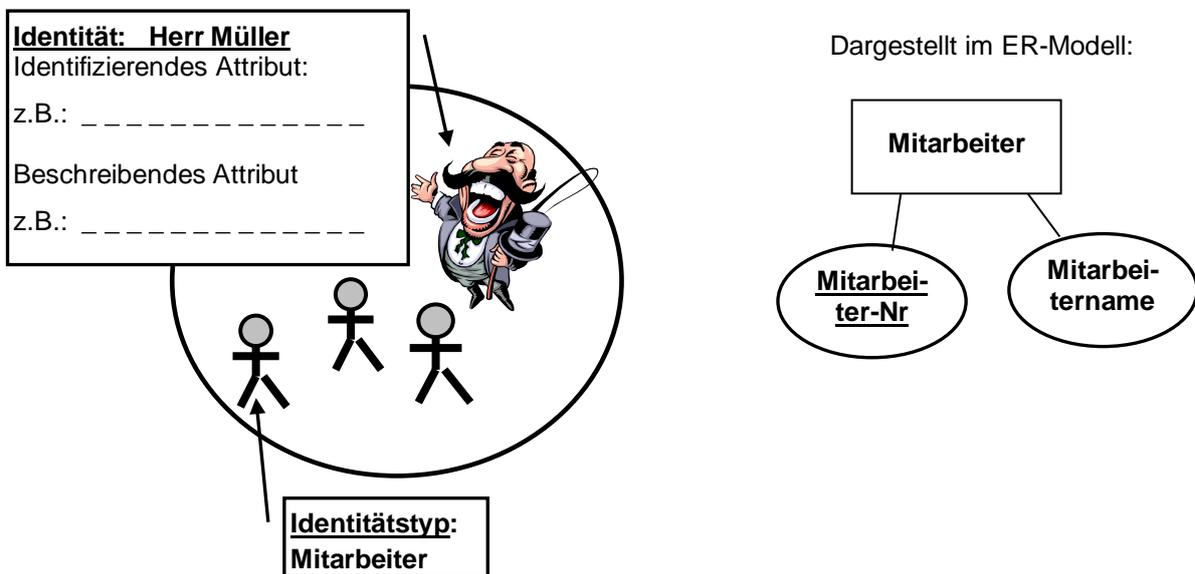
Die **Entität (Entity)** ist das konkrete, individuell identifizierbare Objekt bzw. Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt, für das auf einem Datenträger Sachverhalte festgehalten sind.

Der **Entitätstyp** ist die Typisierung gleichartiger Entitäten.

- z.B.:
- „Herr Müller“ ist die Identität, Mitarbeiter ist der Identitätstyp
 - „Auspuff“ ist die Identität, Auto-Teil ist der Identitätstyp

Attribute sind die Eigenschaften eines Entitätstyps. Man unterscheidet:

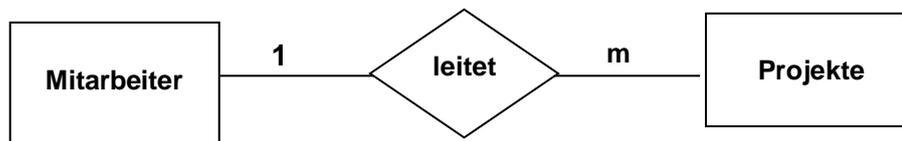
- Identifizierende Attribute (z.B. Kundennummer, Artikelnummer, Personennummer, ...)
- Beschreibende Attribute (z.B. Kundenname, Firmenname, Artikelbezeichnung, ...)



Beziehungstyp – konkrete Beziehung

Zwischen den Entitätstypen besteht eine Beziehung („relationship“). Dabei gibt es verschiedene **Beziehungstypen**.

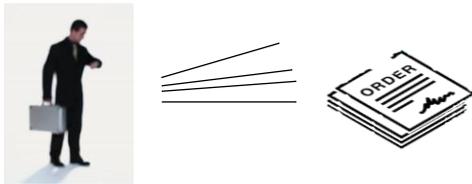
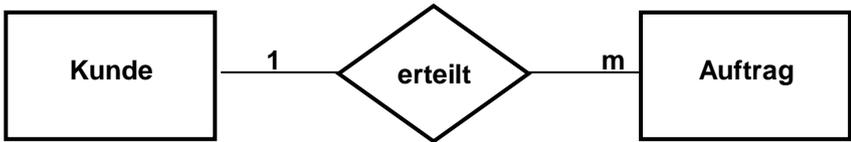
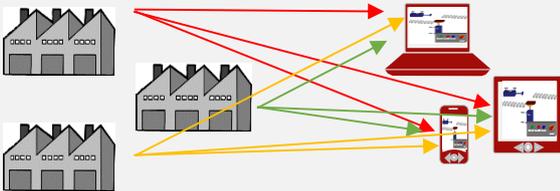
- z.B.: Ein Angestellter leitet mehrere Projekte, während ein Projekt von genau einem Angestellten geleitet. Hier besteht eine 1:M Beziehung.



Die mögliche Anzahl der an einer Beziehung beteiligten Entitäten kann unterschiedlich sein (wie im obigen Beispiel). Die **Kardinalität** beschreibt die mögliche Anzahl der an einer Beziehung beteiligten Entitäten. Hier: Ein Angestellter leitet mehrere Projekte.

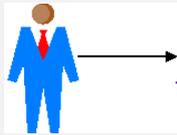
Die verschiedenen und für die Praxis wichtigen Beziehungstypen sind auf den folgenden Seiten dargestellt.

Entity-Relationship-Modell (ER-Modell): Kardinalität einer Beziehung

<p>Die 1:1 Beziehung</p>	<p>Zu jeder Entität von A gibt es genau eine Entität von B und umgekehrt.</p>
	<p>Zu einer Person gehört genau ein Ausweis. Umgekehrt gehört auch zu jedem Ausweis genau eine Person.</p>
<p>ER-Modell:</p>	
<p>Die 1:M Beziehung</p>	<p>Zu jeder Entität von A gibt es eine oder mehrere Entitäten von B. Umgekehrt gibt es zu jeder Entität von B nur eine Identität von A.</p>
	<p>Ein Kunde kann mehrere Aufträge erteilen. Umgekehrt gehört zu einem Auftrag genau ein Kunde.</p>
<p>ER-Modell:</p>	
<p>Die M:N Beziehung</p>	<p>Zu jeder Entität von A gibt es ein oder mehrere Entitäten von B und umgekehrt.</p>
	<p>In einer Fabrik werden n Produkte hergestellt. Zu jedem Produkt können mehrere Fabriken gehören (wenn Produkt aus mehreren Zulieferteile besteht).</p>
	

Die 1:C Beziehung

Es gibt aber auch Entitäten, zwischen denen eine Beziehung stehen **kann**. Für diesen Fall führt man eine konditionale (engl. **conditional** = bedingte) Beziehung ein. Auf der Seite, auf der die Beziehung nur bedingt besteht, wird ein „C“ geschrieben. „C“ kann die Werte „0“, also nicht erfüllt bzw. „1“, also erfüllt, annehmen.



Ein Mitarbeiter bekommt eventuell – abhängig von der Auftragslage seines Arbeitgebers - eine Lohnerhöhung.

ER-Modell:



Die 1:MC Beziehung

Analog der 1:C Beziehung ist es darüber hinaus auch zulässig, wenn es Objekte gibt, denen keine Entsprechung zugeordnet ist (MC steht also für „0...M“).



Eine Person kann beliebig viele Autos besitzen (oder keines).

ER-Modell:

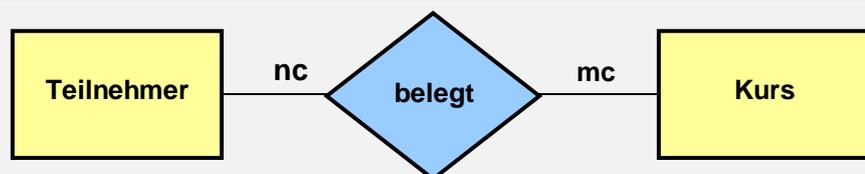


Die MC:NC Beziehung

Es ist auch möglich, dass einer Entitätsmenge aus der N:M Beziehung kein, ein oder mehrere Elemente aus der anderen Entitätsmenge zugeordnet. In diesem Fall steht das Konditionszeichen „C“.

(Bei einer Volkshochschule wäre diese Darstellung richtig, wenn ein Teilnehmer auch dann in der Datenbank verbleibt, wenn er momentan keine Kurse belegt, bzw. wenn ein Kurs auch noch gespeichert bleibt auch wenn dieser gerade nicht durchgeführt wird.)

ER-Modell:



Merke: In relationalen Datenbanken führt eine M:N Beziehung zu redundanten Informationen; damit ist die Konsistenz des Datenbestandes gefährdet!
Eine M:N – Beziehung ist nicht darstellbar!!

Für die Auflösung von N:M Beziehungen ist eine Beziehungsentität (d.h. „Beziehungstabelle“) erforderlich. Im Relationalen-Modell nimmt dabei die Tabelle, welche die Beziehungsentität darstellt, die Primärschlüssel der beteiligten Entitätstypen (hier Teilnehmer-Nr. und Kurs-Nr.) als Fremdschlüssel auf. Die beiden Fremdschlüssel zusammen ergeben häufig den Primärschlüssel in der neuen Tabelle.

Erläuterung am Beispiel Verwaltung ‚Kursbelegung an einer Hochschule‘:

Fall 1: Ein Schüler belegt mehrere Fächer

Tabelle: (n) Schüler

Teiln_Nr	Name	Vorname	KursNr
2001	Hugendubel	Michael	100
2001	Hugendubel	Michael	101
2002	Müller	Franz	102

Tabelle: (m) Fächer

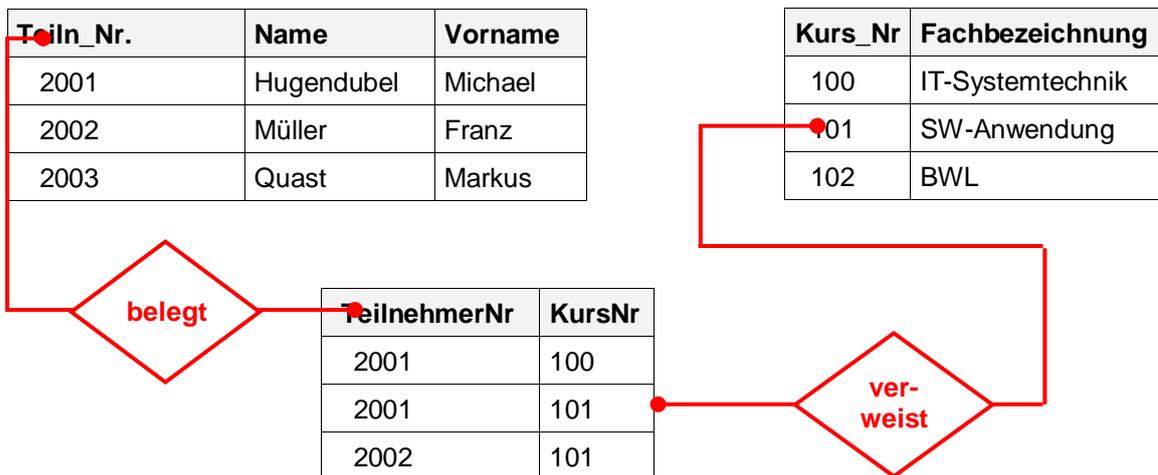
Kurs_Nr	Fachbezeichnung	Beginn
100	IT-Systemtechnik	10.04
101	SW-Anwendung	12.03
102	BWL	12.03



Welche redundante Information ist hier enthalten? Welches Problem ergibt sich hierbei?

Michael Hugendubel belegt in der Tabelle ‚Schüler‘ mehrere Positionen. Wenn er austritt oder seinen Namen ändert, dann muss die ganze Tabelle nach ihm durchsucht werden.

Lösung: Auflösung der M:N Beziehung im Relationalen-Modell durch Einfügen einer Beziehungstabelle (Beziehungsrelation, Beziehungsentität):



ER-Modell:



Ein Teilnehmer ist mehrfach eingetragen in eine Belegungsliste. Alle Kurse der Belegungsliste haben jeweils nur eine Kurzbeschreibung.

Beziehungen und referentielle Integrität einer relationalen Datenbank

Bedeutung des Wortes ‚Referenz‘:

Im Zusammenhang mit relationalen Datenbanken ist „Referenz“ nur ein anderes Wort für die Beziehung zwischen zwei Tabellen (bzw. Entitäten), die im ER-Modell mit Hilfe einer Raute symbolisiert wird.

Integrität einer Datenbank

Integrität bedeutet im normalen Sprachgebrauch so viel wie Unbescholtenheit oder Unverletzlichkeit. Eine Person ist integer, wenn sie nicht korrupt ist. Eine Datenbank besitzt Integrität, wenn ihr Datenbestand nicht korrupt ist oder inkonsistent, also unregelmäßig oder „unsauber“ ist.

Verletzung der referentiellen Integrität

Die referentielle Integrität einer Datenbank wird z.B. dann verletzt, wenn in einer Detailtabelle ein Datensatz vorhanden ist, für den in der Haupttabelle kein entsprechender Eintrag existiert. Beispiel:

		Kunde				
Haupttabelle	KundeNr	Nachname	Vorname	Straße	PLZ	Ort
	1	Aumeister	Anton	Meranerstr. 16	81455	Dinkelshofen
	1	Kessidis	Karl	Blumenstr. 5	71533	Bad Limburg
	3	Müller	Martin	Nansenstr. 66	86338	Mering
	4	Kein Eintrag				
	...					

		Auftrag				
Detailtabelle	AuftragsNr	KNr	ArtikelNr	Menge	Auftragsdatum	Sachbearb.
	100	2	2	1	8.4.2009	Hg
	101	1	3	3	12.5.2009	Sr
Dieser Eintrag verletzt die referentielle Integrität	102	4	1	4	13.7.2009	Mü
	103	3	4	4	20.12.2009	Dm
	...					

Ein solcher korrupter oder inkonsistenter Zustand kann dadurch entstehen, dass entweder eine falsche Kundennummer beim Erstellen eines neuen Auftrags verwendet wurde, oder dass in der Zwischenzeit der entsprechende Datensatz aus der Haupttabelle „Kunden“ gelöscht wurde.

Prüfung der referentiellen Integrität durch das DBMS

MS-Access (und auch andere DBMS) besitzt die Funktionalität, dafür zu sorgen, dass die referentielle Integrität zwischen einer Haupt- und einer Detailtabelle nicht verletzt wird. Wird bei der Erstellung einer Beziehung zwischen Zwei Tabellen die Eigenschaft „mit referentiellen Integrität“ ausgewählt, so bringt MS-Access eine Fehlermeldung, sobald der Benutzer versucht, einen inkonsistenten Zustand herzustellen.

2.3. Normalformen 1 bis 3 für relationale Datenbanken

Sie haben bereits erste Erfahrung mit relationalen Datenbanken gemacht. Sie haben eine große Datenbasis mit Hilfe von mehreren Tabellen zerlegt. Dabei mussten Sie die Datenbank so entwickeln, dass diese Datenbank keine Redundanz enthält.

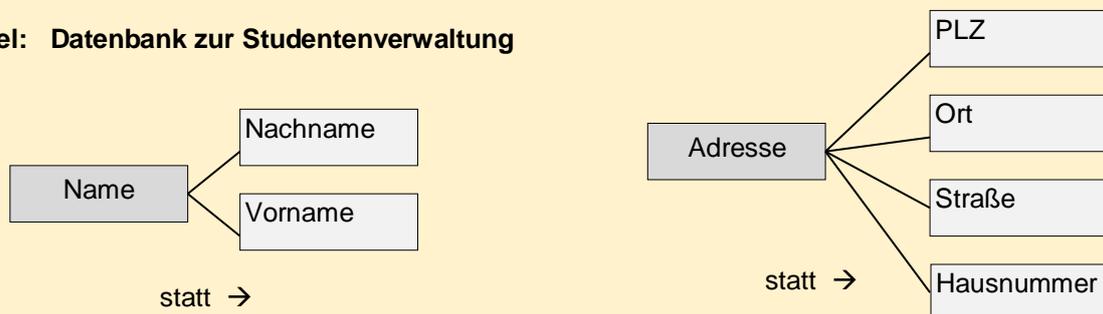
Mit Hilfe von sogenannten Normalformen (NF) wird eine Tabelle, die Redundanzen enthält, in ein System von mehreren Tabellen zerlegt, die über gemeinsame Datenfelder in Beziehung zueinander stehen.

Erste Normalform (1NF)

Eine Relation (Tabelle) ist in der ersten Normalform, wenn alle Datenfelder mit zusammengesetzten Inhalten in ihre einzelnen Bestandteile zerlegt sind.

D.h. alle Attributwerte (Merkmalswerte) sind nicht weiter zerlegbar. Man sagt, sie sind atomar.

Beispiel: Datenbank zur Studentenverwaltung

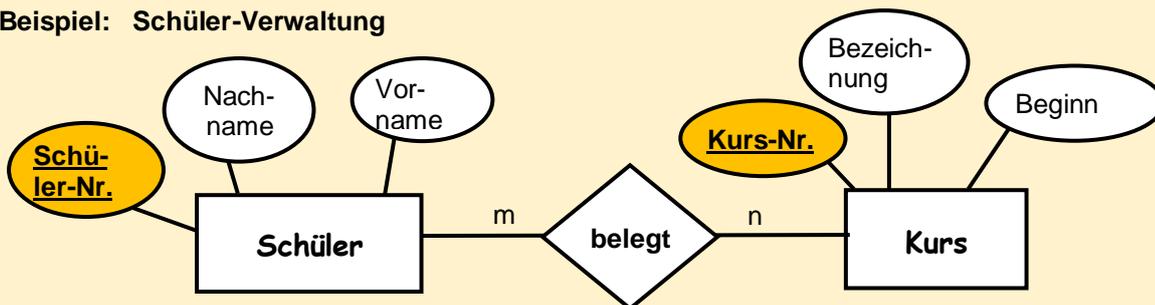


Zweite Normalform (2NF)

Eine Relation (Tabelle) ist in der zweiten Normalform, wenn sämtliche Datenspalten ausschließlich vom Primärschlüssel abhängen

Ist eine Datenspalte nicht vom Primärschlüssel oder nur von einem Teil des Primärschlüssels abhängig, dann muss für diese Datenspalte eine neue Tabelle eingerichtet werden.

Beispiel: Schüler-Verwaltung



<u>Schüler-Nr.</u>	Nachname	Vorname
<u>Zahl</u>	Text	Text
2001	Huendubel	Michael
2002	Müller	Franz
2003

<u>Kurs-Nr.</u>	Fachbezeichnung
<u>Zahl</u>	Text
100	IT-Systemtechnik
101	SW-Anwendung
102	BWL

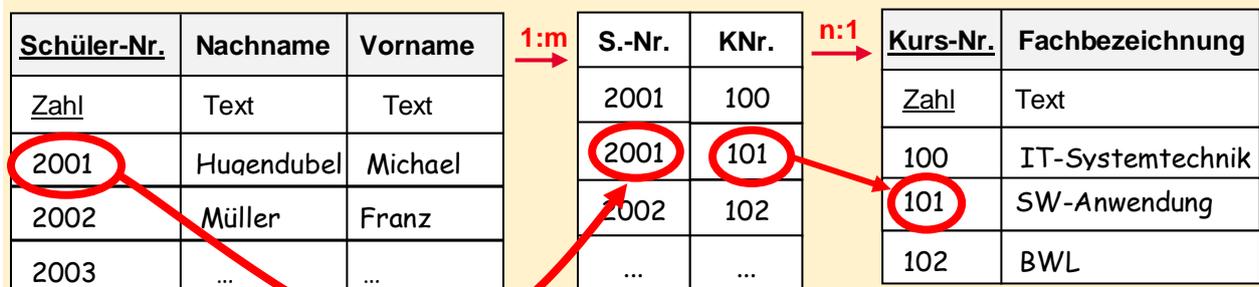
Tabelle erfüllt die 2. Normalform!

Dritte Normalform (3NF)

In der 3. Normalform dürfen in einer Tabelle nur Attribute stehen, die untereinander direkt abhängig sind. Attribute, die innerhalb der Tabelle nicht abhängig sind, werden in eine eigene Tabelle ausgelagert.

Beispiel: Schüler-Verwaltung

In der 2. Tabelle ist das Attribut ‚K-Nr‘ nicht abhängig von den restlichen Attributen. Durch Auslagerung in eine eigene Tabelle wird nebenbei eine m:n – Beziehung aufgelöst.



2.4. Anomalien von Datenbanken

Es gibt drei Anomalien, die bei einem unsauber realisierten Datenbanken-Entwurf auftauchen können: -
 Löschanomalie

- Einfügeanomalie
- Aktualisierungsanomalie / Änderungsanomalie

Erklärung am folgenden Beispiel:

Lehrerkürzel	Lehrername	Fachname	Fachleiterkürzel
Hei	Heine	Informatik	Pr
Hei	Heine	Mathe	Pr
Zwi	Zwitech	Informatik	Al
Sc	Schöllner	Musik	Al
Wg	Wagner	Deutsch	Mo
Mü	Müller	Deutsch	Mo

 Was kann man zu dieser Tabelle sagen?

Die Tabelle enthält viele Redundanzen.

 Wie wirkt sich dies auf die Datenintegrität aus?

Die Datenbank zu pflegen, wird schwerer. Wenn z.B. ein Lehrer seinen Nachnamen wechselt, dann muss die ganze Tabelle nach diesem Lehrer durchsucht werden.

Einfügeanomalie

... ein neues Fach ‚Kunst‘ eingeführt werden soll, jedoch dafür kein Lehrer verfügbar ist? Nun – das Fach kann nicht eingeführt werden. Man muss warten, bis ein Lehrer einsetzbar ist. Anders ausgedrückt: dem Fach ‚Kunst‘ kann **kein** Primärschlüssel (hier: Lehrerkürzel) eingefügt werden.

Löschanomalie

... Lehrer Schöllner pensioniert wird. Wenn Schöllner die Schule verlässt, dann muss das Fach Musik gelöscht werden. Außerdem geht die Information verloren, dass ‚Al‘ Fachleiter für Musik ist.

Allgemein ausgedrückt: Bei der Löschanomalie kann es passieren, dass beim bewussten Löschen eines Datensatzes unbewusst Daten verloren gehen, die man später benötigen könnte.

Änderungs- / Updateanomalie

... Fiiifkus (‚FI‘) anstatt ‚Pr‘ Fachleiter für Informatik und Mathe wird, dann müssen Änderungen in zwei Datensätzen gemacht werden.

Allgemein ausgedrückt: Es entstehen Inkonsistenz von Daten. Man muss per Hand alle Einträge mühsam aktualisieren; es darf kein Fehler unterlaufen, ansonsten führt es zu Inkonsistenz.

2.5. Aufgaben zu Kapitel 2

Lösungen Siehe im Anhang..

Aufgabe 1:

Erstellen Sie ein ER-Modell zu den nachfolgenden Beziehungen!

Trainer - Mannschaft	Firma – Mitarbeiter
Schiedsrichter – Spiel	Raum – Veranstaltungen
Mannschaft - Spieler	Finger – Hand

Aufgabe 2:

In der zentralen KFZ-Datenbank in Flensburg werden zu jedem anmeldeten KFZ verschiedene Informationen gespeichert. Unter anderem sind dort Informationen zum Fahrzeughalter und zum Hersteller eines Fahrzeuges gespeichert.

Zeichnen Sie für die Entitäten **KFZ, Hersteller und Fahrzeughalter** das **Entity-Relationship-Diagramm (ERD)** und vergessen Sie nicht die entsprechenden Kardinalitäten anzugeben.

Aufgabe 3:

Für eine kleine Firma soll ein Datenbankschema entworfen werden. Die Firma ist in mehreren Abteilungen unterteilt. Jeder Mitarbeiter der Firma gehört einer der existierenden Abteilungen an. Die Eigenschaft der Firma ist, dass die Mitarbeiter ihre Tätigkeit in verschiedene Projekte ausüben. Jeder Mitarbeiter kann in verschiedenen Projekten arbeiten und dabei unterschiedliche Aufgaben wahrnehmen.

- 3.1 Zeichnen Sie das entsprechende Entity-Relationship-Modell. Lösen Sie falls notwendig alle m:n Beziehungen in das relationale Modell auf.
- 3.2 Bestimmen Sie die Attribute der Entitäten und kennzeichnen Sie alle Primärschlüssel und alle Fremdschlüssel einer Entität.
- 3.3 Beschreiben Sie die Tabellen mit Hilfe der Relationen-Schreibweise. Kennzeichnen Sie alle Primärschlüssel und alle Fremdschlüssel einer Entität.

Aufgabe 4:

Für eine Tanzschule soll ein Datenmodell entwickelt werden. Die Tanzschule besitzt mehrere Tanzkurse und eine Vielzahl von Tanzschülern unterschiedlicher Altersgruppen. Jeder Tanzschüler hat eine Mitgliedsnummer, die auf seinen Mitgliedsausweis aufgedruckt ist.

Die Tanzschule wird von verschiedenen Tanzlehrern angehalten. Manche Tanzkurse werden mit mehreren Tanzlehrern durchgeführt.

- 4.1 Bestimmen Sie für das Datenmodell der Tanzschule die Entitäten und deren Attribute. Die 1. Normalform soll erfüllt sein.
- 4.2 Zeichnen Sie das Entity-Relationship-Modell (ERM) und lösen Sie darin die m:n-Beziehung auf.
- 4.3 beschreiben Sie die Tabellen mit Hilfe der Relationen-Schreibweise. Kennzeichnen Sie alle Primärschlüssel und alle Fremdschlüssel in eindeutiger Weise!

Aufgabe 6:

Welche Aufgabe hat ein Primärschlüssel in Access?

3. Kapitel: Arbeiten mit Access 2013

3.1 Anlegen einer Datenbank

3.2 Objekte in ACCESS2013

3.3 Tabellen erstellen

3.4 Tabellen in Beziehung setzen

3.5 Abfragen

3.5.1 Erstellen einer einfachen Abfrage

3.5.2 Kreuzabfrage

3.6 Formulare

3.6.1 Formulare erstellen in der Entwurfsansicht

3.6.2 Unterformulare – anwenden und erstellen

3.7 Steuerelemente in Formulare

3.7.1 Steuerelemente mit Makros

3.7.2 Steuerelemente mit VBA

3.8 VBA in ACCESS

3.9 Berichte

3.10 Datenaustausch nach Excel und zurück

3.11 Aufgaben zu Kapitel 3

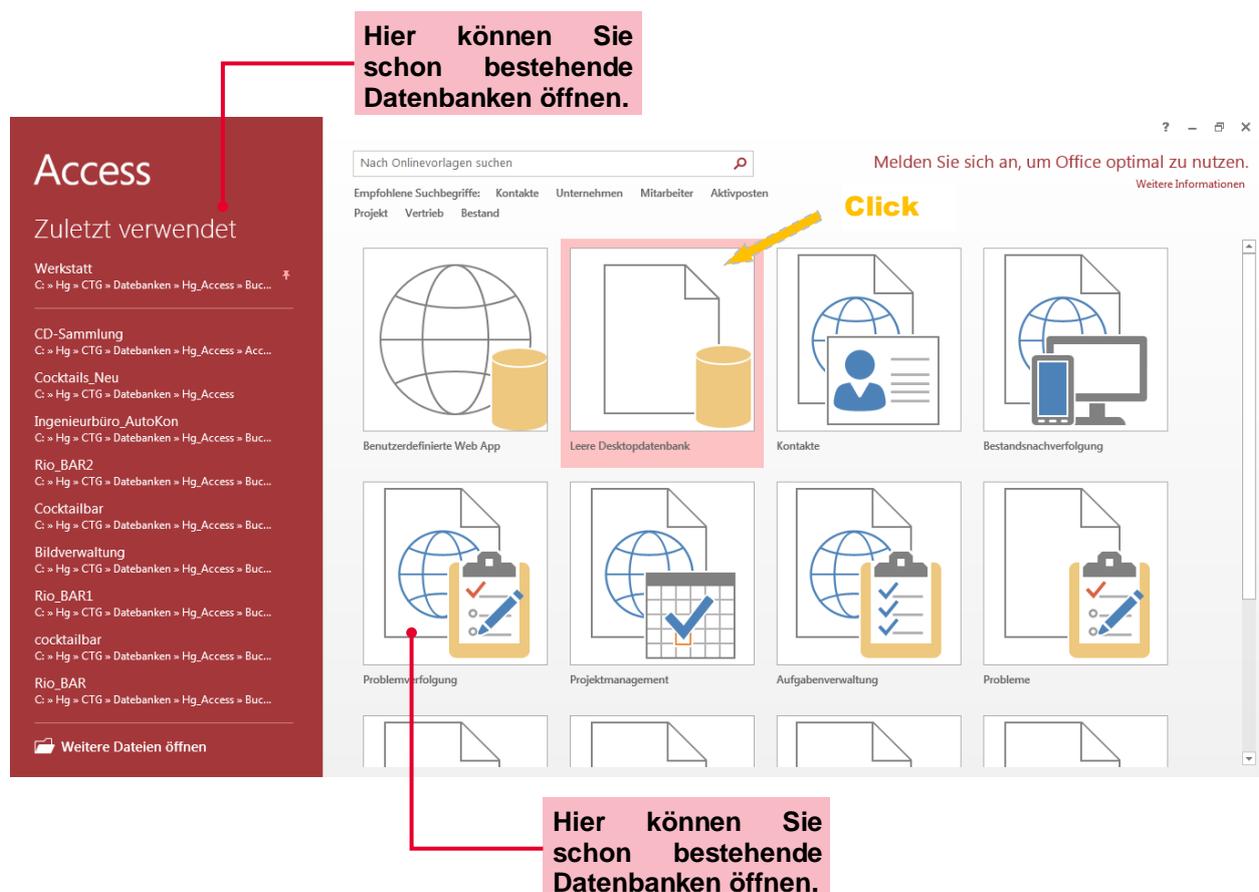
3.1. Anlegen einer Datenbank in ACCESS2013

Sie starten ACCESS in der Windows-Oberfläche mittels Mausklick auf das nebenstehende Icon. Sofort werden Sie von ACCESS gefragt, ob Sie eine bestehende Datenbank öffnen möchten oder eine Neue erstellen wollen.



Falls Sie eine neue Datenbank erstellen wollen, haben Sie die Möglichkeit, Vorlagen zu benutzen, welche ACCESS anbietet.

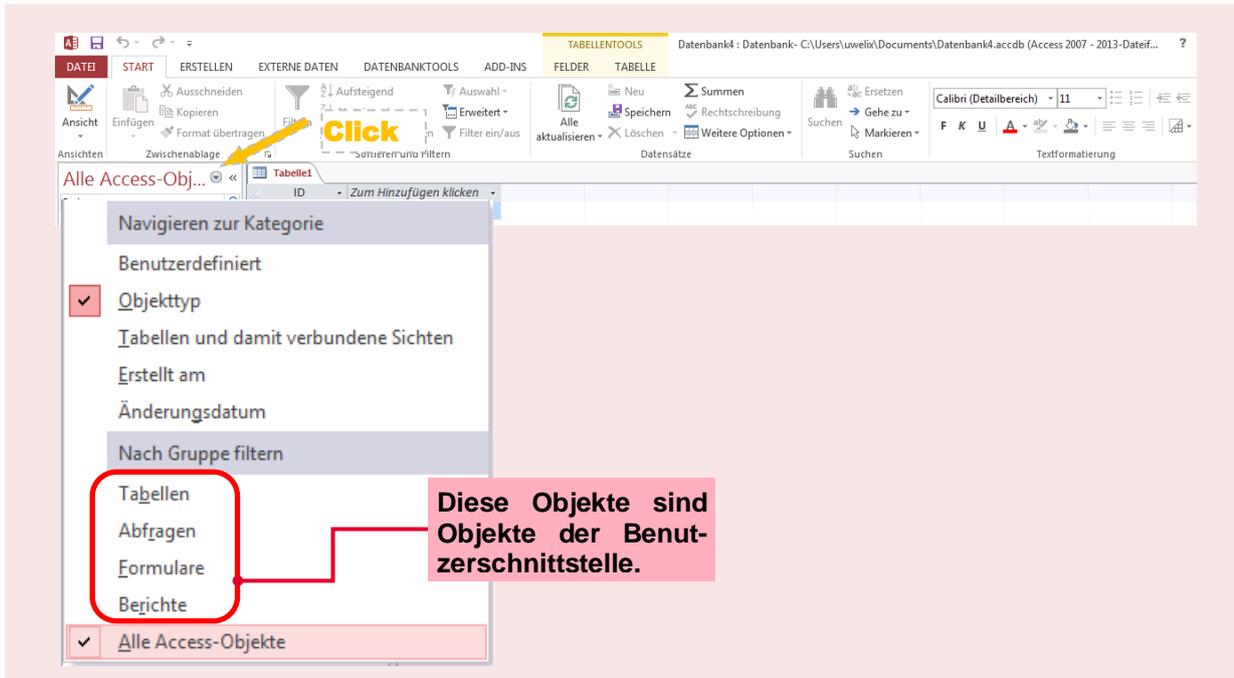
Da aber Vorlagen oft nicht genau den Bedürfnissen des Nutzers entsprechen und zum Anpassen der Vorlagen auch Kenntnisse über ACCESS benötigt werden, soll an dieser Stelle eine neue Datenbank erstellt werden. Zudem soll das Lernziel dieses Kapitels sein, den Aufbau und Erstellen einer Datenbank anhand von ACCESS zu verstehen.



Im Folgenden müssen Sie selbstverständlich noch einen Datennamen vergeben und den Ablageort (d.h. Pfad, unter welchem die Datenbank zu finden ist) angeben.

3.2. Objekte in ACCESS2013

Zunächst soll ein Überblick gegeben werden, welche Objekte ACCESS anbietet und z.T. auch notwendig sind, um eine Datenbank einzurichten. Diese hier genannten Objekte werden Sie genauer in den folgenden Kapiteln kennenlernen.



Objekte einer Access-Datenbank:

- Tabellen:** Sammlung von Daten
- Abfragen:** Auswahl bestimmter Daten einer / mehrerer Tabellen. Sie können Sortierungen und Filter angeben.
- Formulare:** Darstellung von Daten in ansprechender Form mittels Masken. Objekte, wie Buttons und Auswahlfenster können Ereignisse zugeordnet werden.
- Berichte:** Präsentation der Daten für die Ausgabe (Drucker)

Zusätzlich können Sie weitere Objekte anlegen bzw. programmieren, was auch in anderen Microsoft-Programmen wie Excel, etc. möglich ist:

- Makros:** Programme in Stapelform (Batch)
- Module/Subroutines:** Programme in einer höheren Programmiersprache (VBA = „Visual Basic für Anwendungen“)

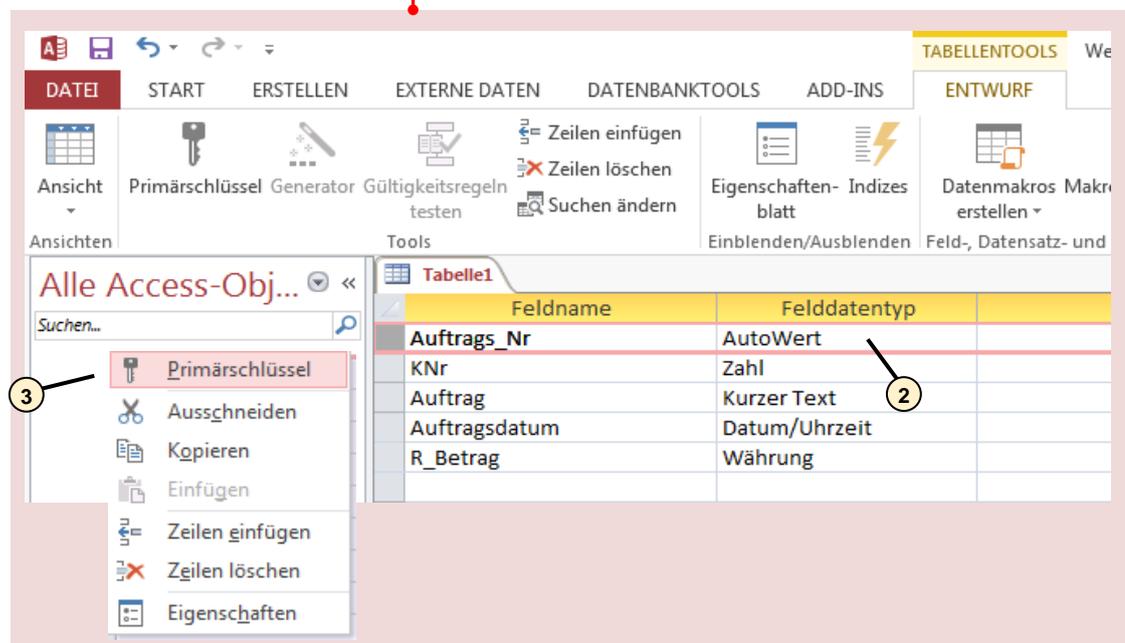
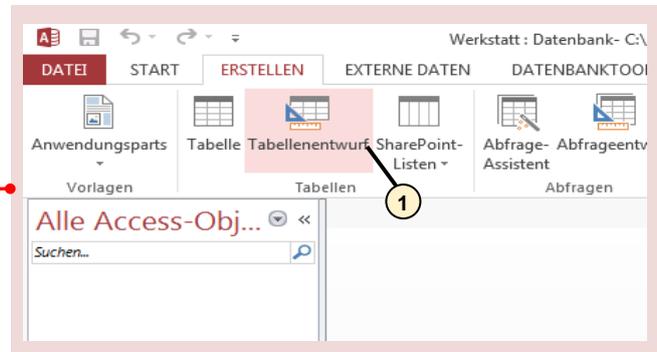


Im Weiteren sollen diese Objekte am praktischen Beispiel kennengelernt werden. Die Datenbank hierzu findet sich auf der **Lösungs-CD** unter dem Namen **„Werkstatt“**.

3.3. Tabellen erstellen

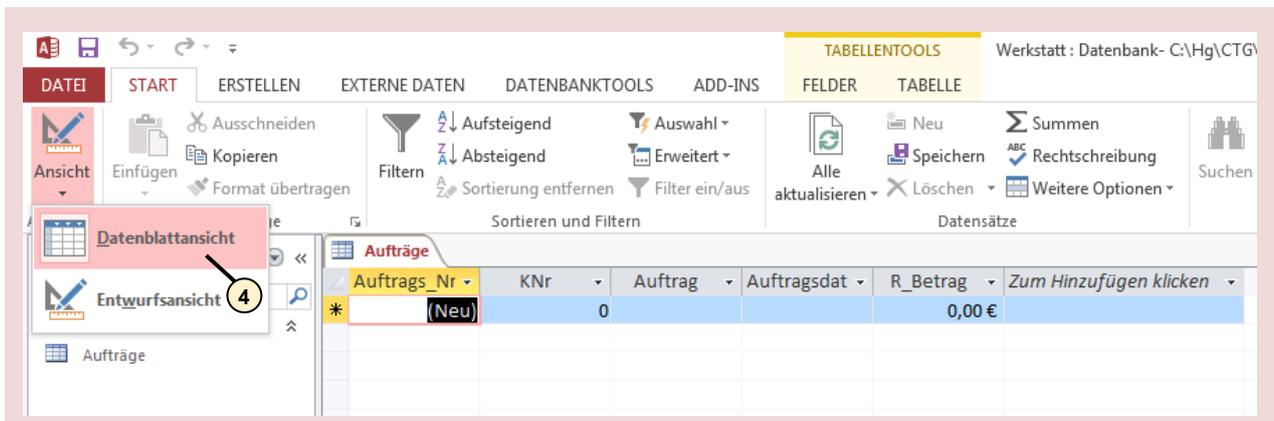
Hier soll eine Datenbank entwickelt werden für eine Auto-Werkstatt. D.h. für die Kundendaten sowie Rechnungen soll jeweils eine eigene Tabelle erstellt werden. Diese Datenbank finden Sie in der Lösungs-CD unter dem Namen ‚Werkstatt‘.

- 1 Erstellen Sie ein neues (leeres) Datenblatt und speichern Sie diese unter dem Namen ‚Werkstatt‘. Zunächst müssen Sie die Tabelle ‚Aufträge‘ in der Entwurfsansicht erstellen.



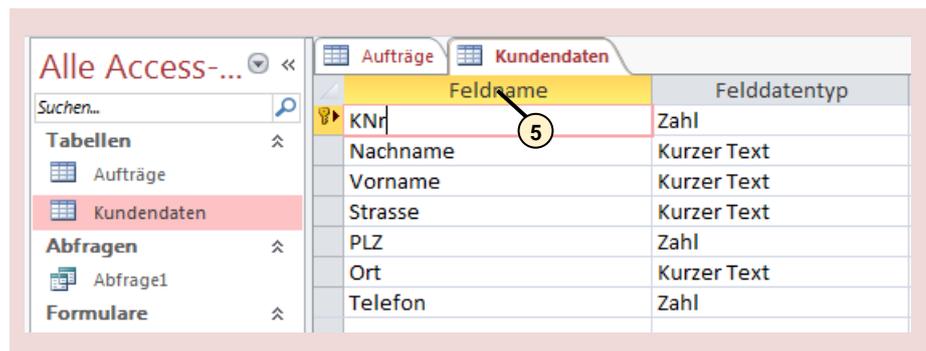
- 2 Die Feldnamen sind wie abgebildet zu sehen. Das Feld ‚Auftrags_Nr‘ soll den Primärschlüssel bekommen. Hierzu wählen Sie dieses Feld an und öffnen das Kontextmenü (rechte Maustaste).
- 3 Mit einem Mausklick vergeben Sie den Primärschlüssel.

Um konkrete Daten einzugeben, müssen Sie in die Datenblattansicht wechseln. ACCESS verlangt, dass Sie zuerst die Tabelle speichern. Vergeben Sie dieser Tabelle den Namen ‚Aufträge‘. Füllen Sie nun die Tabelle mit Daten.



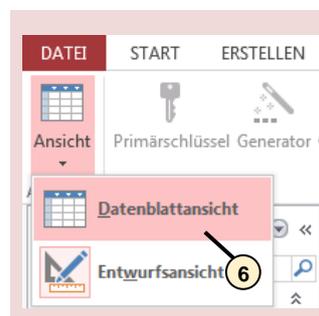
- ④ Zur Eingabe von konkreten Daten in die Tabelle muss in die Datenblattansicht umgeschaltet werden:

Sie müssen noch eine zweite Tabelle für diese Datenbank haben. Erstellen Sie die Tabelle ‚Kunde‘ nach folgender Entwurfsansicht:



- ⑤ Der Kundennummer (abgekürzt mit ‚KNr‘) soll als identifizierende Eigenschaft den Primärschlüssel vergeben werden.

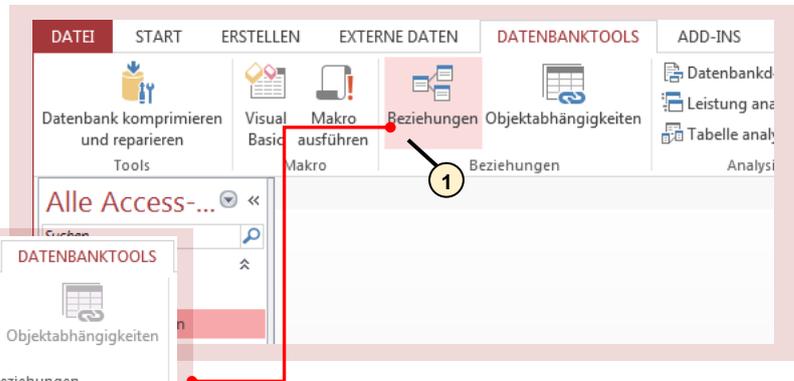
- ⑥ Speichern Sie diese Tabelle unter ‚Kundendaten‘ und wechseln Sie die Ansicht in ‚Datenblattansicht‘. Füllen Sie nun auch diese Tabelle mit konkreten Werten.



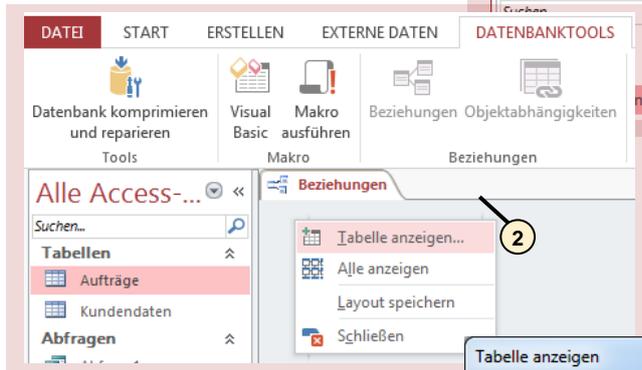
3.4. Tabellen in Beziehung setzen

Um abfragen zu können, welcher Auftrag zu welchem Kunden gehört und umgekehrt, muss die Auftrags-tabelle mit der Kundentabelle in Beziehung – sprich Relation – gesetzt werden. Nicht umsonst wird Access auch als ‚Relationale Datenbank‘ bezeichnet.

- 1 Wandern Sie in die Registerkarte ‚DATENBANKTOOL‘ und klicken Sie auf die Schaltfläche ‚Beziehungen‘.



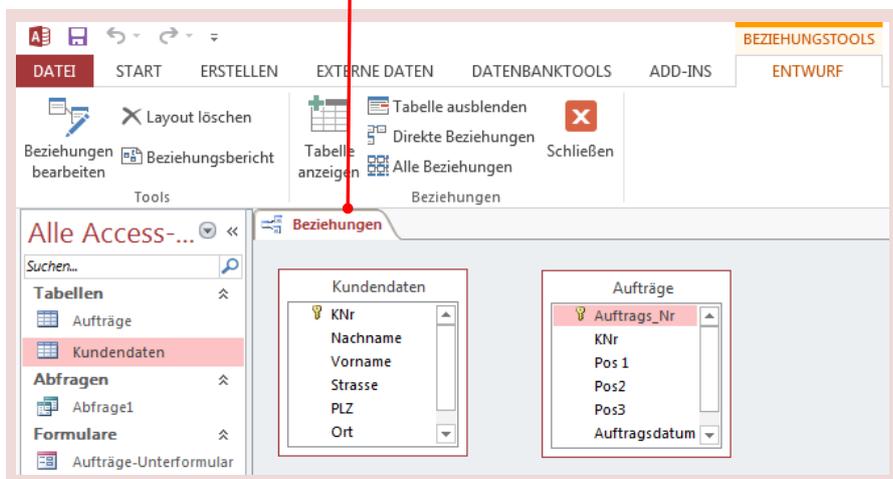
- 2 Es öffnet sich das Fenster für die Beziehungen. Allerdings wie können Sie die relevanten Tabellen auswählen? Klicken Sie am einfachsten die rechte Maustaste.



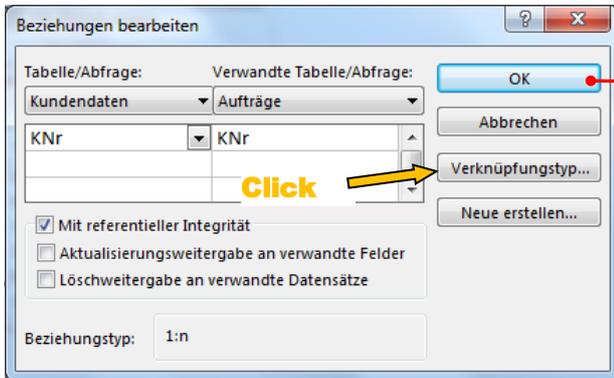
- 3 In dem sich nun öffneten Auswahlfenster wählen Sie die Tabellen, welche Sie in Beziehung setzen wollen.



Die zwei Tabellen, welche zu wählen sind, erscheinen nun in einem Beziehungsfenster. Allerdings noch ohne Beziehung.



Klicken Sie auf das Feld ‚KNr‘ der Kundentabelle und wandern Sie mit gedrückter linker Maustaste in die Auftrags-tabelle auf das Feld ‚Auftrags_Nr‘. Eine Beziehung wird nun erteilt. Aber welche Eigenschaften wollen Sie dieser Beziehung vergeben? Hierzu öffnet sich folgendes Dialogfenster:



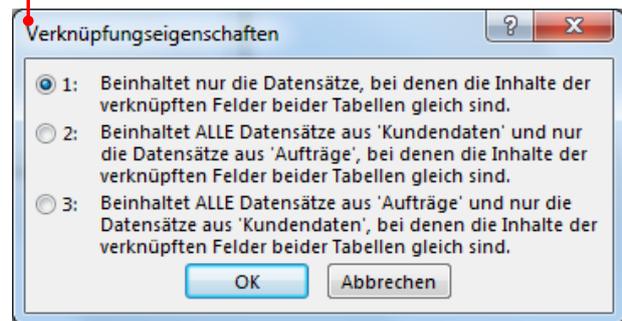
Sie haben nun weitere Optionen, wie diese Beziehung setzen. Mit der Option ‚Mit referentieller Integrität‘ prüft die Datenbank, ob Ihr Datensatz Schlüssig ist. Z.B. ist die Integrität nicht gegeben, wenn zu einem Auftrag kein Kunde gehört.

- Standardmäßig werden nur Datensätze angezeigt, bei denen der Inhalt der verknüpften Felder gleich ist.

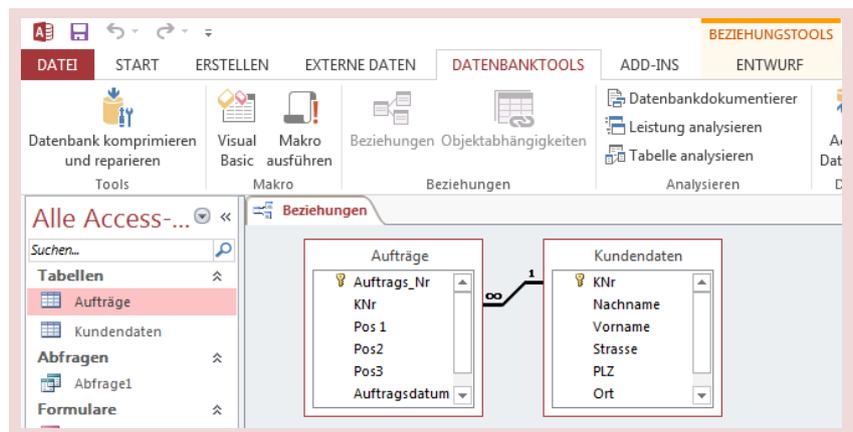
Bsp.: Es existiert eine Kdr 2 in der Auftrags-tabelle und in der Kundentabelle.

Wenn nun aber die Datenbank nicht gepflegt wird, dann kann es passieren, dass ein Datensatz aus Tabelle ‚Auftrag‘ existiert, aber in der Tabelle ‚Kundendaten‘ nicht mehr (z.B. Kunde ist verstorben und wurde deshalb gelöscht). Dieser Datensatz kann mit der Option 1 nicht angezeigt werden.

- Wenn Sie nun auch diese Datensätze anzeigen wollen, dann wählen Sie Option 2.



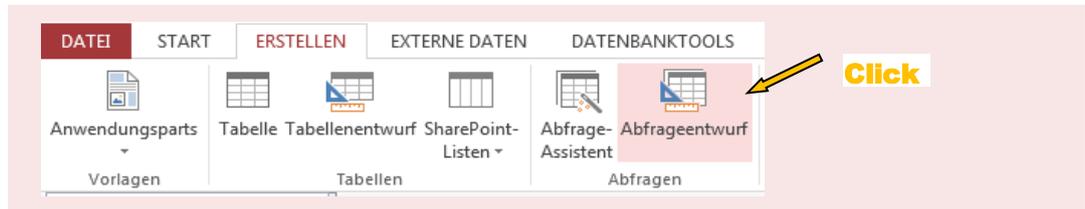
Mit der Wahl der optionalen Integrität ist eine 1:n – Beziehung erschaffen worden, d.h. ein Kunde kann 1 oder mehrere Aufträge erteilt haben.



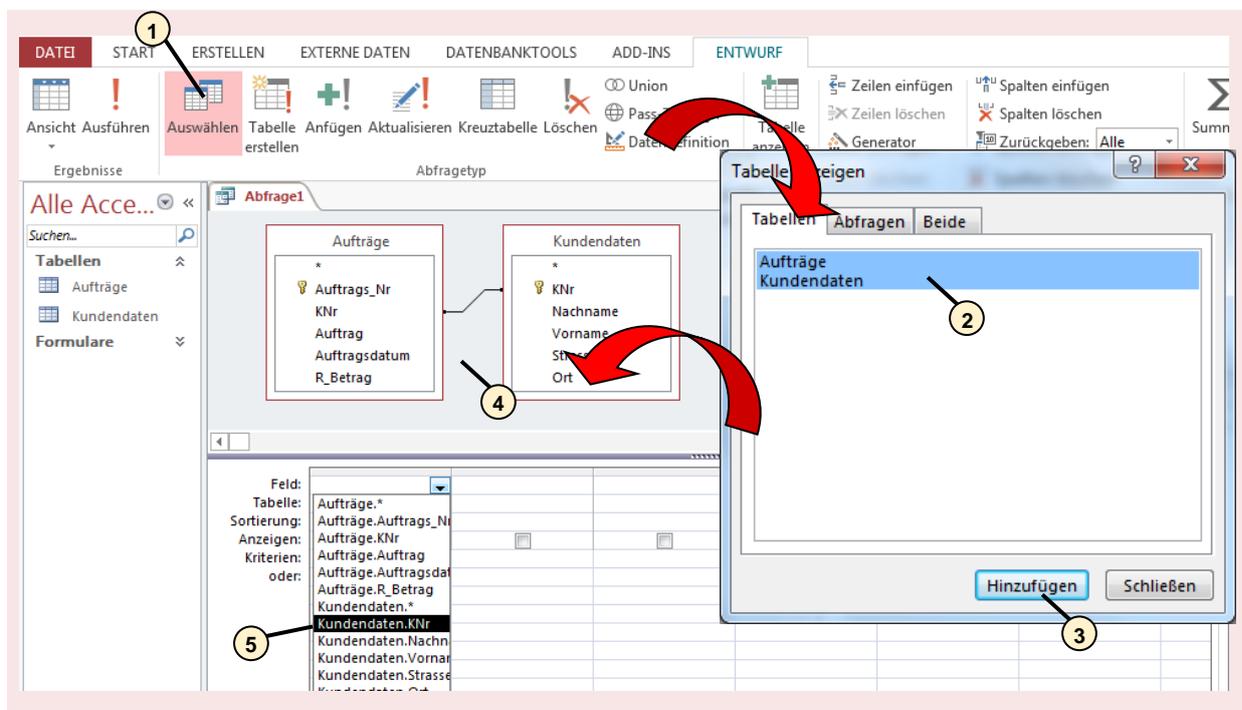
3.5. Abfragen erstellen

Abfrageentwurf

Nun mag es für manchen Anwender von Nachteil sein, wenn die gewünschten Informationen auf verschiedenen Tabellen verteilt sind. So ist der Buchhaltung für die Rechnungserstellung unwichtig sein, welche Auftragsnummer sich hinter dem ‚Ölwechsel‘ verbirgt, für den Werkstattmeister aber nicht.



Es ist nun mühsam, die einzelnen Tabellen durchzuschauen, um die gewünschte Information zu erhalten. Hier setzen Sie Abfragen ein - **Abfragen dienen zur Auswertung von Tabellen**. In der Entwurfsansicht einer sogenannten Abfrage können Sie nun selbst wählen, welche Informationen für den Nutzer der Datenbank angezeigt werden sollen.



- ① Die Daten haben Sie schon vorher in Tabellen abgespeichert. Genau auf diese Tabellen müssen Sie wieder zugreifen, wenn Sie eine Abfrage starten möchten.
- ② Um eine Abfrage auf mehreren Tabellen zu starten, müssen die Tabellen miteinander verknüpft sein. Üblicherweise haben Sie die Tabellen im Vorfeld schon in Beziehung gesetzt (Siehe Kapitel 3.4). Falls die Tabellen noch nicht in Beziehung stehen, können Sie dies auch im Abfrage-Fenster wie oben abgebildet durchführen. Hier wählen Sie die Tabellen an.
- ③ Die gewählten Tabellen im Dialogfenster bestätigen.
- ④ Es erscheinen die Tabellen mit ihren Verknüpfungen werden
- ⑤ Hier wählen Sie die gewünschten Daten.

Im Entwurfsmodus wählen Sie die gewünschten Spalten, welche in der Abfrage dann angezeigt werden. Sie sehen, dass Sie aus beiden Tabellen wählen können.

Hier entscheiden Sie im Nachhinein, ob die Datensätze angezeigt werden.

In der Datenblattansicht haben Sie die aktuellen Datensätze sichtbar.

Auftragsdat	Nachname	Vorname
01.05.2013	Kummer	Simon
22.06.2012	Kummer	Simon
08.04.2013	Müller	Martin
20.01.2013	Hofmann	Andreas

3.5.1. Kreuzabfragen erstellen

Kreuztabellenabfragen dienen zur Gruppierung und Sortierung von Daten sowie zur Anwendung von Funktionen wie z. B. der Berechnung von Summen, Durchschnittswerten oder Anzahlen. Der Unterschied zu herkömmlichen Abfragen ist der, dass Kreuztabellenabfragen Spalten- und Zeilenköpfe als Kriterien für die in einem Raster angeordneten Daten verwenden.

Dabei können Sie selbst die Spalten- und Zeilenköpfe festzulegen. Die gewünschte Information liegt - als Kreuzungspunkt der Spalte und der Zeile - in angepasster Form vor. Der einfachste Weg, eine Kreuzabfrage zu erstellen, geht über den Abfrage-Assistenten. Die weiteren Arbeitsschritte sollen anhand einem konkreten Beispiel dargestellt werden.

Arbeitsauftrag: Kreuzabfrage für Cocktailbar, in welcher Sie die Drinks nach Typ (Alkoholisch, nicht alkoholisch, Café-Getränk) abfragen können.

Neue Abfrage

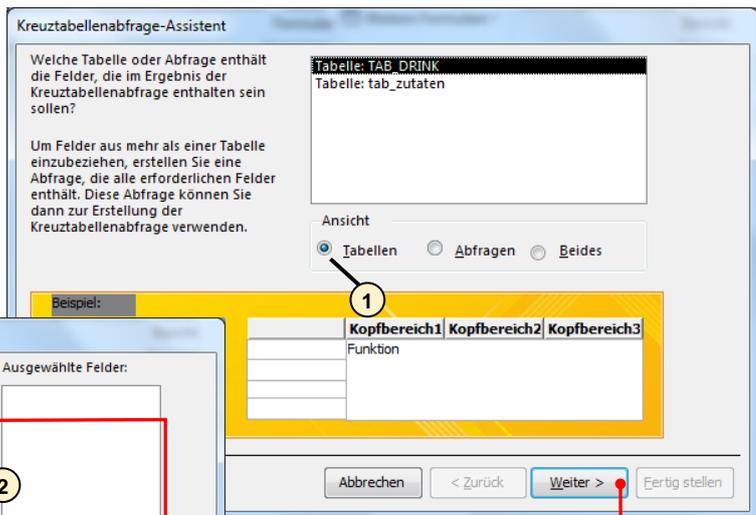
Dieser Assistent erstellt eine Kreuztabellenabfrage, die Datensätze in kompaktem datenblattähnlichen Format anzeigt.

- Auswahlabfrage-Assistent
- Kreuztabellenabfrage-Assistent**
- Abfrage-Assistent zur Duplikatsuche
- Abfrage-Assistent zur Inkonsistenzsuche

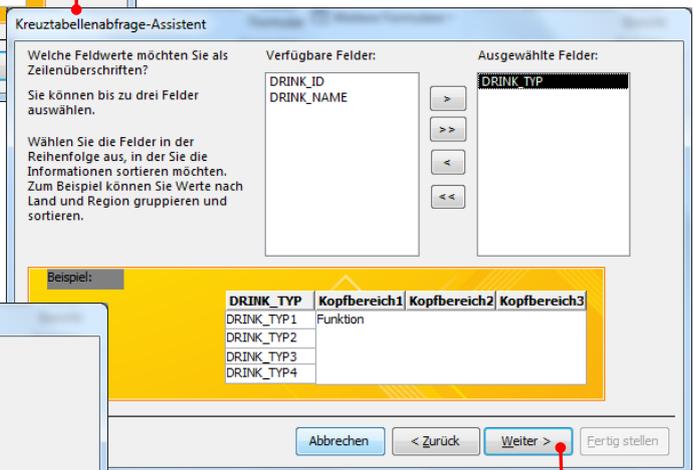
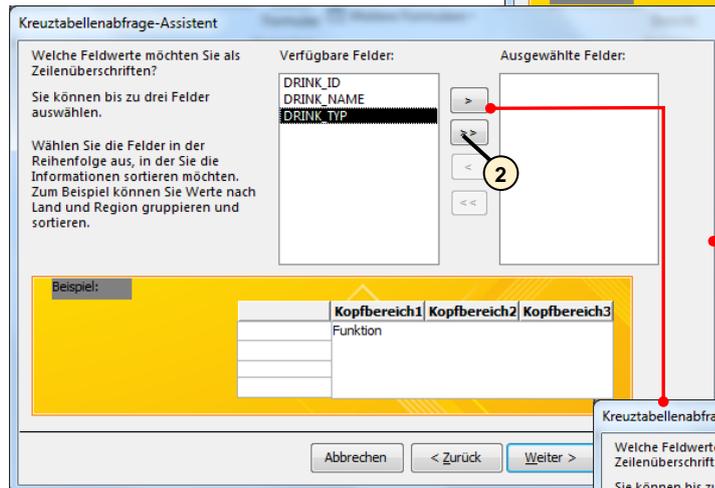
OK Abbrechen

Nach Anwahl des Kreuztabellen-Assistenten öffnet sich das erste Fenster dieses Assistenten.

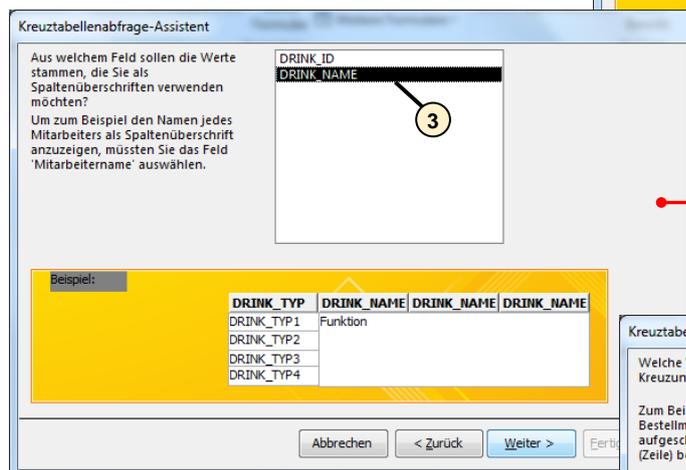
- 1 Wählen Sie hier die Datenquelle. In unserem Fall eine Tabelle, welche die Drink-Typen gespeichert hat.



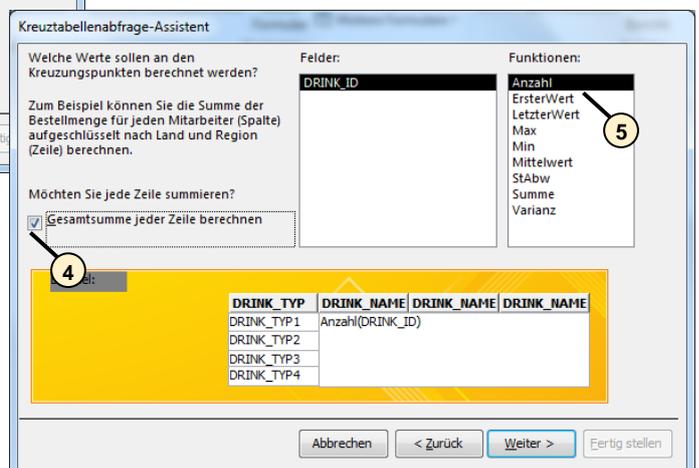
- 2 Hier legen Sie bereits die Struktur der Kreuztabelle fest, nämlich die Spalten. Eine Vorschau erhalten Sie zeitgleich.



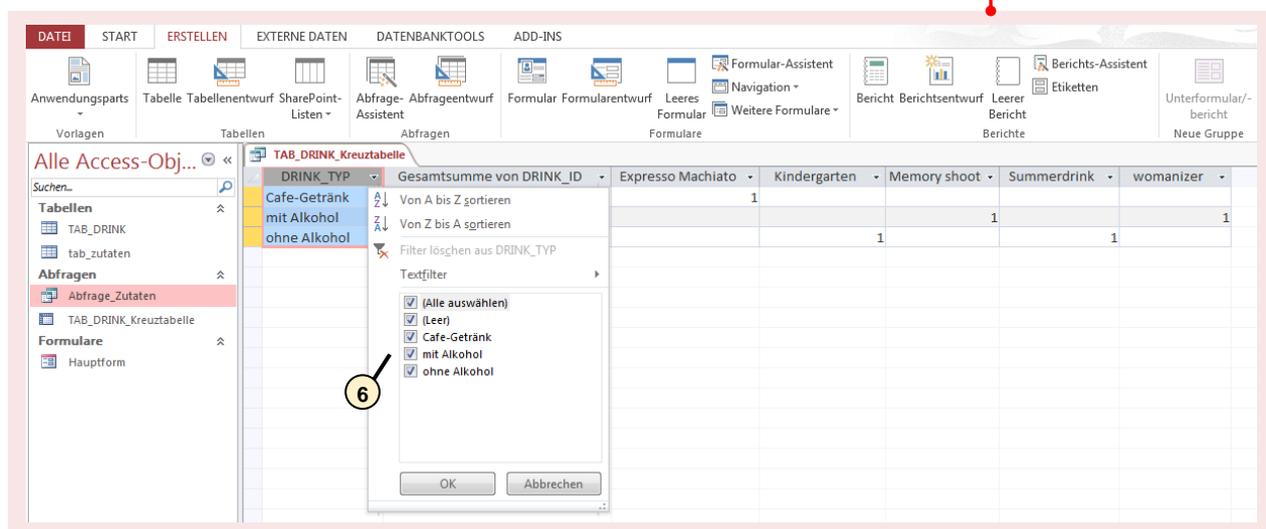
- 3 Als Spaltenbeschriftung legen Sie die Drink-Namen fest.



- 4 Wenn Sie die Anzahl der Drinks wissen wollen, welche Ihrem Kriterium entspricht, dann setzen Sie hier ein Häkchen.
- 5 Natürlich muss unter die Funktion 'Anzahl' gewählt sein.



- ⑥ Ein letztes Bestätigen mit der Option, sich die Abfrage-Ergebnisse sofort anzeigen zu lassen, und Sie haben den letzten Schritt beendet.



Hier können Sie die Kriterien festlegen, nach welchen dann gesucht wird. Eine ‚1‘ in der Tabelle bedeutet: Kriterium erfüllt.

3.6. Formulare

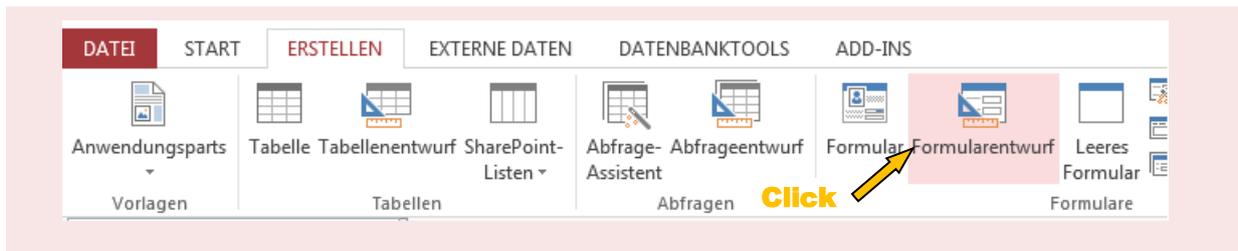
Definition
Formular

Ein Formular ist eine Schnittstelle für den Anwender, auf eine bequeme Weise Daten zu lesen, ändern und neu einzugeben in einer ansprechender Optik.

Beispiel: Formular zur Kunden- und Rechnungsverwaltung

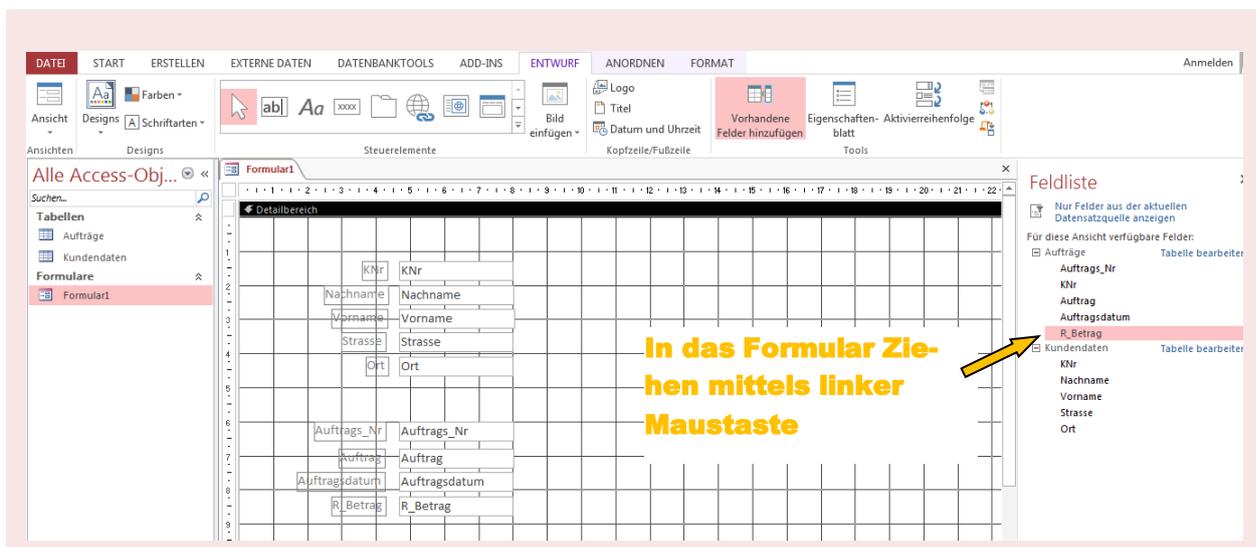
3.6.1. Erstellen von Formulare in der Entwurfsansicht

Gewünscht ist ein Formular angepasst auf diese Datensätze mit entsprechenden Möglichkeiten. Daher erstellen Sie das Formular in der Layout-Ansicht und nicht über den Formular-Assistenten.

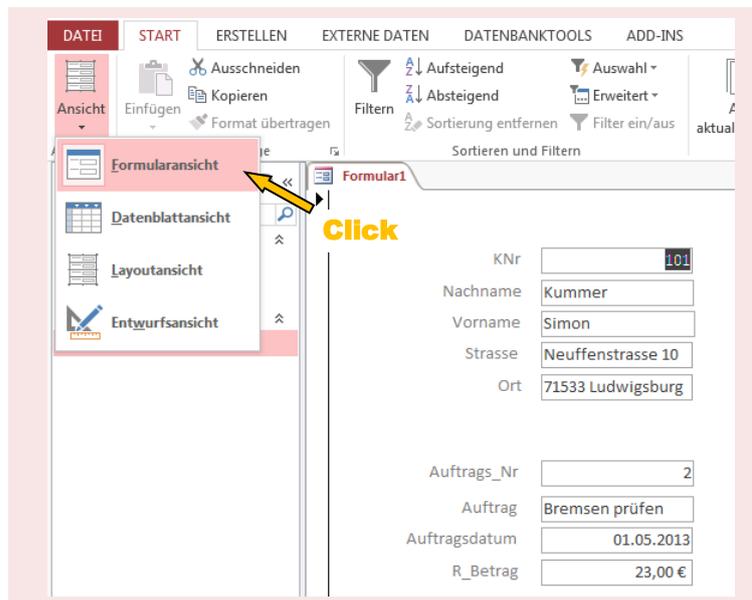


Um die Datenfelder ins Formular zu ziehen, öffnen Sie die die zwei Tabellen ‚Aufträge‘ und ‚Kundendaten‘, die Sie zuvor schon erstellt haben. Auf diese Weise können Sie die Daten einsehen, welche Sie als Feld in das Formular bringen können.

In diesem Beispiel soll der Kunde an oberster Zeile stehen und darunter aufgeführt die Aufträge, welcher er erteilt hat. Speichern Sie dieses Formular z.B. unter dem Namen ‚Formular1‘.



Nun können Sie Ihr Formular schon austesten. Hierfür müssen Sie in die Ansicht auf ‚Formularansicht‘ schalten. Über die untere Schaltleiste können Sie die einzelnen Datensätze einsehen.

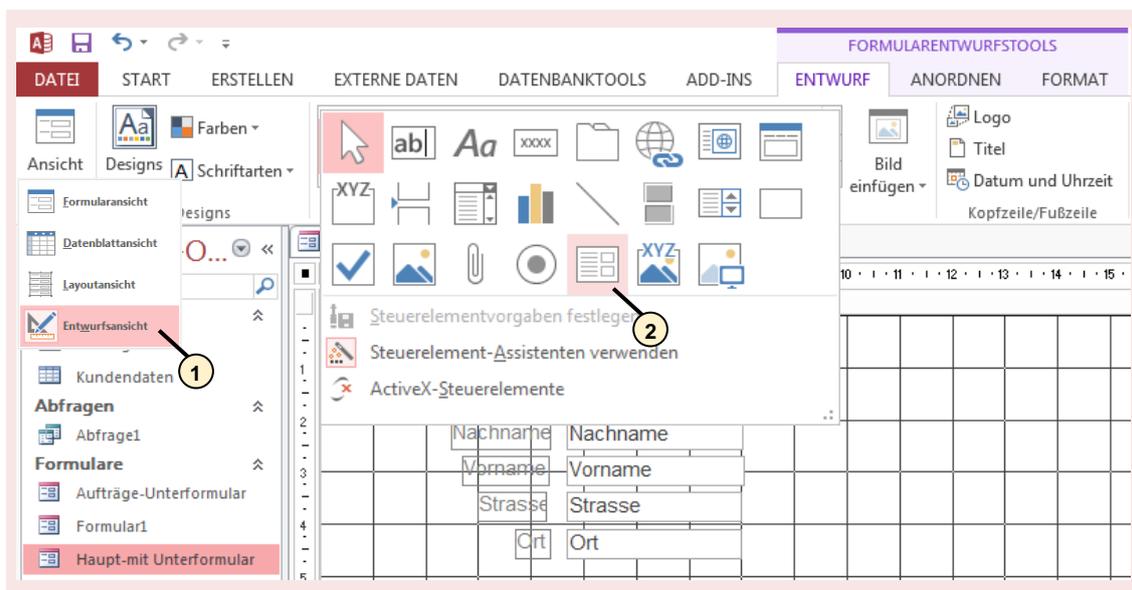


3.6.2. Unterformulare – erstellen und anwenden



Nun kann ein Kunde ein oder mehrere Aufträge erteilt haben (d.h. eine 1:n-Beziehung liegt vor). Es wäre wünschenswert, wenn unter einem Kundennamen alle von ihm erteilten Aufträge ersichtlch wären.

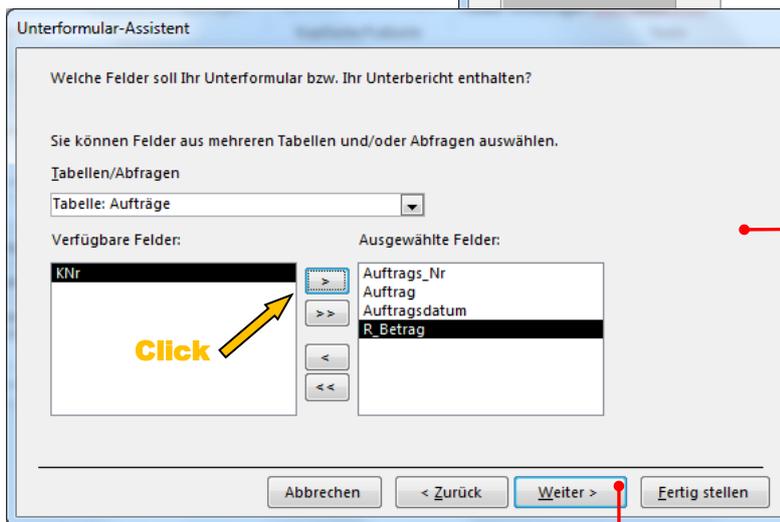
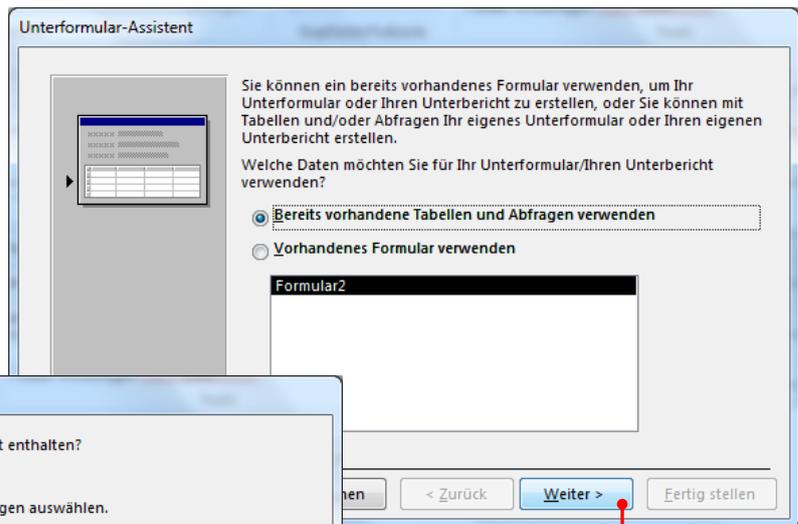
Lösung: Verwendung von ‚**Unterformularen**‘.



- ① Sie können Unterformulare **nur** in das Hauptformular einfügen, wenn das Hauptformular in Entwurfs-Ansicht ist.
- ② Über dieses Icon wählen Sie das Unterformular. Halten Sie die linke Maustaste gedrückt und wandern Sie in das Hautformular. Mit Ziehen der linken Maustaste bestimmen Sie die Gröé des Unterformulars.

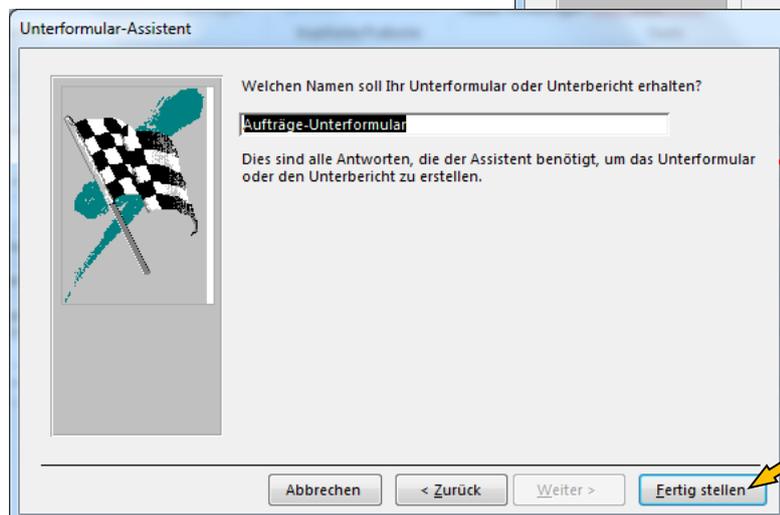
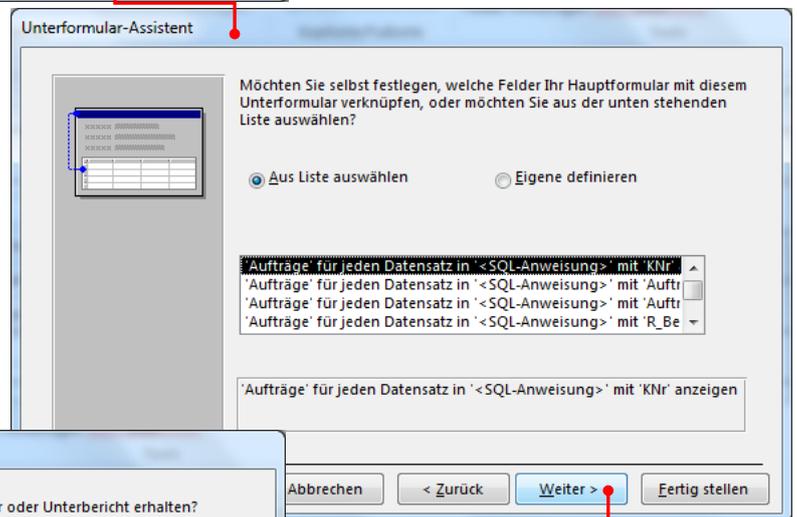
Nach Bestätigen eines Hinweises öffnet sich nebenstehendes Dialogfenster.

Für das neu zu erstellende Unterformular sollen Daten aus der schon vorhandenen Tabelle ‚Aufträge‘ verwendet werden. Da wählen Sie die Option 1.



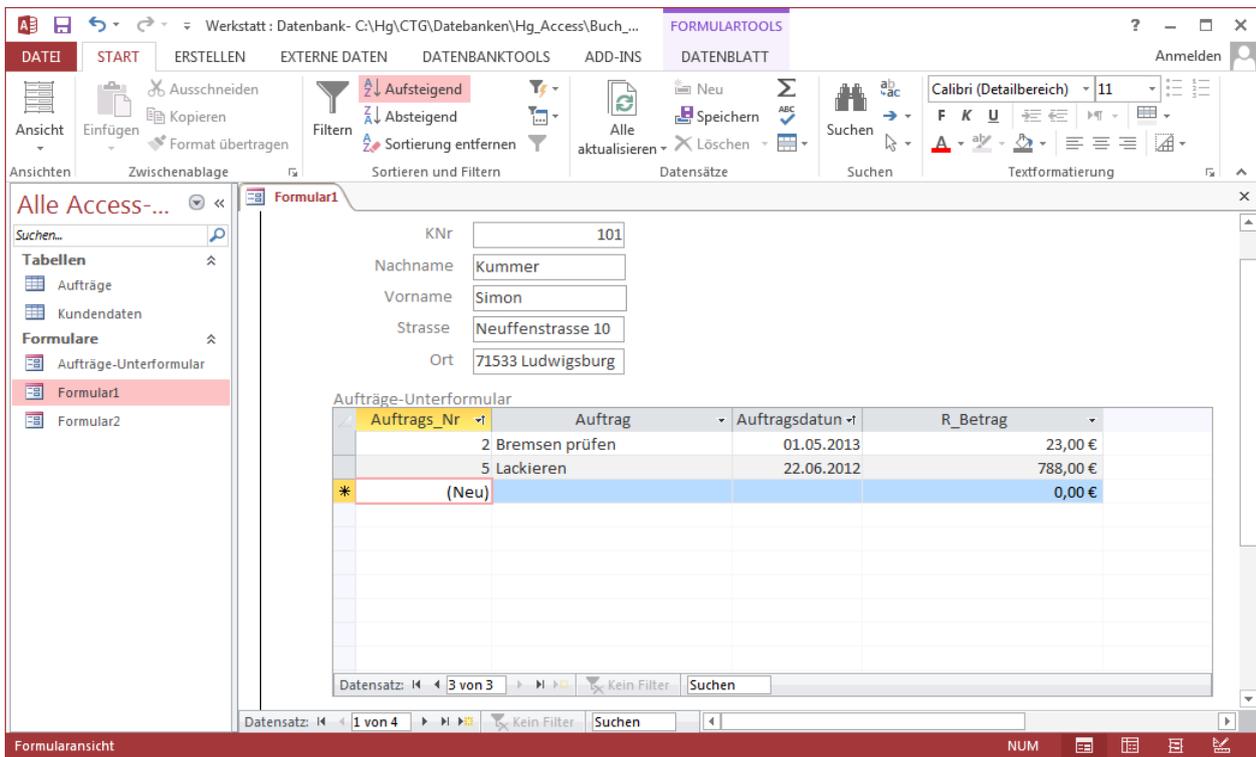
Hier legen Sie fest, welche Daten im Unterformular übernommen werden sollen.

Das Haupt- und Unterformular müssen miteinander verknüpft werden. Da zu einem bestimmten Kunden, d.h. Kundennummer sämtliche Aufträge angezeigt werden sollen, ist ‚KNr‘ die Verknüpfung.



Abgespeichert muss natürlich auch. Vergeben Sie hierfür einen Namen.

Das Ergebnis präsentiert sich in der Formularansicht wie folgt:



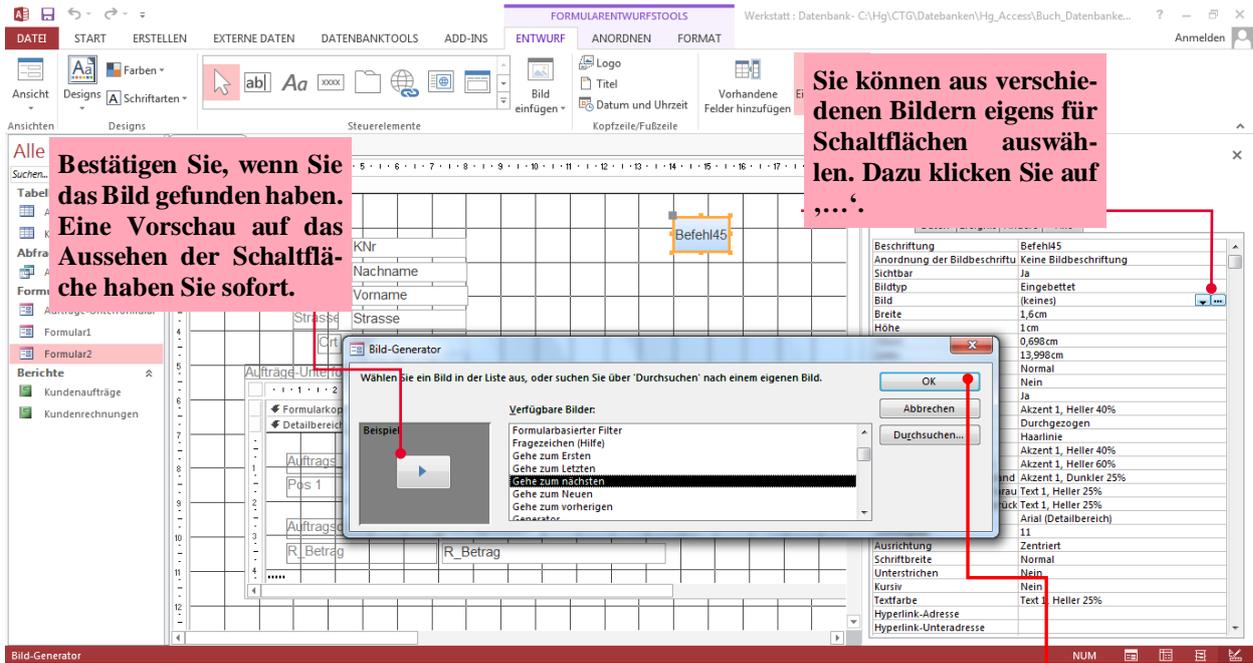
Sie können innerhalb es Hauptformular – sprich der Angabe des Kunden – zu den einzelnen Aufträgen springen, d.h. innerhalb dem Unterformular wandern.

3.7. Steuerelemente in Formulare

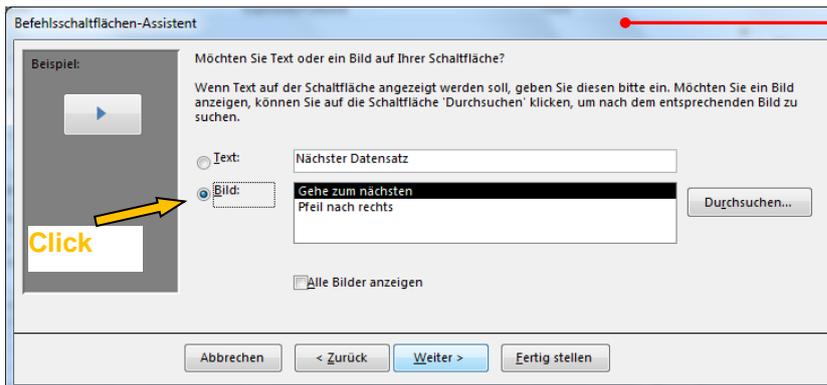
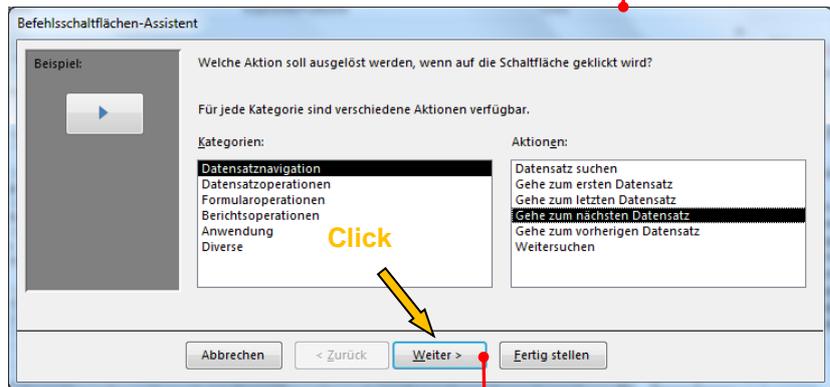
3.7.1. Steuerelemente mit Makros



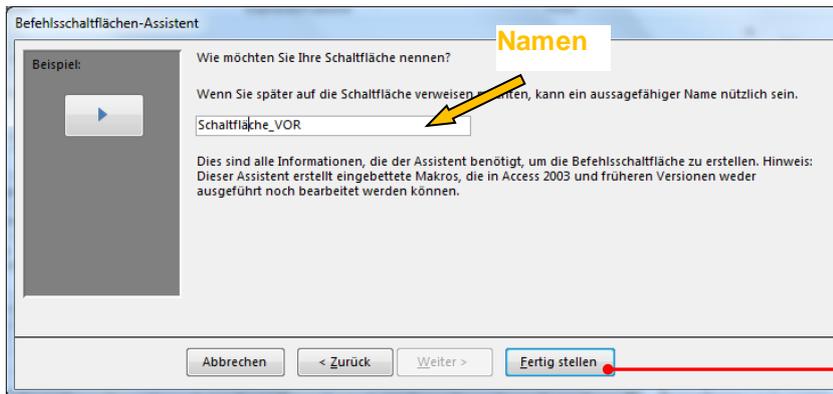
Über Steuerelemente können Sie Formulare bedienerfreundlich gestalten. Die Möglichkeiten sind vielfältig. In einem einfachen Beispiel erhalten Sie einen kleinen Einblick. Es soll über Schaltflächen zum nächsten bzw. vorherigen Datensatz umgeschaltet werden können. Setzen Sie in der Entwurfsansicht das Steuerelement ‚Schaltfläche‘  in das Formular.



Die Aktion, welche beim Anklicken dieser Schaltfläche ausgeführt wird, definieren Sie im nächsten Fenster.

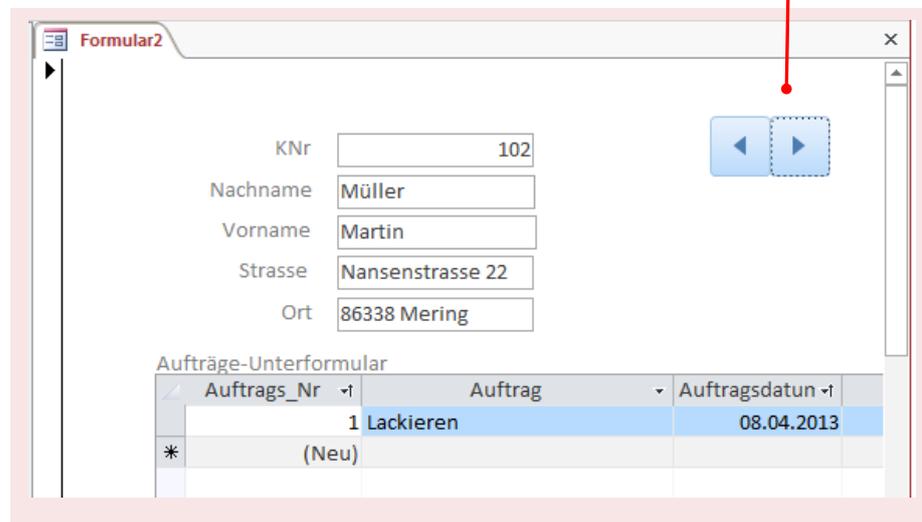


Sie können nun bestimmen, ob die Schaltfläche als Beschriftung ein Text oder ein Zeichen hat.



Die Schaltfläche können Sie einen Namen vergeben.

Ebenso richten Sie eine Schaltfläche für das Zurückwandern in den vorherigen Datensatz ein. Das Ergebnis ist ansprechend – oder etwa nicht?



3.7.2. Steuerelemente mit VBA

VBA Visual Basic for Application

Um die Funktionalität zu erweitern, können Sie einem Objekt (meist Schaltfläche) ein VB-Code hinterlegen. Dies soll an einem einfachen Beispiel gezeigt werden. Da dieses Thema ‚Visual-Basic in ACCESS‘ sehr umfangreich ist, soll dies im nachfolgenden Kapitel näher erklärt werden.

3.8. Berichte



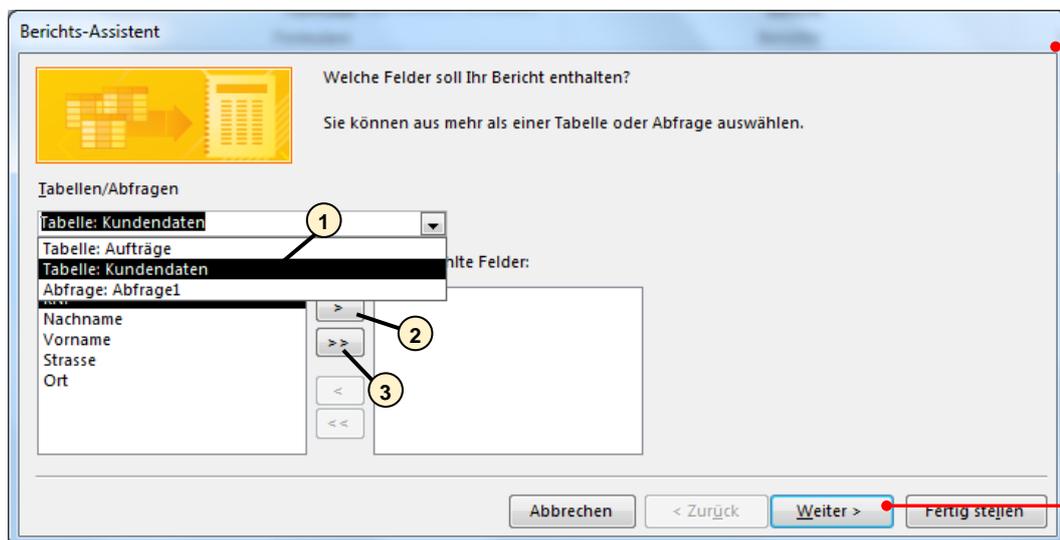
Berichte dienen zum Ausdrucken auf Papier. Sie ähneln den Formularen sehr – allerdings kann man mit den Steuerelementen Daten nur einsehen.

Erstellen von Berichten

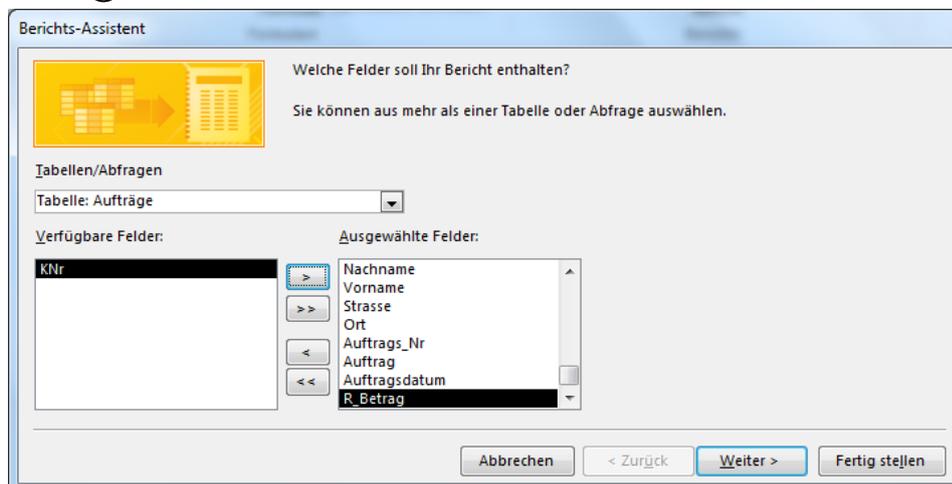
Als Beispiel dient die Datenbank ‚Werkstatt‘, welche sich auf der Lösungs-CD findet. Der Bericht soll zum Schluss sämtliche Aufträge eines Kunden anzeigen und natürlich ausdrucken können.

Eine sehr einfache Weise, einen Bericht zu erstellen, ist den **Berichtsassistenten** zu verwenden. Sie können nach Fertigstellung immer noch über die Entwurfsansicht den erstellten Bericht nach Ihren Wünschen anpassen.

Über die Registerkarte ‚Erstellen‘ können Sie den Berichtsassistenten starten.



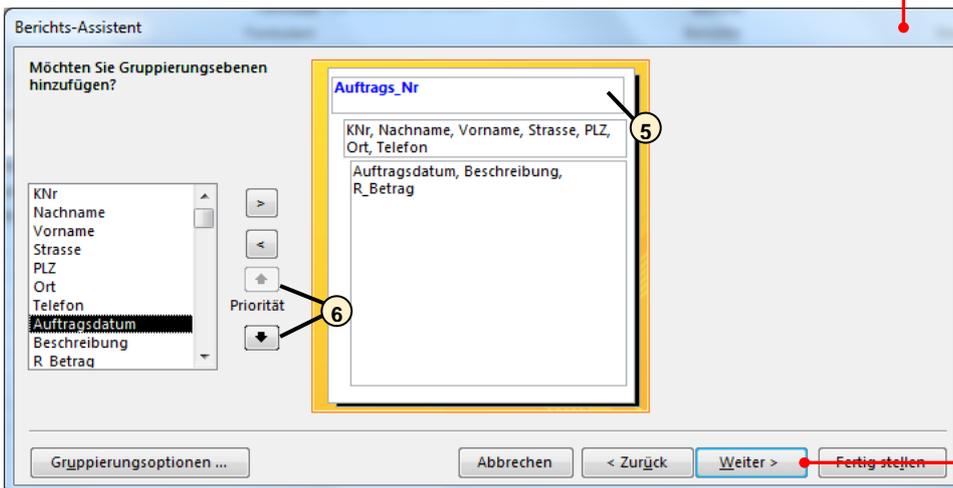
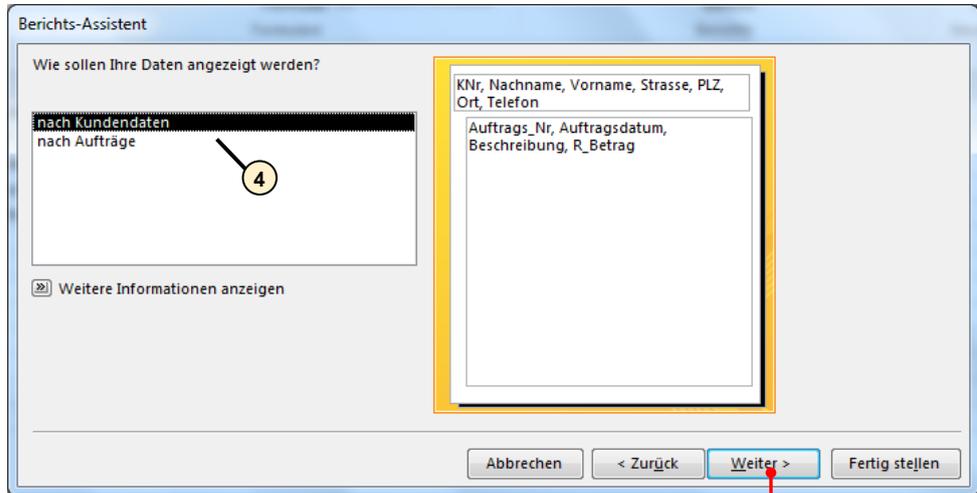
- ① Wenn Sie über den Berichts-Assistenten einen Bericht erstellen, dann können Sie Daten von mehreren Tabellen in ein Bericht einbinden. Oder Sie binden eine Abfrage in Ihr Bericht ein, welche ohnehin mehrere Tabellen umfassen kann.
- ② Die Datenfelder können Sie einzeln auswählen und in den Bericht übernehmen.
- ③ Oder mit einem Mausklick alle Felder übernehmen.



Hier nun alle Datenfelder der Tabelle ‚Kunde‘ übernommen und nachfolgend die Datenfelder der Tabelle ‚Aufträge‘. Auf das Datenfeld KNr (Fremdschlüssel) der Tabelle ‚Aufträge‘ wird verzichtet, da die Aufträge ohnehin einem Kunden zugeordnet sind.

Wollen Sie einen Berichtskopf haben oder sollen verschiedene Anzeigeebenen eingerichtet werden? Dann können Sie dies in den folgenden zwei Fenstern einrichten.

④ Die Daten sollen angezeigt werden nach Kunden, schließlich kann jeder Kunde mehrere Aufträge haben. Eine Vorschau haben Sie. Bestätigen Sie, indem Sie weiterklicken.

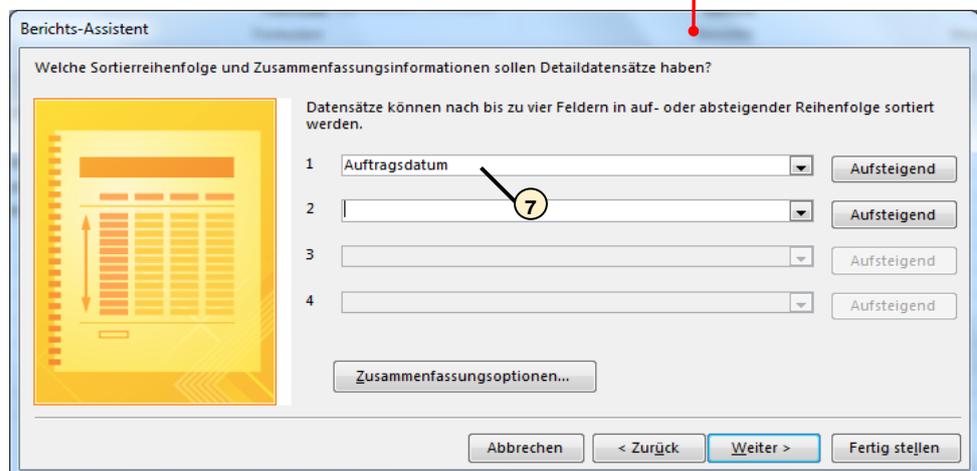


⑤ Wollen Sie weitere Gruppierungsebenen hinzufügen? Es würde Sinn machen, die Auftragsnummer oben an den Berichtskopf zu platzieren.

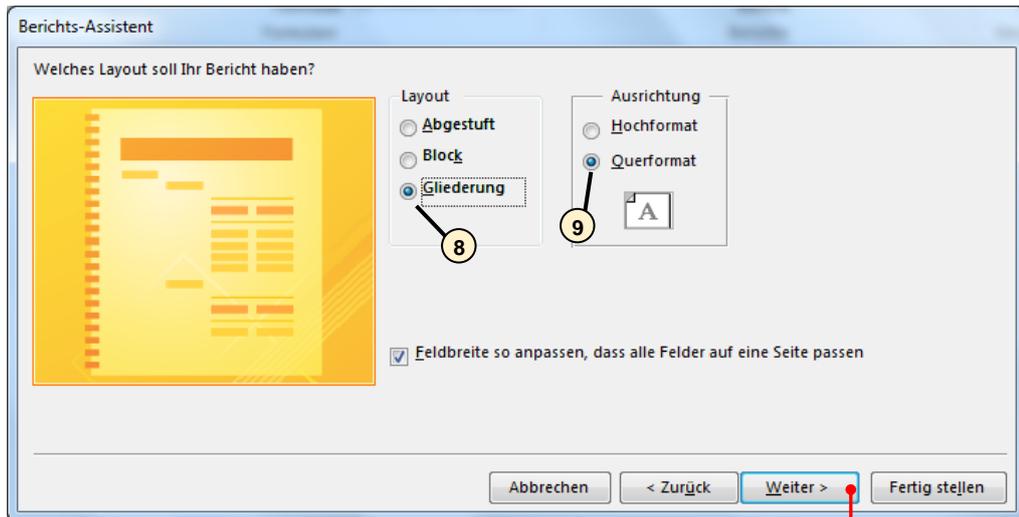
⑥ Die Reihenfolge der einzelnen Ebenen können Sie über diese zwei Schaltflächen ändern.

Es macht Sinn, sich die Aufträge sortiert nach Datum anzeigen zu lassen. Haben Sie mehrere Aufträge am gleichen Tag? Dann definieren Sie Sortierkriterien. Siehe hierzu nachfolgendes Fenster.

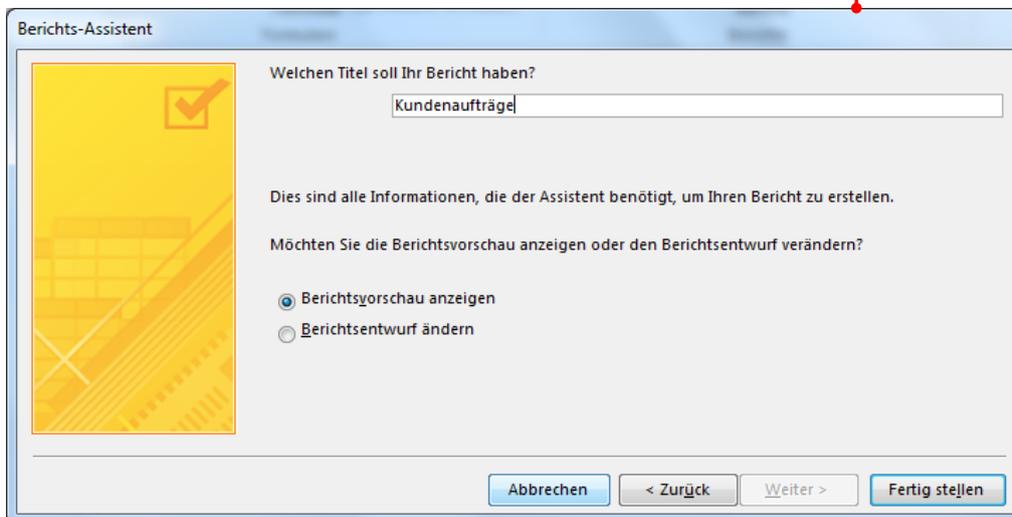
⑦ Hier werden die Aufträge nach dem Datum sortiert.



Wenn Sie die hierarchische Wertung der Daten betonen wollen, dann nehmen Sie Layout ‚Gliederung‘. Eine Vorschau ansehen Sie zeitgleich im Dialogfenster. Den Bericht, welchen Sie erhalten, können Sie immer noch in der Entwurfsansicht anpassen.



Damit haben Sie Ihren Bericht formatiert. Der letzte Schritt wäre die Namensvergebung.



3.9. Datenaustausch zu Excel



Selbstverständlich können Sie in Access Daten über die Zwischenablage kopieren. Allerdings sollte dann die Datenmenge nicht zu groß und die Daten im externen Programm tabellenartig formatiert sein.

Sie können aber auch Excel-Tabellen direkt in Access einbinden. Somit können Sie die Werte in den Feldern der eingebundenen Tabellen ohne weiteres bearbeiten, auch größere Datenmengen sind kein Problem.

The screenshot shows the Microsoft Access interface with the 'EXTERNE DATEN' (External Data) ribbon selected. The ribbon includes options for 'Importieren und Verknüpfen' (Import and Link), such as 'Excel', 'Access', and 'ODBC-Datenbank'. A file explorer window is open, showing a folder named 'Datenaustausch zu Escel' containing a file named 'Bekanntnisse'. A dialog box titled 'Externe Daten - Excel-Tabelle' is displayed, prompting the user to choose the source and target of the data. The dialog box contains the following text:

Wählen Sie Quelle und Ziel der Daten aus

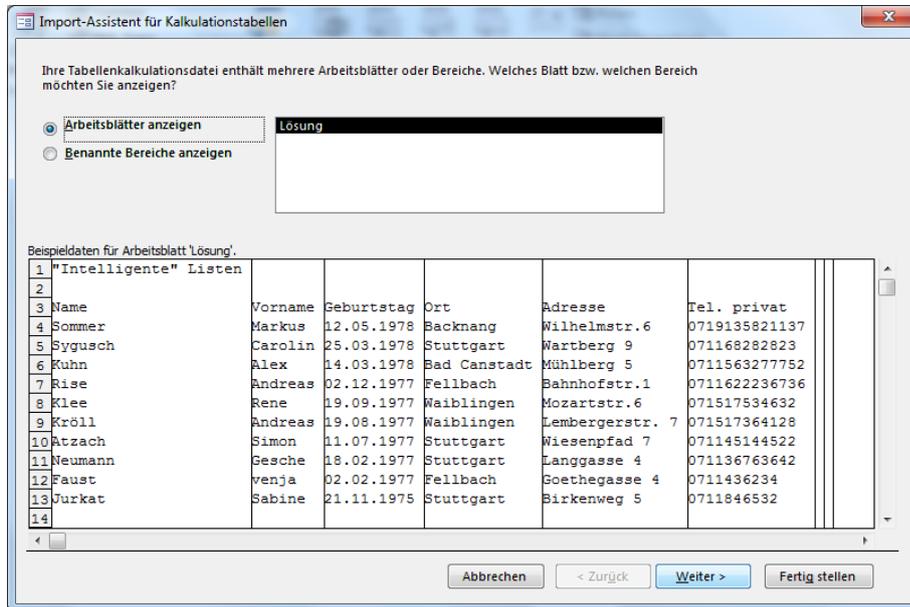
Geben Sie die Quelle für die Definition der Objekte an.

Dateiname: C:\Buch_Datenbanken2013\Lsg_CD\Datenaustausch zu Escel\Bekanntnisse.xlsx

Geben Sie an, wie und wo Sie die Daten in der aktuellen Datenbank speichern möchten.

- Importieren Sie die Quelldaten in eine neue Tabelle in der aktuellen Datenbank.**
Wenn die angegebene Tabelle nicht vorhanden ist, wird sie durch Access erstellt. Wenn die angegebene Tabelle bereits vorhanden ist, werden die Inhalte durch Access möglicherweise mit den importierten Daten überschrieben. Änderungen an den Quelldaten werden nicht in die Datenbank übernommen.
- Erstellen Sie eine Verknüpfung zur Datenquelle, indem Sie eine verknüpfte Tabelle erstellen.**
In Access wird eine Tabelle erstellt, die eine Verknüpfung zu den Quelldaten in Excel enthält. Änderungen, die an den Quelldaten in Excel vorgenommen werden, werden in der verknüpften Tabelle übernommen. Die Quelldaten können jedoch nicht in Access geändert werden.

Buttons: OK, Abbrechen



Es folgen weitere Dialogfenster, in denen Access z.B. nach der Spaltenüberschrift fragt, in welcher Form ein Primärschlüssel angelegt werden soll oder nach dem Namen der nun neuen Tabelle. Diese Dialogfenster sind aber selbsterklärend.

3.10. Aufgaben zu Kapitel 3

Aufgabe 1:

Was müssen Sie beachten, wenn Sie eine Abfrage von mehreren Tabellen machen?

Aufgabe 2:

Für wen sind Formulare vor allem gedacht?

Aufgabe 3:

Wann verwenden Sie Unterformulare?

Aufgabe 4:

In welcher Ansicht müssen Sie arbeiten, um Unterformulare zu erstellen und einzufügen?

4. VBA in ACCESS

VB (Visual Basic) ist eine Programmiersprache, die in jeder Office-Anwendung eingebettet ist. Office-Anwendung verwendet werden kann. Durch den Einsatz von VBA erweitern Sie die Funktionalität Ihrer Datenbank erheblich.

Vorteile von VB in Access

- Mit Visual Basic können Sie Fehlerbehandlung durchführen
- Sie können Schleifen oder andere Strukturen programmieren
- Es können Variablen definiert werden
- Sie können einen VB-Code dokumentieren, in Foren posten oder per E-Mail versenden
- Sie können Ihr Programm debuggen, was bei Makros nicht möglich ist
- Makros können Funktionen der Windows-API oder anderer Bibliotheken nicht direkt verwenden- Visual Basic sehr wohl
- Mit VBA können Sie Systemobjekte wie Tabellen und Abfragen erstellen oder bearbeiten, Transaktionen verarbeiten usw

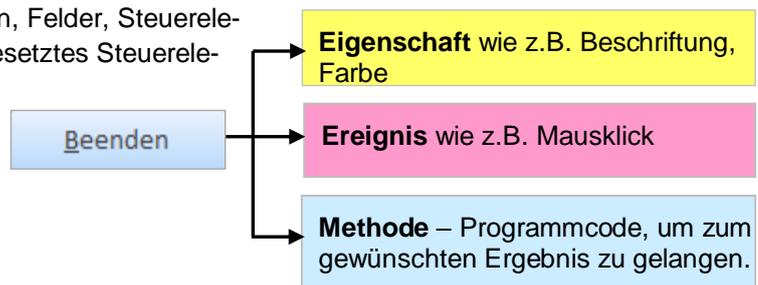
4.1. Begriffe in ACCESS

Visual Basic ist ereignis- und objektorientiert und setzt eigentlich die Anwendung der Maus voraus. Fenster sind vordefiniert und können bequem mit der Maus angeordnet werden.

Ein VBA-Programm setzt sich zusammen aus Funktionen und Sub-Prozeduren, welchen bestimmten Objekten zugeordnet sind.

Objekte können sein: Formblatt, Abfragen, Felder, Steuerelemente, etc. Nehmen Sie ein häufig eingesetztes Steuerelement ‚Schaltfläche‘.

Sie setzen die den Button an eine bestimmte Position im Formblatt, definieren Aussehen und weisen eine bestimmte Aktion zu, welche beim Anklicken ausgeführt werden soll.



Visual Basic ist objektorientiert

Visual Basic versteht seine Elemente (Fenster, Schaltflächen, Listen, etc.) als Objekte, die Eigenschaften haben und denen man bestimmte Methoden und Ereignisse zuordnen kann.

Visual Basic ist ereignisorientiert

Programme unter Visual Basic warten auf Ereignisse. Sie können ein Programm starten und es passiert nicht. Klicken Sie auf eine Schaltfläche, dann passiert etwas, weil der dazugehörige Programmcode abläuft.

Eigenschaften

Jedes Objekt hat Eigenschaften, wie Farbe, Größe, Rahmen, etc. Sie sind so wichtig, dass es dafür ein eigenes Fenster gibt.

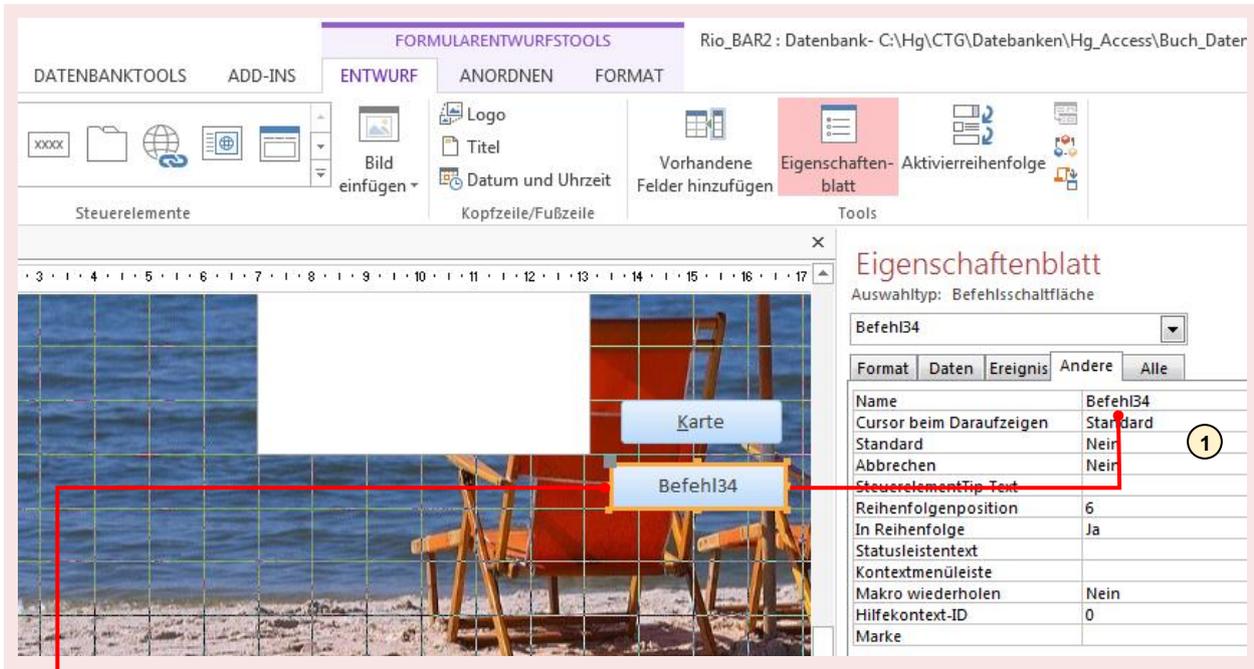
Ereignisse

Jedes Objekt hat unterschiedliche Möglichkeiten, auf Ereignisse zu reagieren. Ein häufiges Ereignis ist z.B. der Klick mit der Maus auf eine Schaltfläche.

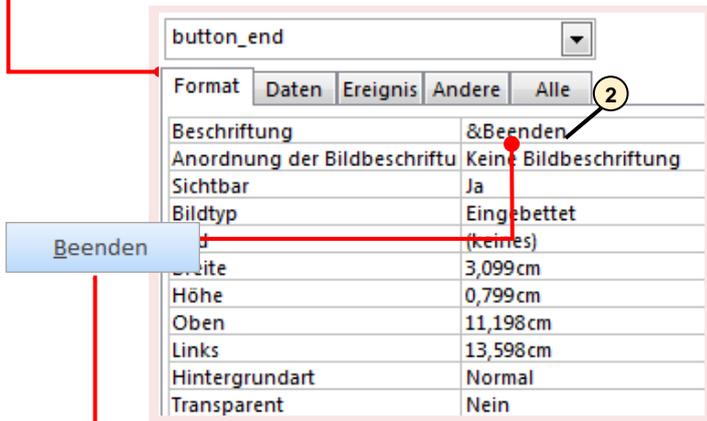
Methoden

Methoden sehen den Eigenschaften ähnlich, bedeuten aber die Anweisungen an ein Objekt, sich auf eine bestimmte Weise zu verhalten. Z.B. eine bestimmte Position auf der Form einzunehmen oder einer Liste neue Elemente hinzuzufügen.

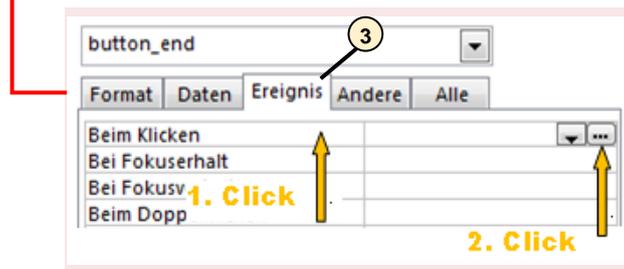
Praktisches Beispiel: Arbeitsauftrag „Coktail-Bar“. Dieses Beispiel findet sich auch auf der Lösungs-CD, wird im Kapitel 5 noch näher beschrieben.



- ① Die erste Eigenschaft, welche zu verändern ist, sei der Objektname. Ändern Sie den Name von „Befehl34“ auf „button_end“. Unter diesem Namen wird dieses Objekt dann auch in Ihrem VB-Programm angesprochen, d.h. Ihr Programm wird dadurch übersichtlicher.



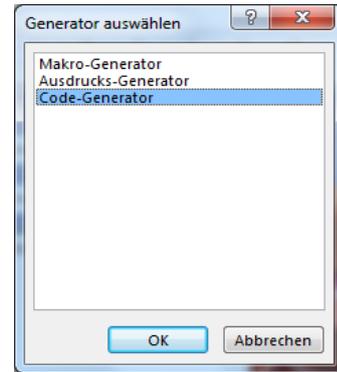
- ② Nun soll jede Bedienoberfläche für den Anwender bedienfreundlich sein. Dazu gehört auch die Beschriftung (engl. „caption“). Sobald Sie die Eigenschaft ‚Beschriftung‘ ändern, wird dies im Formular übernommen.



- ③ Die Angabe des Ereignisses, bei welchem Access beendet wird, geben Sie in der Registerkarte ‚Ereignis‘ an.

Um hinter der Beendigung des Formulars steht ein Programmcode. Sie können nun die Quelle dieses Generators angeben.

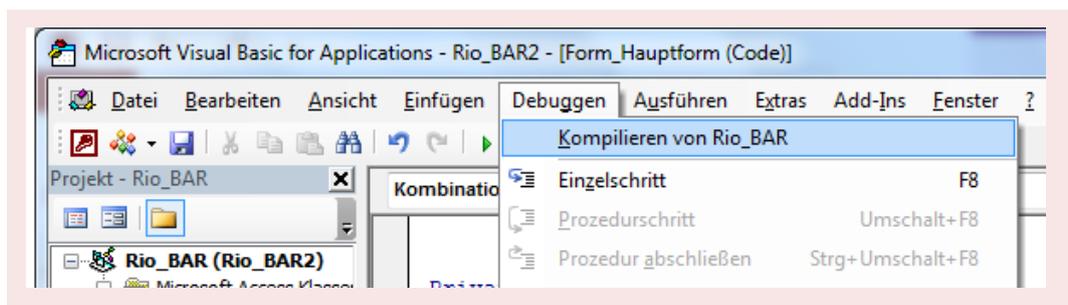
Mit Doppelklick oder Anwählen und Bestätigen öffnet sich der VBA-Editor.



Hier wurde schon der Programmrumpf angelegt, welcher der Schaltfläche ‚Beenden‘ zugewiesen ist und in diesem Sie Ihr Programmcode einfügen.

Unter dem Befehl ‚DoCmd‘ sind Befehle zusammengefasst, welche auf ein Fenster oder File anwendbar sind. Sobald Sie das Punktzeichen eingegeben haben, öffnet sich ein Auswahlfenster der Methoden zugehörend zu ‚DoCmd‘. Sie wählen die Methode ‚CloseDatabase‘.

Bevor Sie Ihr Programm austesten, kompilieren Sie es. Damit wird Ihr Programm in ein Maschinencode übersetzt. Syntaxfehler und so manche Semantikfehler werden erkannt und angezeigt.



Der Befehl ist nun über die Schaltfläche ‚Beenden‘ im Formblatt aufrufbar.

4.2. Variablen und Konstanten

Variablen können als Platzhalter angesehen werden, in denen Werte eingesetzt werden können. Die in Visual Basic am häufigsten verwendeten Variablentypen:

Variablentyp	Inhalt	Werteberich
Zahlenvariablen		
Byte	8 Bit-Ganzzahl vorzeichenlos	0 bis 255
Integer	32-Bit-Ganzzahl	-32.768 bis 32.767
Long	Ganzzahl (4 Bytes)	-2.147.483.648 bis 2.147.483.647
Single	Fließkomma	-3,402823E38 bis 3,402823E38
Double	Fließkomma doppelter Genauigkeit	-1,79769313486231E208 bis 1,79769313486231E208
Zeichen und Zeichenketten		
Char	VB Unicode	Ein einzelnes Zeichen
String	VB Unicode	Zeichenkette (Text) – bis zu 2 Milliarden Zeichen
Sonstige Variablen		
Date	Datum und Zeit	1.Jan 100 bis 31 Dez 9999
Currency	Währung	-922.337.203.685.477,5808 bis 922.337.203.685.477,5808
Boolean	1 Bit	True (wahr bzw. 1) oder False (falsch bzw. 0)
Object	Verweis auf ein Objekt	Button, TextBox, etc.
Variablen betreffend ACCESS		
CurrentDb()	aktuell geöffnete Datenbank	
Recordset	Datensatz	

ACCESS als Datenbank kennt noch weitere Variablentypen. Hier zu nennen ist der Typ ‚Recordset‘. Führen Sie die Deklaration wie folgt durch:

```
Dim rs As ADO.Recordset
```

Hier ist es wichtig, die betreffende Bibliothek explizit vor der Benennung des Variablentyps zu setzen, um Überschneidungen zu vermeiden. ‚Recordset‘ ist in der Bibliothek² DAO sowie in der Bibliothek ADO (an ACCESS 2000) zu finden.

Konstanten sind in Ihrem Wert unveränderlich. Im Quelltext kann ein Wert zugewiesen werden, welcher sich dann aber zur Laufzeit nicht mehr verändert. Bekanntes Beispiel einer Konstanten: π (Pi) = 3,14159...

In Visual-Basic vordefinierte Konstanten müssen nicht mehr deklariert werden. Konstanten, die mit ‚vb‘ beginnen gelten für alle Visual-Basic- Versionen, die mit ac beginnen, gibt es nur in ACCESS

z.B. vbRed meint die "Farbe rot", vbNewLine meint "gehe in die nächste Zeile"
acNext steht in Access für "nächster Datensatz"

² Eine Bibliothek ist eine Sammlung von Funktionen und Prozeduren, auf welche zugegriffen werden kann, um diese in das eigene Programm einzubinden.

Damit Variablen und Konstante im VBA-Programm verwendet werden können, müssen Sie vorab bekannt gegeben werden, sprich **deklariert** werden. Hierbei wird zugleich der Variablentyp bzw. Typ der Konstante festgelegt.

Beispiel:

```
Dim Zahl1 As single 'bedeutet: Der Platzhalter Zahl1 beinhaltet Kommazahlen
Const Pi As Single = 3.14159 'Hier wird die Konstante Pi als Kommazahl deklariert und
                             gleichzeitig ein Wert zugewiesen
```

Zur Fehlervermeidung können Sie angeben, in welchen Programmteilen eine Variable oder Konstante bekannt ist, d.h. Sie bestimmen über die Deklaration den **Gültigkeitsbereich**:

```
'Deklaration globaler Variablen
Dim p, d As Single
'Deklaration globaler Konstante
Const Pi As Single = 3.14159
```

```
Private Sub Befehl4_Click()
    ,Deklaration lokaler Variable
    Dim A As Single

    A = Pi*d*d/4
    P = F * A
    .....
End Sub
```

Eine **globale** Variable wird außerhalb jeglichem Unterprogramm oder Funktion deklariert und ist im gesamten Programm bzw. Programmteilen bekannt.

Eine **lokale** Variable ist nur bekannt, d.h. gültig innerhalb der Funktion, in welcher sie deklariert wurde.

Aufgabe:

Welcher Fehler hat folgender Programmcode?

```
Dim Erg As single
Erg = "1" + "1"    \\ "1" ist vom Typ char - Typenkonflikt!
```

4.3. Zugriffe auf Datensätze mittels VBA

Die Funktionen, Welche Microsoft bereitstellt, um auf eine Datenbank und deren Datensätze zuzugreifen, sind in der Bibliothek DAO (Data Access Object) abgelegt, welche Sie als User verwenden können. Die Bibliothek DAO ist seit 2007 der Standardzugriff.

Option Compare Database

```
Dim db As DAO.Database
Dim rs As DAO.Recordset
```

```
Private Sub ds_speichern_Click()
    Set db = CurrentDb
    Set rs = db.OpenRecordset("Tabelle1", dbOpenTable)
    Do While Not rs.EOF
        If (rs.Fields("Nr") = Nr.Value) Then
            rs.Edit
            rs.Fields("datum") = datum.Value
            rs.Fields("Kurzinfo") = Kurzinfo.Value
            rs.Fields("Uhrzeit") = Uhrzeit.Value
            rs.Update
        End If
        rs.MoveNext
    Loop
    db.Close
    Set db = Nothing
End Sub
```



Eine Beispiels-Datenbank befindet sich auf der Lösungs-CD im Verzeichnis ‚Zugriff auf Datensätze‘!

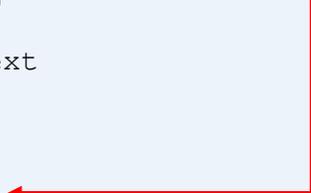
4.4. Fehlersteuerung

Kann es im Programmablauf zu Fehlern kommen (z.B. "nächster Datensatz" wird angewählt, obwohl schon der letzte aktuell ist), kommt eine Fehlermeldung.

Das Verhalten bei absehbaren Fehlern kann gesteuert werden.

```
Private Sub NaechsterSatz_Click()  
    On Error GoTo Err_Meldung  
  
    DoCmd.GoToRecord , , acNext  
    Exit_NaechsterSatz_Click:  
        Exit Sub  
  
    Err_Meldung:  
        MsgBox "Kein weiterer Datensatz"  
End Sub
```

Bei Fehler springe
zur Marke



5. Kapitel: Tipps und Tricks

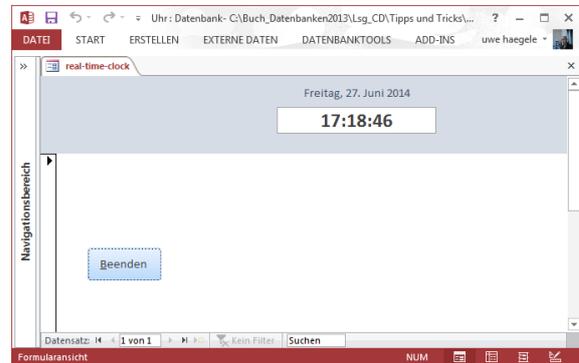
4.1 Bedingte Formatierungen

4.2 Echtzeit-Uhr in Ihrem Formular

5.1. Bedingte Formatierungen

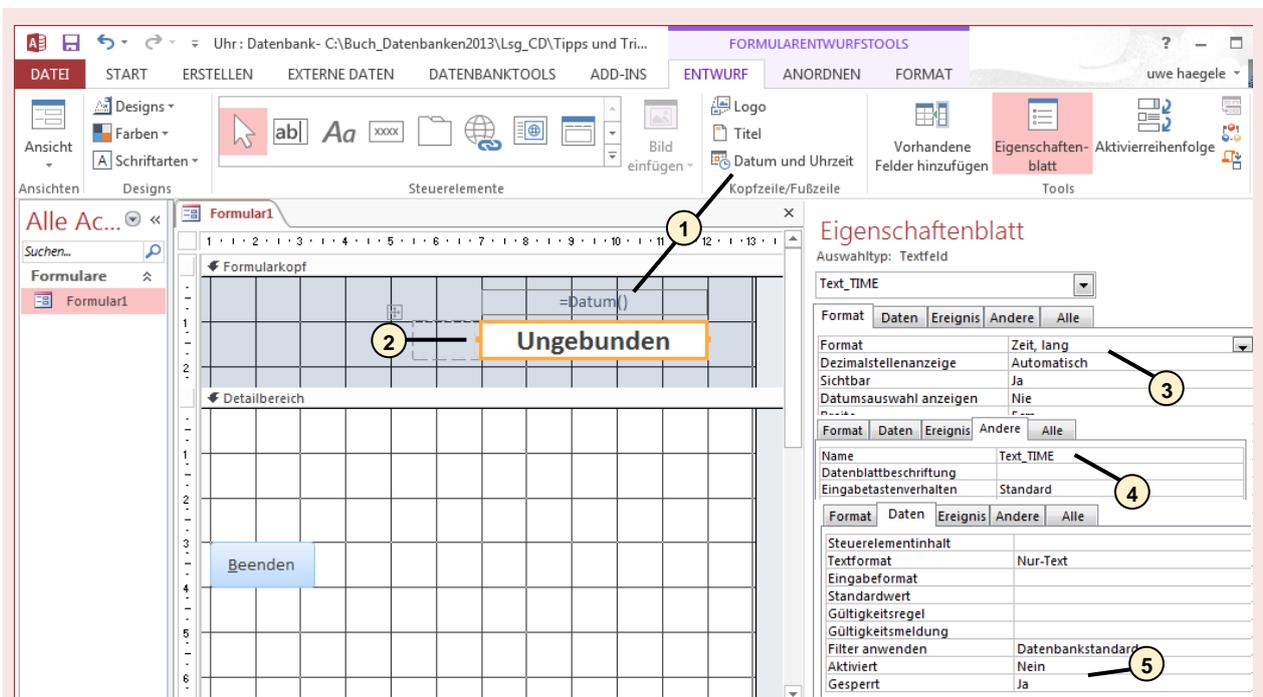
5.2. Echtzeit-Uhr in Ihrem Formular

Eine sekundengenaue Echtzeit-Uhr in Ihrem Formular – ein echter Hingucker. Sicher gibt es auch die Uhrzeit-Funktion Zeit(); diese zeigt die Uhrzeit jedoch nur beim Öffnen des Access-Files genau an.



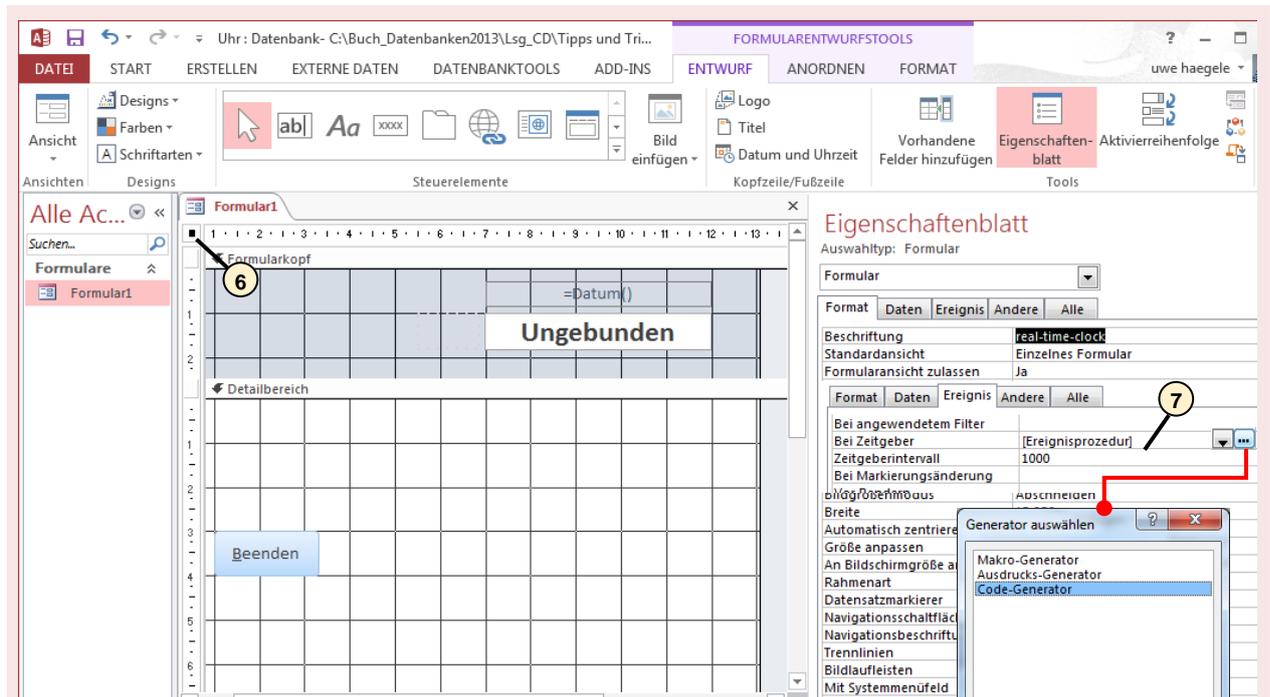
Die Arbeitsschritte hierzu kleinschrittig aufgezeigt:

Schritt 1: Objekte ins Formular platzieren und formatieren



- ① Platzieren Sie zuerst die Elemente in Ihrem neu angelegten Formular. Setzen Sie zuerst das Datumfeld in Ihr Formular – hierbei öffnet sich automatisch der Formulkopf. Sie haben die Möglichkeit, das Format der Datumsanzeige zu bestimmen.
- ② Das eigentliche Element für diese Aufgabe ist ein einfaches Textfeld, welches Sie aber entsprechend formatieren müssen:
- ③ Innerhalb der Registerkarte ‚Format‘ geben Sie den Inhalt als Zeitangabe an. Ein Pop-Up-Menü gibt Ihnen Auswahlmöglichkeiten. Nehmen Sie eine Formatierung, welche lediglich die Uhrzeit anzeigt und nicht das Datum.
- ④ Da dieses Textfeld im VB-Programm noch angesprochen wird, ändern Sie dessen Name sinnvoll.
- ⑤ Für den Anwender soll ersichtlich sein, dass dieses Element zur reinen Anzeige dient und keine Eingabe möglich ist. Daher sperren Sie dieses Element und deaktivieren Sie es.

Schritt 2: Formular formatieren



- ⑥ Um die Eigenschaften eines Formulars ändern zu können, klicken Sie auf die linke obere Ecke. Als bald erscheint ein kleines schwarzes Rechteck und das Eigenschaftsfenster des Formulars öffnet sich.
- ⑦ Für die Funktion der Echtzeit-Uhr wichtige Einstellungen finden sich in der Registerkarte ‚Ereignis‘. Die Idee ist folgende: Nach einem Zeitintervall von einer Sekunde soll eine Aktualisierung der Anzeige stattfinden. Die Angabe Zeitgeberintervall 1000 steht für 1 Sekunde und soll als Trigger dienen für eine Ereignisprozedur.

Schritt 3: VBA-Code anlegen

Option Compare Database

```
Private Sub Form_Timer()
```

```
    Me.Text_TIME.Value = Format(Time, "hh:mm:ss")
```

```
End Sub
```

Der Rumpf der Sub-Routine wird automatisch angelegt. Sie müssen lediglich den rot markierten Programm-Code einfügen.

Erläuterungen zum Programm:

- Private Diese Sub-Routine kann nicht von außerhalb der Datenbank aufgerufen werden.
- Me Der Code bezieht sich auf ein Element im aktuellen Formular.
- .Value Hiermit ist die Eigenschaft ‚Value‘ = Wert gemeint
- = Zuweisung
- Format() Angabe des Formats der Zeitangabe

6. Kapitel: Sonstiges

5.1. Datenbank pflegen und sichern

5.1.1 Datenbank komprimieren

5.1.2 Datenbank mit Kennwort verschlüsseln

6 Speichern unter ACCDE-Datei

7 SkyDrive

6.1. Datenbank pflegen und sichern

Je größer Ihr Datenbestand ist, desto wichtiger ist es, die Datenbank zu pflegen und zu sichern. Hier werden einige Möglichkeiten aufgezeichnet, welche Sie regelmäßig durchführen sollten.

6.1.1. Datenbank komprimieren

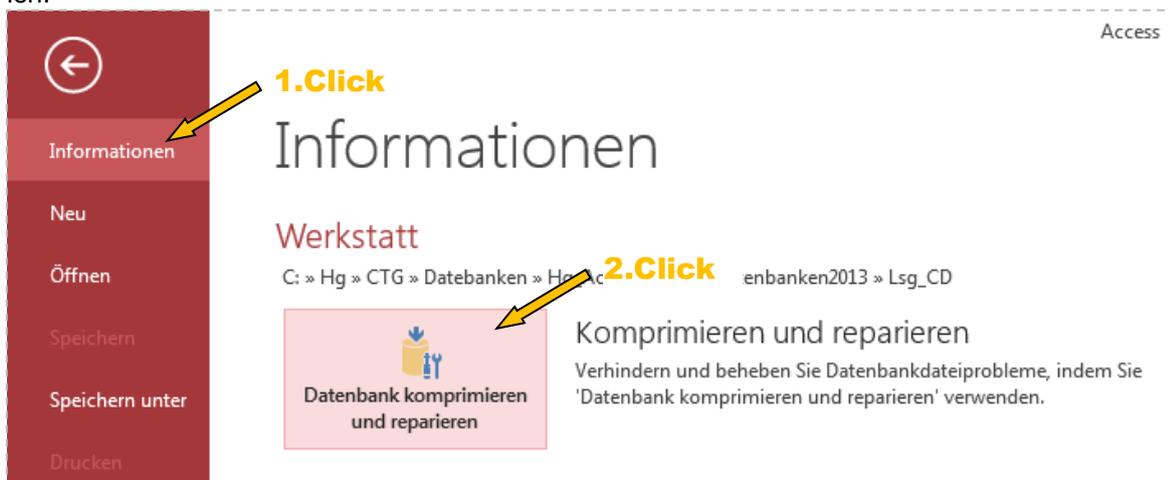
Datenbanken haben einen hohen Speicherbedarf. Dabei belegen leere Datenfelder genauso viel Speicher wie voll Beschriebene. Haben Sie umfangreiche Formulare und Berichte angelegt, dann bläht sich der Speicherbedarf weiter auf.

Hier sollten Sie diese Datenbank komprimieren- auch hinsichtlich eines Backups, welches Sie auf ein externes Speichermedium ablegen wollen.

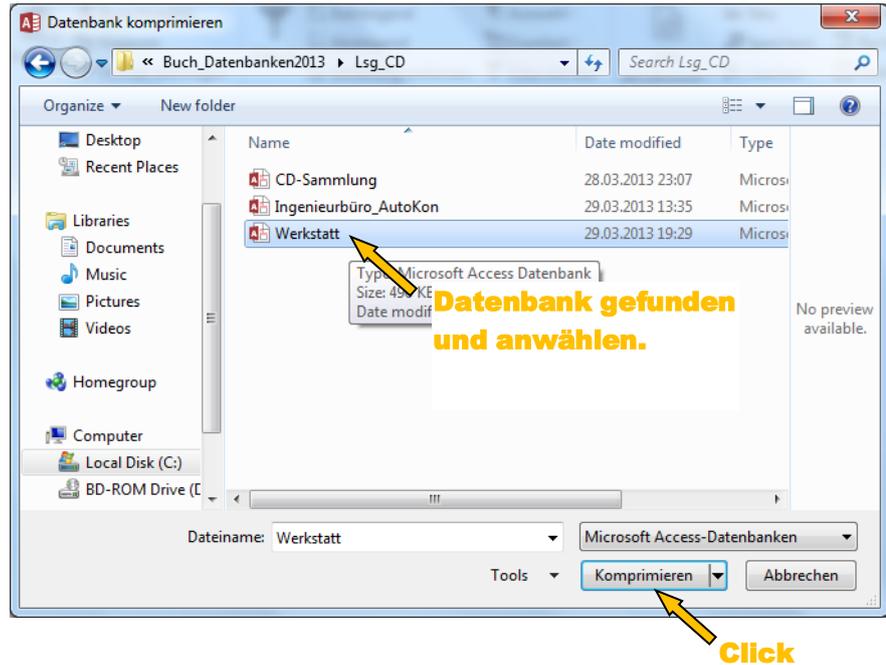
Um eine Datenbank zu komprimieren, gehen Sie in das Backstage-Menü. Haben Sie bereits eine Datenbank geöffnet, können Sie sofort die Schaltfläche zum Komprimieren und reparieren anklicken. Der Prozess erfolgt dann kommentarlos.



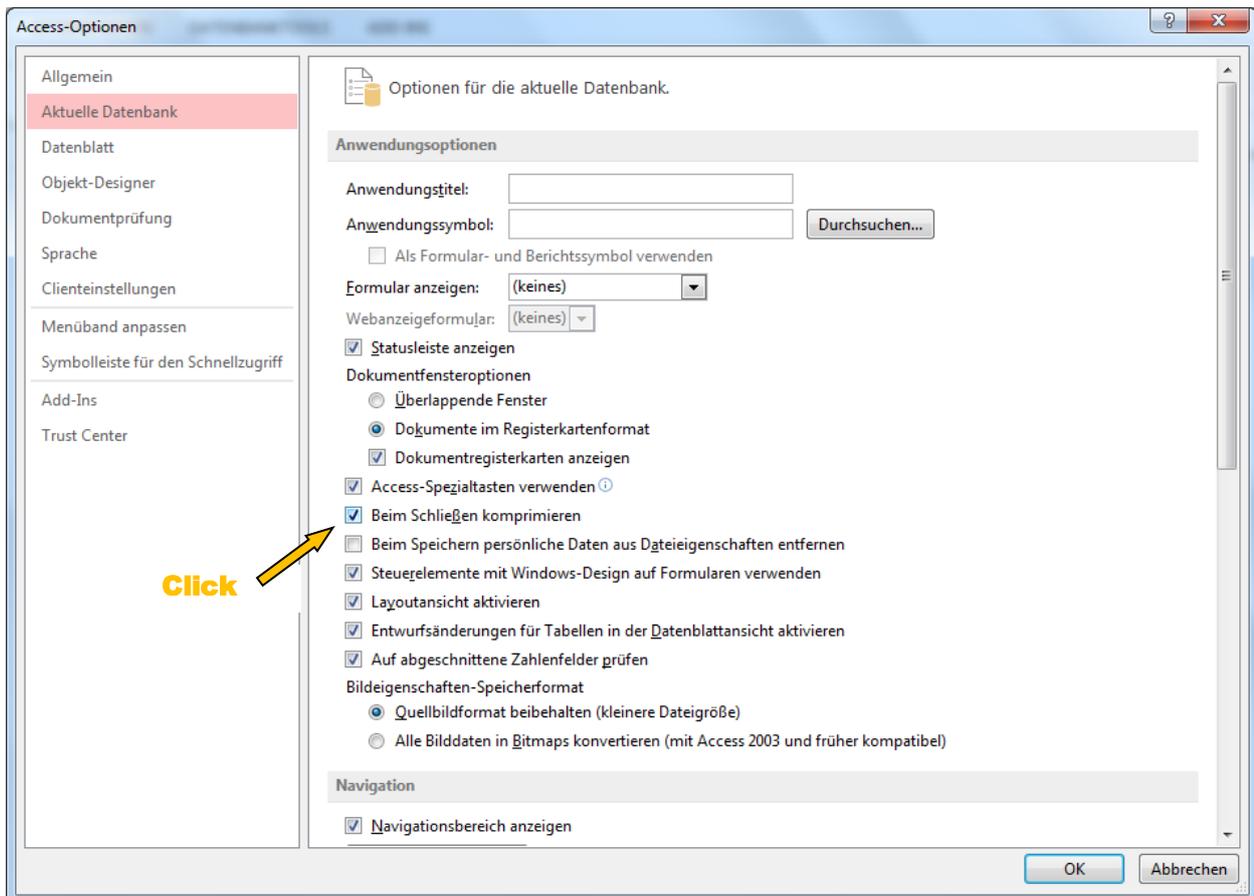
Haben Sie keine Datenbank aktuell geöffnet, dann müssen Sie die zu komprimierende Datenbank auswählen.



Nun geben Sie die Datenbank (Datei) an, welche Sie komprimieren wollen und bestätigen Sie.



Sie können mit wenigen Klicks das Komprimieren **automatisch** beim Schließen einer Datenbank durchgeführt wird. Dazu wandern Sie ins Dialog-Fenster *ACCESS-Optionen* und aktivieren Sie die besagte Option:

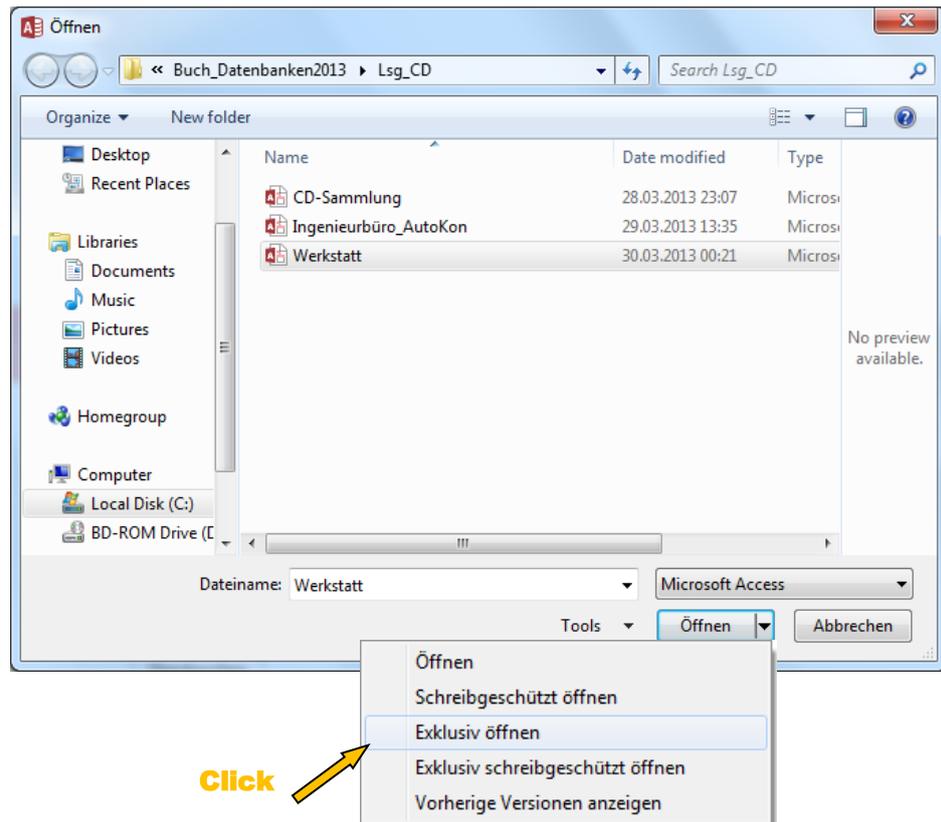


6.1.2. Datenbank mit Kennwort verschlüsseln

Der Schutz der Daten vor Unbefugten ist ein wichtiges Thema. So können viele Dateien auf der DOS-Ebene angesehen werden, wenn die Datenbank nicht verschlüsselt wurde.

Vorgehensweise:

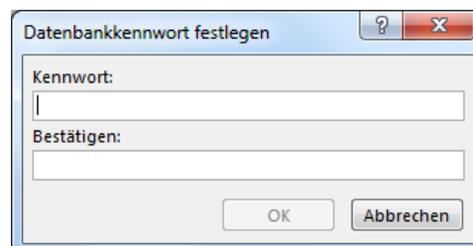
Sie können eine Datenbank nur verschlüsseln, wenn diese im **Exklusiv**-Modus geöffnet ist. Wenn Sie die zu verschlüsselnde Datenbank schon geöffnet haben, dann schließen Sie diese. Sie müssen über die Schaltfläche *Durchsuchen* das Dialogfeld ‚Öffnen‘ öffnen und diese Datei suchen.



Nun können Sie in der Registerkarte **Datei** die Schaltfläche zum Verschlüsseln betätigen. Sofort öffnet sich ein Dialogfenster zum Erfragen eines Kennwortes.



Nach dem Schließen kann die Datenbank nur mit diesem hier vergebenen Kennwort geöffnet werden.



Sie können die Verschlüsselung wieder aufheben. Dazu müssen Sie die Datenbank wieder im Exklusiv-Modus öffnen. In der Registerkarte **Datei** können Sie mit dem Passwort die Verschlüsselung aufheben.

6.2. Speichern unter ACCDE-Datei

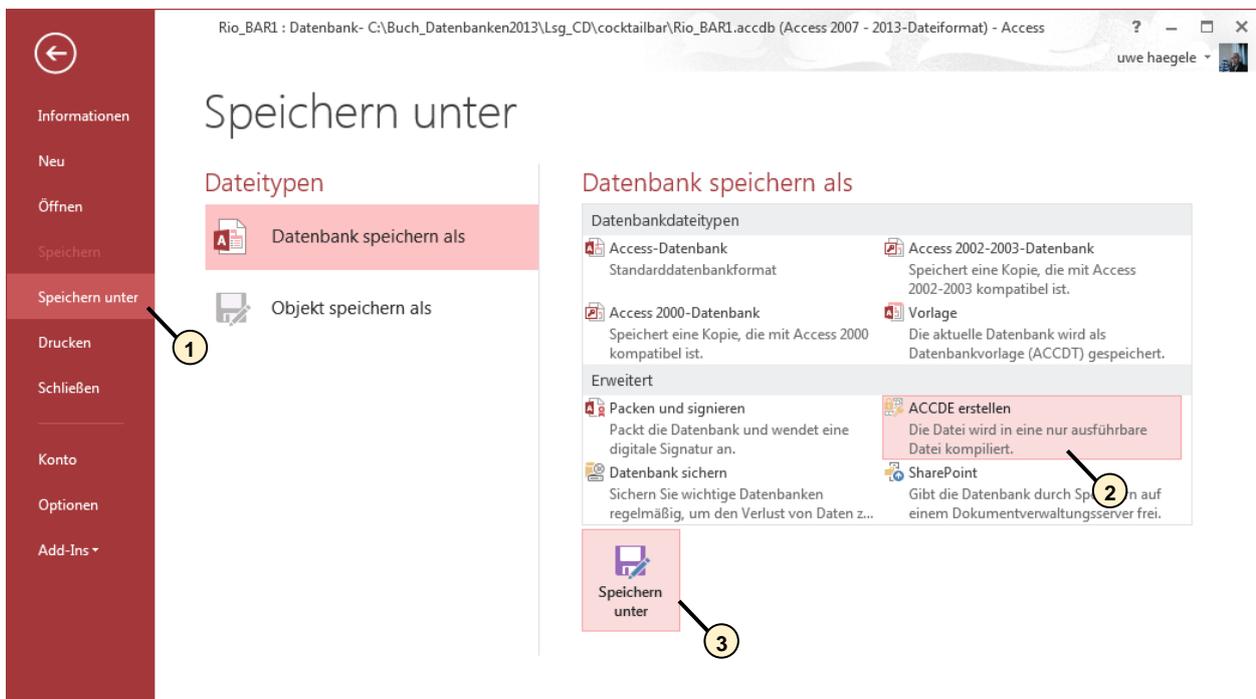
Der vollständige Dateiname von ‚accde‘ ist „**Access Execute Only Database**“ und wird unterstützt von MS-Access ab der Version 2007³. Diese Dateien werden auch als ausführbare Access-Dateien bezeichnet. Um diese Dateien zu öffnen, benötigen aber trotzdem Access oder Access Runtime-Versionen, welche Sie kostenlos aus dem Netz herunterladen können.

Vorteile eines accde-Files?

Im Unterschied zu den accdb Dateien kann der Nutzer einer accde Datei keine Änderungen am Design der Formulare oder Reports vornehmen. Sofern die Datei VBA-Skripte, so sind diese nur in kompilierter Form vorhanden, d.h. der VB-Code ist nicht einsehbar und somit auch nicht änderbar. Zudem ist die Dateigröße kleiner.

Erstellen eines accde-Files?

- ① Hierzu wandern Sie ins Backstage-Menü in den Menü-Punkt ‚Speichern unter‘.
- ② Wählen Sie das gewünschte Datei-Format.
- ③ Zum Schluss ein Klick zur Bestätigung.



³ Bei früheren Versionen von Microsoft Access ist diese Art von Dateien bekannt unter der Endung ‚mde‘.

6.3. SkyDrive

Ein ⁴ ist ein persönlicher Datenspeicher von immerhin 7 Gbyte im Internet. Man kann ihn ähnlich, wie einen USB-Stick oder eine 2. Festplatte nutzen. Der Zugriff erfolgt über ein Login und ein Passwort. Da sich die Daten im Netz befinden, spricht man auch von einer ‚Cloud‘ ⁵ von Microsoft.

Office 2013 hat den Zugriff auf diese Cloud komfortable in die einzelnen Programme eingebunden. Dadurch wird das SkyDrive zu einem festen und alltagstauglichen Bestandteil von Office.



Abbildung: Mobil dank SkyDrive

Vorteile von SkyDrive

- Die Daten auf dem SkyDrive sind weitestgehend vor Verlust geschützt, d.h. es kann keine Festplatte oder kein USB-Stick verlorengehen.
- Kostenlos für all Diejenigen, welche Office 2013 legal erworben haben.
- Man hat grundsätzlich auf die Daten im SkyDrive Zugriff, sobald man sich ins Internet eingeloggt hat, von wo auch immer. Daher kann man diese Daten mit verschiedenen Computern abrufen, sie nutzen und bearbeiten.

Nachteile von SkyDrive

- Da die Daten in einem virtuellen Laufwerk abgelegt sind, ist der Ablegeort der Dateien nicht genau bestimmt und damit die Datensicherheit nicht mehr gewährleistet.
- Die Dateien in dieser Cloud können leicht nach Stichwörtern, etc. gescannt werden und damit ein Persönlichkeitsprofil von Ihnen erstellt werden.

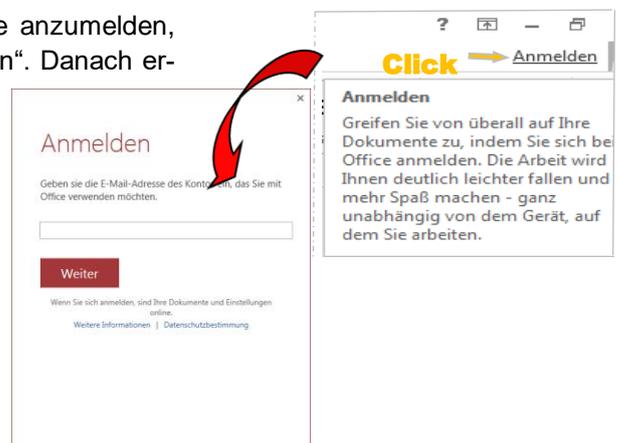
Voraussetzung für SkyDrive

Voraussetzung, dass Sie SkyDrive nutzen können, ist dass Sie eine E-Mail-Adresse über ein Microsoft-Konto haben. Zudem will Microsoft an dieser Stelle, dass Ihr Rechner Cookies erlaubt – dies können Sie über die Systemeinstellungen Ihres Rechners freischalten.

Anmelden bei SkyDrive

Die Schaltflächen, um sich bei SkyDrive anzumelden, befinden sich oben rechts auf „Anmelden“. Danach erfolgt die Abfrage der Zugangsdaten.

Das sind die E-Mail-Adresse sowie das Kennwort Ihres Microsoft-Kontos. Es kann z.B. das Konto sein, über das Sie Office bezogen haben.



⁴ auf Deutsch „Laufwerk im Himmel“

⁵ auf Deutsch „Wolke“

7. Kapitel: Ausgewählte Arbeitsaufträge

6.1. Arbeitsauftrag: Kundenrechnungen verwalten

Lerninhalte: Tabellen, Schlüssel (Primärschlüssel, Fremdschlüssel), Relationen-Schreibweise, ER-Modell, Relationen-Schreibweise, Formulare, Berichte

6.2. Arbeitsauftrag: CD-Sammlung

Lerninhalte: Tabellen, Schlüssel (Primärschlüssel, Fremdschlüssel), Relationen-Schreibweise, ER-Modell, Relation bzw. Relationen-Schreibweise, Auflösung von $m:n$ Beziehungen, Berichte, Formulare

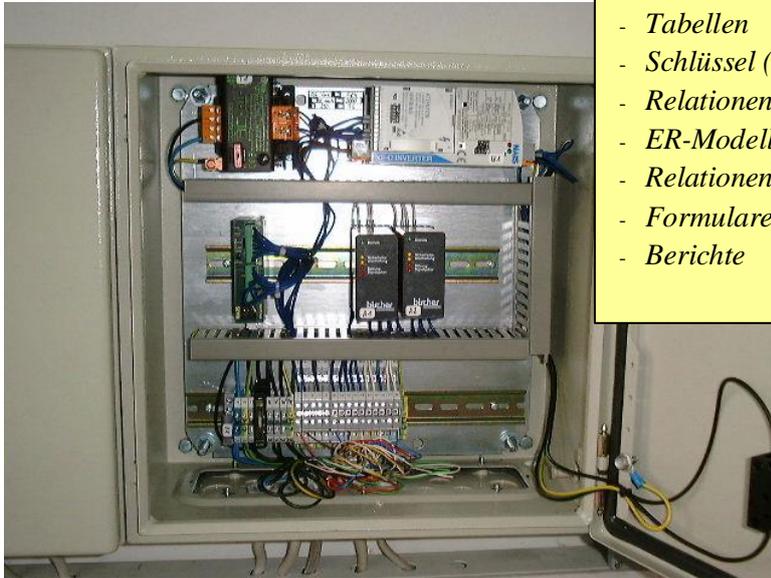
6.3. Arbeitsauftrag: Showroom

Lerninhalte: Relationale Datenbanken, Tabellen, Schlüssel (Primärschlüssel), ER-Modell, **Bildverwaltung**

6.4. Arbeitsauftrag: Cocktailbar

Lerninhalte: Tabellen, Schlüssel (Primärschlüssel, Fremdschlüssel), Relationen-Schreibweise, ER-Modell, **Bildverwaltung, Steuerelemente und Visual Basic**

Arbeitsauftrag: Kundenrechnungen



Lerninhalte:

- Tabellen
- Schlüssel (Primärschlüssel, Fremdschlüssel)
- Relationen-Schreibweise
- ER-Modell
- Relationenschreibweise
- Formulare
- Berichte

Sie führen ein eigenes Ingenieurbüro „AutoKon GmbH“. Hierbei bauen Sie SPS-Steuerungen und programmieren diese.

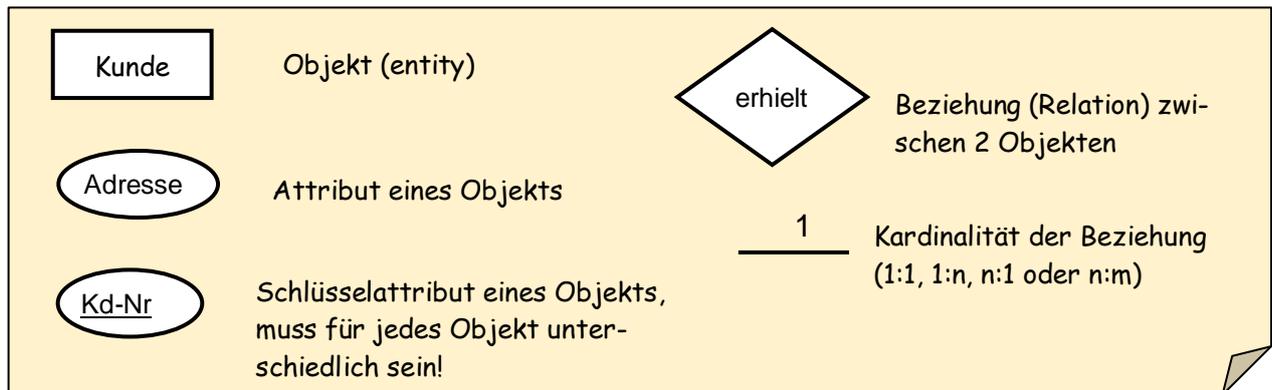
7.1. Arbeitsauftrag: Kundenrechnungen verwalten

Die Rechnungs-Daten sind derzeit in „einfachen“ Excel-Dateien gespeichert. Um einen Überblick die Aufträge und Kunden zu haben, wollen Sie eine benutzerfreundliche Datenbank erstellen, welche eine einfache Pflege der Daten für die Abwicklung von Kundenaufträgen erlaubt.

Rechnungssendung:

<p>Ingenieurbüro AutoKon GmbH Prachtallee 18 81455 Nürnberg</p>	<p>Tel.: 081334 / 235945 FAX: 081334 / 4573245</p>																														
<p>- RBS Stuttgart Stuttgart</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Auftragsbeschreibung:</td> <td>SPS-Steuerung für Transfer</td> </tr> <tr> <td>Kundennummer:</td> <td>00024</td> </tr> <tr> <td>Rechnungsdatum:</td> <td>12.01.13</td> </tr> </table>	Auftragsbeschreibung:	SPS-Steuerung für Transfer	Kundennummer:	00024	Rechnungsdatum:	12.01.13																								
Auftragsbeschreibung:	SPS-Steuerung für Transfer																														
Kundennummer:	00024																														
Rechnungsdatum:	12.01.13																														
<p>Rechnung</p> <p>Sehr geehrter Herr Moschner Vielen Dank für Ihren Auftrag über die Erstellung und Programmierung einer SPS-Steuerung für die Transferstraße. Der Kostenvoranschlag ist wie folgt:</p>																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Pos.</th> <th style="width: 40%;">Einzelkosten</th> <th style="width: 25%;">Bemerkung</th> <th style="width: 10%;">Preis/Einheit</th> <th style="width: 20%;">Gesamt</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Materialkosten</td> <td>Siehe Anlage</td> <td style="text-align: center;">-</td> <td style="text-align: right;">1360,00</td> </tr> <tr> <td>2</td> <td>Arbeitszeit</td> <td>40 Stunden</td> <td style="text-align: right;">45,00</td> <td style="text-align: right;">1800,00</td> </tr> <tr> <td>3</td> <td>Verbrauchsmaterial</td> <td>1</td> <td style="text-align: right;">20,00</td> <td style="text-align: right;">20,00</td> </tr> <tr> <td>4</td> <td>Fahrtkosten</td> <td>4 Stunden</td> <td style="text-align: right;">25,00</td> <td style="text-align: right;">100,00</td> </tr> <tr> <td colspan="4"></td> <td style="text-align: right;"><u>3280,00</u></td> </tr> </tbody> </table>		Pos.	Einzelkosten	Bemerkung	Preis/Einheit	Gesamt	1	Materialkosten	Siehe Anlage	-	1360,00	2	Arbeitszeit	40 Stunden	45,00	1800,00	3	Verbrauchsmaterial	1	20,00	20,00	4	Fahrtkosten	4 Stunden	25,00	100,00					<u>3280,00</u>
Pos.	Einzelkosten	Bemerkung	Preis/Einheit	Gesamt																											
1	Materialkosten	Siehe Anlage	-	1360,00																											
2	Arbeitszeit	40 Stunden	45,00	1800,00																											
3	Verbrauchsmaterial	1	20,00	20,00																											
4	Fahrtkosten	4 Stunden	25,00	100,00																											
				<u>3280,00</u>																											
<p>Die Preise verstehen sich zuzüglich der gesetzlichen Mehrwertsteuer. Bitte zahlen Sie die Rechnung innerhalb 10 Tagen nach Rechnungseingang.</p> <p>Mit freundlichen Grüßen</p> <p>AutoKon GmbH</p>																															

Schritt 1: Erstellen des Entity-Relationship-Modells



Es lassen sich drei Entitäten samt ihren Attributen definieren:

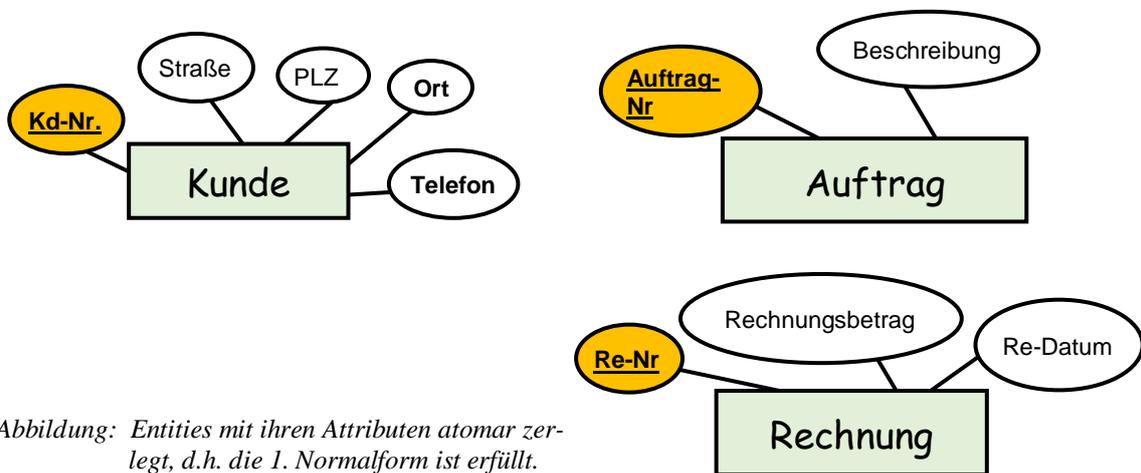


Abbildung: Entitäten mit ihren Attributen atomar zerlegt, d.h. die 1. Normalform ist erfüllt.

Nun geht es darum, darzustellen, wie diese Identitätstypen zueinander in Beziehung stehen.

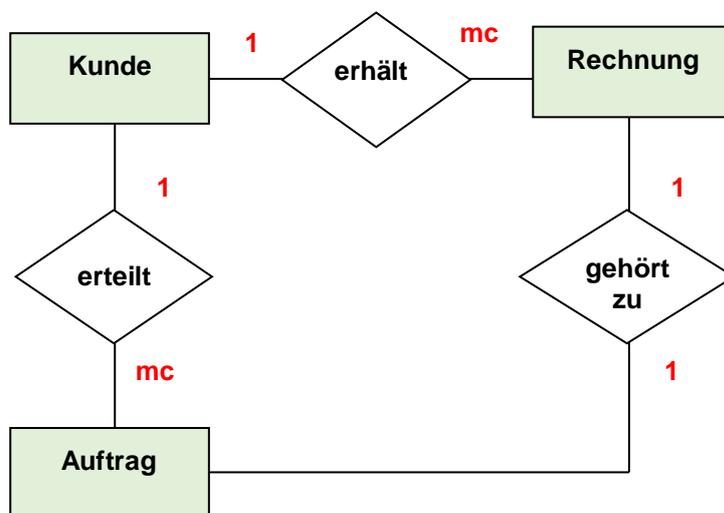


Abbildung: ER-Modell mit den Kardinalitäten

Schritt 2: Beschreibung der DB in Relationenschreibweise

Setzen Sie Primärschlüssel und Fremdschlüssel durch Unterstreichen des betreffenden Attributes innerhalb des Datensatzes und zeichnen Sie die Verknüpfung.

Kunde (KNr, Nachname, Vorname, Straße, Postleitzahl, Ort)

Auftrag (A_Nr, Beschreibung, KNr)

Rechnung (Re-Nr, Rechnungsbetrag, KNr)

Legende: Primärschlüssel Fremdschlüssel



Schritt 3: Tabellen erstellen und in Beziehung setzen

Die Tabellen hierzu werden wie folgt aussehen:

<u>KNr</u>	Nachname	Vorname	Straße	PLZ	Ort
Zahl	Kurzer Text	Kurzer Text	Kurzer Text	Zahl	Kurzer Text
0001	Heinrich	Max	Mozartstr. 1	83221	Ludwigsberg

<u>A_Nr.</u>	<u>KNr.</u>	Beschreibung
Zahl	Zahl	Kurzer Text
2341	0001	Schaltschrank

<u>Re_Nr.</u>	<u>KNr.</u>	Re_Betrag
Zahl	Zahl	Währung
11.23	0001	1022,00

Erstellen Sie die Tabellen, verknüpfen Sie diese und erstellen Sie Formulare zur einfachen Bedienung des Anwenders. Diese Datenbank findet sich auf der Lösungs-CD.

CD-Sammlung

Lerninhalte:

- Tabellen
- Schlüssel (Primärschlüssel, Fremdschlüssel)
- Relationen-Schreibweise
- ER-Modell
- Relation bzw. Relationenschreibweise
- Auflösung von m:n Beziehungen
- Berichte
- Formulare



7.2. Arbeitsauftrag: CD-Liste verwalten

Sie laden sich legal Musik-Clips aus dem Internet und speichern diese auf eine CD. Um einen Überblick zu bekommen, welche CDs welche Lieder beinhalten, erstellen Sie eine Datenbank.



Dieser Arbeitsauftrag befindet sich auf der Lösungs-CD!

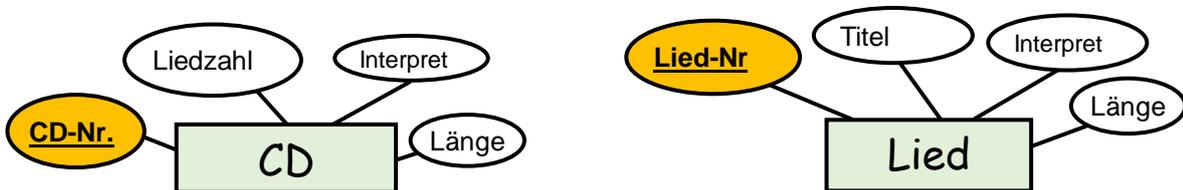
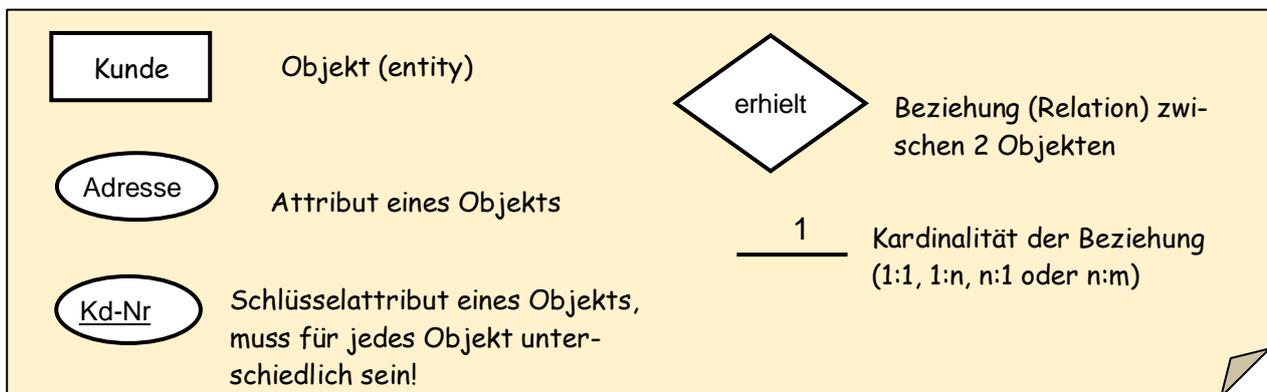
Um einen Überblick der Lieder auf Ihre CDs zu haben, wollen Sie eine benutzerfreundliche Datenbank erstellen.



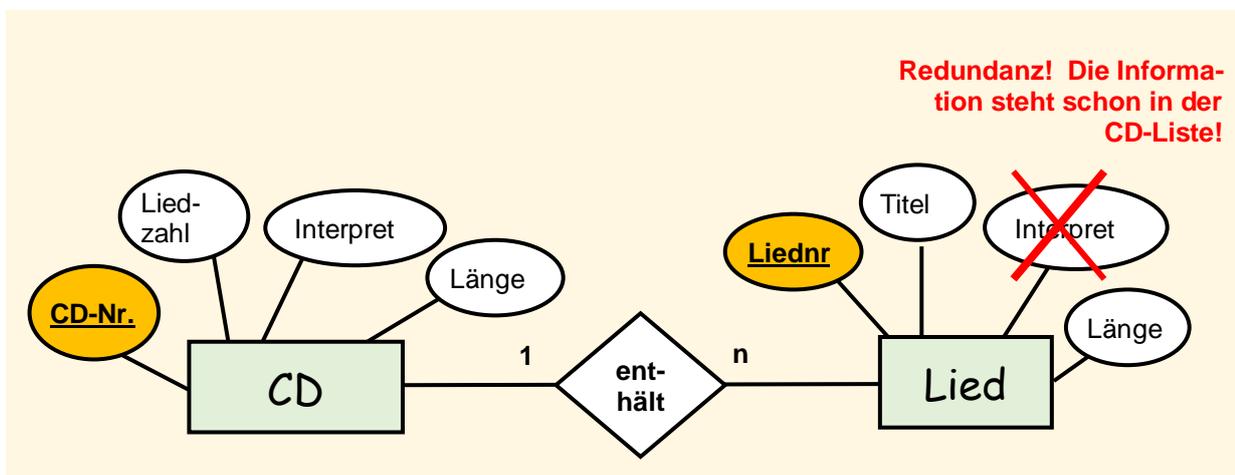
Zunächst soll die Einschränkung gemacht werden, dass sich auf einer CD immer nur Lieder eines Interpreten befinden.

Schritt 1: Erstellen des Entity-Relationship-Modells

Erstellen Sie das ER-Modell ausgehend von der Entität ,CD':



Die Entitäten ,CD' und ,Lied' müssen noch in Beziehung gesetzt werden:



Schritt 2: Beschreibung der DB in Relationenschreibweise

CD (CD-Nr., Interpret, Länge, Liedzahl)



Lied (LiedNr., CD-Nr., Titel, Länge)

Legende:

Primärschlüssel

Fremdschlüssel

Schritt 3: Tabellen erstellen und in Beziehung setzen

<u>CD-Nr.</u>	Interpret	Länge	Liedzahl
<u>Zahl</u>	Text	Zahl/Zeit	Zahl
1	Abba	49:32	9
2	B. Adams	66:12	5
3

<u>LiedNr.</u>	<u>CD-Nr.</u>	Titel	Länge	Interpret
<u>Zahl</u>	Zahl	Text	Zahl/Zeit	Text
1	2	Summer of `69	3:35	B. Adams
2	2	Heaven	4:03	
3	1	Waterloo	4:06	Abba

Fremdschlüssel „CD-Nr.“:

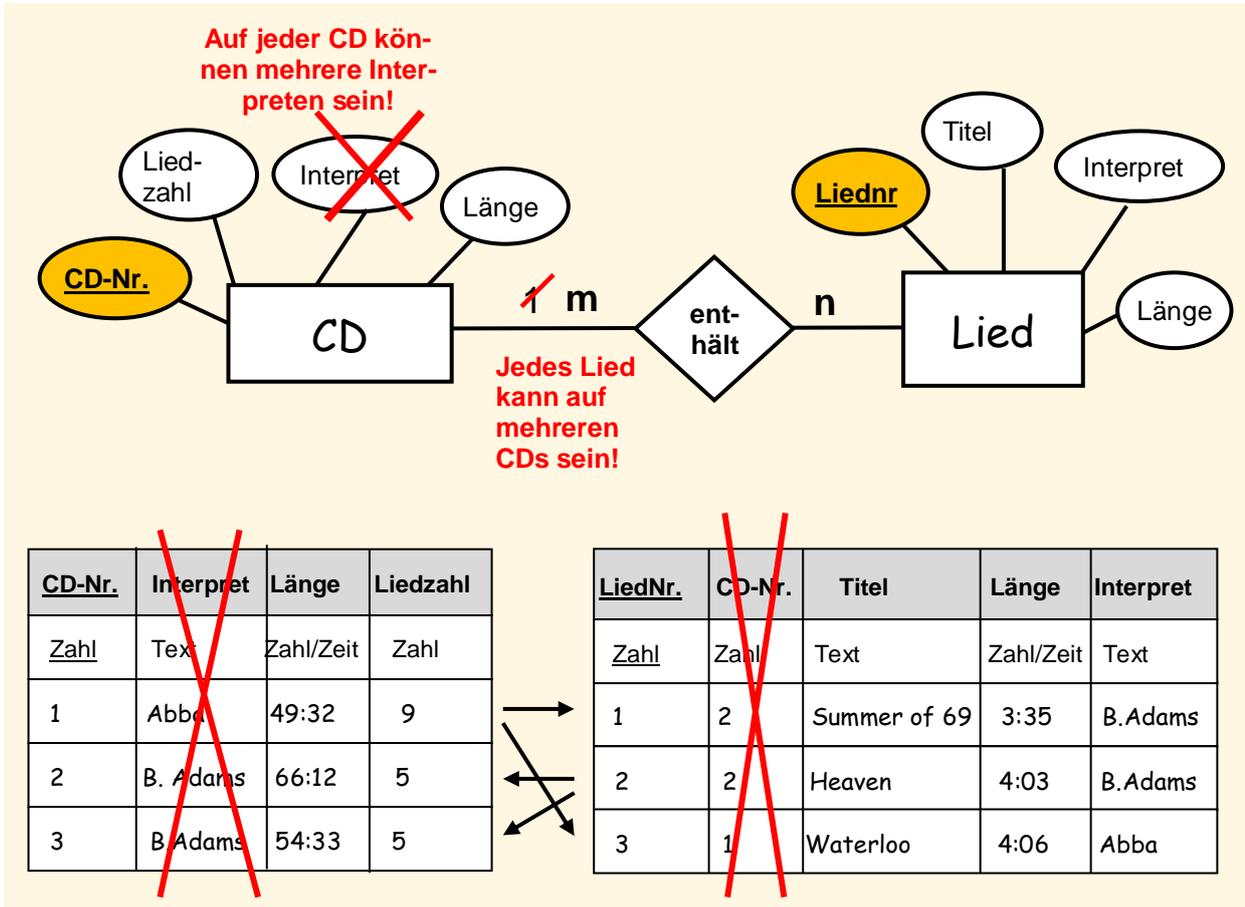
Der Eintrag im Datenfeld CD-Nr. der Tabelle ‚Lied‘ muss als Primärschlüssel in der Tabelle ‚CD‘ vorkommen!

Erst jetzt erstellen Sie die Tabellen in ACCESS und setzen Sie diese in Beziehung. In ACCESS geschieht dies über „Schlüssel“.



Erweiterung: Auf jeder CD können verschiedene Interpreten sein und ein Lied kann sich auf mehreren CDs befinden

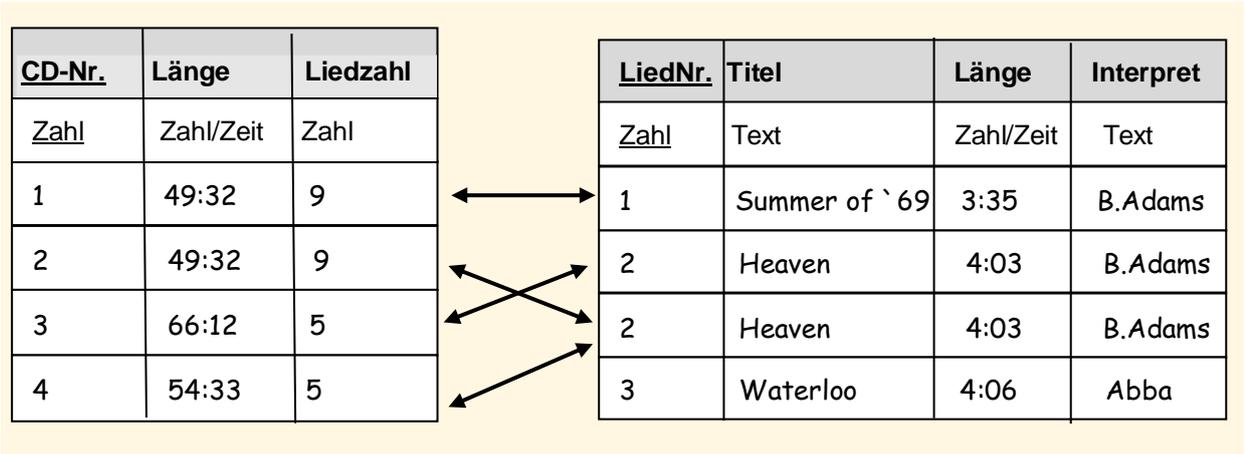
Schritt 1: Erstellen des Entity-Relationship-Modells



Wieso müssen die Spalten „Interpret“ und „CD-Nr.“ entfallen?

Die Spalten Interpret und CD-Nr. müssen entfallen, da die Einschränkung „zu jeder CD gehört ein Interpret“ entfällt.

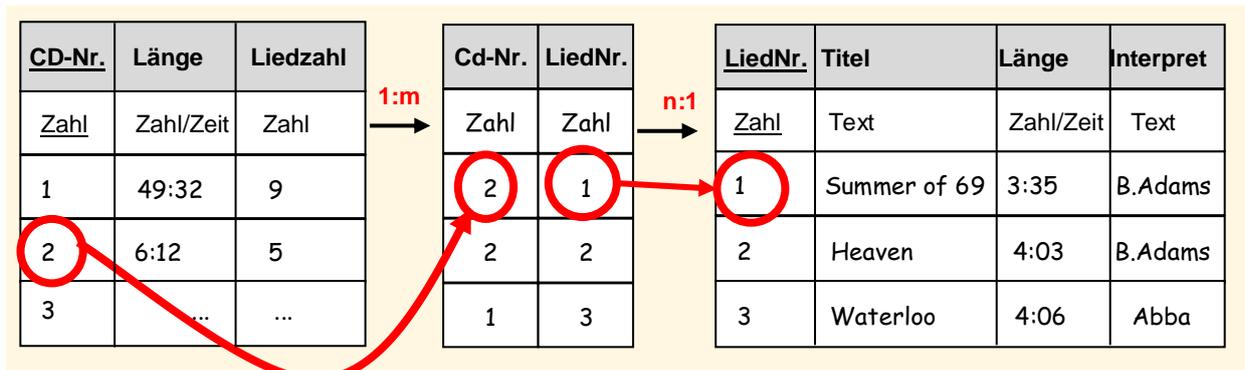
Wir schreiben die obigen Tabellen um, so dass jeder Pfeil einer Tabelle eindeutig auf die jeweils andere Tabelle zeigt:



✎ Welches Problem liegt nun vor?

Ein Lied kann nicht mehr eindeutig einer CD zugeordnet werden. So findet sich z.B. das Lied mit Nr.2. auf der CD 2 und 3.

Lösung: Auflösung einer m:n-Beziehung in eine 1:m und n:1 Beziehung mittels einer Beziehungstabelle:



Schritt 2: Beschreibung der DB in Relationenschreibweise

CD (CD-Nr., Interpret, Länge, Liedzahl)

Beziehungstabelle (CD-Nr., LiedNr.)

Lied (LiedNr., CD-Nr., Titel, Länge)

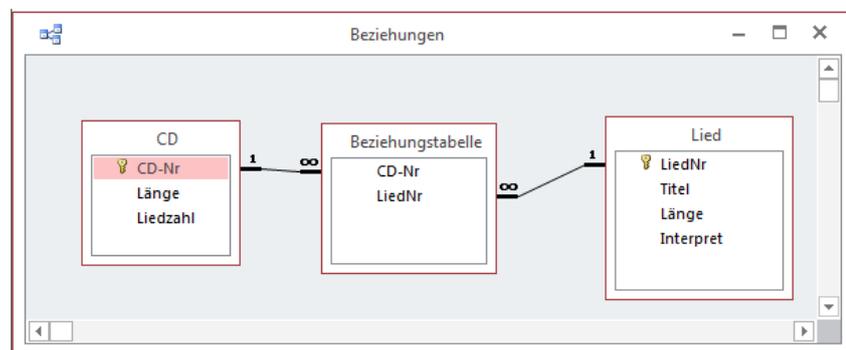
Legende:

Primärschlüssel

Fremdschlüssel

Schritt 3: Implementieren der Datenbank in ACCESS2013

Nachdem die Tabellen definiert wurden und mittels Schlüssel verknüpft wurden, präsentiert sich die Beziehungstabelle wie folgt:



Diese Datenbank findet sich auf der Lösungs-CD!

Showroom eines Autohauses



Lerninhalte:

Relationale Datenbanken

- Tabellen
- Schlüssel (Primärschlüssel)
- Entity-Relationship-Modell
- **Bildverwaltung**

7.3. Arbeitsauftrag: Bilder.- und Datenverwaltung

Kein Autohaus kann erfolgreich ihre Autos verkaufen, wenn nicht ein ansprechender Showroom vorhanden ist. Ihre Aufgabe ist es, eine Datenbank über die ausgestellten Autos zu entwickeln. Neben den Kennwerten, wie Leistung, Hubraum, etc. soll beim Aufruf eines Autos, d.h. Datensatz, ein Bild dieses Autos im Formular erscheinen.

Info:

Generell bieten sich zwei Möglichkeiten in ACCESS an, Bilder anzuzeigen und zu verwalten:

- **Bilder in eine Tabelle importieren**

Das Bild – als OLE⁶-Objekt – wird in einer Tabelle selbst abgespeichert. **Nachteil** ist, dass sich der Speicherbedarf der Tabelle unverhältnismäßig vergrößert.

- **Bildaufwurf durch Pfad**

Der betreffende **Bildpfad** wird beim Aufruf des Datensatzes an ein ungebundenes Objekt weitergeleitet. Diese Möglichkeit sollten Sie **unbedingt nutzen**, wenn Sie **viele Bilder** haben.

Anhand dieses Arbeitsauftrags sollen beide Möglichkeiten aufgezeigt werden.



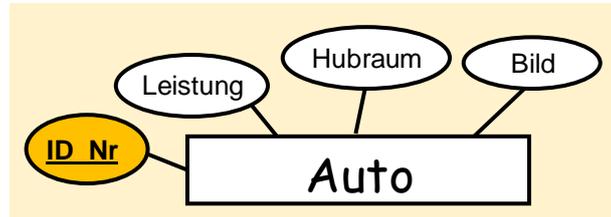
Dieser Arbeitsauftrag befindet sich auf der Lösungs-CD!

⁶ OLE = ‚Objekt Linking and Embedding (Objekt-Verknüpfung und –Einbettung)‘ ist eine von Microsoft entwickelte Technik, um Objekte (Grafik, Tabellen, etc.) in ein anderes Dokument einzubetten.

Lösungsmöglichkeit 1: Showroom – Bilder werden als OLE-Objekt in Tabelle abgespeichert

Schritt 1: Erstellen des Entity-Relationship-Modells

Da hier der Schwerpunkt in der Bilderverwaltung und im Einsatz von Steuerelementen liegen soll, soll das ER-Modell hier einfach gestaltet sein.



Schritt 2: Tabellenentwurf und Füllen mit Werten

Zuerst müssen Sie die Tabelle in Entwurfsansicht erstellen:

- ① Die Identifikation-Nummer des Bildes kann ein Autowert darstellen. Diese wird den Primärschlüssel bilden.
- ② Wichtig: Felddatentyp von Bild ist **OLE-Objekt**.

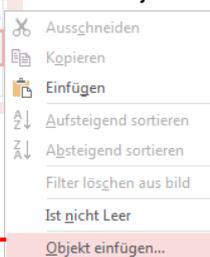
Abbildung: Tabelle in Entwurfsansicht

Feldname	Felddatentyp
ID_Nr	AutoWert
Automarke	Kurzer Text
Leistung	Kurzer Text
Hubraum	Kurzer Text
bild	OLE-Objekt

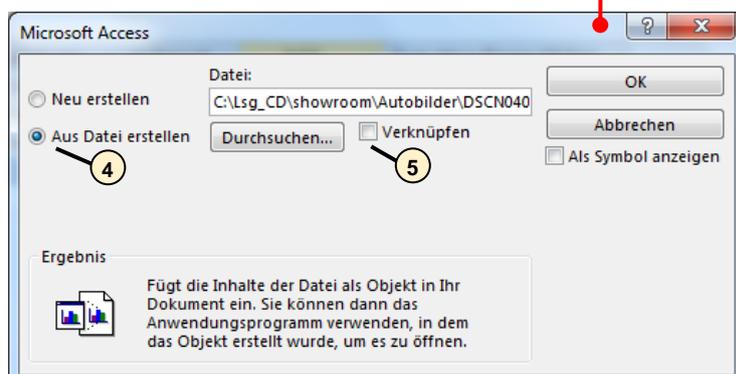
Um konkrete Werte einzugeben, schalten Sie um in Datenblattansicht. Der Wert ‚ID_Nr‘ wird automatisch vergeben, die Werte ‚Automarke‘, ‚Leistung‘, ‚Hubraum‘ sind als kurzer Text definiert. Die Angabe, wie das Bild einzubinden ist, gestaltet sich wie folgt:

ID_Nr	Automarke	Leistung	Hubraum	bild
1	FIAT 500 C Lounge	51 kW	1348 ccm	Bitmap Image
2	FIAT 500 L Pop Star	70 kW (95 PS)	1368 ccm	Bitmap Image
3	FIAT Punto 6 More	57 kW (77 PS)	1368 ccm	Bitmap Image
5				
*	(Neu)			

- ③ Wählen Sie das Feld ‚bild‘ an, öffnen Sie das Kontextmenü (rechte Maustaste), um ein Objekt einzufügen.



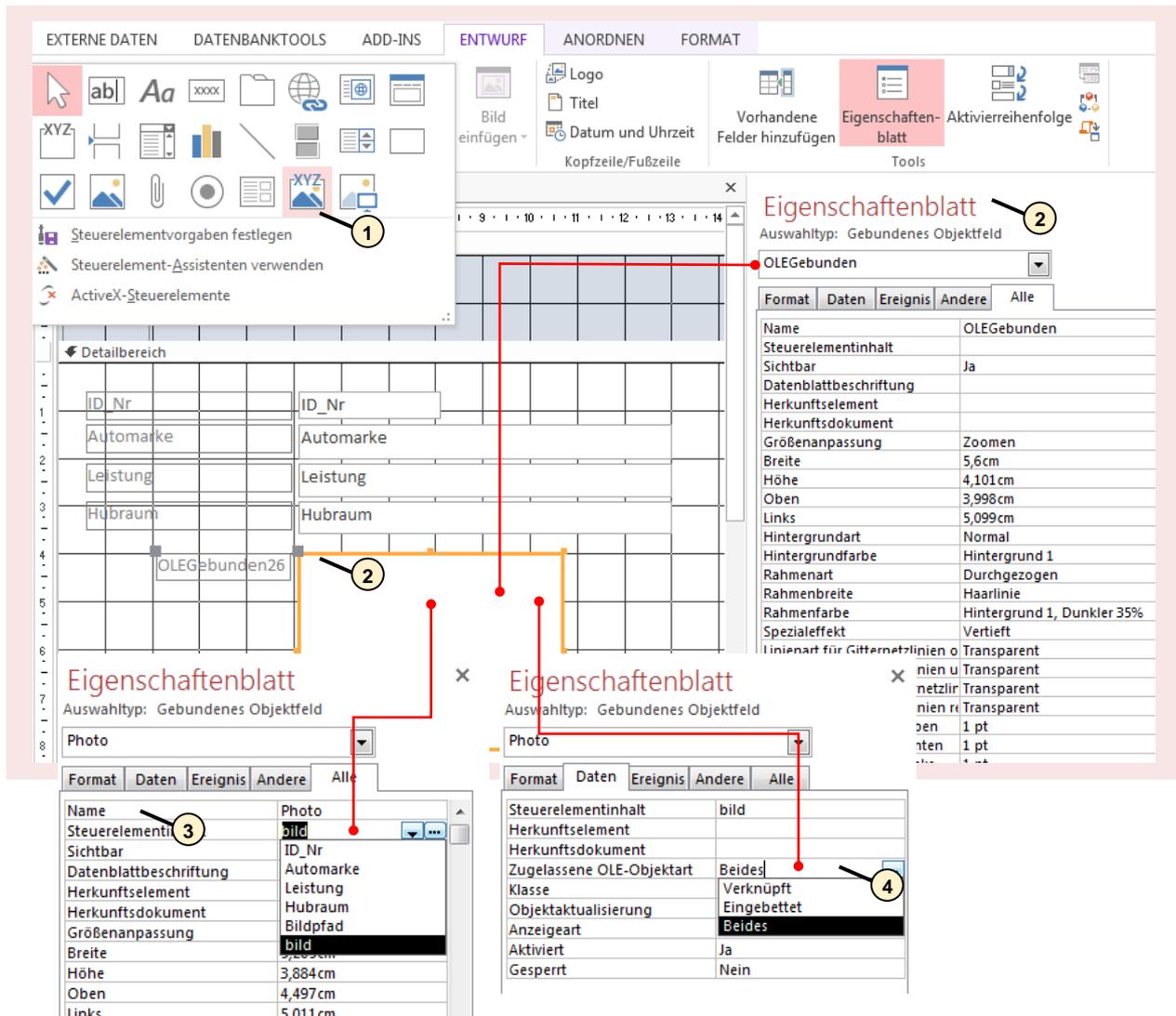
- ④ Haben Sie die Fotos als Datei schon vorhanden, dann wählen Sie diese Option und klicken Sie auf den Botton „Durchsuchen“, um den Speicherort anzugeben.



- ⑤ Sie können mit Doppelklick aus ACCESS heraus das Bild editieren. Wenn Sie ein Programm hierzu angeben wollen, dann verknüpfen Sie dieses.

Schritt 3: Entwurf Formular

Erstellen Sie ein Formular in der Entwurfsansicht und fügen Sie die Felder der Tabelle ‚Auto‘ (ID-Nr, ‚Automarke‘, ‚Leistung‘, ‚Hubraum‘) in gewohnter Weise ein. Das Bild selbst stellt ein OLE-Objekt dar. Hierzu fügen Sie ein Steuerelement Typ ‚OLEGebunden‘ ins Formular ein.



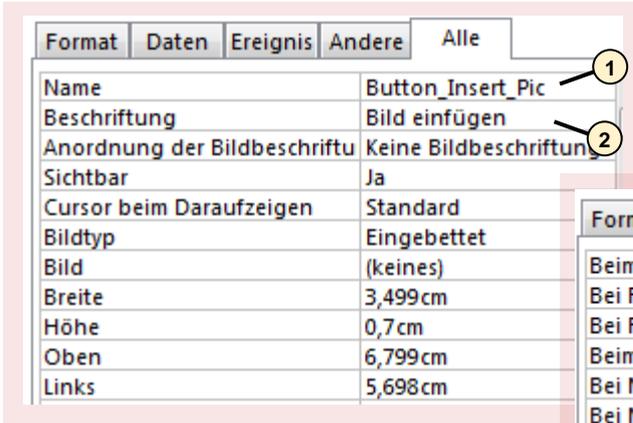
- ① Sie müssen sich in der Entwurfsansicht befinden. Über dieses Icon platzieren Sie ein gebundenes Objektfeld in Ihr Formular, in welchem ein Bild einbetten kann.
- ② Daraufhin öffnet sich das ‚Eigenschaftsblatt‘, in welchem Sie sämtliche, auf dieses Objekt angewandte, Einstellungen vornehmen können. Sie können dieses Eigenschaftsfenster jederzeit öffnen, indem Sie auf den Rand des Objektes klicken.
- ③ Passen Sie die Eigenschaften dieses gebundenen Textfeldes an. Dazu gehört der Name und als Steuerelementinhalt das OLE-Feldobjekt ‚bild‘.
- ④ Weiterhin geben Sie hier schon an, ob das Objekt, in diesem Falle ein Bild, mit einem (Bildverarbeitungs-) Programm verknüpft werden soll. Dann können Sie aus ACCESS heraus durch Doppelklick das eingebundene Bild bearbeiten.

Fertig: Damit haben können Sie nun innerhalb des Formulars die Datensätze samt den Bildern anschauen.

Optimierung: neue Bilder über das Formular Einfügen

Bild einfügen

Was für Textfelder möglich ist, nämlich die Eingabe von Werten über das Formular, soll auch für das Bild möglich sein. Dazu fügen Sie eine Schaltfläche ein, ändern die Eigenschaft und weisen einen VBA-Code zu.



- ① Ändern Sie den Namen, unter welchem dieses Objekt im Programm angesprochen wird.
- ② Die Beschriftung dient dem Bediener.



- ③ Unter der Registerkarte ‚Ereignis‘ soll eine Prozedur ablaufen. Klicken Sie in das Feld neben ‚Beim Klicken‘. Es erscheint die Schaltfläche , welche Sie anklicken, um das Programmfenster zu öffnen.

Der VBA-Code, welcher der Schaltfläche ‚Bild einfügen‘ zugeordnet ist, ist abgebildet:

```

Microsoft Visual Basic for Applications - showroom - [Form_Auto (Code)]
Datei Bearbeiten Ansicht Einfügen Debuggen Ausführen Extras Add-Ins Fenster ?
Projekt - Datenbank1
Datenbank1 (showroom)
  Microsoft Access Klassenobjekte
    Form_Auto
  Button_Insert_Pic
    Option Compare Database
    Private Sub Button_Insert_Pic_Click()
      On Error Resume Next
      Me!Photo.Action = acOLEInsertObjDlg
    End Sub
    
```

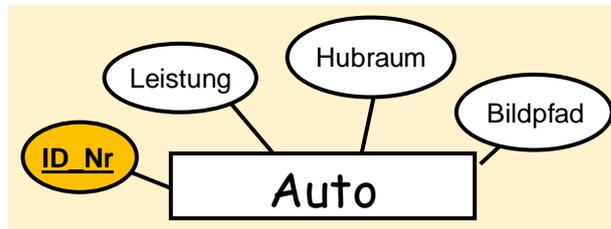
Bemerkung:

Bei der Anweisung ‚On Error Resume Next‘ handelt es sich um eine ungesteuerte Fehlerbehandlung, d.h. der Fehler wird einfach „übersprungen“.

Lösungsmöglichkeit 2: Showroom – Bilder einbetten durch Pfadangabe

Schritt 1: Erstellen des Entity-Relationship-Modells

Im Entity-Relationship-Modell ersetzen Sie die Eigenschaft ‚Bild‘ durch die Eigenschaft ‚Bildpfad‘.



Schritt 2: Tabellenentwurf und Füllen mit Werten

- ① Die Identifikation-Nummer des Bildes sei Autowert.
- ② Die restlichen Felder sind vom Typ ‚kurzer Text‘. Hierzu gehört natürlich auch die Pfadangabe des Bildes.

Abbildung: Tabelle in Entwurfsansicht

Feldname	Felddatentyp
ID_Nr	AutoWert ①
Automarke	Kurzer Text
Leistung	Kurzer Text
Hubraum	Kurzer Text
Bildpfad	Kurzer Text ②

Abbildung: Tabelle in Entwurfsansicht

Schalten Sie um in die Datenblattansicht!
Hier können Sie konkrete Werte eingeben.



Datenblattansicht der Tabelle 'Auto2' mit den folgenden Daten:

ID_Nr	Automarke	Leistung	Hubraum	Bildpfad
1	FIAT 500 C Lounge	51 kW	1348 ccm	.\Autobilder\FIAT 500 C Lounge.bmp
2	FIAT 500 L Pop Star	70 kW (95 PS)	1368 ccm	.\Autobilder\FIAT 500 L Pop-Star.bmp
3	FIAT Punto 6 More	57 kW (77 PS)	1368 ccm	.\Autobilder\FIAT Punto.bmp
* (Neu)				

- ③ Bei der Pfadangabe empfiehlt sich eine relative Pfadangabe, d.h. Pfad ausgehend vom aktuellen Projektpfad. Dadurch ist das Bild auch zu finden, wenn Sie das Projekt in einem anderen Verzeichnis ablegen.

Schritt 3: Entwurf Formular

Erstellen Sie ein neues Formular in der Entwurfsansicht. Zunächst setzen Sie die vorhandenen Felder der Tabelle ‚Auto2‘ so wie in den vorhergehenden Arbeitsaufträgen ein und positionieren Sie die Felder an der gewünschten Stelle.

Was nun dieses Formular interessant macht, ist der Aufruf der Bilder über das Steuerelement ‚Bild‘. Richten Sie dieses Steuerelement wie folgt ein:

Der Steuerelementinhalt ist der ‚Bildpfad‘. Über eine Auswahlliste können Sie auf die Tabellenfelder zugreifen.

Nun ist diese Datenbank in Ihrer Minimalfunktion, d.h. ohne Erweiterung um einen VBA-Code, schon lauffähig.

Cocktailbar



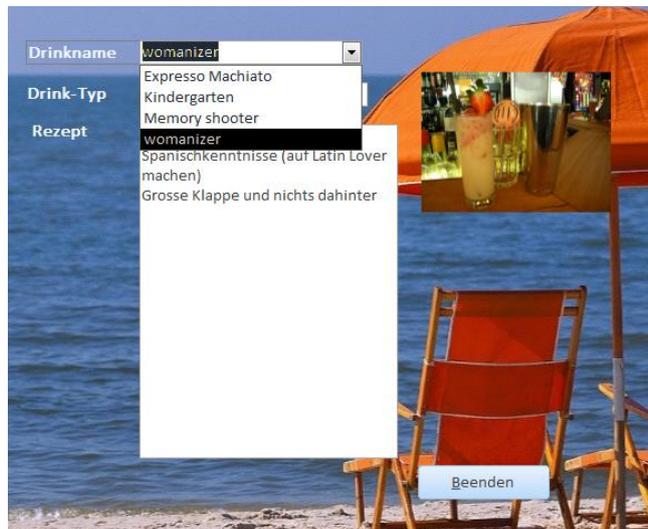
Lerninhalte:

- Tabellen
- Schlüssel (Primärschlüssel, Fremdschlüssel)
- Relationen-Schreibweise
- ER-Modell
- Bildverwaltung
- Steuerelemente und Visual Basic

7.4. Arbeitsauftrag: Datenbank für Cocktailbar

Sie wollen eine Datenbank für Ihre Cocktailbar einrichten. Dazu wollen Sie neben dem Rezept auch ein Bild zum jeweiligen Cocktail speichern.

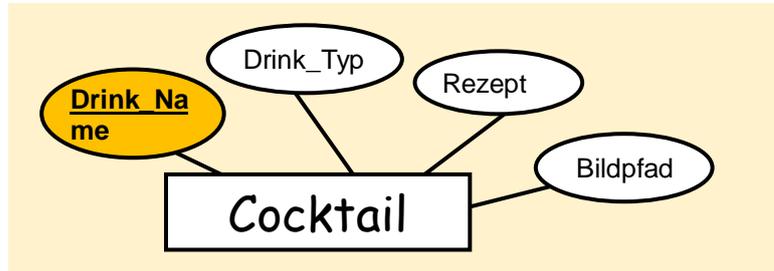
Die Drinks anwählen wollen Sie über ein Kombinationsfeld. Die Drinks selbst sollen über ihren Namen gewählt werden.



Dieser Arbeitsauftrag befindet sich auf der Lösungs-CD!

Schritt 1: Erstellen des Entity-Relationship-Modells

Da hier der Schwerpunkt in der Bilderverwaltung und im Einsatz von Steuerelementen liegen soll, soll das ER-Modell hier einfach gestaltet sein.



Schritt 2: Tabellenentwurf und Füllen mit Werten

Erstellen Sie eine Tabelle in der Entwurfsansicht. Die Felddatentypen sind aus dem ER-Modell zu sehen und wurden in der folgenden Abbildung umgesetzt.

- Da auch ein Drink über sein Name identifiziert werden kann, sei dem Attribut ‚DRINK_NAME‘ der Primärschlüssel zugewiesen.
Die Bilder sind über deren Pfadangabe gespeichert.

Feldname	Felldatentyp
DRINK_NAME	Kurzer Text
DRINK_TYP	Kurzer Text
Rezept	Langer Text
Bildpfad	Kurzer Text

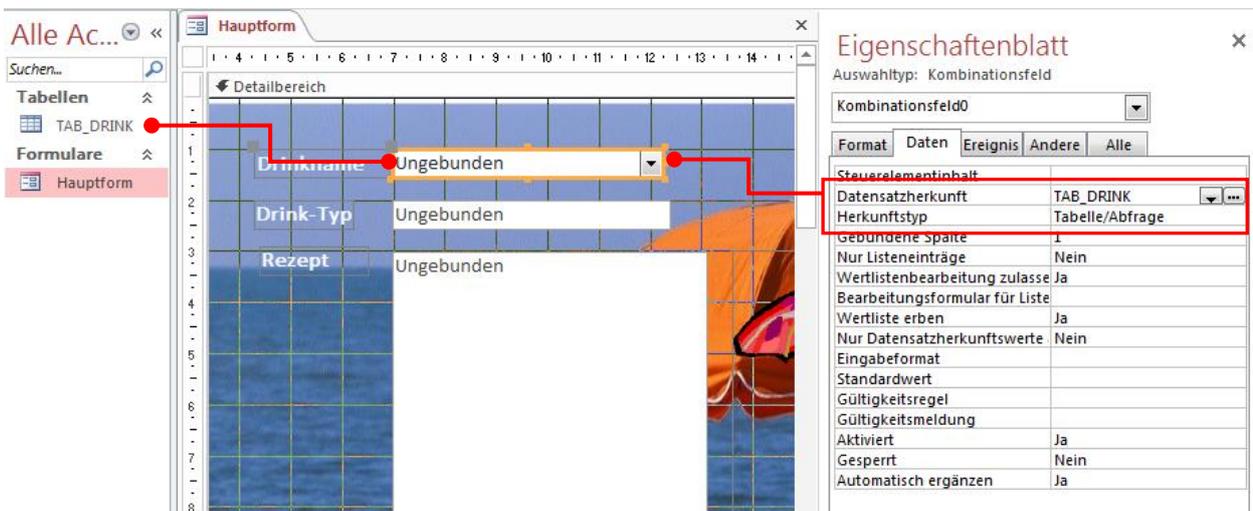
Abbildung: Tabelle in Entwurfsansicht

Um konkrete Werte einzugeben, schalten Sie um in die Datenblattansicht!

DRINK_NAME	DRINK_TYP	Rezept	Bildpfad
Espresso Machiato	Cafe-Getränk	cafemachiato	\cocktailpics\expresso machiato.bmp
Kindergarten	ohne Alkohol	Viel Milch	\cocktailpics\kindergarten.bmp
Memory shooter	mit Alkohol	Hochprozentiger Schnaps	\cocktailpics\memory shooter.bmp
womanizer	mit Alkohol	süsse Worte,	\cocktailpics\womanizer.bmp
*			

- Auch hier die relative Pfadangabe, ausgehend vom aktuellen Projektpfad – beginnend mit einem Backslash ‚\‘.

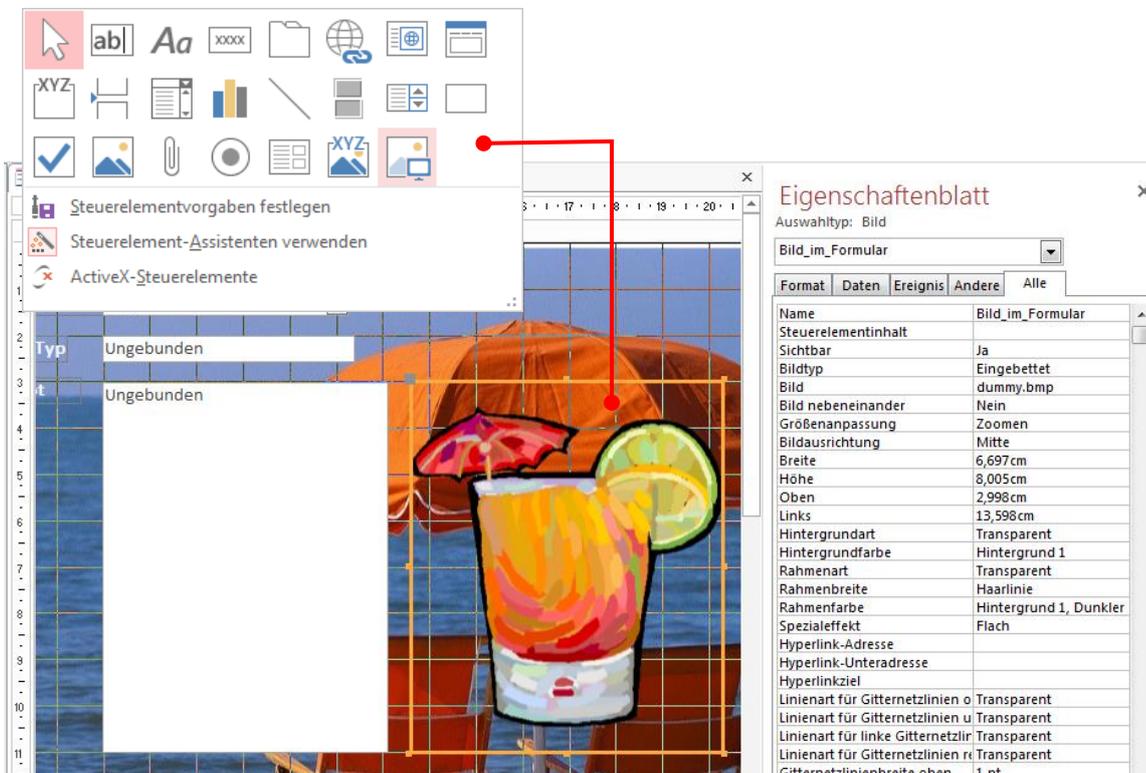
Schritt 3: Entwurf Formular



Bei Anwahl eines neuen Drinks soll sofort die zugehörige Info für den Benutzer im Formular erscheinen. Also startet die Ereignisprozedur *nach der Aktualisierung*.



Wenn nun noch ein Foto des Cocktails erscheint, wäre das Formular perfekt:



Um das VBA-Programm übersichtlich zu gestalten, ändern Sie den Namen des Objektes ‚Bild‘ um z.B. in ‚Bild_im_Formular‘. Bilder sollen eingebettet werden. Beim Starten haben Sie noch kein Cocktail gewählt, daher soll zunächst ein ‚Dummy‘-Bild erscheinen. Auch dies können Sie im Eigenschaftsfenster vorgeben.

Der zugehörige VBA-Code:

```
Private Sub Befehl31_Click()
    FollowHyperlink CurrentProject.Path & "☞menü.doc"
End Sub

Private Sub button_end_Click()
    DoCmd.CloseDatabase
End Sub

Private Sub Kombinationsfeld0_AfterUpdate()
    On Error GoTo fehler
    Dim db As DAO.Database 'Bibliothek DAO (Data Access Objects) liefert dem VBA-Entwickler
alles,
    Dim rs As DAO.Recordset 'was er für den Zugriff auf die in einer Datenbank enthalte
    Dim Krit As String
    Dim Picpfad As String

    Set db = CurrentDb() 'Verweis auf aktuell geöffnete Datenbank
    Set rs = db.OpenRecordset("Tab_Drink", DAO.dbOpenSnapshot)
    Krit = "Drink_Name=" & Kombinationsfeld0 & ""
    rs.FindFirst Krit

    If rs.NoMatch Then
        MsgBox "Der von Ihnen eingegebene Name wurde in der Datenbank nicht gefunden."
    Else
        Textfeld_Drinktyp = rs!Drink_Typ
        Textfeld_Kurzinfo = rs!Rezept
        'Mögl.1: Bildname wird zusammengesetzt aus Inhalt des Kombinationsfeld und "bmp"
        Picpfad = CurrentProject.Path & "☞cocktailpics☞" & Me!Kombinationsfeld0.Column(0) &
".bmp"
        Me!Bild_im_Formular.Picture = Picpfad
        'Mögl.2: Bildpfad und Bildname wird aus Tabelle gelesen
        Me!Bild_im_Formular.Picture = CurrentProject.Path & rs!Bildpfad
    End If
ende:
    Exit Sub
fehler:
    MsgBox Err.Description
    Resume ende
End Sub
```

8. Lösungen zu den einzelnen Kapiteln

Lösungen Kapitel 2:

Aufgabe 1:

Trainer - Mannschaft



Eine Mannschaft kann mehrere Trainer haben.

Firma – Mitarbeiter



Eine Firma hat mehrere Mitarbeiter.

Schiedsrichter – Spiel



Mehrere Schiedsrichter sind für ein Spiel abgestellt.

Raum – Veranstaltungen



In einem Raum ist immer nur eine Veranstaltung.

Mannschaft - Spieler



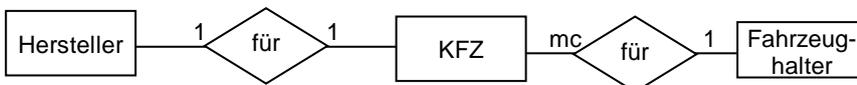
Eine Mannschaft besteht aus mehreren Spieler.

Finger – Hand



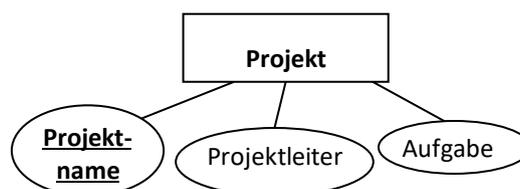
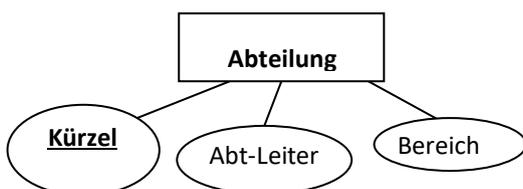
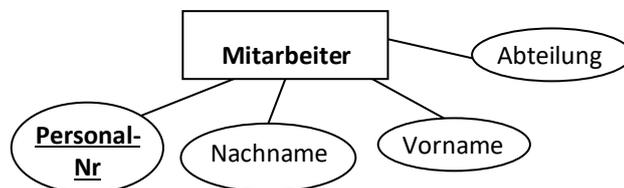
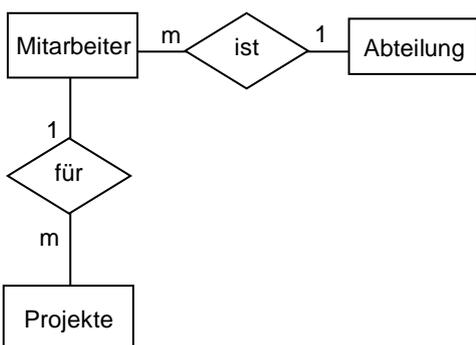
Mehrere Finger besitzt die Hand.

Aufgabe 2:



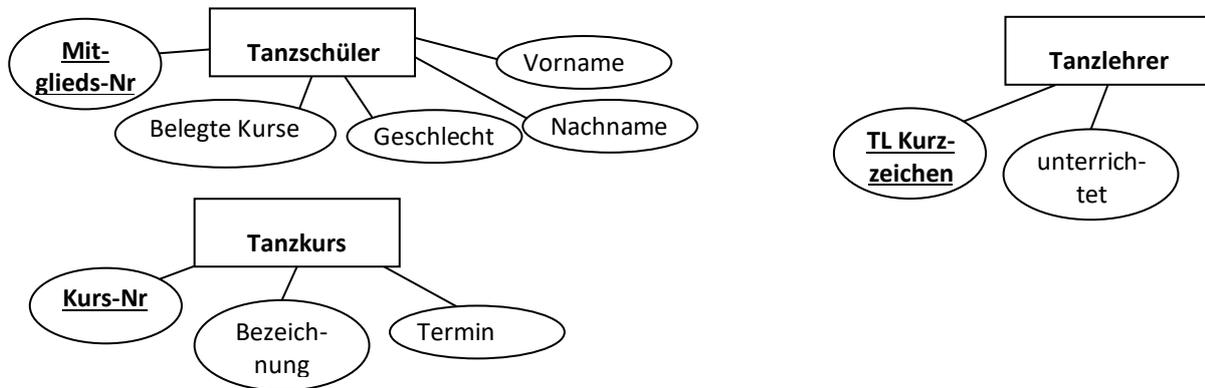
Ein Fahrzeughalter kann mehrere Autos führen ein KFZ hat nur einen Hersteller.

Aufgabe 3:

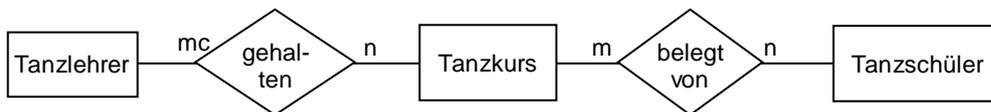


Aufgabe 4:

4.1: Bestimmung der Entitäten und deren mögliche Attribute:

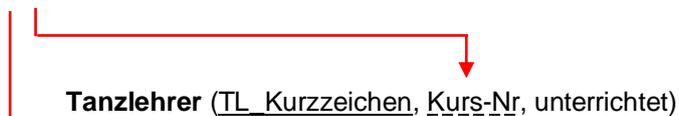


4.2 ER-Modell der Tanzschule



4.3 Relationenschreibweise

Tanzkurs (Kurs-Nr, TL-Kurzzeichen, Mitglieds-Nr, Bezeichnung, Termin)



Tanzschüler (Mitglied-Nr, Kurs-Nr, belegte-Kurse, Geschlecht, Nachname, Vorname)

Legende: Primärschlüssel Fremdschlüssel

Lösungen Kapitel 3:

Aufgabe 1:

Die Daten müssen vorher schon in Tabellen abgespeichert sein. Zudem müssen diese Tabellen miteinander verknüpft sein.

Aufgabe 2:

Formulare sind vor allem für Endanwender gedacht, da diese bequem über diese Bedienschnittstelle auf die gewünschten Daten zugreifen können.

Aufgabe 2:

Formulare sind vor allem für Endanwender gedacht, da diese bequem über diese Bedienschnittstelle auf

Aufgabe 3:

Unterformulare verwenden Sie bei 1:n – Beziehungen.

Aufgabe 4:

Ein Unterformular kann nur im Hauptformular eingefügt werden, wenn diese in der Entwurfsansicht vorliegt.

9. Stichwortverzeichnis:

A	K
Abfrage 30	Kardinalität 15
Abfrageentwurf 30	Konsistenz 5
ACCDE-Format 54	Konstanten 43
Access Executive Only Database 54	Konzeptionelle Ebene 6
Anomalien 21	Kreuztabellenabfrage 31
ANSI 6	- Assistent 31
Attribut 9	
	L
B	Logikschicht 7
Benutzerschicht 6, 7	Logische Schicht 6
Berichte 45	
Beziehungstabelle 17	M
Bilderverwaltung 66	Makro 38
	Mehrbenutzerfähigkeit 5
C	Mehrfachspeicherung 5
Cloud 55	
Const 44	N
	Normalform 19
D	
Datenaustausch zu Excel 48	O
Datenbank	Object Linking and Embedding 66
Kennwort 53	OLE 66
komprimieren 51	OLEGebunden 68
Datenbankmanagementsystem 5	On Error Resume Next 69
Datensatz 9	
DBMS 5	P
Dim 44	physikalische Schicht 7
Drei-Schichten-Architektur 6	Physikalische Schicht 6
	Primärschlüssel 8
E	
Entity-Relationship-Modell 12	R
ER-Modell 12	Redundanz 5
Exklusiv-Modus 53	referentielle Integrität 18
Externe Ebene 6	Referenz 18
	Relation 8
F	relationale Datenban 8
Fehlersteuerung 44	Relationenschreibweise 12
Formulare 34	
Unterformular 35	S
Fremdschlüssel 12	SkyDrive 55
	Steuerelement 38
G	
grafic user interface 7	T
GUI 7	Tabellen 26
	Tupel 9
I	
Integrität 5, 18	
Interne Ebene 6	

V

Variablen 43

VBA 39
Verschlüsselung 53
Visual Basic 39

