

# Übungen zu Wirtschaftsinformatik 2

## LE 06 – Relationales Modell (Teil 3)

### SQL

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre>

# Haftungsausschluss

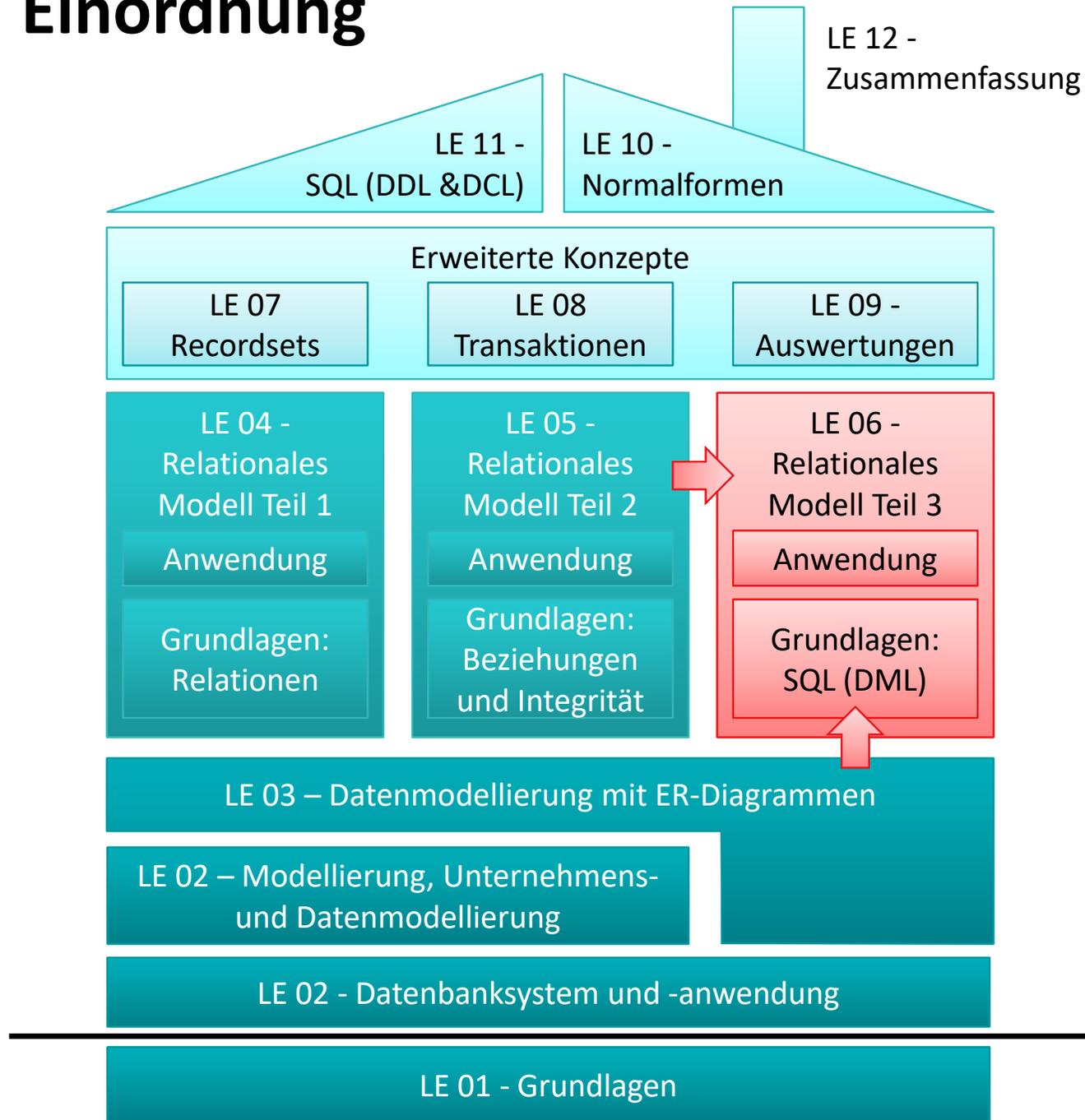
**Alle nachfolgenden Ausführungen und Inhalte sind gewissenhaft erarbeitet worden.**

**Dennoch kann weder die inhaltliche Richtigkeit, noch die Übereinstimmung mit den tatsächlichen Bedürfnissen des Nutzers garantiert werden. Der Nutzer verpflichtet sich, sie vor ihrer Verwendung inhaltlich auf Richtigkeit und auf Brauchbarkeit für den konkreten Einsatzzweck zu prüfen.**

**Es kann keine Gewährleistung für die rechtliche oder technische Wirksamkeit der Ausführungen und Inhalte übernommen werden - insbesondere können hieraus auch keine haftungsrelevanten Ansprüche hergeleitet werden.**

- Grundbestandteile der Datenbanksprache SQL kennenlernen
- SQL-Anweisungen der Data Manipulation Language (DML) nutzen
  - Abfrage von Daten mit SQL zur Umsetzung von Operationen der Relationalen Algebra
  - Hinzufügen, Löschen und Ändern von Daten mit SQL
- Praktische Anwendung in MS Access üben
  - Verwendung der verschiedenen SQL-Anweisungen in verschiedenen Abfragearten von MS Access
  - Integration von SQL-Anweisungen in eigene Funktionen, Prozeduren und Formulare

# Einordnung



# Inhalt

## Lernziel, Lehrinhalte und Einordnung

### SQL-Anweisungen zur Auswahl und Manipulation

- Einfache Anweisungen
- Auswählen von Produkten einer Produktkategorie
- Einfügen von Produkten zum Warenkorb
- Ändern der Anzahl eines Produktes im Warenkorb
- Löschen eines Produktes aus dem Warenkorb
- Verbinden (Join) von Benutzern und Kunden

### Ausblick

# Inhalt

Lernziel, Lehrinhalte und Einordnung

## **SQL-Anweisungen zur Auswahl und Manipulation**

- Einfache Anweisungen
- Auswählen von Produkten einer Produktkategorie
- Einfügen von Produkten zum Warenkorb
- Ändern der Anzahl eines Produktes im Warenkorb
- Löschen eines Produktes aus dem Warenkorb
- Verbinden (Join) von Benutzern und Kunden

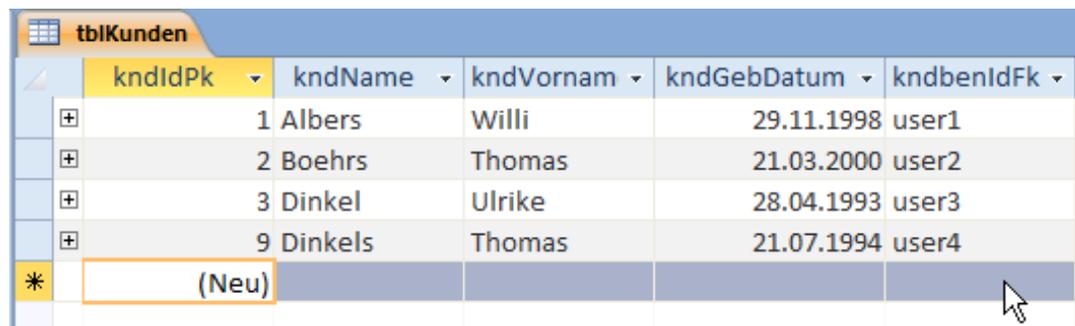
## **Ausblick**

# Auswahl und Manipulation mit SQL

# Auswahl und Manipulation mit SQL: Übung 6.1 ~~BHT~~

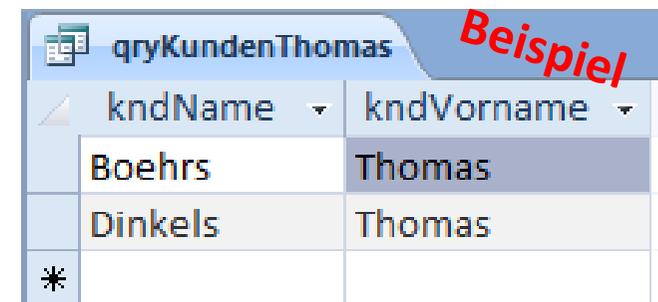
## Ü6.1 (Teil 1)

- Sofern noch nicht vorhanden, erfassen Sie in ihrer Kundentabelle die folgenden Testdaten für Kunden. Sollten dazu weitere Datensätze in der Tabelle `tblBenutzer` benötigt werden, legen Sie diese zuvor an.



kndIdPk	kndName	kndVorname	kndGebDatum	kndbenIdFk
1	Albers	Willi	29.11.1998	user1
2	Boehrs	Thomas	21.03.2000	user2
3	Dinkel	Ulrike	28.04.1993	user3
9	Dinkels	Thomas	21.07.1994	user4
*	(Neu)			

- Ermitteln Sie alle Kunden, deren Vorname "Thomas" ist. Gehen Sie wie folgt vor
  - Erstellen Sie einen neuen Abfrageentwurf,
  - schließen Sie den Dialog zum Hinzufügen von Tabellen und
  - wechseln Sie gleich in die SQL-Ansicht.
  - Erfassen Sie eine SQL-Abfrage die alle Kunden ermittelt, deren Vorname "Thomas" ist.
  - Führen Sie die Abfrage aus



**Beispiel**

kndName	kndVorname
Boehrs	Thomas
Dinkels	Thomas
*	

# Auswahl und Manipulation mit SQL: Übung 6.1 ~~BHT~~

## Ü6.1 (Fortsetzung Teil 1)

- Erstellen Sie für jede der folgenden Abfragen einen neuen Abfrageentwurf und wechseln Sie in die SQL-Ansicht
  - Ermitteln Sie alle Kunden, die vor dem 01.01.2000 geboren sind
  - Ermitteln Sie alle Kunden, deren Vorname "Thomas" ist und die vor 01.01.2000 geboren wurden
  - Ermitteln Sie alle Kunden, deren Vorname "Thomas" ist oder die vor dem 01.01.2000 geboren wurden
- Warum liefert die Verknüpfung mit dem Und-Operator weniger Ergebnisse als die Verknüpfung mit dem Oder-Operator?

	kndIdPk	kndName	kndVorname	kndGebDatum	kndbenIdFk
+	1	Albers	Willi	29.11.1998	user1
+	2	Boehrs	Thomas	21.03.2000	user2
+	3	Dinkel	Ulrike	28.04.1993	user3
+	9	Dinkels	Thomas	21.07.1994	user4
*		(Neu)			

# Auswahl und Manipulation mit SQL: Übung 6.1

## Ü6.1 (Teil 2)

- Erstellen Sie für die folgenden Abfragen einen neuen Abfrageentwurf und wechseln Sie in die SQL-Ansicht.
  - Ermitteln Sie alle Produkte mit Bezeichnung und Preis sowie der Bezeichnung ihrer Produktkategorie. Produkte ohne Kategorie und Kategorien ohne Produkte sollen nicht im Ergebnis enthalten sein
  - Passen Sie ihre vorherige Lösung an: Produkte ohne Kategorie sollen im Ergebnis enthalten sein.
  - Passen Sie ihre vorherige Lösung noch einmal an: Produktkategorien ohne Produkte sollen im Ergebnis enthalten sein.

# Auswahl und Manipulation mit SQL: Übung 6.1 ~~BHT~~

## Ü6.1 (Teil 3)

- ...
- Sie wollen mit einer SQL-Anweisung Boehrs, Vera (geb. 03.12.1997) in die Tabelle der Kunden einfügen

	kndIdPk	kndName	kndVorname	kndGebDatum	kndbenBenutzernameFk
+	1	Albers	Willi	29.11.1998	user1
+	2	Boehrs	Thomas	21.03.2000	user2
+	4	Dinkels	Thomas	01.05.2001	user4
→+	9	Boehrs	Vera	03.12.1997	user10
*		(Neu)			

- Warum müssen Sie zuerst einen neuen Benutzer in der Tabelle **tblBenutzer** für Vera anlegen?
- Erstellen Sie in der SQL-Ansicht eines neuen Abfrageentwurfs/oder einer "Einfügeabfrage" eine SQL-Anweisung die einen weiteren Benutzer "user10" mit Passwort "test10" in die Tabelle Benutzer einfügt
- Erstellen Sie in der SQL-Ansicht eines neuen Abfrageentwurfs/oder einer "Einfügeabfrage" eine SQL-Anweisung die Boehrs, Vera mit einer Beziehung zu user10 in die Tabelle Kunden einfügt

– ...

# Auswahl und Manipulation mit SQL: Übung 6.1 ~~BHT~~

## Ü6.1 (Fortsetzung Teil 3)

- ...
- Fügen Sie ein neues Produkt "Weihnachtsbauständer" hinzu mit folgenden Angaben
  - der Beschreibung "30cm Durchmesser"
  - zum Preis von 12 EUR
  - ohne Foto
  - mit einem Lagerbestand von 10 Stück
- Der Weihnachtsbaumständer soll der neuen Produktkategorie "Saisonartikel" zugeordnet sein.

# Auswahl und Manipulation mit SQL: Übung 6.1 ~~BHT~~

## Ü6.1 (Teil 4)

- ...
- Erstellen Sie in der SQL-Ansicht eines neuen Abfrageentwurfs/oder in der SQL-Ansicht einer "Aktualisierungsabfrage" eine SQL-Anweisung die bei allen Kunden mit dem Namen "Boehrs" den Namen in "Böhrs" ändert.

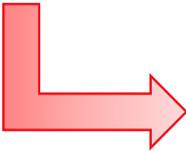
– ...

*Beispiel: Vorher*

kndIdPk	kndName	kndVorname	kndGebDatum	kndbenBenutzernameFk
1	Albers	Willi	29.11.1998	user1
2	Boehrs	Thomas	21.03.2000	user2
4	Dinkels	Thomas		
9	Boehrs	Vera		
*	(Neu)			

*Beispiel: Nachher*

kndIdPk	kndName	kndVorname	kndGebDatum	kndbenBenutzernameFk
1	Albers	Willi	29.11.1998	user1
2	Böhrs	Thomas	21.03.2000	user2
4	Dinkels	Thomas	01.05.2001	user4
9	Böhrs	Vera	03.12.1997	user10
*	(Neu)			



# Auswahl und Manipulation mit SQL: Übung 6.1 ~~BHT~~

## Ü6.1 (Fortsetzung Teil 4)

- ...
- Erhöhen Sie den Preis aller Produkte in der Kategorie "Dünger" um 2 EUR!

# Auswahl und Manipulation mit SQL: Übung 6.1 ~~BHT~~

## Ü6.1 (Teil 5)

– ...

– Sie wollen mit einer SQL-Anweisung den Benutzer "user10" löschen.

- Warum müssen Sie zuerst Vera Böhrs aus der Tabelle Kunden löschen?
- Erstellen Sie in der SQL-Ansicht eines neuen Abfrageentwurfs/ oder einer "Löschabfrage" eine SQL-Anweisung, die Vera Böhrs aus der Tabelle Kunden löscht.
- Erstellen Sie in der SQL-Ansicht eines neuen Abfrageentwurfs/oder einer "Löschabfrage" eine SQL-Anweisung, die user10 aus der Tabelle löscht.

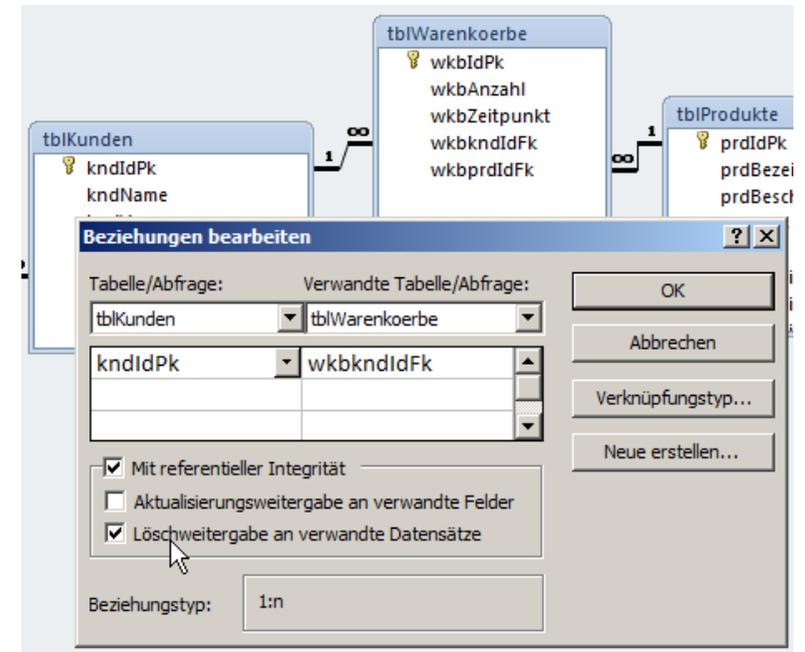
– ...

tblBenutzer			
benBenutzernamePk	benPasswor	benIstAdmi	Zum Hinz
admin1	test1	<input checked="" type="checkbox"/>	
admin2	test2	<input checked="" type="checkbox"/>	
user1	test1	<input type="checkbox"/>	
user10	test10	<input type="checkbox"/>	
tblKunden			
kndIdPk	kndName	kndVorname	kndGebDatum
9	Boehrs	Vera	03.12.1997
* (Neu)			
user2	test2	<input type="checkbox"/>	
user3	test3	<input type="checkbox"/>	
user4	test4	<input type="checkbox"/>	

# Auswahl und Manipulation mit SQL: Übung 6.1 BHT

## Ü6.1 (Teil 5)

- ...
- Aktivieren Sie an der Beziehung von Kunde zu Warenkorb die Löschweitergabe (Beziehungsansicht > Dialog "Beziehung bearbeiten" > "Löschweitergabe an andere Felder" aktivieren)
- Erstellen Sie in der SQL-Ansicht eines neuen Abfrageentwurfs/oder einer "Löschabfrage" eine SQL-Anweisung, die einen Kunden aus der Tabelle Kunden löscht, zu dem es einen Warenkorbeintrag gibt.
- Was stellen Sie fest?



**Wichtig!** Löschen Sie nicht den Kunden, dessen Primärschlüssel von der Funktion gibAktuellerKundeId() zurückgegeben wird.

```
'Ü1.10
' Öffentliche Funktion zum Setzen der Kundennummer
Public Function gibAktuellerKundeId() As Long
' Vorübergehend mit festem Wert
' Let gibAktuellerKundeId = lngKundeNr
Let gibAktuellerKundeId = 2
End Function
```

# Auswahl und Manipulation mit SQL

# Auswahl von Tupeln mit SQL

# Auswahl von Tupeln mit SQL: Übung 6.2

## Ü6.2 (Teil 1)

– Erweitern Sie das Formular ihres Produktkatalogs um eine Aufklappliste im Formularkopf, die alle Produktkategorien zeigt



- Folgen Sie dem Assistenten, der sich bei Aufklapplisten öffnet
- Wählen Sie als Datenquelle die Tabelle tblProduktKategorien
- Als Ergebniswert soll die Aufklappliste den Primärschlüssel aus der Tabelle liefern
- Es soll nur die Bezeichnung der Produktkategorie sichtbar sein

– ...

## Ü6.2 (Teil 2)

- Überlegen Sie, wie Sie den Wert des in der Aufklappliste ausgewählten Eintrags erhalten. (Hinweis: Dieser Wert entspricht dem Primärschlüssel der ausgewählten Produktkategorie.)
- Überlegen Sie sich eine SQL-Abfrage, die nur solche Produkte auswählt, die einer bestimmten Kategorie zugeordnet sind.
- Wie kann in VBA der Wert aus der Aufklappliste in der SQL-Abfrage berücksichtigt werden? Notieren Sie Ihre Abfrage.
- ...

## Ü6.2 – Lösungsvorschlag

- Ansatz: SQL-Anweisung als String speichern und mit Werten aus Variablen zusammenbauen.
- Beispiel:

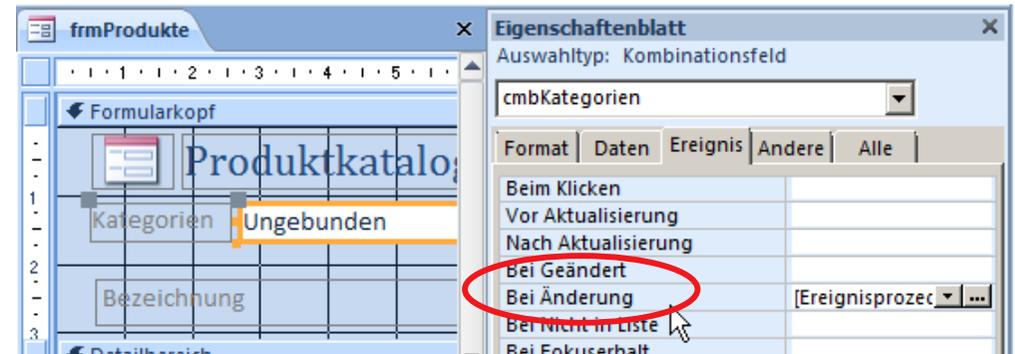
```
'...  
  
Dim strSql As String ' Variable für SQL-Abfrage  
Dim intKat As Integer ' Variable für ausgewählte Kategorie  
  
' Ermitteln des aktuellen Wertes der Aufklappliste,  
' liefert den Primärschlüssel der ausgewählten Kategorie  
Let intKat = Me.cmbKategorie.Value  
  
' SQL-Abfrage zur Auswahl der Produkte einer Kategorie mit dem Wert  
' aus der Aufklappliste verbinden  
Let strSql = "SELECT * FROM tblProdukte WHERE prdkatIdFk=" & intKat & ";"  
  
'...'
```

# Auswahl von Tupeln mit SQL: Übung 6.2

## Ü6.2 (Teil 3)

– ...

– Implementieren Sie eine Ereignisprozedur, die bei der Änderung der Auswahl in dieser Liste aufgerufen wird und



- aus der Aufklappliste den dort aktuell ausgewählten Wert ermittelt und in einer Variablen speichert,
- diese Variable dann in Ihre SQL-Anweisung (von oben) eingebaut und die gesamte Anweisung als String gespeichert
- die SQL-Anweisung im Direktbereich ausgibt
- dem Formular die SQL-Anweisung dann als Datenquelle zuweist (Nutzen Sie dafür die Eigenschaft Me.RecordSource)
- die Daten des Formulars neu lädt (mit Me.Requery).

– Hinweis: Wenn etwas nicht funktioniert vergleichen Sie, ob die Ausgabe im Direktbereich mit Ihrer beabsichtigten SQL-Abfrage übereinstimmt.

# Auswahl von Tupeln mit SQL

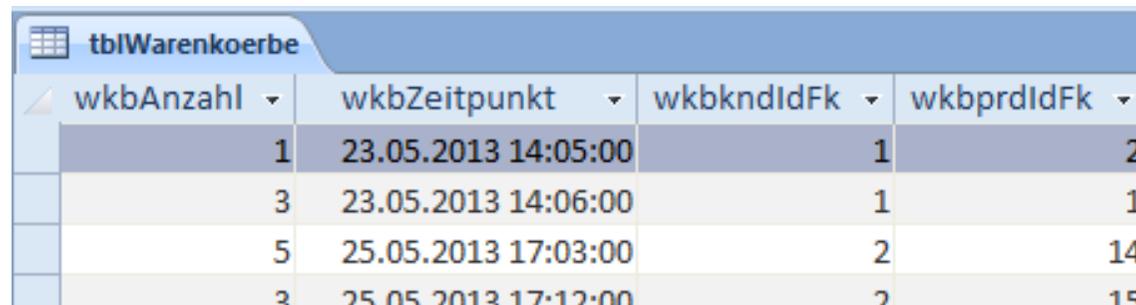
# Einfügen von Tupeln mit SQL

# Einfügen von Tupeln mit SQL: Übung 6.3

## Ü6.3 (Teil 1)

- Überlegen Sie sich eine SQL-Anweisung, mit der der aktuelle Artikel in den Warenkorb gelegt werden kann, d.h. in der Tabelle **tblWarenkoerbe** eine neue Zeile hinzugefügt wird
  - mit der Anzahl 1
  - mit dem aktuellen Zeitpunkt (Rückgabewert der VBA-Funktion **Now( )**)
  - mit einem Fremdschlüssel für die Beziehung zum Produkt (aktueller Artikel, dessen Primärschlüssel in einem versteckten Feld auf der Oberfläche enthalten ist)
  - und einem Fremdschlüssel für die Beziehung zum Kunden (Rückgabewert der Funktion **gibAktuellerKundeId( )**)

– ...



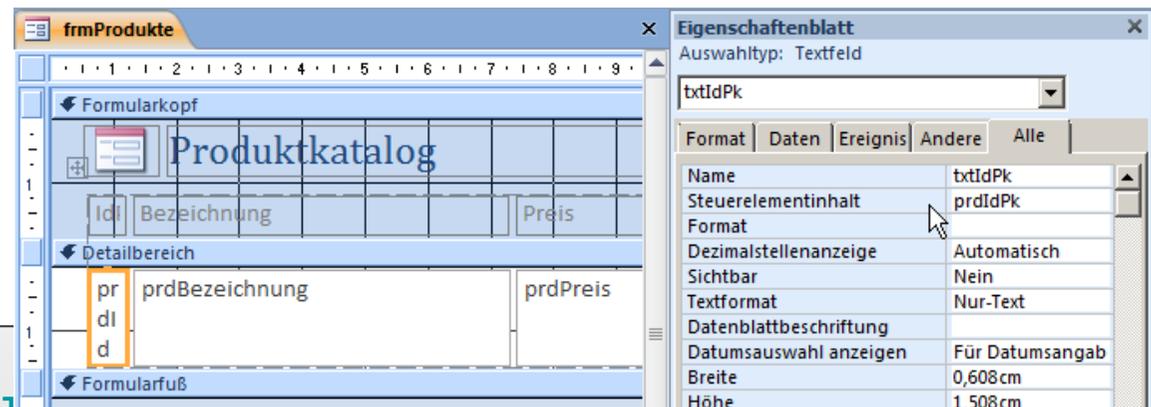
wkbAnzahl	wkbZeitpunkt	wkbkndIdFk	wkbprIdFk
1	23.05.2013 14:05:00	1	2
3	23.05.2013 14:06:00	1	1
5	25.05.2013 17:03:00	2	14
3	25.05.2013 17:12:00	2	15

# Einfügen von Tupeln mit SQL: Übung 6.3

## Ü6.3 – Lösungsvorschlag

– Hinweis: Im Formular **frmProdukte** existiert ein verstecktes Feld, das den Primärschlüsselwert des Produktes enthält.

– Beispiel:



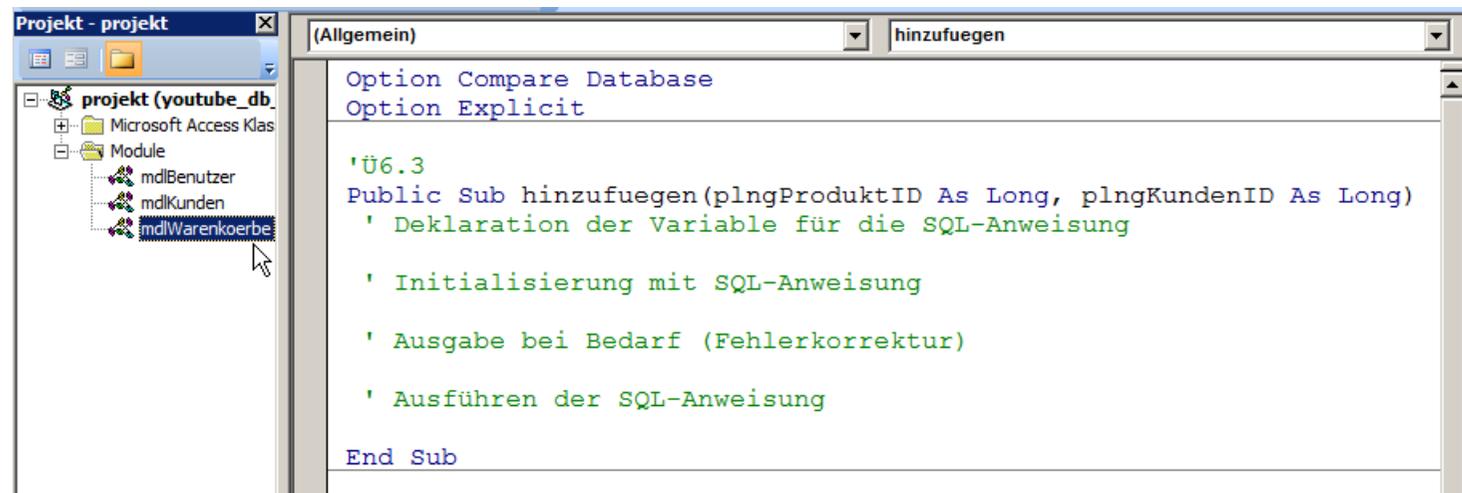
```
' ...  
Dim strSql As String ' Variable für SQL-Anfrage  
Dim lngPrd As Long ' Variable für Primärschlüsselwert des akt. Produktes  
  
' Ermitteln des Primärschlüssels des aktuellen Produktes  
Let lngPrd = Me.txtIdPk.Value  
  
' SQL-Anweisung zum Einfügen in Warenkorb  
Let strSql = "INSERT INTO tblWarenkoerbe " & _  
            "(wkbAnzahl, wkbZeitpunkt, wkbprdIdFk, wkbkndIdFk) VALUES " & _  
            "(1, '" & Now() & "', " & lngPrd & ", 2 );"  
  
' ...
```

# Einfügen von Tupeln mit SQL: Übung 6.3

## Ü6.3 (Teil 2)

- ...
- Legen Sie das Modul **mdlWarenkoerbe** und darin eine öffentliche Prozedur **hinzufuegen** an, die
  - als Parameter den Primärschlüsselwert eines Produktes und eines Kunden bekommt (ProudktID, KundenID)
  - diese Parameterwerte in die SQL-Anweisung (von oben) einbaut
  - die SQL-Anweisung ausführt

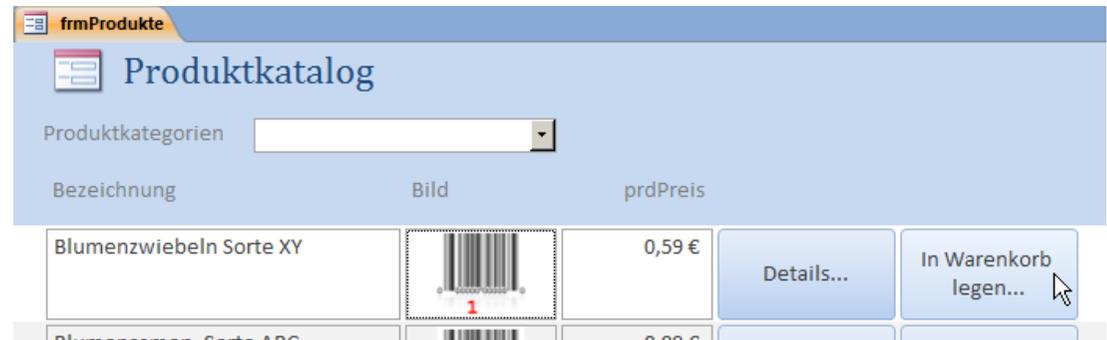
– ...



# Einfügen von Tupeln mit SQL: Übung 6.3

## Ü6.3 (Teil 3)

- ...
- Erweitern Sie ihren Produktkatalog im Detailbereich um eine Schaltfläche "In Warenkorb legen"
- Legen Sie eine Ereignisprozedur für das Klicken auf diese Schaltfläche an, so dass
  - aus dem versteckten Feld mit dem Primärschlüssel des Produktes der Wert gelesen und in einer Variable gespeichert wird
  - in einer zweiten Variable den Rückgabewert der Funktion `mdlKunden.gibAktuellerKundeID( )`
  - und dann die Prozedur `mdlWarenkoerbe.hinzufuegen` mit diesen Variablen als Parameter aufgerufen wird
  - und anschließend
    - ein eventuell noch geöffnetes Formular des Warenkorbs geschlossen wird, weil es noch "alte" Daten zeigt
    - und das Formular des Warenkorbs erneut geöffnet wird, damit es die aktuellen Daten zeigt



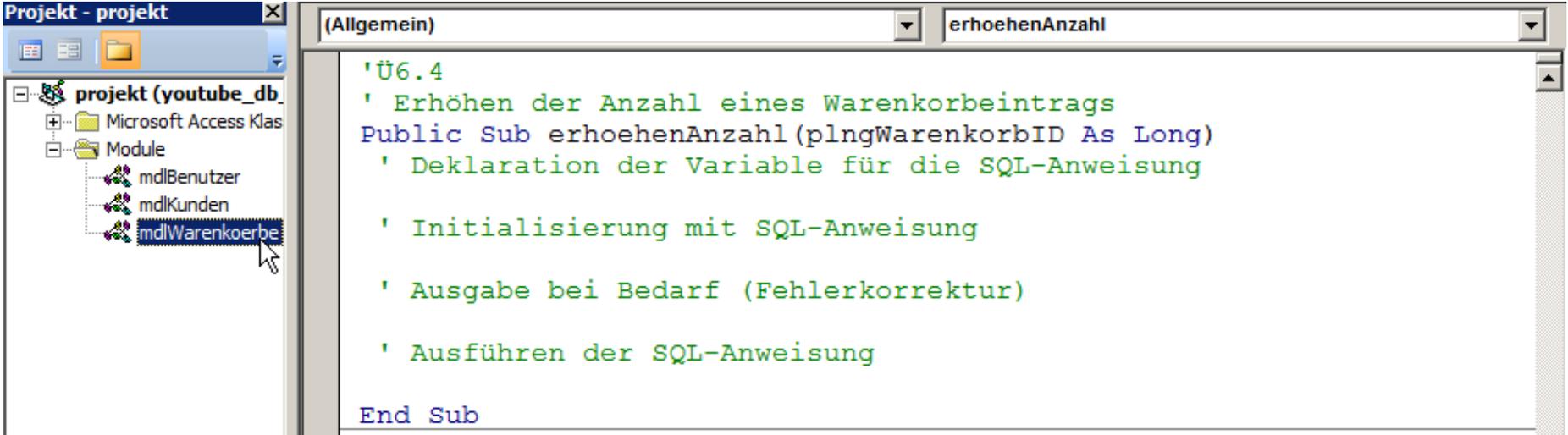
# Einfügen von Tupeln mit SQL

# Ändern von Tupeln mit SQL

# Ändern von Tupeln mit SQL: Übung 6.4

## Ü6.4 (Teil 1)

- Überlegen Sie sich eine SQL-Anweisung, mit der die Menge des aktuellen Artikels um eins erhöht werden kann.
- Implementieren Sie im Modul **mdlWarenkoerbe** eine Prozedur **erhoehenAnzahl**,
  - die als Parameter einen Primärschlüsselwert des Warenkorbeintrags übergeben bekommt
  - und diesen Parameter in ihre SQL-Anweisung einbaut und ausführt



```
(Allgemein) | erhoehenAnzahl
'Ü6.4
' Erhöhen der Anzahl eines Warenkorbeintrags
Public Sub erhoehenAnzahl(plngWarenkorbID As Long)
' Deklaration der Variable für die SQL-Anweisung

' Initialisierung mit SQL-Anweisung

' Ausgabe bei Bedarf (Fehlerkorrektur)

' Ausführen der SQL-Anweisung

End Sub
```

# Ändern von Tupeln mit SQL: Übung 6.4

## Ü6.4 (Teil 2)

- ...
- Erweitern Sie das Formular des Warenkorbs im Detailbereich um eine Schaltfläche zum Erhöhen der Menge eines Produktes im Warenkorb
- Implementieren Sie die Ereignisprozedur so, dass beim Klick auf die Schaltfläche
  - aus dem versteckten Feld auf der Oberfläche der Primärschlüssel des Warenkorbeintrags ermittelt und
  - anschließend die Prozedur `mdlWarenkoerbe.erhoeheAnzahl` mit der ID als Parameter aufgerufen wird
  - und das Formular aktualisiert wird.



The screenshot shows a web application window titled 'frmWarenkorb' with a sub-header 'Ihr Warenkorb'. Below the header is a button '<< Weiter einkaufen'. The main content is a table with columns: 'Bezeichnung', 'Einzelpreis', 'Anzahl', and 'Gesamt'. The table contains two rows of items. The first row is 'Blumenzwiebeln Sorte XY' with a price of 0,80 € and a quantity of 8, resulting in a total of 6,40 €. The second row is 'Blumendünger "Sabine"' with a price of 0,99 € and a quantity of 4, resulting in a total of 3,96 €. In the 'Anzahl' column of the first row, there is a small blue box containing a '+' sign, which is circled in red. The second row also has a similar '+' sign in its 'Anzahl' column.

Bezeichnung	Einzelpreis	Anzahl	Gesamt
Blumenzwiebeln Sorte XY	0,80 €	8	6,40 €
Blumendünger "Sabine"	0,99 €	4	3,96 €

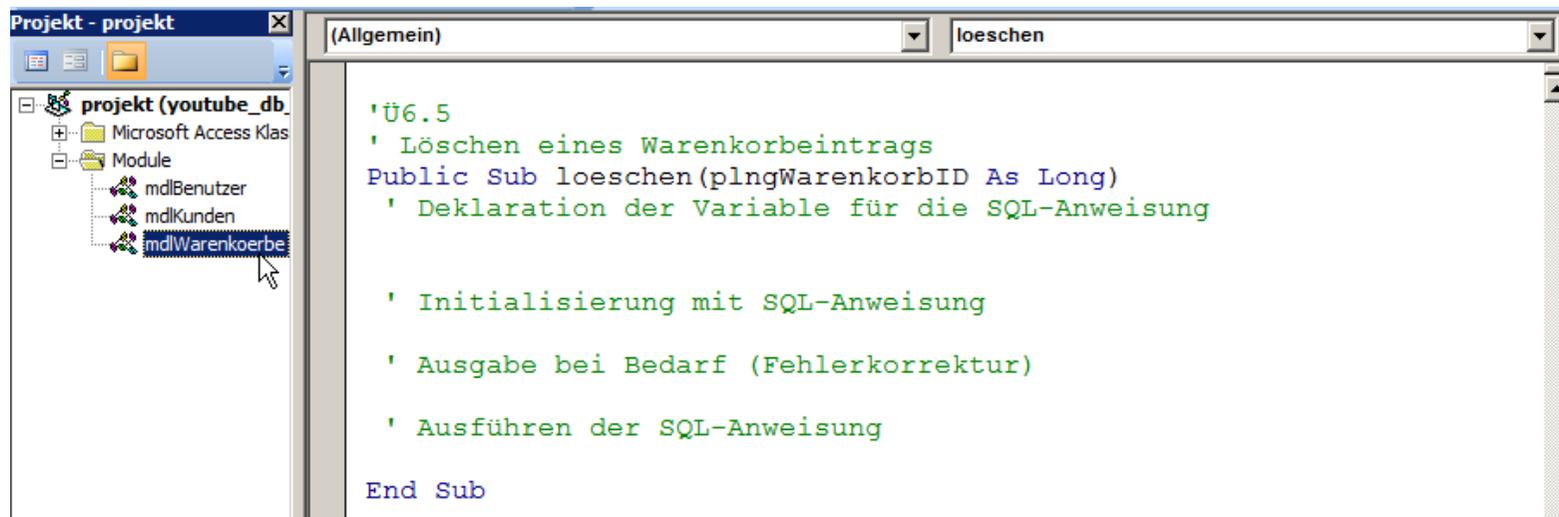
# Ändern von Tupeln mit SQL

# Löschen von Tupeln mit SQL

# Löschen von Tupeln mit SQL: Übung 6.5

## Ü6.5 (Teil 1)

- Überlegen Sie sich eine SQL-Anweisung, mit der ein Eintrag aus dem Warenkorb gelöscht werden kann.
- Implementieren Sie im Modul **mdlWarenkoerbe** eine Prozedur **loeschen**,
  - die als Parameter den Primärschlüssel des zu löschenden Warenkorbeintrags übergeben bekommt
  - und diesen Parameter in ihre SQL-Anweisung einbaut und ausführt



```
(Allgemein) loeschen

'Ü6.5
' Löschen eines Warenkorbeintrags
Public Sub loeschen(plngWarenkorbID As Long)
' Deklaration der Variable für die SQL-Anweisung

' Initialisierung mit SQL-Anweisung

' Ausgabe bei Bedarf (Fehlerkorrektur)

' Ausführen der SQL-Anweisung

End Sub
```

# Löschen von Tupeln mit SQL: Übung 6.5

## Ü6.5 (Teil 2)

- Erweitern Sie das Formular des Warenkorbs im Detailbereich um eine Schaltfläche zum Löschen eines Eintrags im Warenkorb
- Implementieren Sie die Ereignisprozedur so, dass beim Klick auf die Schaltfläche (nach vorheriger Warnmeldung)
  - aus dem versteckten Feld auf der Oberfläche den Primärschlüsselwert des Warenkorbeintrags ermittelt und
  - anschließend die Prozedur `mdlWarenkoerbe.loesche` mit dem Primärschlüsselwert als Parameter aufgerufen wird



# Löschen von Tupeln mit SQL

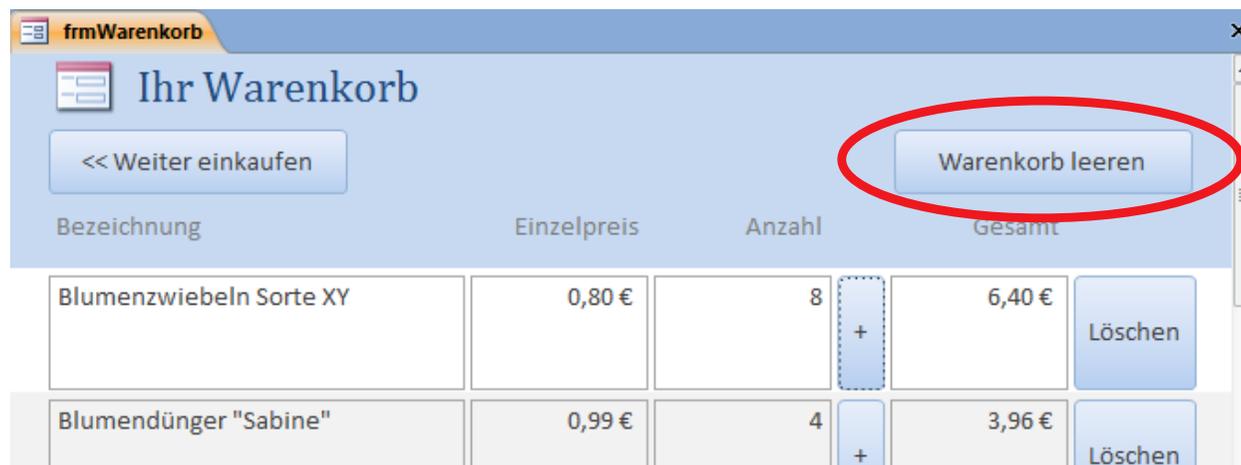
## Ü6.6 (Teil 1)

- Überlegen Sie sich eine SQL-Anweisung, mit der alle Warenkorbeinträge eines Kunden gelöscht werden können.
- Implementieren Sie im Modul `mdlWarenkoerbe` eine Prozedur `loeschenWarenkorb`,
  - die als Parameter den Primärschlüssel eines Kunden übergeben bekommt
  - Initialisieren Sie eine Variable vom Typ `String` mit einem SQL **DELETE**-Kommando, in dessen **WHERE**-Bedingung Sie den Wert des Parameters integrieren
  - Geben Sie den Wert der Variable im Direktbereich aus
  - Führen Sie die in der Variable enthaltene SQL-Anweisung auf der Datenbank aus

# Löschen von Tupeln mit SQL: Übung 6.6

## Ü6.6 (Teil 2)

- Erweitern Sie das Formular des Warenkorbs im Detailbereich um eine Schaltfläche zum Leeren des Warenkorbs
- Implementieren Sie die Ereignisprozedur so, dass beim Klick auf die Schaltfläche (nach vorheriger Warnmeldung)
  - der Rückgabewert der Funktion `gibAktuellerKundeID()` in einer Variable gespeichert wird
  - die Prozedur `mdlWarenkoerbe.loeschenWarenkorb` mit dieser Variable als Parameter aufgerufen wird
  - das Formular aktualisiert wird



# Löschen von Tupeln mit SQL

# Verbinden (Join) von Relationen mit SQL

# Verbinden (Join) von Relationen in SQL: Übung 6.7 BHT

**Ü6.7: Überlegen Sie sich eine SQL-Abfrage die Datensätze aus den Tabellen Benutzer und Kunden kombiniert, so dass**

- alle Spalten aus Tabelle Benutzer und die zugehörige KundenID (~~kndIdPk~~ aus Tabelle Kunden) ausgegeben werden und
- alle Datensätze aus Benutzer enthalten sind, selbst wenn sie keinen Join-Partner in der Tabelle Kunden haben

**Probieren Sie die SQL-Abfrage in der SQL-Ansicht einer Auswahlabfrage aus.**

**Speichern Sie diese Abfrage als:**

Wichtig, weil in weiteren Übungen damit gearbeitet wird.

**qryBenutzerUndKundenID**

# Verbinden (Join) von Relationen mit SQL

# Inhalt

Lernziel, Lehrinhalte und Einordnung

## SQL-Anweisungen zur Auswahl und Manipulation

- Einfache Anweisungen
- Auswählen von Produkten einer Produktkategorie
- Einfügen von Produkten zum Warenkorb
- Ändern der Anzahl eines Produktes im Warenkorb
- Löschen eines Produktes aus dem Warenkorb
- Verbinden (Join) von Benutzern und Kunden

## Ausblick



# Inhalt

## Lernziel, Lehrinhalte und Einordnung

## SQL-Anweisungen zur Auswahl und Manipulation

- Einfache Anweisungen
- Auswählen von Produkten einer Produktkategorie
- Einfügen von Produkten zum Warenkorb
- Ändern der Anzahl eines Produktes im Warenkorb
- Löschen eines Produktes aus dem Warenkorb
- Verbinden (Join) von Benutzern und Kunden

## Ausblick

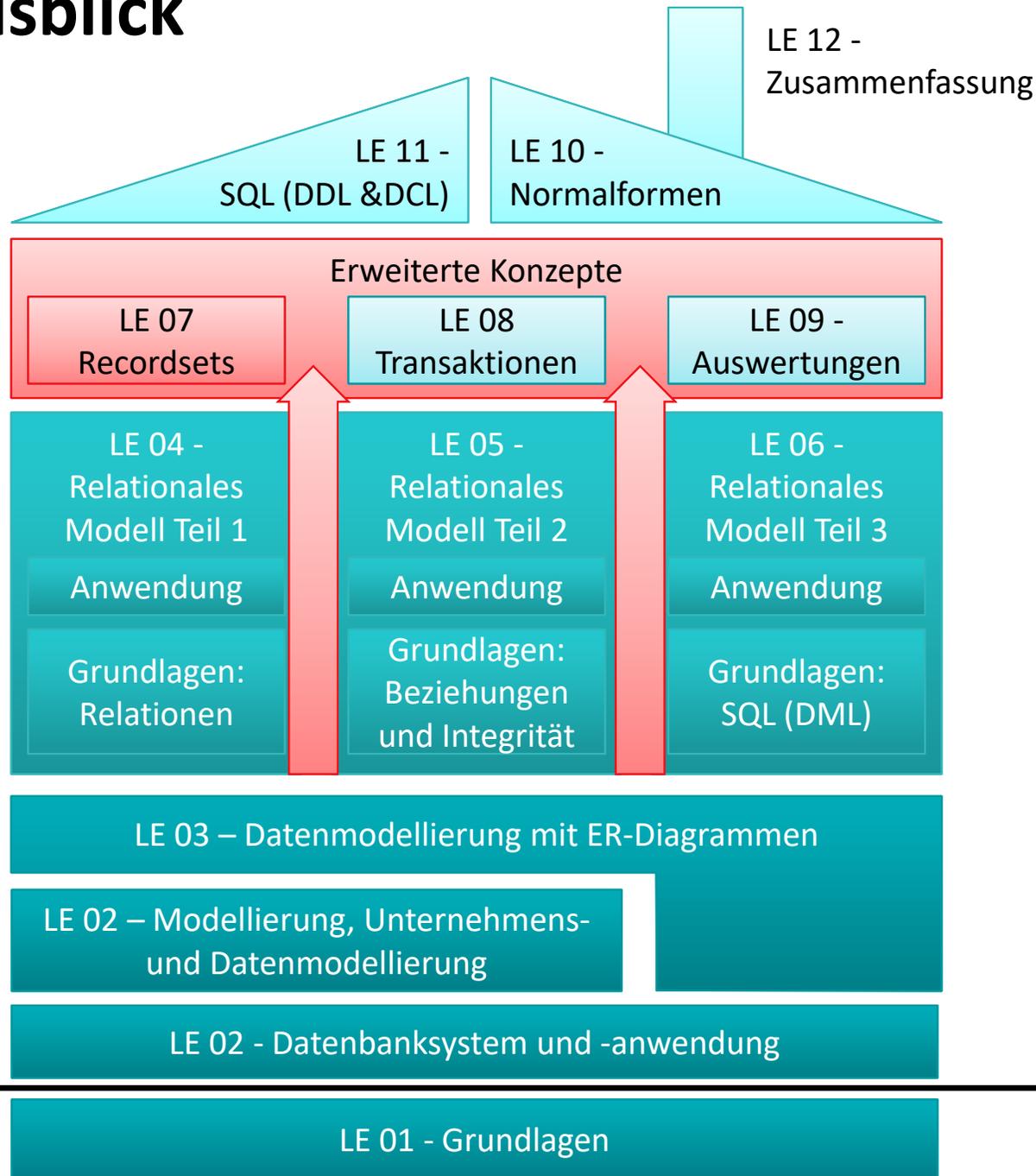
# Haftungsausschluss

**Alle nachfolgenden Ausführungen und Inhalte sind gewissenhaft erarbeitet worden.**

**Dennoch kann weder die inhaltliche Richtigkeit, noch die Übereinstimmung mit den tatsächlichen Bedürfnissen des Nutzers garantiert werden. Der Nutzer verpflichtet sich, sie vor ihrer Verwendung inhaltlich auf Richtigkeit und auf Brauchbarkeit für den konkreten Einsatzzweck zu prüfen.**

**Es kann keine Gewährleistung für die rechtliche oder technische Wirksamkeit der Ausführungen und Inhalte übernommen werden - insbesondere können hieraus auch keine haftungsrelevanten Ansprüche hergeleitet werden.**

# Ausblick



# Übungen zu Wirtschaftsinformatik 2

## LE 06 – Relationales Modell (Teil 3)

### SQL

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre>

# Codeblöcke (Syntax und Beispiele)

```
' Generelle Syntax mit Angabe des gewünschten Inhalts  
' z.B. vbDirectory, vbHidden, vbSystem  
Let <strElement> = Dir(<Pfad>, <GewünschteInhalte>)  
Let <strElement> = Dir() ' Nächstes (im vorherigen Pfad)
```

```
Dim intResult As Integer ' Rückgabewert  
Dim i As Integer ' Schleifenvariable  
Dim oFd As Object ' Variable für FileDialog  
Set oFd = Application.FileDialog(3) ' Initial. als Dateiauswahl = 3  
  
oFd.AllowMultiSelect = True ' Konfiguration, z.B. Mehrfachauswahl  
Let intResult = oFd.Show ' Dialog anzeigen und Ergebnis merken  
  
If intResult = 0 Then  
    Exit Sub ' Abbruch durch Benutzer  
End If  
  
' Schleife über alle ausgewählten Dateien  
For i = 1 To oFd.SelectedItems.Count  
    Debug.Print oFd.SelectedItems(i)  
Next
```

		META Condensed normal  Meta Condensed normal
<b>#0098A1</b> HKS 51 100% CMYK 100   0   40   0 RGB 0   152   161	70 %	HKS 13 100% CMYK 0   94   91   0 RGB 239   24   30
<b>#39B7BC</b> HKS 51 70% CMYK 70   0   28   0 RGB 57   183   188	50 %	HKS 51 10% CMYK 10   0   4   0 RGB 235   246   246
<b>#BEE2E2</b> HKS 51 30% CMYK 30   0   12   0 RGB 190   226   226	15 %	

# Cliparts

