

Übungen zur Wirtschaftsinformatik 2

LE 07 – Recordsets als Datenbankschnittstelle

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi2>

Lernziel und Lehrinhalte

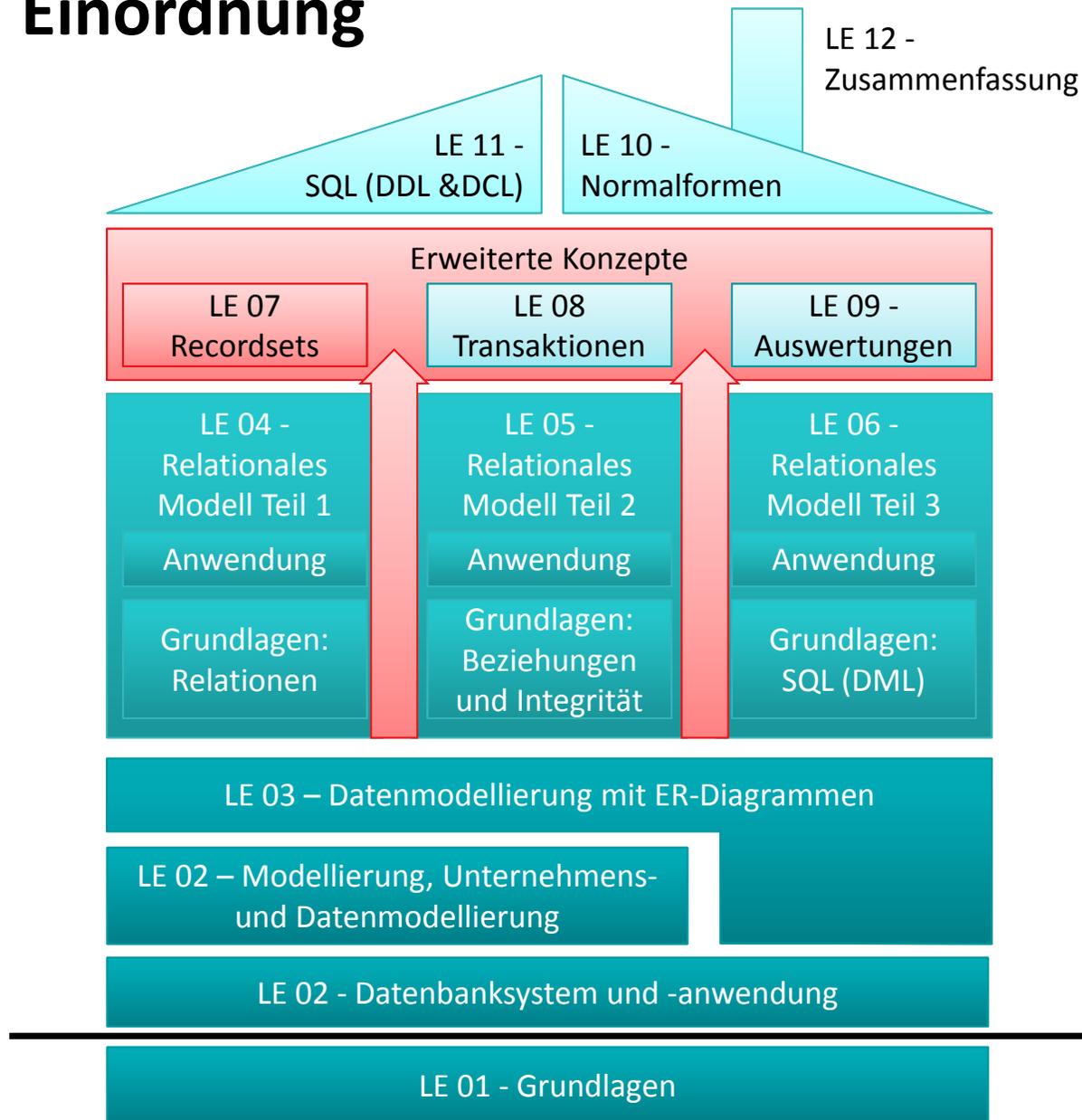
Lernziel

- Nutzung von Recordsets für die Manipulation von Daten verwenden
- Grundlegende Datenbankoperationen anhand einfacher Beispiele durchführen

Lehrinhalte

- Schnittstellen für die Verbindung von Programmiersprachen mit relationalen Datenbanken
- Recordsets als Beispiel einer Datenbankschnittstelle für VBA

Einordnung



Inhalt

Lernziel, Lehrinhalt und Einordnung

Übungen zu Recordsets

- Grundlegende Übungen
 - Einstieg in Recordsets
 - Navigation in Datensätzen per Zeiger
 - Mehrere Datensätze finden und ändern
 - Hinzufügen eines Datensatzes
 - Genau einen Datensatz finden und löschen
- Weiterführende Übungen zum Online-Shop
 - Ändern und Hinzufügen
 - Kombination mit Abfragen
 - Kombination mit SQL

Ausblick

Inhalt

Lernziel, Lehrinhalt und Einordnung

Übungen zu Recordsets

- Grundlegende Übungen
 - Einstieg in Recordsets
 - Navigation in Datensätzen per Zeiger
 - Mehrere Datensätze finden und ändern
 - Hinzufügen eines Datensatzes
 - Genau einen Datensatz finden und löschen
- Weiterführende Übungen zum Online-Shop
 - Ändern und Hinzufügen
 - Kombination mit Abfragen
 - Kombination mit SQL

Ausblick

Einstieg: Übung Ü7.1

Ü7.1: Schreiben Sie eine private Prozedur `zeigeErstesProdukt()` im neuen Modul `mdlProdukte`

- Fügen Sie ein neues Modul `mdlProdukte` hinzu und schreiben Sie die private Prozedur `zeigeErstesProdukt()`
- Initialisieren und öffnen Sie ein Recordset für den Zugriff auf alle Datensätze der Tabelle `tblProdukte`
- Geben Sie die Bezeichnung und den Preis des ersten Produkts im Direktbereich aus

Navigation per Zeiger: Übung Ü7.2

Ü7.2: Schreiben Sie im Modul `mdlKunden` eine private Prozedur `zeigeAlleKunden()`

- Initialisieren und öffnen Sie ein Recordset für den Zugriff auf alle Datensätze der Tabelle `tblKunden`
- Navigieren Sie rückwärts über die Datensätze im Recordset, d.h. von hinten nach vorn und geben Sie jeden Kunden mit Primärschlüssel, Name und Vorname aus.
- Hinweis: Der letzte Kunde im Recordset soll als erstes, oben im Direktbereich stehen.

Navigation per Zeiger: Übung Ü7.2

Ü7.2: Lösungsansatz

- Wie gelangt man an das Ende des Recordset?
- Wie gelangt man vom Ende des Recordset zum vorherigen Element?
- Wie kann man prüfen, ob man das erste Element des Recordset bearbeitet hat und nun der Beginn erreicht ist?

Hinweis: Endlosschleifen lassen Access wie abgestürzt wirken. Sollte ihr Access nicht reagieren, versuchen Sie die folgende Tastenkombination, um eine Endlosschleife zu unterbrechen.

Strg + Pause/Unterbr

bzw.

Ctrl + Break

Mehrere finden und ändern: Ü7.3

Ü7.3: Schreiben Sie im Modul mdlProdukte eine private Prozedur `erhohePreisInKategorie()`

- Initialisieren und öffnen Sie ein Recordset für den Zugriff auf alle Datensätze der Tabelle `tblProdukte`, so dass die Datenbankverbindung Änderungen erlaubt
- Finden Sie alle Produkte der Kategorie 3 (Gartengeräte) und erhöhen Sie den Preis dieser Produkte um 1 EUR, indem Sie
 - den aktuellen Preis aus dem gefundenen Datensatz lesen
 - den neuen Preis berechnen (aktueller Preis + 1 EUR)
 - mit dem neuen Preis den gefundenen Datensatz aktualisieren

Neuen Datensatz hinzufügen: Ü7.4

Ü7.4: Schreiben Sie im Modul mdlProdukte einen private Prozedur hinzufuegenProdukt()

- Initialisieren und öffnen Sie ein Recordset für den Zugriff auf alle Datensätze der Tabelle **tblProdukte**, so dass die Datenbankverbindung Änderungen erlaubt
- Schalten Sie das Recordset in den Anfügemodus und fügen Sie das folgende Produkt (ohne Foto) hinzu
 - Bezeichnung: Spaten 'Mike'
 - Beschreibung: Stabiler Spaten, ideal für Gartenarbeit
 - Preis: 12,99 EUR
 - Lagerbestand: 5 Stück
 - Kategorie: Gartengeräte
- Beenden Sie das Anfügen und Übertragen Sie die Änderungen in die Datenbank

Neuen Datensatz hinzufügen: Ü7.5

Ü7.5: Ändern Sie im Modul mdlProdukte die private Prozedur hinzufuegenProdukt() aus Übung Ü7.4

- Fügen Sie jetzt das Produkt zur neuen Kategorie Fahrräder hinzu
 - Bezeichnung: E-Bike V Pro Max
 - Beschreibung: 70km Reichweite
 - Preis: 1250 EUR
 - Lagerbestand: 3 Stück
 - Kategorie: Fahrräder
- Was müssen Sie zuerst einfügen? Die neue Kategorie oder das neue Produkt?

Einen finden und löschen: Ü 7.6

Ü7.6: Schreiben Sie im Modul mdlKunden eine private Prozedur loeschenKunde(), um die Kundin Ulrike Dinkel (geb. 28.04.1993) zu löschen

- Initialisieren und öffnen Sie ein Recordset für den Zugriff auf alle Datensätze der Tabelle **tblKunden**, so dass die Datenbankverbindung Änderungen erlaubt
- Finden Sie genau die eine Kundin Ulrike Dinkel
- Löschen Sie die Kundin

Hinweis

- Wie muss die Bedingung lauten, damit genau die Kundin Ulrike Dinkel (geb. 28.04.1993) zu finden (und nicht zufällig andere Kundinnen mit gleichem Namen oder Geburtsdatum)?

Inhalt

Lernziel, Lehrinhalt und Einordnung

Übungen zu Recordsets

– Grundlegende Übungen

- Einstieg in Recordsets
- Navigation in Datensätzen per Zeiger
- Mehrere Datensätze finden und ändern
- Hinzufügen eines Datensatzes
- Genau einen Datensatz finden und löschen

– Weiterführende Übungen zum Online-Shop

- Ändern und Hinzufügen
- Kombination mit Abfragen
- Kombination mit SQL

Ausblick



Inhalt

Lernziel, Lehrinhalt und Einordnung

Übungen zu Recordsets

– Grundlegende Übungen

- Einstieg in Recordsets
- Navigation in Datensätzen per Zeiger
- Mehrere Datensätze finden und ändern
- Hinzufügen eines Datensatzes
- Genau einen Datensatz finden und löschen

– Weiterführende Übungen zum Online-Shop

- Ändern und Hinzufügen
- Kombination mit Abfragen
- Kombination mit SQL

Ausblick

Ändern und Hinzufügen: Übung Ü7.7

Ü7.7 (Teil 1): Weniger-Schaltfläche im Formular des Warenkorbs

- Implementieren Sie im Modul `mdlWarenkoerbe` eine öffentliche Prozedur `reduziereAnzahlperRecordset`,
 - die als Parameter die ID des Warenkorbeintrags (Wert des Primärschlüssels) übergeben bekommt, die vom Typ Long ist
 - ein Recordset mit allen Datensätzen der Tabelle `tblWarenkoerbe` füllt
 - im Recordset den Datensatz mit der übergebenen ID des Warenkorb findet
 - die Anzahl bei diesem Datensatz ausliest und prüft, ob die Anzahl größer als 1 ist
 - Wenn ja, soll die Anzahl um 1 reduziert werden
 - Wenn nicht, soll der Datensatz gelöscht werden (weil nicht weiter reduziert werden kann)
- ...

Ändern und Hinzufügen: Übung Ü7.7

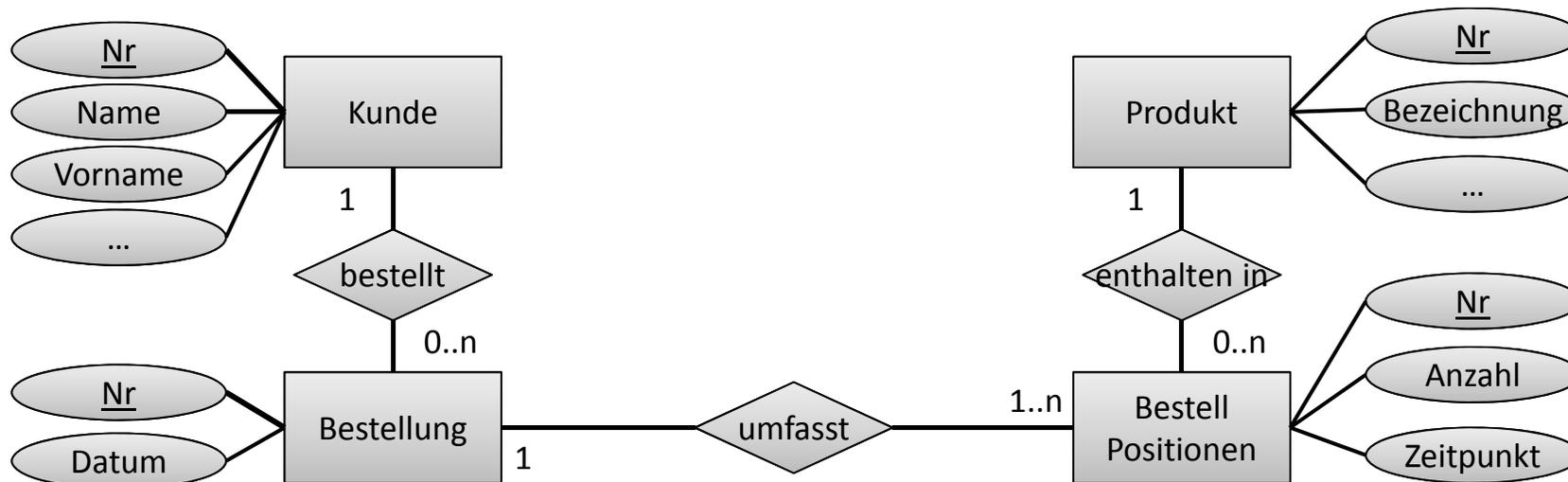
Ü7.7 (Teil 2): Weniger-Schaltfläche im Formular des Warenkorbs

- ...
- Sofern noch nicht vorhanden, erweitern Sie das Formular des Warenkorbs im Detailbereich um eine Schaltfläche zum Reduzieren der Menge eines Produktes
- Implementieren bzw. ändern Sie die Ereignisprozedur so, dass beim Klick
 - aus dem versteckten Feld im Formular die ID des Warenkorbeintrags ermittelt und
 - anschließend die Prozedur `mdlWarenkoerbe.reduziereAnzahlperRecordset()` mit der ID als Parameter aufgerufen
 - und das Formular aktualisiert wird.

Vergleichen Sie, welche Implementierung einfacher ist (mit SQL wie in Ü6.4 oder mit Recordsets)? Warum war der Einsatz von Recordsets dennoch sinnvoll?

Wiederholung (LE05 - Beziehungen): Übung Ü7.8

Ü7.8: Überführen Sie das folgende ER-Diagramm in Tabellen und Beziehungen in Ihrer Datenbank. Nutzen Sie dabei die vorhandenen Relationen Produkte und Kunde.



Ändern und Hinzufügen: Übung Ü7.9

Ü7.9 (Teil 1): Implementieren Sie die private Funktion `erzeugeBestellung` im Modul `mdlBestellungen`

- die als Parameter den Primärschlüssel eines Kunden übergeben bekommt
- indem Sie per Recordset in der Tabelle `tblBestellungen` einen neuen Datensatz erzeugen.
- Verwenden Sie im neuen Datensatz als Wert für den Fremdschlüssel des Kunden den übergebenen Parameterwert
- Setzen Sie im neuen Datensatz den Wert für das Datum auf `Now()`
- Setzen Sie den Rückgabewert der Funktion auf den Wert des Primärschlüssels, bevor Sie die Bearbeitung des Recordsets mit `Update()` und `Close()` beenden

Ändern und Hinzufügen: Übung Ü7.9

Ü7.9 (Teil 2): Testen Sie die Funktion `erzeugeBestellung()` indem

- Sie eine private Prozedur `erzeugeBestellungTest()` schreiben und
- in dieser Prozedur die Funktion `erzeugeBestellung()` aufrufen
- als Parameter den Wert des Primärschlüssels eines existierenden Kunden (aus Tabelle `tblKunden`) übergeben
- und den Rückgabewert der Funktion im Direktbereich ausgegeben
- Prüfen Sie, ob ein Datensatz mit aktueller Zeitangabe und Fremdschlüssel zum übergebenen Kunden in der Tabelle `tblBestellungen` eingefügt wurde

Kombination mit Abfragen: Übung Ü7.10

Ü7.10: Implementieren Sie die private Prozedur `erzeugeBestellpositionen` im Modul `mdlBestellungen`

- Initialisieren Sie ein Recordset mit der Abfrage `qryWarenkorb` (aus Übung Ü5.6 und Ü5.7)

The screenshot shows the Microsoft Access interface for a query named 'qryWarenkorb'. At the top, there is a diagram showing the relationships between three tables: `tblKunden`, `tblWarenkoerbe`, and `tblProdukte`. `tblKunden` is connected to `tblWarenkoerbe` with a 1-to-many relationship. `tblWarenkoerbe` is connected to `tblProdukte` with a 1-to-many relationship. Below the diagram is the design grid for the query. The design grid has columns for each field and rows for 'Feld:', 'Tabelle:', 'Sortierung:', 'Anzeigen:', and 'Kriterien:'. The fields are: `kndIdPk`, `wkbIdPk`, `wkbAnzahl`, `wkbZeitpunkt`, `prdIdPk`, `prdBezeichnung`, `prdPreis`, and `Gesamtpreis: [prdPreis]*[wkbAnzahl]`. The fields `wkbAnzahl`, `wkbZeitpunkt`, and `prdIdPk` are highlighted with red boxes. The 'Kriterien:' row contains the criteria `=gibAktuellerKundeld()`.

Feld:	kndIdPk	wkbIdPk	wkbAnzahl	wkbZeitpunkt	prdIdPk	prdBezeichnung	prdPreis	Gesamtpreis: [prdPreis]*[wkbAnzahl]
Tabelle:	tblKunden	tblWarenkoerbe	tblWarenkoerbe	tblWarenkoerbe	tblProdukte	tblProdukte	tblProdukte	
Sortierung:				Aufsteigend				
Anzeigen:	<input checked="" type="checkbox"/>							
Kriterien:	=gibAktuellerKundeld()							
oder:								

- Gehen Sie über alle Datensätze im Recordset und geben Sie den Schlüssel des Produktes, die Anzahl und den Zeitpunkt im Direktbereich aus

Kombination mit Abfragen: Übung Ü7.11

Ü7.11 (Teil 1): Erweitern Sie die Prozedur `erzeugeBestellungen` (aus vorheriger Ü7.10) so, dass

- die Funktion den Primärschlüssel einer Bestellung als Parameter übergeben bekommt
- zusätzlich ein weiteres Recordset zum Zugriff auf die Tabelle `tblBestellungen` initialisiert wird
- über alle Datensätze im Recordset des Kundenwarenkorb gegangen wird und jeweils
 - ein neuer Datensatz im `Bestellungen`-Recordset angelegt wird
 - der Fremdschlüssel des Produktes, die Anzahl und der Zeitpunkt aus dem Warenkorb in den neuen Datensatz im `Bestellungen`-Recordset kopiert wird und
 - für den neuen Datensatz im `Bestellungen`-Recordset der Fremdschlüssel der Bestellung auf den als Parameter übergebenen Wert der `BestellungID` setzen
 - den neuen Datensatz im `Bestellungen`-Recordset speichern

Kombination mit Abfragen: Übung Ü7.11

Ü7.11 (Teil 1): Lösungsansatz

prdBezeichnung	prdPreis	wkbAnzahl	Gesamt	wkbIdPk	prdIdPk	knIdPk	wkbZeitpunkt
Blumenzwiebeln Sorte XY	0,80 €	6	4,80 €	3	14	2	25.05.2013 17:03
Rasendünger "Güner Karl"	8,50 €	1	8,50 €	10	2	2	23.11.2013 19:07
Sonnenblumensamen, Sor	1,99 €	1	1,99 €	11	16	2	23.11.2013 19:08

bspIdPk	bspAnzahl	bspZeitpunkt	bspPrdIdFk	bspBstIdFk
1	3	01.01.2000	1	1
2	1	01.01.2000	24	1
3	2	01.01.2000	1	2
4	1	01.01.2000	15	2
5	2	01.02.2000	15	3
...
13	1	23.11.2013 19:07:56	2	4
14	1	23.11.2013 19:08:01	2	4

Parameter
BestellungID

Kombination mit Abfragen: Übung Ü7.11

Ü7.11 (Teil 2): Testen Sie Prozedur `erzeugeBestellpositionen`, indem

- Sie eine private Prozedur `erzeugeBestellpositionenTest()` schreiben und
- in dieser Prozedur zuerst die Funktion `setzeAktuellerKundeId()` aufrufen und den Primärschlüssel eines existierenden Kunden als Parameter übergeben und
- dann die Funktion `erzeugeBestellpositionen()` aufrufen und als Parameter den Wert des Primärschlüssels einer existierenden Bestellung (aus Tabelle `tblBestellungen`) übergeben.
- Prüfen Sie, ob der Warenkorb des aktuellen Kunden in die Tabelle `tblBestellPositionen` übernommen wurde und als Fremdschlüssel in jedem Datensatz die übergeben Bestellung verwendet wird

Kombination mit SQL: Übung Ü7.12

Ü7.12 (Teil 1): Benutzeranmeldung

- Ändern Sie im Modul `mdlBenutzer` vorhandene Funktion "anmelden" (aus Ü1.11)
- Deklarieren Sie eine Variable vom Typ `String` und initialisieren Sie sie mit folgender SQL-Abfrage (aufbauend auf Ü6.6, `qryBenutzerUndKundenID`), die Sie mit den übergebenen Parametern für Benutzernamen und Passwort ergänzen

```
"SELECT tblBenutzer.*, tblKunden.kndIdPk " & _  
"FROM tblBenutzer LEFT JOIN tblKunden " & _  
"ON tblBenutzer.benBenutzername = tblKunden.kndbenIdFk " & _  
"WHERE tblBenutzer.benBenutzername='" & pstrBenutzer & "' " & _  
"AND tblBenutzer.benPasswort='" & pstrPasswort & "';"
```

- Initialisieren und öffnen Sie ein Recordset mit dieser SQL-Abfrage
- Hinweis: Übernehmen Sie die Abfrage per Copy&Paste. Wenn das nicht geht, achten Sie auf die Leerzeichen! Die Spaltenbezeichnungen müssen zu ihrer Benutzer-Tabelle und die Parameter zu Ihrer Funktion passen.
- ...



Kombination mit SQL: Übung Ü7.12

Ü7.7 (Teil 2): Benutzeranmeldung

- ...
- Prüfen Sie ob genau ein Datensatz gefunden wurde (mittels `rcsBenutzer.RecordCount`, wobei Sie auf die bekannten Fallstricke beim Umgang mit `RecordCount` achten müssen!)
- Wenn genau ein Datensatz gefunden wurde
 - gehen mit `MoveFirst` zum ersten Datensatz (notwendig wegen `RecordCount`-Fallstrick)
 - setzen Sie den Rückgabewert der Funktion anmelden auf `True`
 - ermitteln Sie aus dem Datensatz das Kennzeichen, ob der Benutzer ein Administrator ist und speichern Sie es in der Variable `bolIstBenutzerAdmin` (aus Ü1.11)
 - wenn der Benutzer kein Admin ist, ermitteln Sie aus dem Datensatz die Kundennummer
 - rufen Sie mit der Kundennummer die Prozedur `setzeAktuellerKundeId()` im Modul `mdlKunden` auf
- In allen anderen Fällen setzen Sie den Rückgabewert der Funktion anmelden auf `False`

Kombination mit SQL: Übung Ü7.12

Ü7.7 (Teil 3): Prüfen Sie, dass der aktuelle Kunde in Abfragen und Prozeduren verwendet wird

- Prüfen Sie, ob die Abfrage `qryWarenkorb` (aus Übung Ü5.7) die Funktion `gibAktuellerKundeID()` als Selektionskriterium verwendet und nur noch den Warenkorbinhalt des aktuell angemeldeten Kunden zeigt.

Kombination mit SQL: Übung Ü7.12

Ü7.7 (Teil 4): Prüfen Sie, dass der aktuelle Kunde in Abfragen und Prozeduren verwendet wird

- Prüfen Sie, ob die Ereignisprozeduren
 - für das Hinzufügen von Produkten zum Warenkorb **btnInWarenkorbLegen_Click()** (aus Übung Ü6.3)
 - für das Löschen des Warenkorb eines Kunden **btnLeeren_Click()** (aus Übung Ü6.6)
- den Rückgabewert der Funktion **gibAktuellerKundeID()** verwendet, um die Funktionen im Modul **mdlWarenkoerbe** aufzurufen

Kombination mit SQL: Übung Ü7.12

Ü7.7 (Teil 5): Hinweis: Sie müssen sich nun immer anmelden, bevor Sie die betroffenen Formulare oder Abfragen in Ihrer Datenbank öffnen. Andernfalls enthalten sie keine Datensätze oder eine Fehlermeldung wegen einer Fremdschlüsselverletzung.



Inhalt

Lernziel, Lehrinhalt und Einordnung

Übungen zu Recordsets

- Grundlegende Übungen
 - Einstieg in Recordsets
 - Navigation in Datensätzen per Zeiger
 - Mehrere Datensätze finden und ändern
 - Hinzufügen eines Datensatzes
 - Genau einen Datensatz finden und löschen
- Weiterführende Übungen zum Online-Shop
 - Ändern und Hinzufügen
 - Kombination mit Abfragen
 - Kombination mit SQL

Ausblick



Inhalt

Lernziel, Lehrinhalt und Einordnung

Übungen zu Recordsets

– Grundlegende Übungen

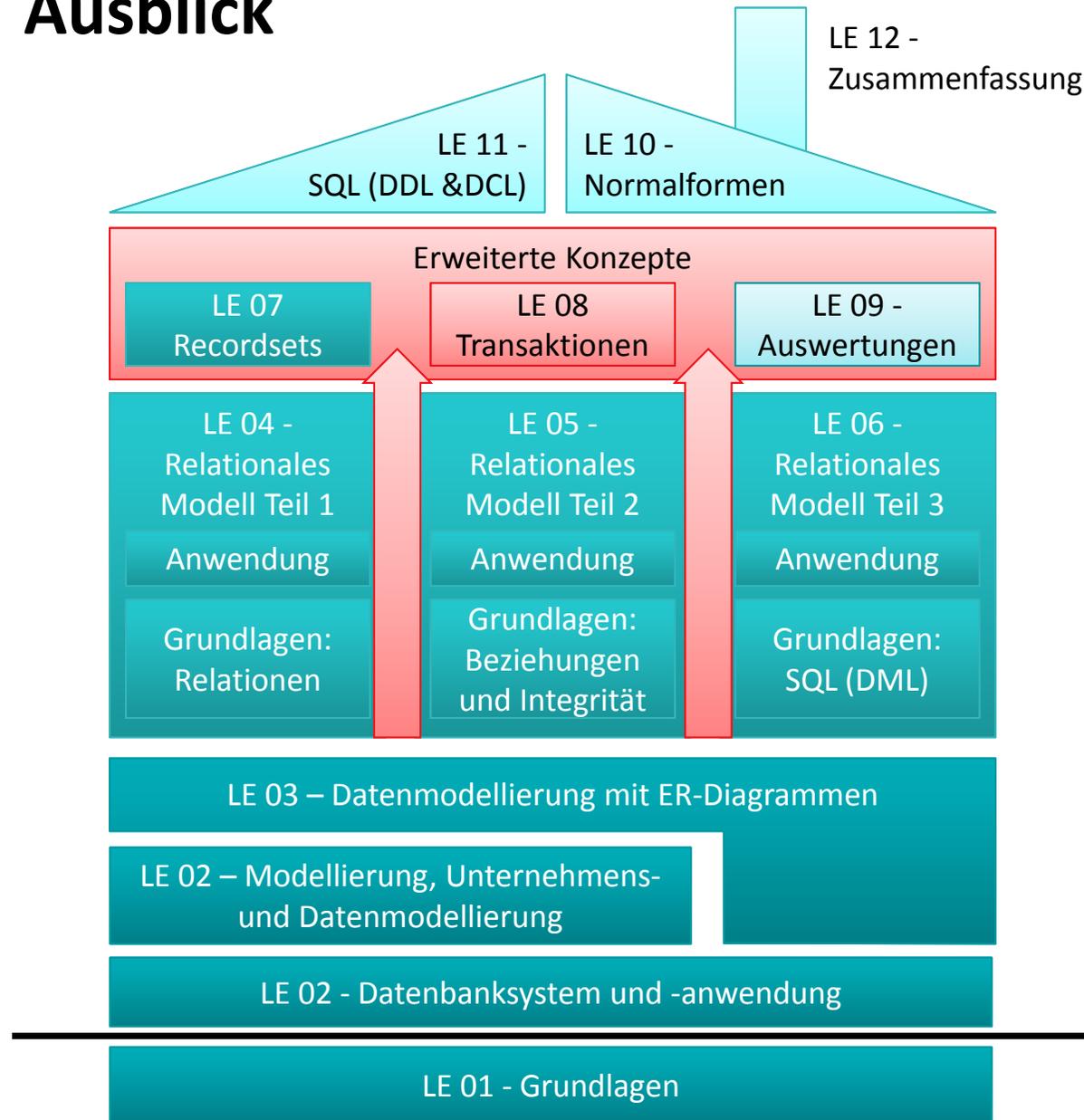
- Einstieg in Recordsets
- Navigation in Datensätzen per Zeiger
- Mehrere Datensätze finden und ändern
- Hinzufügen eines Datensatzes
- Genau einen Datensatz finden und löschen

– Weiterführende Übungen zum Online-Shop

- Ändern und Hinzufügen
- Kombination mit Abfragen
- Kombination mit SQL

Ausblick

Ausblick



Übungen zum Kurs Datenbanken

LE 07 – Recordsets als Datenbankschnittstelle

Prof. Dr. Thomas Off

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