

Übungen zur Wirtschaftsinformatik 2

LE 08 – Transaktionen

Prof. Dr. Thomas Off

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

Lernziel und Inhalt

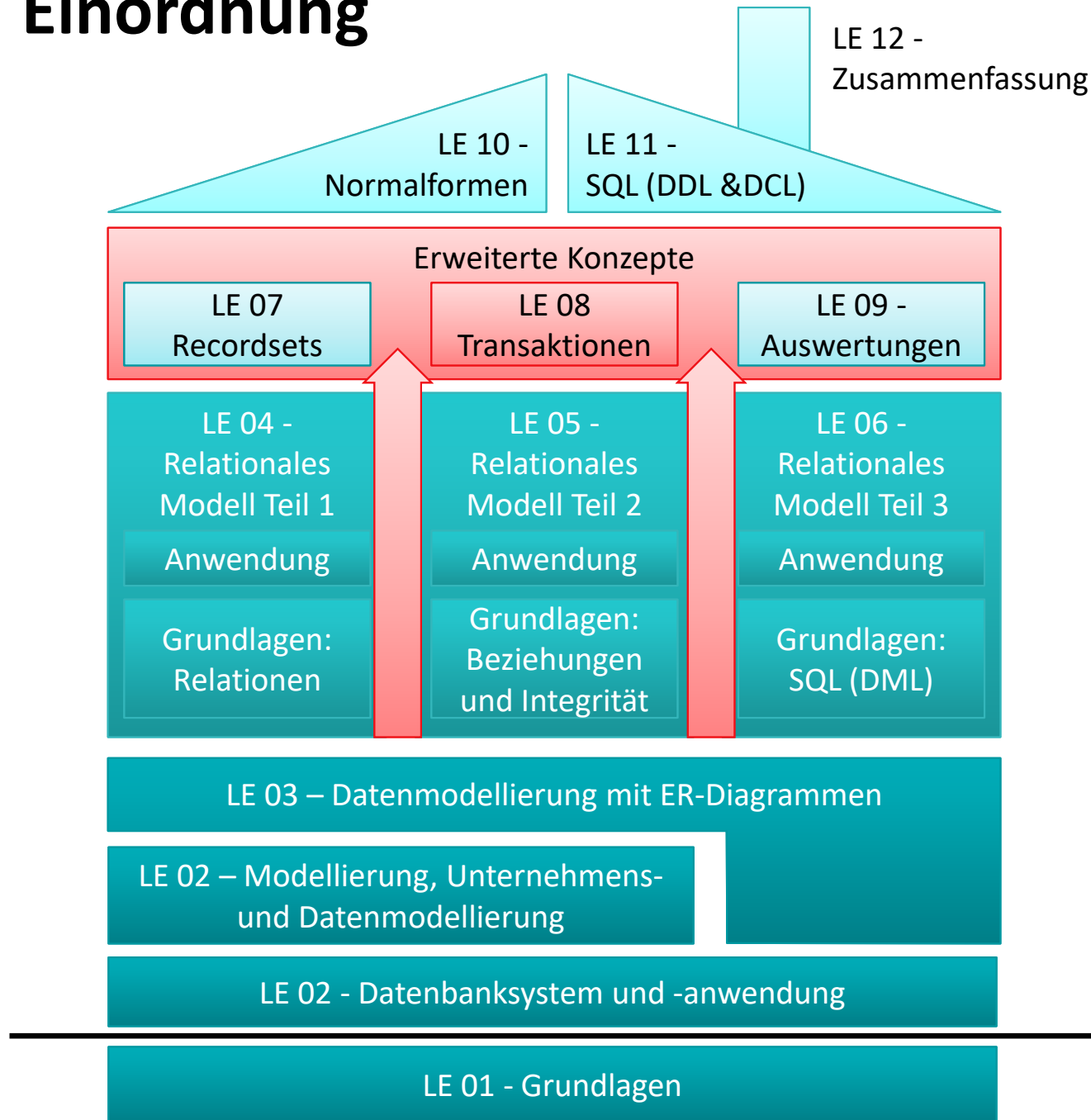
Lernziel

- Einsatzszenarien für Transaktionen kennenlernen
- Exemplarisches Einsatzszenario identifizieren und aufbauen
- Transaktionen mit Commit und Rollback praktisch anwenden

Lehrinhalt

- Wiederholung von elementaren Datenbankoperationen
- Aufbau eines Szenarios für die Verwendung von Transaktionen
- Implementierung von Transaktionen

Einordnung



Inhalt

Lernziel, Lehrinhalte und Einordnung

Anwendung von Transaktionen mit MS Access

- Grundlegende Übungen
 - Implementierung der Lagerbestandsverwaltung
 - Absicherung mit Transaktionen
- Weiterführende Übungen
 - Implementierung des Bestellprozesses als Szenarios mehrerer zusammengehöriger Datenbankoperationen
 - Erzeugen einer Bestellung
 - Kopieren des Warenkorbs in Bestellungenpositionen
 - Leeren des Warenkorbs
 - Absicherung mit Transaktionen

Ausblick

Inhalt

Lernziel, Lehrinhalte und Einordnung

Anwendung von Transaktionen mit MS Access

- Grundlegende Übungen
 - Preisanpassungen um erweitern Transaktionen
 - Implementierung der Lagerbestandsverwaltung
- Weiterführende Übungen
 - Implementierung des Bestellprozesses als Szenarios mehrerer zusammengehöriger Datenbankoperationen
 - Erzeugen einer Bestellung
 - Kopieren des Warenkorbs in Bestellungenpositionen
 - Leeren des Warenkorbs
 - Absicherung mit Transaktionen

Ausblick

Ü8.1: Preisanpassung mit Transaktionen absichern

- Ändern Sie im Modul mdlProdukte die Lösung der Aufgabe Ü7.3, so dass der Preis der Produkte der Kategorie um 5 EUR verringert wird.
- Ergänzen Sie die Lösung Ü7.3 anschließend um Transaktionen, so dass immer Preis bei allen Produkten der ausgewählten Kategorie angepasst wird oder bei keinem.
- Führen Sie die Prozedur mehrfach aus.
- Was stellen Sie fest? Woher stammt die Meldung, die Sie sehen?

Ü8.2: Lagerbestand beim Hinzufügen zum Warenkorb reduzieren

- Ergänzen Sie die vorhandene Prozedur `hinzufuegen()` im Modul `mdlWarenkoerbe`
- Fügen Sie eine weitere SQL-Anweisungen ein die den Lagerbestand beim Hinzufügen eines Artikels zum Warenkorb reduziert
- Nutzen Sie eine SQL **UPDATE**-Anweisung, bei der Sie in der **WHERE**-Bedingung den als Parameter übergebenen Primärschlüssel des Produktes verwenden
- Warum müssen beide Anweisungen in einer Transaktion erfolgen? Schreiben Sie Ihre Antwort auf diese Frage als Kommentar in den Quellcode.
- Sichern Sie die zusammengehörigen SQL-Anweisungen mit einer Transaktion ab, damit diese atomar ausgeführt werden

Hinweis: Bei einem Online-Shop würde der Lagerbestand erst bei der Bestellung reduziert. Zum Zweck der Übung erfolgt die Lagerbestandsverwaltung hier bereits beim Hinzufügen und Entfernen von Artikeln aus dem Warenkorb.

Transaktionen: Übung Ü8.3

Ü8.3 (Teil 1): Weitere Funktionen zur Anpassung des Lagerbestands

- Ergänzen Sie weitere vorhandenen Funktionen im Modul `mdlWarenkoerbe` um SQL-Anweisungen zum
 - Reduzieren des Lagerbestands beim Erhöhen der Anzahl des Artikels im Warenkorb
 - Erhöhen des Lagerbestands bei Löschen eines Artikels aus dem Warenkorb
- Nutzen Sie die `DLookup()`-Funktion, um z.B. den Primärschlüssel des Produktes zu ermitteln, der zum Wareneintrag gehört

```
' Beispiel
Let lngPrdId = DLookup("wkbprdIdFk", _
                    "tblWarenkoerbe", _
                    "wkbIdPk=" & plngWarenkorbID)
```

wkbIdPk	wkbAnzahl	wkbZeitpunkt	wkbprdIdFk	wkbkndIdFk
1	6	01.01.2000	1	1
2	4	01.01.2000	2	1
3	1	01.01.2000	3	1

Transaktionen: Übung Ü8.3

Ü8.3 (Teil 2): Weitere Funktionen zur Anpassung des Lagerbestands

- ...
- Warum müssen beide Anweisungen in einer Transaktion erfolgen? Schreiben Sie Ihre Antwort auf diese Frage als Kommentar in den Quellcode.
- Sichern Sie die zusammengehörigen SQL-Anweisungen mit einer Transaktion ab, damit diese atomar ausgeführt werden

Hinweis: Bei einem Online-Shop würde der Lagerbestand erst bei der Bestellung reduziert. Zum Zweck der Übung erfolgt die Lagerbestandsverwaltung hier bereits beim Hinzufügen und Entfernen von Artikeln aus dem Warenkorb.

Ü8.4 (Teil 1): Lagerbestand beim Reduzieren der Anzahl im Warenkorb erhöhen

- Ergänzen Sie die vorhandene Funktion `reduziereAnzahlperRecordset ()` im Modul `mdlWarenkoerbe` um das Erhöhen des Lagerbestands,
- indem Sie nachdem der Zeiger im Recordset Warenkorb auf den richtigen Warenkorbbeitrag zeigt dort den Fremdschlüssel des Produktes lesen und in einer Variable speichern und
- nach Abschluss aller Änderungen am Recordset des Warenkorbs (d.h. nach dem Close) ein neues Recordset für den Zugriff auf die Tabelle `tblProdukte` öffnen
 - den Recordset-Zeiger auf den ersten (und einzigen) Eintrag mit dem Schlüssel des Produktes setzen
 - das Recordset in den Änderungsmodus schalten, die Änderung des Lagerbestands vornehmen und abschließen

Transaktionen: Übung Ü8.4

Ü8.4 (Teil 2): Weitere Funktionen zur Anpassung des Lagerbestands

- ...
- Welche Anweisungen müssen in einer Transaktion erfolgen? Schreiben Sie Ihre Antwort auf diese Frage als Kommentar in den Quellcode.
- Sichern Sie die zusammengehörigen Anweisungen mit einer Transaktion ab, damit diese atomar ausgeführt werden

Hinweis: Bei einem Online-Shop würde der Lagerbestand erst bei der Bestellung reduziert. Zum Zweck der Übung erfolgt die Lagerbestandsverwaltung hier bereits beim Hinzufügen und Entfernen von Artikeln aus dem Warenkorb.

Inhalt

Lernziel, Lehrinhalte und Einordnung

Anwendung von Transaktionen mit MS Access

- Grundlegende Übungen
 - Implementierung der Lagerbestandsverwaltung
 - Absicherung mit Transaktionen
- Weiterführende Übungen
 - Implementierung des Bestellprozesses als Szenarios mehrerer zusammengehöriger Datenbankoperationen
 - Erzeugen einer Bestellung
 - Kopieren des Warenkorbs in Bestellungspositionen
 - Leeren des Warenkorbs
 - Absicherung mit Transaktionen

Ausblick



Inhalt

Lernziel, Lehrinhalte und Einordnung

Anwendung von Transaktionen mit MS Access

- Grundlegende Übungen
 - Implementierung der Lagerbestandsverwaltung
 - Absicherung mit Transaktionen
- Weiterführende Übungen
 - Implementierung des Bestellprozesses als Szenarios mehrerer zusammengehöriger Datenbankoperationen
 - Erzeugen einer Bestellung
 - Kopieren des Warenkorbs in Bestellungenpositionen
 - Leeren des Warenkorbs
 - Absicherung mit Transaktionen

Ausblick

Wiederholung (LE01 – Grundlagen): Übung Ü8.5 ~~BHT~~

Ü8.5 (Teil 1): Implementieren Sie das Bestellen eines gefüllten Warenkorbs

- Implementieren Sie im Modul `mdlBestellungen` die Funktion `bestellen()` die als Parameter den Primärschlüssel eines Kunden übergeben bekommt und als Rückgabewert einen Long liefert
- Prüfen Sie mit der Domänenfunktion `DCount()`, ob Artikel im Warenkorb des als Parameter übergebenen Kunden vorhanden sind (wenn nicht, Rückgabewert der Funktion auf `-1` setzen und Funktion vorzeitig verlassen)
- Rufen Sie in allen anderen Fällen die Funktion `erzeugeBestellung()` (aus Ü7.4) auf und speichern Sie den Rückgabewert dieser Funktion als ID der neu erzeugten Bestellung in einer Variable speichern
- ...

Wiederholung (LE01 – Grundlagen): Übung Ü8.5 ~~BHT~~

Ü8.5 (Teil 2): Implementieren Sie das Bestellen des eines gefüllten Warenkorbs

- ...
- Rufen Sie die Funktion **erzeugeBestellpositionen()** aus Ü7.5 und Ü7.6 auf und übergeben Sie ihr als Parameter die IDs der neu erzeugten Bestellung
- Rufen Sie die neue Funktion **loeschenWarenkorb()** aus Ü6.6 auf und übergeben Sie als Parameter die ID des aktuellen Kunden
- Geben Sie als Rückgabewert der Funktion **bestellen()** die ID der neuen Bestellung zurück und beenden Sie die Funktion
- ...

Wiederholung (LE01 – Grundlagen): Übung Ü8.5 ~~BHT~~

Ü8.5 (Teil 3): Implementieren Sie das Bestellen des eines gefüllten Warenkorb

- ...
- Erweitern Sie Ihr Formular des Warenkorb um eine Schaltfläche "Bestellen"
 - Ermitteln Sie in der Ereignisprozedur den Primärschlüsselwert des aktuellen Kunden
 - Rufen Sie die neue Funktion `bestellen()` auf und übergeben Sie den Primärschlüssel des aktuellen Kunden
 - Speichern Sie den Rückgabewert der Funktion in einer Variable
 - Aktualisieren Sie das Formular des Warenkorb, indem Sie es neu laden
 - Zeigen Sie eine Fehlermeldung, wenn die Bestellung nicht erfolgreich war, d.h. die Funktion `bestellen()` einen Rückgabewert < 0 liefert; andernfalls eine Erfolgsmeldung.
 - Schließen Sie das Formular des Warenkorb

Ü8.5: Bestellvorgang mit Transaktion absichern

- Warum müssen welche der zuvor programmierten Anweisungen in einer Transaktion erfolgen? Schreiben Sie Ihre Antwort auf diese Frage als Kommentar in den Quellcode.
- Sichern Sie den Bestellvorgang mit Transaktionen ab, so dass
 - die Prüfung des Warenkorbhalts
 - das Erzeugen einer neuen Bestellung
 - das Kopieren des Warenkorbhalts in die Bestellpositionen und
 - das Leeren des Warenkorbs sollen durch eine Transaktion abgesichert werden, so dass sie als atomare Einheit ausgeführt werden
 - Beachten Sie den Erfolgsfall (Commit) und den Fehlerfall (Rollback)

Inhalt

Lernziel, Lehrinhalte und Einordnung

Anwendung von Transaktionen mit MS Access

- Grundlegende Übungen
 - Implementierung der Lagerbestandsverwaltung
 - Absicherung mit Transaktionen
- Implementierung des Bestellprozesses
 - Aufbau eines Szenarios mehrerer zusammengehöriger Datenbankoperationen als Wiederholung
 - Erzeugen einer Bestellung
 - Kopieren des Warenkorbs in Bestellungspositionen
 - Leeren des Warenkorbs
 - Absicherung mit Transaktionen

Ausblick



Inhalt

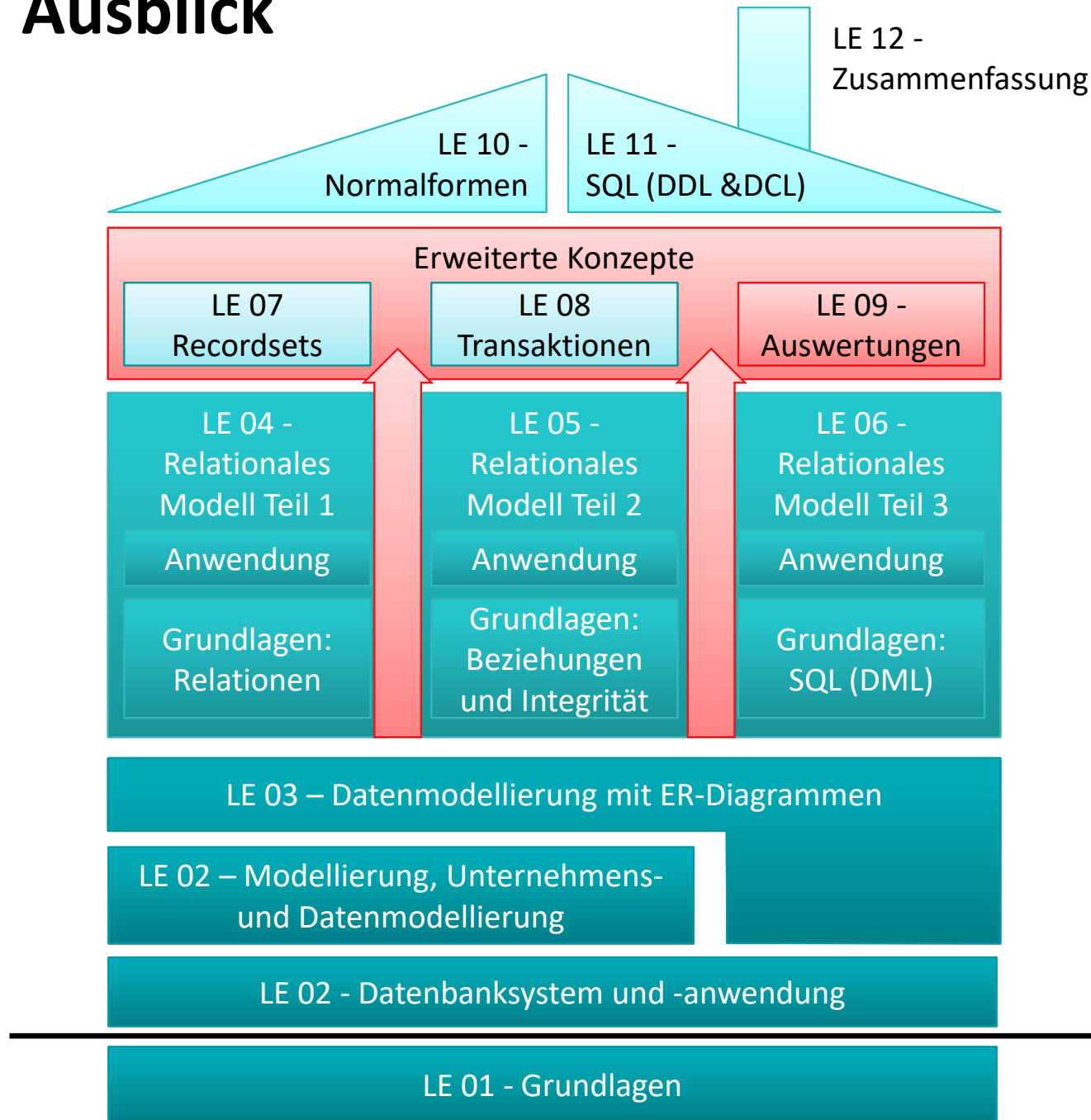
Lernziel, Lehrinhalte und Einordnung

Anwendung von Transaktionen mit MS Access

- Grundlegende Übungen
 - Implementierung der Lagerbestandsverwaltung
 - Absicherung mit Transaktionen
- Implementierung des Bestellprozesses
 - Aufbau eines Szenarios mehrerer zusammengehöriger Datenbankoperationen als Wiederholung
 - Erzeugen einer Bestellung
 - Kopieren des Warenkorbs in Bestellungenpositionen
 - Leeren des Warenkorbs
 - Absicherung mit Transaktionen

Ausblick

Ausblick



Übungen zur Wirtschaftsinformatik 2

LE 08 – Transaktionen

Prof. Dr. Thomas Off

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