

28 + 5 * 3 = 43 ^{8^1}
 Note ~~2.3~~ 2.0 _{Sty Sty}

Klausur Datenbanksysteme I : Name:
 Termin: Donnerstag, 24.07.2008: Matr.-Nr:

Die Teilnahme an dieser Klausur ist Pflicht. Ihre Semesternote wird aus der von Ihnen im Semester erreichten Gesamtpunktzahl errechnet. Die Übungsaufgaben ergeben maximal 30 Punkte, diese Klausur ergibt maximal 70 Punkte, Summe 100 Punkte. Bestanden beginnt bei 51 Punkten.

7 Aufgabe 1: Konzepte/Begriffe
 (7 Punkte)

Erläutern Sie kurz folgende Begriffe: Trigger, Stored procedure, Index, Sperre

7 Aufgabe 2: Datenbankentwurf
 (7 Punkte)

Wie modellieren Sie eine m:n-Beziehung „Leser leihen Buchexemplare“ mit einem ER-Graph? Skizzieren Sie die möglichen Tabellen mit je einer Datenzeile
 Wie lauten die erzeugenden SQL-Befehle mit Primärschlüssel, Feldbedingungen, referentieller Integrität?

18 Aufgabe 3: SQL
 (21 Punkte)

Kunde	knr	kname	kanschrift
	101	Peter Müller	13467 Berlin
	102	Eva Braun	23245 Hamburg
	103	Hans Otto	72179 Stuttgart

bucht	bnr	knr	znr	dauer	anreise
	401	101	301	1	15/1/2005
	402	102	303	3	15/2/2005
	403	103	301	2	16/1/2005

Zimmer	znr	beschreibung	hotel	tagespreis
	301	EZ	201	67.00
	302	DZ	203	85.00
	303	Suite	201	155.00

Hotel	hnr	hname	hanschrift
	201	Excelsior	79000 Ulm
	202	Flora	44000 Münster
	203	Zentral	68000 Oppeln

Einzeltabellenabfragen:

- 1 Gesucht sind Name und Anschrift aller Berliner Kunden.
- 2 Gesucht ist eine Liste der Zimmer mit Wochenpreis < 600 absteigend nach Preis sortiert.
 + Wochenpreis ausgeben

Joinabfragen:

- 3 Welcher Kunde hat welche Zimmer in welchem Hotel gebucht? *Knummer reicht*
- 4 In welchem Hotel gibt es Doppelzimmer?

Gruppierungen:

- 5 Was erbringen die Zimmer in welchem Hotel für Mieteinnahmen?
- 6 Was haben die Kunden für Mietkosten?

Änderungen:

- 7 Kunde Hanna Barth bucht ein Einzelzimmer für 2 Tage im Hotel Excelsior.
- 8 Die Zimmerpreise des Hotels Flora werden um 10 % erhöht.

Sichten:

- 9 Erzeugen Sie eine View Hotelzimmer der Hotels in Ulm und ihrer Zimmer mit den neuen Überschriften (Hotel, Anschrift, Zimmerbeschreibung, Zimmerpreis).
- 10 Erzeugen Sie eine Tabelle bucht mit Primärschlüssel bnr (Buchungsnummer), grösser als 401, knr (Kundennummer) muss in Tabelle kunde existieren, znr (Zimmernummer) muss in Tabelle Zimmer existieren, dauer und anreise haben keine Bedingungen.

Trigger

- 11 Schreiben Sie einen Trigger t, der bei jeder neuen Buchung die Gesamtzahl der freien Zimmer um 1 reduziert. Die Gesamtzahl ist am Anfang 520.

Prozeduren

- 12 Schreiben Sie eine Parameterprozedur p, die für alle Einzelzimmer die Zimmerpreise um einen eingegebenen Prozentsatz erhöht.

18

Aufgabe 4: Visuelle Definition

(7 Punkte)

Erstellen Sie zur Access-Abfrage die SQL-Form

7

lieferung : Auswahlabfrage

Feld:	nachname	artikelname	bestellgewicht: [bestellmenge]*[artikelgewicht]
Tabelle:	Person	Artikel	
Sortierung:	Aufsteigend		
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:	"Meier" Oder "Müller"		
oder:			

Aufgabe 5: Access-Programmierung (14 Punkte)

Gegeben sind ein Formular und zwei Programme.
Was tun die Programme ?
Was ist der Unterschied ?
Bewerten Sie die beiden Lösungen

The screenshot shows a Microsoft Access form. On the left, there are two text boxes labeled 'Text0:' and 'Text2:'. To the right of these text boxes are two buttons labeled 'Befehl4' and 'Befehl5'. At the bottom of the form, there is a status bar that reads 'Datensatz: 14' followed by navigation icons and '1 von 1'.

1. Programm:

```
Private Sub Befehl4_Click()  
Dim db As DAO.Database  
Set db = CurrentDb()  
Dim rs As DAO.Recordset  
Dim sql As String  
sql = "select * from kunde where kname = " + Text0 + " and kpasswort = " + Text2 + ""  
MsgBox (sql)  
Set rs = db.OpenRecordset(sql)  
If rs.EOF Then  
    MsgBox ("nicht OK")  
Else  
    MsgBox ("OK")  
End If  
End Sub
```

2. Programm:

```
Private Sub Befehl5_Click()  
Dim db As DAO.Database  
Set db = CurrentDb()  
Dim rs As DAO.Recordset  
Dim sql As String  
sql = "select * from kunde"  
MsgBox (sql)  
Dim gefunden As Boolean  
gefunden = False  
Set rs = db.OpenRecordset(sql)  
rs.MoveFirst  
While Not rs.EOF  
    If rs![kname] = Text0 And rs![kpasswort] = Text2 Then  
        gefunden = True  
    End If  
    rs.MoveNext  
Wend  
If gefunden Then  
    MsgBox ("OK")  
End Sub
```

```

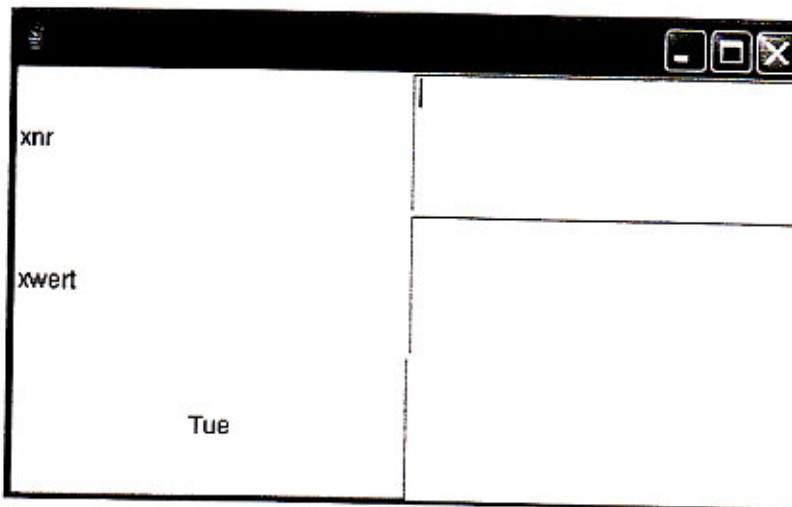
Eise
  MsgBox ("nicht OK")
End If
End Sub

```

Aufgabe 6: Java-Programmierung (14 Punkte)

- +7 Wo ist der Fehler im Programm? *weitere Fehler (2 oder 3)*
 - Was ist die Ausgabe des Programms, wenn der Fehler korrigiert ist? *in Zahlen für die Beispiele*

bestellung : Tabelle				
	anr	pnr	apreis	amenge
▶	1	1	10,00 DM	20
	2	1	5,00 DM	5
	1	2	4,00 DM	6
	2	2	4,00 DM	8
*	0	0	0,00 DM	0



```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;

public class a2 extends Frame {
  Panel ein = new Panel(new GridLayout(0,2));
  TextField xnr = new TextField();
  TextField xwert = new TextField();
  Button tue;
  ResultSet rs;
  Statement s;

  public a2()
  {
    add("Center", ein);
    ein.add(new Label("xnr"));
    ein.add(xnr);
    ein.add(new Label("xwert"));
    ein.add(xwert);
    ein.add(tue = new Button("Tue"));
  }
}

```

```

setSize(400,250);
setVisible(true);
try {
    Connection c = DriverManager.getConnection("jdbc:odbc:klausuraccessdb", "", "");
    s = c.createStatement();
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch (Exception ex) {ex.printStackTrace();}

addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        System.exit(0); });
tue.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        tue();});
}

private void tue()
{
    try {
        String q = "SELECT sum (apreis * amenge) FROM bestellung WHERE pnr = " + xnr.getText();
        rs = s.executeQuery(q);
        rs.next();
        xwert.setText(rs.getString(1));
    }
    catch (Exception ex) {ex.printStackTrace();}
}

public static void main(String[] args)
{
    new a2();
}
}

```

51 + 2 = 53 sty

24.07.08

Aufgabe 1

Trigger: Die im Trigger gespeicherte Anweisungsfolge wird nach einem INSERT-, UPDATE, oder DELETE-Ereignis automatisch ausgeführt.

Stored procedure: Mehrere Anweisungen werden unter einem Namen gespeichert, auf Anforderung ausgeführt

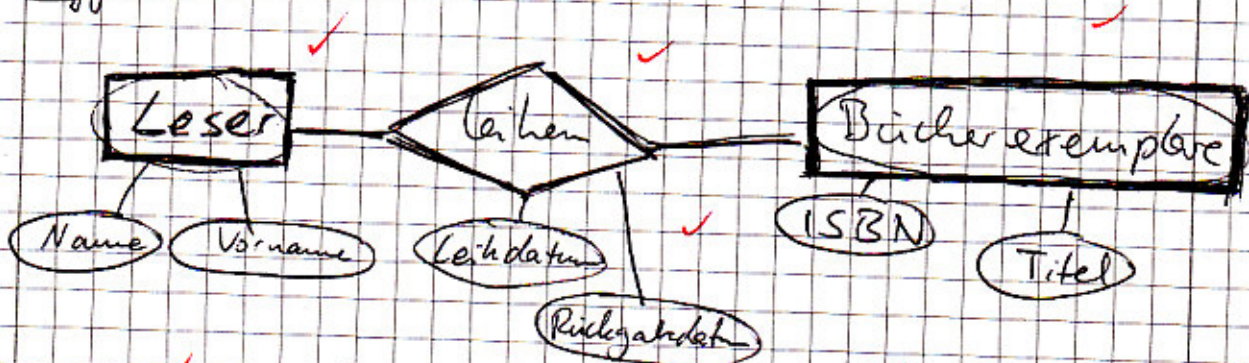
Index: Hilfsmittel, um Datensätze in einer Tabelle schneller zu finden.

Sperre: Funktion der Datenbank, die gleichzeitigen Zugriff verschiedener Benutzer auf die selben Datensätze verhindert.

- exclusive-Sperre: lässt gar keinen Zugriff anderer zu

- shared-Sperre: Lesen ist anderen Benutzern erlaubt.

Aufgabe 2



Leser:

L#	Name	Vorname
1	Schmitt	Isolde

Buecher:

B#	isbn	titel
1	15-3458-230	Der Seher

ausleihen:

A#	L#	B#	leih	rueck
1	1	1	15/07/08	15/09/08

create table Leser (L# integer primary key, name varchar
vorname varchar (20))

create table Buecher (B# integer primary key, isbn char (20),
titel varchar (50))

create table ausleihen (A# integer primary key, L# integer
foreign key (L#) references Leser, B# integer foreign
key (B#) references Buecher, leih date, ruede date)

Aufgabe 3: Numeriert (siehe Aufgabenslatt)

1) select kname, kauschrift from Kunde where kauschrift
like 'Berlin' (Tagespreis * 7)

~~2) select * from Kunde where (Tagespreis * 7) < 100
order by Tagespreis desc~~

~~3) select kname, beschreibung, kname from Kunde, bucht,
Zimmer, Hotel where ((Kunde.knr = bucht.knr)
and (Zimmer.zur = bucht.zur) and (Hotel.knr
= Zimmer.hotel))~~

4) select lname, lname from Hotel, Zimmer where
((Hotel.knr = Zimmer.hotel) and (beschreibung = 'D?'))

5) select lname, (Sum (dauer * tagespreis)) as mieteinnahmen
from Hotel, Zimmer, bucht where ((Hotel.knr = Zimmer.hotel)
and Zimmer.zur = bucht.zur)) Group by zur

6) select kname, (Sum (dauer * tagespreis)) as mietkosten from
Kunde, bucht, Zimmer where ((Kunde.knr = bucht.knr)
and (Zimmer.zur = bucht.zur)) group by kname

7) insert into Kunde (knr, kname, kauschrift) values
(109, 'Hanna Barth', '75179 Pforzheim')

insert into bucht (bur, knr, zur, dauer, anreise) values
(404, 104, 301, 2, '26/07/2007')

Aufgabe 3

- 8) Update Zimmer Set tagespreis = tagespreis * 1.1 where
(hotel in (select hnr from Hotel where (hname = 'Flora')))
- 9) Create view Hotelzimmer as select hname as Hotel,
hanschrift as Anschrift, beschreibung as Zimmerbeschreibung,
tagespreis as Zimmerpreis from Hotel, Zimmer
where ((Zimmer.hotel = Hotel.hnr) and (hanschrift = 'Ulm'))
- 10) Create table bucht (bnr integer primary key, kur ~~integer~~
integer foreign key (kur) references Kunde, zur integer
foreign key (zur) references Zimmer, dauer integer,
anreise date, Check bnr > 401)
- 11) Create table TfreieZimmer (freieZimmer integer)
insert into TfreieZimmer values (520)
Create trigger t on bucht after insert as update
TfreieZimmer set (freieZimmer = freieZimmer - 1)
- 12) Create procedure p (@prozentsatz integer) as update
Zimmer set tagespreis = tagespreis * (1 +
Convert (Decimal (5, 2), @prozentsatz) / 100)
where beschreibung = 'E2'

~~13) Create view Wochenpreis as select zur, beschreibung, hotel, tagespreis, (tagespreis * 7)
as Wochenpreis where ((tagespreis * 7) < 600)
order by tagespreis desc~~

~~14) Create view KundenZimmer as select kname, zname, hname
from Kunde, Zimmer, Hotel
where ((Kunde.knr = bucht.knr) and (Zimmer.znr
= bucht.znr) and (Hotel.hnr = Zimmer.hotel))~~

Aufgabe 4

Select nachname, artikelname, (bestellmenge * artikelgan-
as bestellgewicht from Person, Bestellung, Artikel
where ((Person.nachname = Bestellung.nachname)
and (Bestellung.artikelname = Artikel.artikelname)
and ((nachname = 'Meier') or (nachname = 'Müller'))
order by nachname

Aufgabe 5

Die Programme sind beide verschiedenartige Login-
Formulare.

- Programm 1:

Mit der Eingabe von Text0 als kname und Text1
als kpassword in der where-Bedingung wird ein SQL-
Select-Befehl auf die Tabelle kunde durchgeführt und
im Recordset rs gespeichert. Wenn dieses gleich null
ist, wird "nicht OK" ausgegeben, ansonsten "OK".

- Programm 2:

Hier werden alle Einträge der Tabelle kunde in rs
gespeichert. Danach wird in einer Schleife vom Programm
überprüft, ob kname und kpassword zu einem Kunden
passen. Die Ausgabe ist gleich wie im 1. Programm:
Wenn der Kunde mit Name und Passwort nicht existiert,
wird "nicht OK" ausgegeben, ansonsten "OK".

24.07.08

Aufgabe 5

Bewertung:

Der Unterschied ist die Art der Übertragung der Daten aus der Datenbank an das Programm.

Beim ersten Programm müssen wenig Daten übertragen werden, sie werden schon mit SQL gefiltert. Beim 2. Programm werden sämtliche Kundendaten übertragen, die Überprüfung wird durch das Programm durchgeführt.

Das führt zu dem Problem, dass bei größeren Kundentabellen große Datenmengen über das Netzwerk übertragen werden müssen, was sehr langsam ist und die Performance des Programms verschlechtern kann.
ausserdem: lastverteilt bei der Suche

Aufgabe 6

- Class.forName(...) muss über der Connection stehen. Erst muss der Treiber angegeben werden.
- Anführungszeichen bei String q: (Select)
Muss heißen ... Where pur = " + cur.getText() + " ;
- Ausgaben: ^{225 DM} ~~229,00~~ für Eingabe 1 ^{pur ist Zahl}
~~57,00~~ DM für Eingabe 2
₅₆