

**Klausur Datenbanken**  
**Wintersemester 2004/2005**  
**Prof. Dr. Wolfgang May**  
**10. Februar 2004, 11-13 Uhr**  
**Bearbeitungszeit: 90 Minuten**

Vorname:

Nachname:

Matrikelnummer:

Bei der Klausur sind **keine Hilfsmittel** (Skripten, Taschenrechner, etc.) erlaubt. Handies müssen ausgeschaltet sein. Papier wird gestellt. Benutzen Sie nur die **ausgeteilten**, zusammengehefteten **Blätter** für Ihre Antworten. Schreiben Sie mit blauem/schwarzem Kugelschreiber, Füller, etc.; Bleistift ist nicht erlaubt.

Zum **Bestehen** der Klausur sind **45** Punkte hinreichend.

	Max. Punkte	Erreichte Punkte
Aufgabe 1 (ER-Modell, Relationales Modell)	24	
Aufgabe 2 (SQL)	20	
Aufgabe 3 (Relationale Algebra)	21	
Aufgabe 4 (Transaktionen)	25	
Summe	90	

**Note:**

Name:

MatNr.:

Die Klausur basiert auf dem folgenden "Bank"-Szenario:

- Kunden haben einen Namen und wohnen in einer Stadt (es gibt keine zwei Kunden gleichen Namens)
- Jeder Kunde unterhält *bis zu drei* Konten, jedes Konto hat eine Kontonummer.
- Auf jedem Konto ist ein bestimmter Geldbetrag,
- Manche Kunden besitzen auf ihren Konten außerdem Aktien von Firmen.
- Die Depotdaten enthalten zu jedem Aktienpaket den Namen der Firma, die Anzahl von Aktien, das Kaufdatum und den damaligen Kaufpreis pro Aktie. Ein Kunde kann Aktien einer Firma zu unterschiedlichen Zeitpunkten und unterschiedlichen Preisen gekauft haben - hier wird angenommen, dass ein Kunde Aktien nicht zweimal am selben Tag Aktien derselben Firma kauft.
- Für jede Firma ist ihr Tätigkeitsbereich und der aktuelle Aktienkurs gespeichert.

Die Datenbank hat das folgende Schema und enthält z.B. die folgenden Werte (die Tupel-Nummern werden in Aufgabe 4 benötigt):

Kunde		Konto		
Name	Stadt	Name	KNr	KStand
Meier	Kassel	Meier	93	7000
Müller	Kassel	Müller	72	15000
Schmidt	Hamburg	Schmidt	18	4.80
Kunze	Hannover	Kunze	42	-10000
		Kunze	13	3.50

Firma		
Name	Bereich	Tageskurs
VW	Auto	26.00
Daimler	Auto	35.00
GM	Auto	1.05
SAP	IT	130.00
Nestle	Ernährung	3500
DeutscheBk	Bank	60

enthält					
	Nummer	Firma	Kaufdatum	Kaufpreis	Anzahl
$e_1$	93	VW	23.2.2000	22.00	200
$e_2$	93	Daimler	25.2.2000	83.50	60
$e_3$	93	Daimler	1.12.2001	20.00	200
$e_4$	72	GM	4.5.1998	1.20	4000
$e_5$	72	SAP	20.6.2000	640.00	10
$e_6$	18	Nestle	20.2.1960	200	10
$e_7$	18	VW	1.8.1980	18.00	300
$e_8$	42	VW	15.6.1970	1.20	2500
$e_9$	13	VW	1.7.1990	30.00	150
$e_{10}$	13	DeutscheBk	3.10.1990	40.00	100
$e_{11}$	13	Nestle	20.3.1995	2800	2



Name:

MatNr.:

---

### Aufgabe 1 (ER-Modell, Relationales Modell [24 Punkte])

- Erstellen Sie ein ER-Diagramm für den obigen Sachverhalt (einschl. Schlüssel und Kardinalitäten) (15 P).
- Unterstreichen Sie die Schlüssel der jeweiligen Relationen (3 P):

Kunde(Name, Stadt)

Konto(Name, KNr, KStand)

Firma(Name, Bereich, Kurs)

enthält(Nummer, Firma, Kaufdatum, Kaufpreis, Anzahl)

- Überstreichen Sie in obiger Aufstellung die Fremdschlüsselattribute (2 P).
- Geben Sie zu der Tabelle “enthält” die Tabellendefinition (in SQL: CREATE TABLE ...) sowie die referentiellen Integritätsbedingungen (in SQL: FOREIGN KEY ... REFERENCES ...) an (2+2 P).  
(die Syntax muss nicht unbedingt SQL sein, aber es muss klar sein, was Sie meinen).



Name:

MatNr.:

---

**Aufgabe 2 (SQL [20 Punkte])**

1. Geben Sie eine SQL-Anfrage an, die die Namen aller Firmen ergibt, die Autos bauen (2P).
2. Geben Sie eine SQL-Anfrage an, die die Menge aller Paare (Kunde, Firma) ergibt, so dass der Kunde Aktien dieser Firma besitzt (3 P).
3. Geben Sie eine SQL-Anfrage an, die für jeden Kunden angibt, wieviele Aktien er insgesamt besitzt (5 P).

Hinweis: `group by knr` ist nicht korrekt, da ein Kunde mehrere Konten haben kann.

4. Geben Sie eine SQL-Anfrage an, die alle Paare (Kunde, Firma) angibt, so dass der Kunde Aktien der Firma hat, die er teurer gekauft hat als sie heute sind (5 P).
5. Geben Sie in natürlicher Sprache an, was die folgende Anfrage ergibt (3 P). Geben Sie weiterhin eine Zeile des Ergebnisses für die oben angegebene Beispieldatenbasis an (2 P).

```
select konto.name, sum(firma.tageskurs * enthaelt.anzahl)
from konto, enthaelt, firma
where konto.knr = enthaelt.nummer
      and enthaelt.firma = firma.name
group by konto.name
```



Name:

MatNr.:

---

**Aufgabe 3 (Relationale Algebra [21 Punkte])**

1. Geben Sie einen Algebra-Ausdruck oder -Baum an, der die Menge der Namen aller Kunden ergibt, die mindestens ein Konto mit negativem Kontostand haben (2 P).
2. Geben Sie einen Algebra-Ausdruck oder -Baum an, der die Menge der Namen aller Kunden ergibt, die Volkswagen-Aktien besitzen (4 P).
3. Geben Sie einen Algebra-Ausdruck oder -Baum an, der die Menge aller existierenden Kontonummern ergibt, die keine Automobilaktien enthalten (5 P).
4. Geben Sie einen Algebra-Ausdruck oder -Baum an, der die Namen aller Kunden angibt, die Aktien *aller in der Datenbank enthaltenen Automobilfirmen* besitzen (5 P).
5. Gegeben seien zwei nichtleere Relationen  $R(A, B)$  und  $S(B, C)$ . Ist für diese Relationen die Gleichung  $\pi[A, C](R \bowtie S) = (\pi[A](R)) \bowtie (\pi[C](S))$ 
  - gültig,
  - abhängig von den Inhalten der Relationen mal gültig, mal nicht gültig,
  - immer ungültig.

Begründen Sie Ihre Entscheidung (5 P).



**Aufgabe 4 (Transaktionen [25 Punkte])**

Sonntag abend, 13.1.2005: Schmidt muss eine Rechnung in Höhe von 10000 E bei Müller bezahlen. Dazu gibt er eine online-Überweisung für Montagmorgen in Auftrag. Da er schon kein Geld mehr auf dem Konto hat, entschließt er sich, Aktien zu verkaufen und vergibt einen online-Auftrag, am nächsten Morgen 3 Nestle-Aktien limitiert mit 3400 E zu verkaufen (da dies unter dem Tageskurs liegt, ist er sich ziemlich sicher, dass der Verkauf klappt). Müller, wissend dass er am nächsten Tag 10000 E von Schmidt überwiesen bekommt, und auch noch Geld auf dem Konto hat, will diese neu anlegen, und schickt einen Kaufauftrag für 3 Nestle-Aktien, limitiert auf 3400 E ab.

Wie der Zufall will, liegen am Montagmorgen keine weiteren Kauf- oder Verkaufsaufträge für Nestle vor, und es geschehen die folgenden Dinge (jeweils mit Angabe der entsprechenden Tupel in den Tabellen):

$T_1$  Überweisung von 10000 E vom Konto von Schmidt (Tupel  $k_3$ ) auf das Konto von Müller ( $k_2$ ).

$T_2$  Ausbuchung von 3 Nestle Aktien aus dem Depot von Schmidt ( $e_6$ ) und Gutschrift von 10200 E auf dem Konto von Schmidt ( $k_3$ ).

$T_3$  Einbuchung von 3 Nestle Aktien in das Depot von Müller ( $e_{12}$ , neues Tupel) und Abbuchung von 10200 E von dem Konto von Müller ( $k_2$ ).

... und jetzt zur Aufgabe:

1. Komplettieren Sie die Transaktionen  $T_1$  und  $T_2$ , die die jeweiligen Tupel  $k_2$ ,  $k_3$ , und  $e_6$  der Datenbank lesen/schreiben; die dritte Transaktion (die u.a. das Tupel  $e_{12}$  schreibt), ist angegeben (4 P):

- $T_1 =$
- $T_2 =$
- $T_3 = W_3 e_{12}; R_3 k_2; k_2 = k_2 - 10200; W_3 k_2$

2. Geben Sie die Tupel  $k_2$ ,  $k_3$ ,  $e_6$  und  $e_{12}$  nach korrekter Ausführung aller Transaktionen an (2 P).

3. Geben Sie einen beliebigen Schedule  $S$  an, der die drei Transaktionen korrekt ausführt. (2 P)

4. Geben Sie einen Schedule  $S'$  an, bei dem die drei Transaktionen nicht korrekt ausgeführt werden. Betrachten Sie dabei nur die Lese- und Schreibaktionen, geben Sie dabei jeweils die gelesenen/geschriebenen Werte an. (5 P)

5. Geben Sie den Dependency-Graphen Ihres Schedules  $S'$  an (berücksichtigen Sie dabei auch die Transaktionen  $T_0$  und  $T_\infty$  wie in der Vorlesung) (4 P).

6. Geben Sie den Konflikt-Graphen Ihres Schedules  $S'$  an (4 P).

7. Woran kann man erkennen, dass  $S'$  nicht serialisierbar ist? (1 P)

8. Zeigen Sie, dass unter Anwendung des 2-Phasen-Locking-Protokolls Ihr Schedule  $S'$  nicht zustandekommen kann. (3 P)

