

# **Klausur**

# **Grundzüge der Informatik III**

**Do, 19.02.2004**

# Aufgabe 1: Datentypen (10 Punkte)

- Was wird unter einem *Datentyp* und einem *Typkonstruktor* verstanden? Grenzen Sie die *atomaren*, *strukturierten* und *mengenwertigen* Datentypen voneinander ab.

# Aufgabe 1: Datentypen

## Lösung 1/3

- **Datentyp: (3 P.)**
  - möglicherweise unendliche Menge von Werten (Wertebereich) und
  - Menge von Operationen auf diesen Werten
  - eine Menge von Integritätsbedingungen, die das Arbeiten auf dem Datentyp reglementieren.
- **Typkonstruktor (type constructor): (1 P.)**
  - eine Art typunabhängige Schablone (Strukturbeschreibung), die genutzt werden kann, um strukturierte Datentypen zu erstellen
- **Atomarer (einfacher) Datentyp: (1 P.)**
  - Werte sind aus Sicht der Anwendung (Benutzer) nicht (weiter) zerlegbar

# Aufgabe 1: Datentypen

## Lösung 2/3

- **Strukturierter (zusammengesetzter) Datentyp: (3 P.)**
  - Ein Typkonstruktor und ein oder mehrere Datentypen (rekursiv) kombiniert ergeben einen strukturierten Datentyp. Werte eines strukturierten Datentyps sind aus gleichartigen oder unterschiedlichen Komponenten zusammengesetzt
  - Über den Komponenten liegt eine Struktur (die des Typkonstruktors), die eine (teilweise) Ordnung festlegen kann
  - Will ein Benutzer auf einen strukturierten Datentyp zugreifen, muss er dessen Struktur beim Zugriff berücksichtigen (D.h. es sind Operationen auf den elementaren Werten und Operationen zum Arbeiten auf der Datenstruktur notwendig)

# Aufgabe 1: Datentypen

## Lösung 3/3

- **Mengenwertiger Datentyp: (2 P.)**
  - ein strukturierter Datentyp, der durch (einmalige) Anwendung des Typkonstruktors Menge entstanden ist. Auf ihn treffen also alle Eigenschaften einer Menge zu, also insbesondere,
    - dass er unsortiert ist,
    - dass alle Elemente vom selben Typ sind und
    - dass er keine Duplikate enthält.

## **Aufgabe 2: Abarbeitung von SQL-Anfragen (12 Punkte)**

- In welcher Reihenfolge wird ein SELECT-FROM-WHERE-Block, erweitert um die HAVING, GROUP BY und ORDER BY-Klauseln abgearbeitet und was geschieht in jeder einzelnen Klausel (knappe, stichpunktartige Beschreibung genügt)?

# Aufgabe 2: Abarbeitung von SQL-Anfragen Lösung

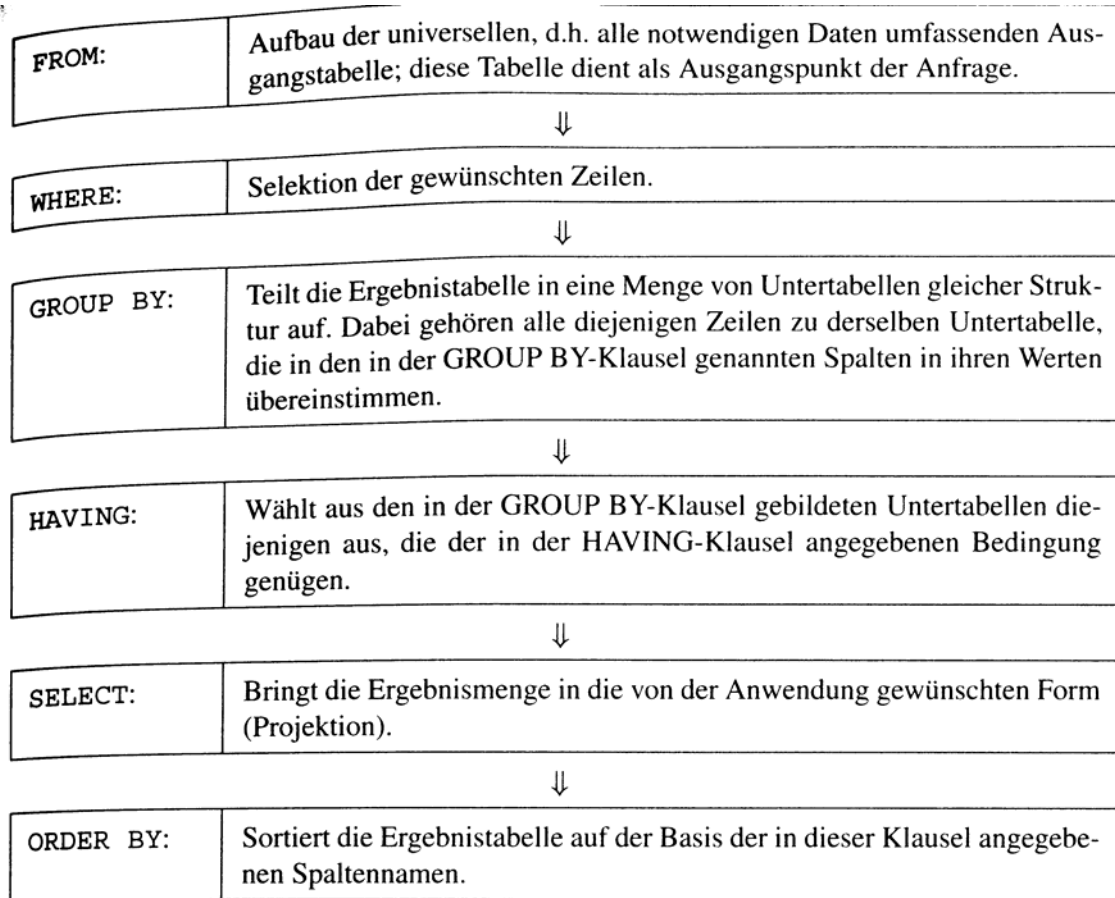


Abb. 7.7: Abarbeitungsreihenfolge und Semantik der von SQL unterstützten Basisklauseln

# Aufgabe 3: Relationale Algebra

## (32 Punkte)

1) Gegeben seien zwei Relationen r und s:

	$\frac{B \quad C \quad D}{4 \quad 5 \quad 9}$		$\frac{A \quad B \quad C}{1 \quad 2 \quad 3}$
r =	6 7 9	s =	1 4 7
	6 3 8		1 6 7
	4 7 8		

Berechnen Sie die Ergebnisrelationen folgender Ausdrücke:

a)  $\sigma_{C=7}(r)$

c)  $r \bowtie s$

b)  $r \bowtie_{(r.B < s.B)} s$

d)  $\pi_{A,D}(r \bowtie s)$



# Aufgabe 3: Relationale Algebra

## Lösung 1/4

- a)

<b>B</b>	<b>C</b>	<b>D</b>
<b>6</b>	<b>7</b>	<b>9</b>
<b>4</b>	<b>7</b>	<b>8</b>

# Aufgabe 3: Relationale Algebra

## Lösung 2/4

- b)

A	s.B	s.C	r.B	r.C	D
1	6	7	4	5	9
1	6	7	4	7	8

# Aufgabe 3: Relationale Algebra

## Lösung 3/4

- c)

A	B	C
1	4	7
1	6	7

# Aufgabe 3: Relationale Algebra

## Lösung 4/4

- d)

A	D
1	8
1	9

# Aufgabe 3: Relationale Algebra

- 2) Gegeben seien folgende Relationen:
  - Patient (Pat-Nr, Vorname, Nachname, Geburtsdatum, Diagnose)
  - Medikament (Med-Nr, Bezeichnung, Preis)
  - Verschreibung (Pat-Nr, Med-Nr, Menge, Dauer)
  - Formulieren Sie die folgenden Anfragen in relationaler Algebra:
    - Geben Sie Pat-Nr, Vor- und Nachnamen aller Patienten aus, die eine Erkältung haben.
    - Geben Sie Med-Nr und Bezeichnungen aller Medikamente aus, die gegen Erkältung verschrieben wurden und mehr als 20 Euro kosten.
    - Finden Sie alle Patienten, die mehr als 3 Tabletten Aspirin verschrieben bekommen haben und geben Sie für diese Patienten den Vor- und Nachnamen sowie die Diagnose aus.
    - Geben Sie Vor- und Nachnamen und Geburtsdatum aller Patienten aus, die keine Medikamente zum Preis von mehr als 30 Euro bekommen haben.

# Aufgabe 3: Relationale Algebra

## Lösung

- $\pi_{\text{Pat-Nr, Vorname, Nachname}} (\sigma_{\text{Diagnose}='Erkältung'} (\text{Patient}))$
- $\pi_{\text{Med-Nr, Bezeichnung}} (\sigma_{\text{Diagnose}='Erkältung' \wedge \text{Preis}>20} (\text{Medikament} \bowtie \text{Verschreibung} \bowtie \text{Patient}))$
- $\pi_{\text{Vorname, Nachname, Diagnose}} (\sigma_{\text{Bezeichnung}='Aspirin' \wedge \text{Menge}>3} (\text{Patient} \bowtie \text{Verschreibung} \bowtie \text{Medikament}))$
- $\pi_{\text{Vorname, Nachname, Geburtsdatum}} (\text{Patient} - (\text{Patient} \bowtie \text{Verschreibung} \bowtie \sigma_{\text{Preis}>30} \text{Medikament}))$

***Alternativen möglich !!!***

# Aufgabe 4: SQL (32 Punkte)

- Gegeben seien folgende Relationen:
    - Produkt (produktNr, produktName, preis, hersteller)
    - Filiale (filialeNr, filialeName, leiterName, ort, umsatz)
    - Kunde (kundenNr, kundenName, stadt)
    - Bestellung (kundenNr, filialeNr, datum, produktNr, anzahl)
- 1) Geben Sie die folgenden SQL-Anweisungen an:

# Aufgabe 4: SQL

- Erzeugen Sie die Tabellen *Produkt* und *Bestellung*. Wählen Sie geeignete Datentypen und berücksichtigen Sie die Schlüsselattribute.



# Aufgabe 4: SQL

- **2 Punkte**

```
CREATE TABLE Produkt(  
    produktNr INTEGER PRIMARY KEY,  
    produktName VARCHAR(20),  
    preis NUMERIC(6,2),  
    hersteller VARCHAR(20) );
```

**2 Punkte**

```
CREATE TABLE Bestellung(  
    kundenNr INTEGER,  
    filialeNr INTEGER,  
    datum DATE,  
    produktNr INTEGER,  
    anzahl INTEGER,
```

**4 Punkte**

```
    CONSTRAINT RechnungPK  
        PRIMARY KEY (kundenNr, filialeNr, datum),  
    CONSTRAINT kundeFK  
        FOREIGN KEY (kundenNr) REFERENCES Kunde,  
    CONSTRAINT filialeFK  
        FOREIGN KEY (filialeNr) REFERENCES Filiale,  
    CONSTRAINT produktFK  
        FOREIGN KEY (produktNr) REFERENCES Produkt );
```

## Aufgabe 4: SQL

- Fügen Sie ein Mobiltelefon von Siemens mit dem Namen SL55 in die Produkt-Relation ein. Er soll 800 Euro kosten und die Produktnummer 7 haben. **(2 Punkte)**

```
INSERT INTO Produkt VALUES(7, 'SL55',  
800.00, 'Siemens');
```

# Aufgabe 4: SQL

- Ändern Sie den Preis dieses Telefons auf 500 Euro. **(2 Punkte)**

```
UPDATE Produkt SET preis=500.00  
WHERE ProduktNr=7;
```

# Aufgabe 4: SQL

Erzeugen Sie eine Sicht, die die Nummern und Namen aller Filialen liefert, deren Leiter gleichzeitig Kunden des Unternehmens sind. (Es wird angenommen, dass Filialenleiter- bzw. Kundennamen die Personen eindeutig identifizieren). Verwenden Sie dabei eine mengenorientierte Unteranfrage. **(4 Punkte)**

```
CREATE VIEW FilialenleiterKunde AS
SELECT filialeNr, filialeName
FROM Filiale
WHERE leiterName IN (SELECT kundenName
                     FROM Kunde);
```

# Aufgabe 4: SQL

Formulieren Sie die folgenden SQL-Anfragen unter Verwendung von geeigneten Aggregatfunktionen:

## Aufgabe 4: SQL

Geben Sie den durchschnittlichen Preis aller Produkte vom Hersteller „Siemens“ an. **(4 Punkte)**

```
SELECT AVG(preis) AS  
durchschnittspreis  
FROM Produkt  
WHERE hersteller = 'Siemens';
```

## Aufgabe 4: SQL

Geben Sie den Gesamtumsatz aller Filialen an, die sich in Essen befinden.

**(4 Punkte)**

```
SELECT SUM(umsatz) As UmsatzEssen  
FROM Filiale  
WHERE ort='Essen';
```

## Aufgabe 4: SQL

Geben Sie für jedes Produkt seine Nummer, den Namen und die Anzahl der Kunden an, die dieses Produkt bestellt haben. **(4 Punkte)**

```
SELECT produktNr, produktName,  
COUNT(DISTINCT kundenNr) AS  
kundenAnzahl  
FROM Produkt NATURAL JOIN  
Bestellung  
GROUP BY ProduktNr;
```



## Aufgabe 4: SQL

Geben Sie für alle Städte, aus denen jeweils mehr als 100 Kunden stammen, den Stadtnamen und die Anzahl der dort ansässigen Kunden an. **(4 Punkte)**

```
SELECT stadt, COUNT(DISTINCT
kundenNr) AS kundenAnzahl
FROM Kunde
GROUP BY stadt
HAVING COUNT(DISTINCT kundenNr)
> 100;
```

# Aufgabe 5:

## Synchronisationsverfahren

### (14 Punkte)

- Gegeben sei das folgende Schedule:
- R5(y) R3(x) W2(z) R1(x) W1(x) R2(x) R4(z)  
W3(z) R4(y) W5(y) W3(y) R1(z)

1)

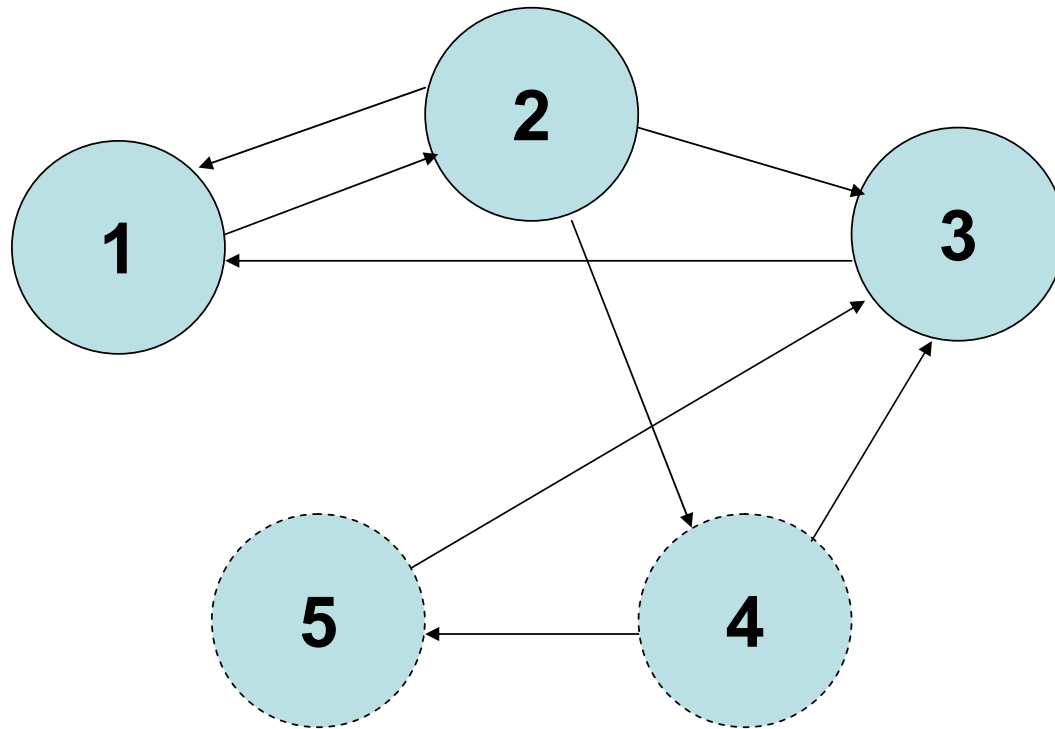
- a) Erstellen Sie für dieses Schedule den Abhängigkeitsgraphen.
- b) Ist sie serialisierbar (begründen Sie Ihre Antwort)?  
Geben Sie wenn möglich die äquivalente serielle Schedule an.
- c) Hat dieses Schedule wirkungslose Transaktionen?  
Nennen Sie alle wirkungslosen Transaktionen, falls vorhanden. Wäre das Schedule serialisierbar, falls wirkungslose Transaktionen ausgelassen würden?

# Aufgabe 5:

## Synchronisationsverfahren

### Lösung

a)



# **Aufgabe 5:**

## **Synchronisationsverfahren**

### **Lösung**

b) Nicht serialisierbar

c) T4 und T5 sind wirkungslos. Immer noch nicht serialisierbar.

# Aufgabe 5:

## Synchronisationsverfahren

2) Welche (um die Sperren erweiterte) Ergebnisschedule würden von dem strikten Zweiphasensperrprotokoll erzeugt?

R5(y) R3(x) W2(z) R1(x) W1(x) R2(x) R4(z)  
W3(z) R4(y) W5(y) W3(y) R1(z)

		<b>x</b>	<b>y</b>	<b>z</b>
<b>R<sub>5</sub>(y)</b>			<b>rl<sub>5</sub>(y)</b> <b>R<sub>5</sub>(y)</b>	
<b>R<sub>3</sub>(x)</b>		<b>rl<sub>3</sub>(x)</b> <b>R<sub>3</sub>(x)</b>		
<b>W<sub>2</sub>(z)</b>				<b>wl<sub>2</sub>(z)</b> <b>W<sub>2</sub>(z)</b>
<b>R<sub>1</sub>(x)</b>		<b>rl<sub>1</sub>(x)</b> <b>R<sub>1</sub>(x)</b>		
<b>[W<sub>1</sub>(x)]</b>	<b>W<sub>1</sub>(x)</b>			
<b><u>R<sub>2</sub></u>(x)</b>	<b>W<sub>1</sub>(x)</b>	<b>rl<sub>2</sub>(x)</b> <b>R<sub>2</sub>(x)</b> <b>ul<sub>2</sub>(x)</b>		<b>ul<sub>2</sub>(z)</b>
<b>R<sub>4</sub>(z)</b>	<b>W<sub>1</sub>(x)</b>			<b>rl<sub>4</sub>(z)</b> <b>R<sub>4</sub>(z)</b>
<b>[W<sub>3</sub>(z)]</b>	<b>W<sub>1</sub>(x),</b> <b>W<sub>3</sub>(z)</b>			

$\underline{R_4(y)}$	$W_1(x),$ $W_3(z)$		$rl_4(y)$ $R_4(y)$ $ul_4(y)$	$ul_4(z)$
$> W_3(z)$	$W_1(x)$			$wl_3(z)$ $W_3(z)$
$\underline{W_5(y)}$	$W_1(x)$		$wl_5(y)$ $W_5(y)$ $ul_5(y)$	
$\underline{W_3(y)}$	$W_1(x)$	$ul_3(x)$	$wl_3(y)$ $W_3(y)$ $ul_3(y)$	$ul_3(z)$
$> W_1(x)$		$wl_1(x)$ $W_1(x)$		
$\underline{R_1(z)}$		$ul_1(x)$		$rl_1(z)$ $R_1(z)$ $ul_1(z)$