

# 1 Thema

## Teilaufgabe 1

### Aufgabe 1

- (a)
- (b)

### Aufgabe 2

- (a)
- (b)
- (c)
- (d)
- (e)
- (f)
- (g)
- (h)

### Aufgabe 3

- (a)
- (b)
- (c)
  - i.
  - ii.
  - iii.
- (a)
- (b)

#### Aufgabe 4

- (a) • **Wachstumsphase/Sperrphase**  
 Sperren werden nur gesetzt, aber dürfen nicht freigegeben werden.
- **Schrumpfphase/Freigabephase**  
 Sperren werden nur freigegeben, aber dürfen nicht angefordert werden.

*Zwei-Phasen-Sperrprotokoll:*

Bedingung an Transaktionen, die Konfliktserialisierbarkeit garantieren.

Alle Sperranforderungen geschehen vor allen Sperrfreigaben.

- (b) Beim *strikten* Zweiphasen-Sperrprotokoll dürfen Sperren erst am Ende der Transaktion freigegeben werden. Dadurch hat ein Abbruch einer Transaktion keine Auswirkungen mehr auf andere Transaktionen. Ein fortgeplanzter Rollback ist also nicht möglich.

Zur Notation:

$\text{lockS}(A)$  := Lesesperre für Datum A

$\text{lockX}(A)$  := Schreibsperre für Datum A

$\text{unlockS}(A)$  := Freigabe der Lesesperre für Datum A

$\text{unlockX}(A)$  := Freigabe Schreibsperre für Datum A

$T_1$	$T_2$	$T_3$	Bemerkung
	<b>BOT</b> $\text{lockX}(x)$ $r_2(x)$		
<b>BOT</b> $\text{lockX}(y)$ $w_1(y)$			
		<b>BOT</b> $\text{lockS}(z)$ $r_3(z)$	
$\text{lockS}(x)$	$\text{lockX}(y)$	$\text{lockX}(x)$	$T_1$ muss warten auf $T_2$ $T_2$ muss warten auf $T_1$ $T_3$ muss warten auf $T_1$

## **Teilaufgabe 2**

### **Aufgabe 1**

(a)

(b)

### **Aufgabe 2**

### **Aufgabe 3**

(a)

(b)

### **Aufgabe 4**

(a)

(b)

### **Aufgabe 5**

## 2 Thema

### Teilaufgabe 1

#### Aufgabe 1

- (a)
- **Atomarität**  
Atomarität bedeutet, dass eine Transaktion immer ganz ausgeführt wird oder gar nicht. Wird sie abgebrochen, so bleibt das System unverändert. Transaktionen sind also unteilbar (atomar).
  - **Isolation**  
Isolation bedeutet, dass sich bei gleichzeitiger Ausführung mehrerer Transaktionen dürfen diese nicht gegenseitig beeinflussen dürfen.
- (b) Bei Vorliegen *physischer* Datenunabhängigkeit eines Datenbanksystems hat eine Änderung an der physischen Speicher- und Zugriffsstruktur keine Auswirkungen auf die logische Struktur des Datenbankschemas.  
Zur Abgrenzung:  
Bei Vorliegen einer *logischen* Datenunabhängigkeit haben Änderungen am Datenbankschema keine Auswirkung auf die Funktionalität der Anwendungen.
- (c) Das ANSI-3-Ebenenmodell besteht aus einer *externen* Ebene, einer *konzeptionellen* Ebene und einer *internen* Ebene:
- **Externe Ebene**  
Benutzersicht auf die Daten.
  - **Konzeptionelle Ebene**  
Logische Gesamtsicht auf Daten und deren Strukturen.
  - **Interne Ebene**  
Physische Sicht auf die Daten.
- (d) Der Begriff *Schlüssel* findet in der Informatik mehrfach Anwendung. In Bezug auf Datenbanksysteme versteht man darunter eine Kombination aus Attributen einer Relation, mittels der sich die Tupel der Relation eindeutig identifizieren lassen. Dabei unterscheidet man unter anderem zwischen den Begriffen *Superschlüssel*, *Schlüsselkandidat* und *Primärschlüssel*:
- **Superschlüssel**  
Bezeichnet eine Attributkombination, von der *alle* Attribute einer Relation funktional abhängen.
  - **Schlüsselkandidat**  
Bezeichnet einen minimalen Superschlüssel. Somit ist keine Teilmenge des

Schlüsselkandidaten ein Superschlüssel.

- **Primärschlüssel**

Bezeichnet den ausgewählten Schlüsselkandidaten (falls mehrere zur Auswahl stehen).

(e) Da  $A$  Schlüsselkandidat ist, ist  $A$  insbesondere auch Superschlüssel. Jede Attributkombination, die  $A$  enthält, ist also wieder Superschlüssel der Relation. Die möglichen Superschlüssel der Relation  $R$  sind also  $\{A\}$ ,  $\{A, B\}$ ,  $\{A, C\}$  und  $\{A, B, C\}$ .

- (f)
- 0.  
Nämlich dann, wenn es keine übereinstimmenden Attributwerte in den Spalten gibt, oder das Selektionsprädikat immer unwahr ist.
  - 63 (=  $9 * 7$ ).  
Nämlich dann, wenn die Relationen in den Spalten überall übereinstimmende Attributwert aufweisen, und das Selektionsprädikat immer wahr ist.
  - 1.  
Nämlich dann, wenn in Relation  $R$  in jedem Tupel das Attribut  $A$  denselben Attributwert hat (auch NULL).
  - 9.  
Nämlich dann, wenn in  $R$  in jedem Tupel paarweise verschiedene Attributwerte für  $A$  vorliegen.

(g) Ein Attribut  $A$  heißt *prim* in einer Relation  $R$  genau dann, wenn  $A$  Teil eines Schlüsselkandidaten von  $R$  ist. Ein primes Attribut ist damit stets ein Schlüsselattribut.

(h)

- **Reflexivität**

$$\beta \subseteq \alpha \Rightarrow \alpha \rightarrow \beta$$

(ist  $\beta$  Teilmenge von  $\alpha$ , so ist  $\beta$  funktional abhängig von  $\alpha$ )

- **Verstärkung**

$$(\alpha \rightarrow \beta) \Rightarrow (\alpha\gamma \rightarrow \beta\gamma)$$

(ist  $\beta$  funktional abhängig von  $\alpha$ , so ist auch  $\beta\gamma$  funktional abhängig von  $\alpha\gamma$  für beliebige  $\gamma$ )

- **Transitivität**

$$(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \gamma) \Rightarrow (\alpha \rightarrow \gamma)$$

(ist  $\beta$  funktional abhängig von  $\alpha$  und  $\gamma$  funktional abhängig von  $\beta$ , so ist  $\gamma$  auch funktional abhängig von  $\alpha$ )

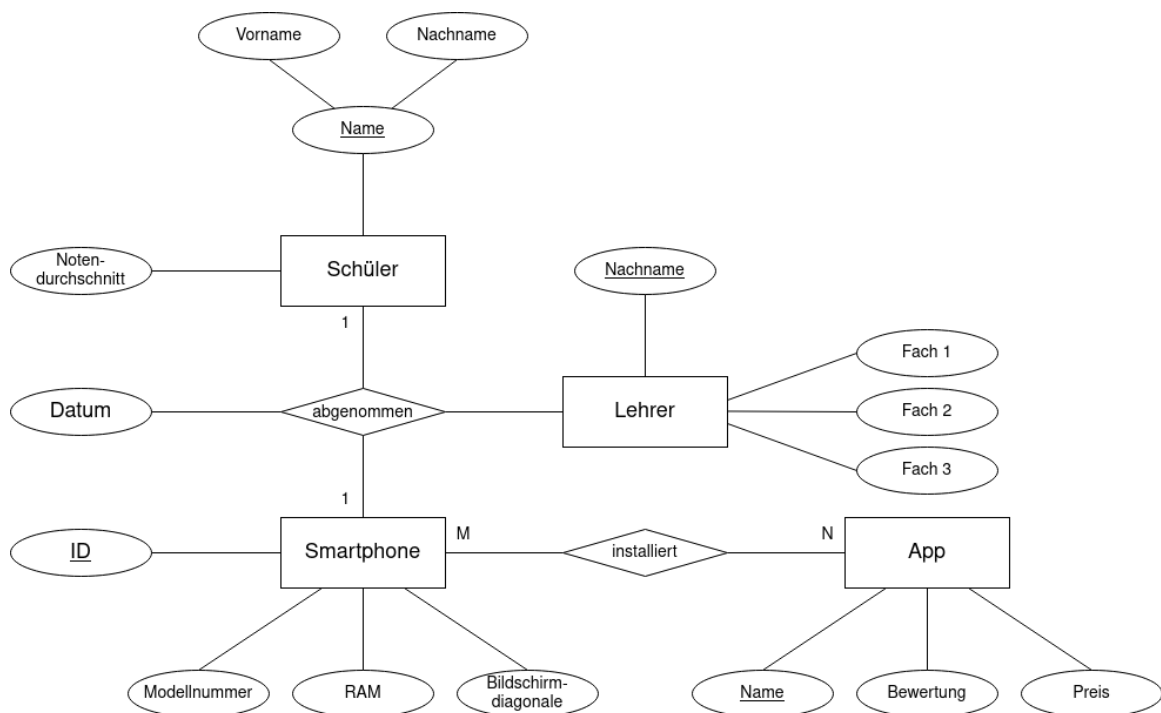
- (i) Aus  $A$  folgt  $C$  und aus  $A$  folgt auch  $D$ , aber weder aus  $A$  noch aus irgendeiner anderen vorliegenden Abhängigkeit folgt  $B$ , deshalb muss  $B$  mit aufgenommen werden in den Schlüsselkandidaten; der einzig mögliche Schlüsselkandidat lautet also  $AB$ . Nun liegen aber partielle Abhängigkeiten vor, es gibt also Nicht-Schlüsselattribute, die voll funktional abhängig sind von lediglich einer *Teilmenge* des Schlüsselkandidaten; nämlich zum Beispiel  $A \rightarrow C$ .

Die Relation erfüllt die Voraussetzung für die 1NF, indem alle Werte atomar vorliegen, verletzt jedoch die 2NF, indem partielle Abhängigkeiten vorliegen. Somit liegt die Relation vor in 1NF.

- (j) Es gilt: Schedule  $S$  seriell  $\Rightarrow$  Schedule  $S$  serialisierbar  
(jede serielle Schedule ist also trivialerweise serialisierbar)

Wir suchen also die Anzahl der möglichen seriellen Schedules. Man hat drei Möglichkeiten für die zuerst auszuführende Transaktion, zwei für die zweite und eine für die letzte Transaktion, also mindestens  $3 \cdot 2 \cdot 1 = 6$  serialisierbare Schedules bei drei Transaktionen.

## Aufgabe 2



Das ER-Diagramm

### Aufgabe 3

- (a) Wir betrachten zunächst einmal die Selektion  $\sigma_{A=2}(X)$  auf die Relation  $X$ :

A	B	C
2	3	4
2	3	2
2	2	1

$\sigma_{A=2}(X)$

B	C	D
2	3	1
2	1	3
3	2	3
2	2	3
1	3	2

$Y$

Anschließend führen wir den Natural Join ( $\bowtie$ ) durch. Beim Natural Join werden gleichnamige Attribute auf Gleichheit der Attributwerte geprüft und in eine neue Relation geschrieben. Das entspricht einer Selektion wie beim Gleichverbund, nur dass beim natürlichen Verbund *alle* übereinstimmenden Attribute auf Gleichheit überprüft werden. In diesem Fall also die Attribute  $B$  und  $C$ . Es kommen also nur diejenigen Tupel aus den Relationen  $\sigma_{A=2}(X)$  und  $Y$  in die neue Relation, die sowohl in den Attributwerten von  $B$  als auch  $C$  übereinstimmen. Die relevanten Tupel wurden in den obigen Relationen farbig markiert.

Insgesamt ergibt sich also:

A	B	C	D
2	3	2	3
2	2	1	3

$\sigma_{A=2}(X) \bowtie Y$

#### Hinweis

Bei einem Natural Join tauchen bei der Ergebnisrelation keine gleichnamigen Attribute auf. Die Attribute  $C$  und  $D$  dürfen also nicht doppelt vorkommen. Anders wäre dies, wenn man erst das Kreuzprodukt bildet und danach eine Selektion durchführt.

- (b) Auch hier teilen wir den Ausdruck auf in Teilausdrücke. Zunächst wenden wir die Projektionen auf beide Relationen an:

Nun wenden wir die Differenz an.

Zur Erinnerung:  $R - S = R \setminus S := \{t \mid t \in R \wedge t \notin S\}$ .

Das heißt also, dass in die neue Relation nur diejenigen Tupel aufgenommen

B	C
2	3
3	4
3	3
2	2
3	2
3	2
1	3
2	1
1	1

$\pi_{B,C}(X)$

B	C
2	3
1	1
2	1
3	2
2	2
1	3

$\pi_{B,C}(Y)$

werden, für die in der jeweils anderen Relation kein identisches Tupel existiert. Wir markieren die Tupel gelb, die in beiden Relationen vorkommen. Die neue Relation  $\pi_{B,C}(X) - \pi_{B,C}(Y)$  besteht also aus den nicht farblich markierten Tupeln der Relation  $\pi_{B,C}(X)$ .



B	C
3	4
3	3

$$\pi_{B,C}(X) - \pi_{B,C}(Y)$$

A	B	C
1	2	3
2	3	4
4	3	3
1	2	2
2	3	2
3	3	2
3	1	2
2	2	1
1	!	1

$X$

Nun wenden wir die Verbundoperation an auf die verkleinerte  $X$ -Relation mit der (nicht verkleinerten) Relation  $X$ . Da die Relation  $\pi_{B,C}(X) - \pi_{B,C}(Y)$  aus einer Teilmenge der Attribute der Relation  $X$  besteht, werden beim Verbund nur die noch fehlenden Attribute mit den Attributwerten hinzugefügt; in diesem Fall das Attribut  $A$ .

Insgesamt ergibt sich also:

A	B	C
2	3	4
4	3	3
3	1	2

$$(\pi_{B,C}(X) - \pi_{B,C}(Y)) \bowtie X$$

(c) Wenden wir zunächst die Projektion  $\pi$  an auf  $X$  und die Selektion  $\sigma$  auf  $Y$ :

A
1
2
4
1
2
3
3
2
1

$$\pi_A(X)$$

B	C	D
1	1	3
2	1	3

$$\sigma_{C=1}(Y)$$

Nun können wir die Verbundoperation ausführen.

Zur Erinnerung:  $A \bowtie_P B$  bedeutet eine Hintereinanderausführung von:

1. Kartesischem Produkt aus den Relationen  $A$  und  $B$  und
2. Restriktion nach Prädikat  $P$ .

### Schritt 1

Zunächst bilden wir das kartesische Produkt aus  $\pi_A(X)$  und  $\sigma_{C=1}(Y)$ :

A	B	C	D
1	1	1	3
2	1	1	3
4	1	1	3
1	1	1	3
2	1	1	3
3	1	1	3
3	1	1	3
2	1	1	3
1	1	1	3
1	2	1	3
2	2	1	3
4	2	1	3
1	2	1	3
2	2	1	3
3	2	1	3
3	2	1	3
2	2	1	3
1	2	1	3

$$\pi_A(X) \times \sigma_{C=1}(Y)$$

### Schritt 2

Nun wenden wir die Restriktion nach Prädikat  $A > P$  an auf  $\pi_A(X) \times \sigma_{C=1}(Y)$  und sind fertig. Die Zellen, auf die das Prädikat zutrifft, sind in obiger Relation bereits farbig markiert.

A	B	C	D
4	1	1	3
4	2	1	3

$$\pi_{A=2}(X) \bowtie_{A>D} \sigma_{C=1}(Y)$$

### (d) Division

Die Division kann man sich vorstellen als Gegenoperation (oder Umkehroperation) zum Kartesischen Produkt.  $R \div S$  enthält diejenigen Attribute aus  $R$ , die in  $S$  nicht vorkommen und diejenigen Attributwertkombinationen, bei denen für jede Zeile aus  $S$  eine entsprechende Zeile in  $R$  vorkommt.

Wir bilden zunächst  $\pi_{A,C}(X)$  und  $\pi_C(Y)$ :

A	C
1	3
2	4
4	3
1	2
2	2
3	2
3	2
2	1
1	1

$\pi_{A,C}(X)$

C
3
1
1
2
3
3

$\pi_C(Y)$

Die Relation  $\pi_C(Y)$  besteht einzig aus dem Attribut  $C$ , und dessen Attributwertmenge ist  $\{1, 2, 3\}$ . Für die Division suchen wir also diejenigen Attributwerte von  $A$  in  $\pi_{A,C}(X)$ , die als Partner Werte aus obiger Menge (also 1, 2 und 3) aufweisen. Diese wurden farbig markiert.

Insgesamt erhalten wir also:

A
1

$\pi_{A,C}(X) \div \pi_C(Y)$

#### Aufgabe 4

- (a)
- (b)
- (c)
- (d)

## Aufgabe 5

(a)

---

```
1 CREATE TABLE App (  
2     Name VARCHAR(255) PRIMARY KEY,  
3     Bewertung FLOAT,  
4     Preis FLOAT,  
5     CHECK (Bewertung >= 0.0 AND Bewertung <= 5.0)  
6 );
```

---

Damit der Wertebereich von 'Bewertung' gesichert ist, wurde eine CHECK-Einschränkung eingebaut. Werden nun Tupel hinzugefügt, deren Bewertung außerhalb des Intervalls [0,5] liegen, so wird dies abgewiesen mit der Fehlermeldung „Check constraint is violated“.

(b)

---

```
1 CREATE TABLE Installiert (  
2     ID INT REFERENCES Smartphone(ID),  
3     Name VARCHAR(255) REFERENCES App(Name),  
4     PRIMARY KEY (ID, Name)  
5 );
```

---

(c)

---

```
1 SELECT DISTINCT i.Name, a.Bewertung  
2 FROM Lehrer l, Eingesammelt e, Installiert i, App a  
3 WHERE l.Name = e.Name AND  
4       e.ID = i.ID AND  
5       i.Name = a.Name AND  
6       l.Name = 'Keating'  
7 ORDER BY a.Bewertung DESC;
```

---

Alternativ mit JOINS:

---

```
1 SELECT DISTINCT i.Name, a.Bewertung  
2 FROM Lehrer l JOIN Eingesammelt e ON l.Name =  
3     e.Name  
4     JOIN Installiert i ON e.ID = i.ID  
5     JOIN App a ON i.Name = a.Name  
6 WHERE l.Name = 'Keating'  
7 ORDER BY a.Bewertung DESC;
```

---

(d)

---

```
1  SELECT AVG(s.Notenschnitt)
2  FROM Schueler s, Eingesammelt e, Smartphone h
3  WHERE s.Vorname = e.Vorname AND
4         s.Nachname = e.Nachname AND
5         e.ID      = h.ID      AND
6         (h.Modellnr = 'A1633' OR h.Modellnr = 'A1688');
```

---

(e) *Tipp: Die Verwendung von Views kann die Aufgabe vereinfachen.*

Wir erstellen zunächst einen View 'AppAnzahl', der uns die Modellnummern der Smartphones mit der Anzahl der installierten Apps ausgibt - absteigend sortiert nach Anzahl der Apps:

---

```
1  CREATE VIEW AnzshlApps
2  AS
3  SELECT Modellnr
4  FROM Smartphone s JOIN Installiert i ON s.ID = i.ID
5  GROUP BY s.ID
```

---

Nun können wir den View verwenden um die Modellnummern derjenigen Smartphones auszugeben, die die meisten Apps haben.

Achtung: Dies können unter Umständen mehrere sein. Deshalb ist ein Abfrage mittels LIMIT(1) nicht sinnvoll.

---

```
1  SELECT ModellNr
2  FROM AppAnzahl
3  WHERE AnzahlApps >= ALL(SELECT AnzahlApps FROM
                           AppAnzahl);
```

---

## Aufgabe 6

(a) Da die Attribute  $C$  und  $D$  von keiner Attributmenge funktional abhängig sind (zu erkennen daran, dass sie auf keiner rechten Seite vorkommen), müssen sie Teil des Schlüsselkandidaten sein. Wir bestimmen die Attributhülle von  $CD$ :

$$CD^+ = \{A, B, C, D, E, F\}$$

Da die Attributhülle von  $CD$  alle Attribute der Relation  $R$  enthält, ist  $CD$  Schlüsselkandidat.

Es kann auch keine weiteren Schlüsselkandidaten geben, denn:

Angenommen es gäbe einen weiteren Schlüsselkandidaten, dann müsste dieser die Attribute  $C$  und  $D$  enthalten (siehe oben) sowie ein weiteres Attribut  $X \in \{A, B, E, F\}$ . Dies widerspricht aber der Minimalität eines Schlüsselkandidaten, da schon eine Teilmenge des Schlüssels  $XCD$  die Eigenschaft erfüllt.

$CD$  ist also der einzig mögliche Schlüsselkandidat der Relation  $R$ .

- (b) Wir bezeichnen im folgenden die Menge der funktionalen Abhängigkeiten mit „FA“.

### Schritt 1: Kanonische Überdeckung $FA_c$

Dazu gehen wir wie folgt vor:

- Linksreduktion und Rechtsreduktion der FDs.
- Entfernung FDs der Form  $\alpha \rightarrow \emptyset$ .
- Zusammenfassung FDs mit gleichen linken Seiten.

Linksreduktion	Rechtsreduktion	Zusammenfassung
$B \rightarrow F$	$B \rightarrow F$	$B \rightarrow F$
$CD \rightarrow E$	$CD \rightarrow E$	$CD \rightarrow E$
$C \rightarrow A$	$C \rightarrow A$	$C \rightarrow A$
$C\emptyset \rightarrow A$	$D \rightarrow \cancel{F}$	$D \rightarrow B$
$D \rightarrow F$	$D \rightarrow B$	
$D \rightarrow B$		

Zur Linksreduktion:  $D$  wird entfernt, da  $A$  schon enthalten ist in der Attributhülle von  $C$ .

Zur Rechtsreduktion:  $F$  wird entfernt, da  $D \rightarrow B \rightarrow F$ .

### Schritt 2: Erzeuge Relationen

Wir erzeugen aus jeder funktionalen Abhängigkeit  $F_i$  in  $FA_c$  eine eigene Relation  $R_i$ :

$R_1(\underline{B}, F),$   
 $R_2(\underline{C}, \underline{D}, E),$   
 $R_3(\underline{C}, A),$   
 $R_4(\underline{D}, B)$

### Schritt 3

Da der Schlüsselkandidat schon in einer der Relationen ( $R_2$ ) vollständig enthalten ist, ist hier nichts weiter zu tun.

Das Relationenschema aus Schritt 2 erfüllt also nach dem Synthesealgorithmus die 3NF.

## **Teilaufgabe 2**

### **Aufgabe 1**

- (a)
- (b)
- (c)
- (d)
- (e)

### **Aufgabe 2**

### **Aufgabe 3**

- (a)
- (b)

### **Aufgabe 4**

- (a)
- (b)
- (c)
- (d)
- (e)