

Probeklausur

Datenbanksysteme

Hinweise zur Probeklausur Die Probeklausur wird in der Vorlesung am **12.02.2016** besprochen. Die in der Probeklausur enthaltenen Aufgaben sind lediglich Beispiel für mögliche Klausuraufgaben. Dies bedeutet insbesondere, dass sowohl in der Klausur Aufgabentypen vorkommen können, die in der Probeklausur nicht enthalten sind, als auch umgekehrt nicht alle Aufgabentypen aus der Probeklausur in der Klausur vorkommen.

Die Probeklausur wird nicht korrigiert, sie muss deshalb auch nicht abgegeben werden.

Zur Selbstkontrolle: Die Probeklausur entspricht in ihrem Umfang grob einer regulären Klausur, ist also auf etwa 90 Minuten ausgelegt.

Hinweise zu diesem Lösungsvorschlag Diese Lösung ist nur eine beispielhafte Lösung. Es wird weder der Anspruch erhoben, dass die angegebenen Lösungen die einzigen noch die besten Lösungen sind. Weiterhin garantieren wir nicht, dass die hier vorgestellten Lösungen fehlerfrei sind. Ein entsprechender Anspruch auf entsprechende Korrektur in einer Prüfung gilt daher nicht.

Aufgabe 1 *ER-Modellierung* (10 Punkte)

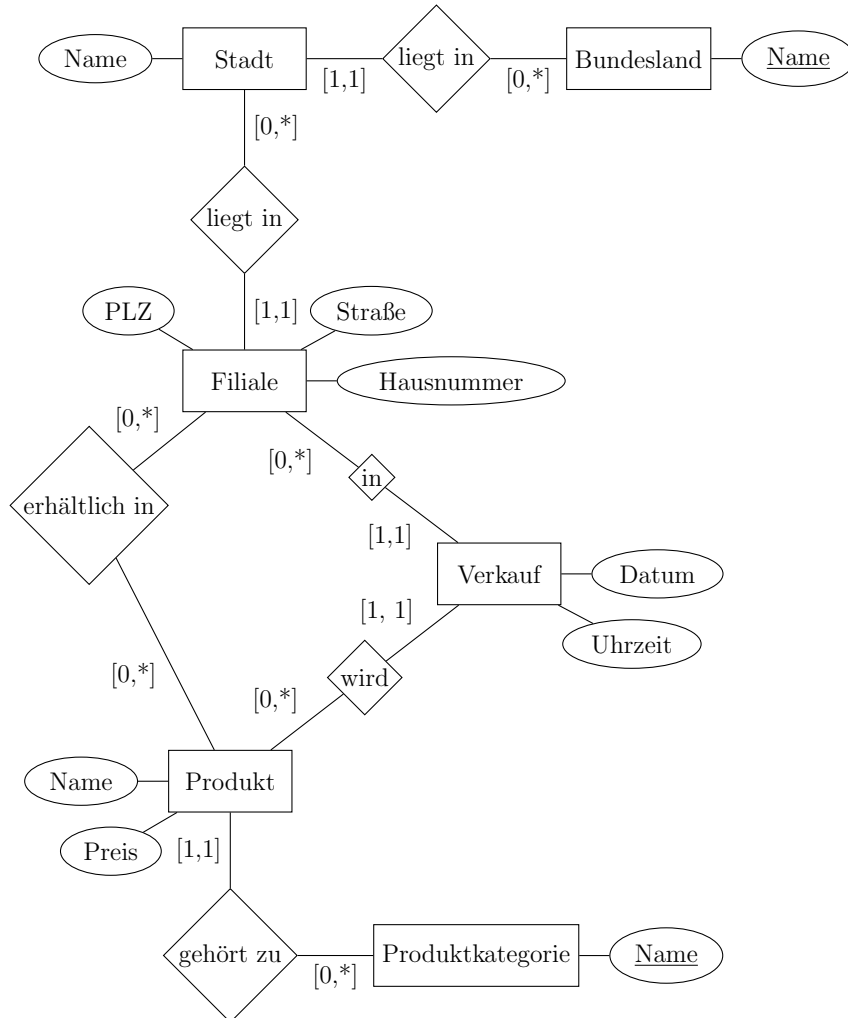
Entwerfen Sie ein ER-Modell zur Verwaltung der Supermarktkette „Centy“. Dabei sollen die folgenden Sachverhalte modelliert werden:

- Die Datenbank soll nach Bundesländern und Städten sortiert sein. Dabei liegt eine Stadt in einem Bundesland.
- Filialen liegen in Städten. Außerdem soll zu jeder Filiale die Adresse, bestehend aus Straße, Hausnummer und PLZ gespeichert werden.
- Es wird mit Produkten aus verschiedenen Produktkategorien gehandelt. So ist beispielsweise „Gouda mittelalt“ ein Lebensmittel und „Taschentücher extra wolzig“ sind eine Haushaltsware. Die Zuordnung von Produkten zu Kategorien soll in der Datenbank möglich sein, die Anzahl der Kategorien ist nicht fix.
- Weiterhin soll zu einem Produkt der Produktname und der Preis des Produktes gespeichert werden.
- Es sind nicht alle Produkte in allen Filialen erhältlich. Deshalb soll die Verfügbarkeit von Produkten in der Datenbank modelliert sein.

- Schließlich soll erfasst werden, welche Produkte an welchem Tag zu welcher Uhrzeit in welcher Filiale verkauft wurden.

Markieren Sie natürliche Schlüssel, falls diese vorhanden sind, durch unterstreichen und geben Sie die Kardinalitäten in Intervallschreibweise an.

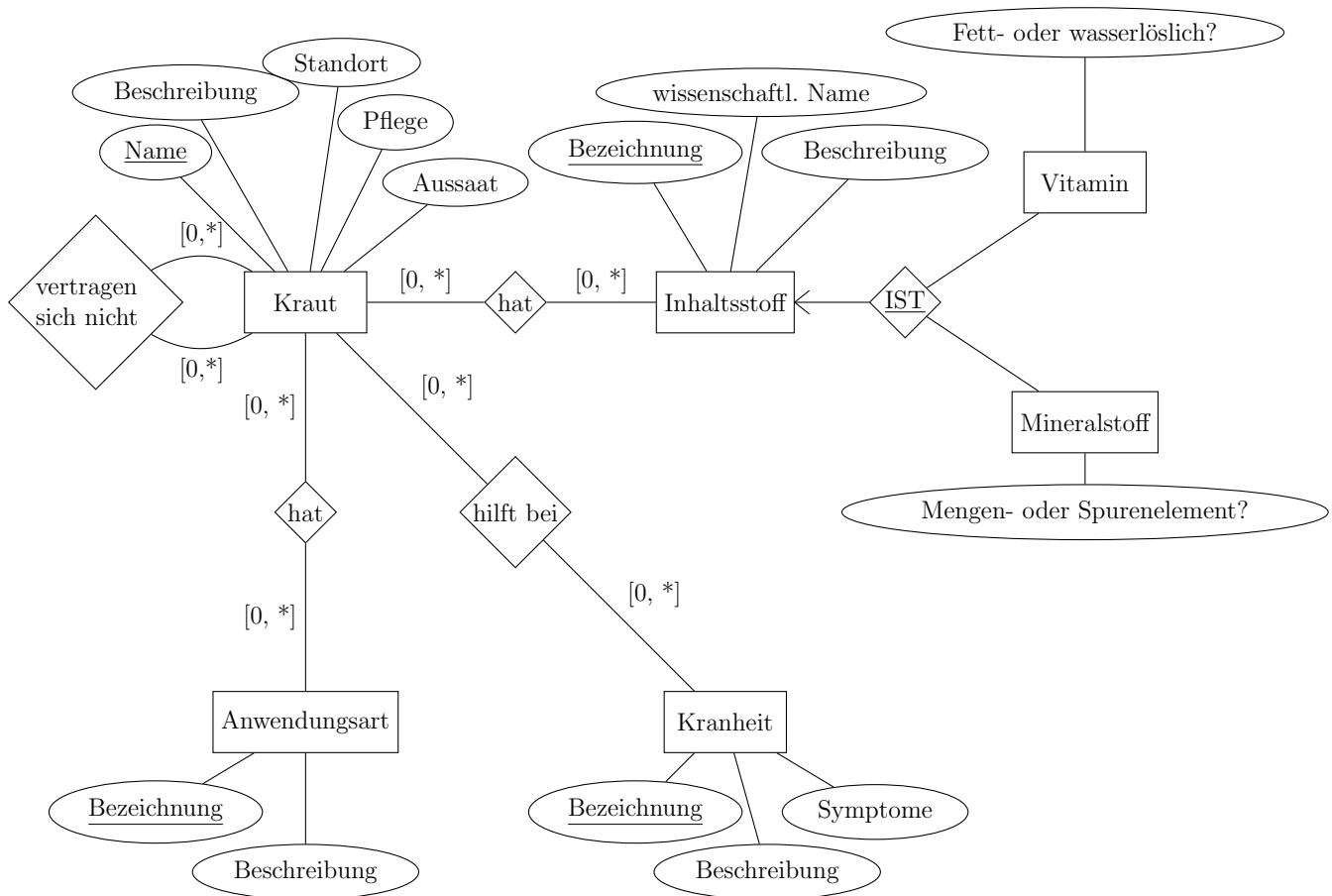
Lösung



Aufgabe 2 Abbildung vom ER-Modell auf das relationale Modell

(10 Punkte)

Gegeben sei das folgende ER-Diagramm, das als Grundlage für eine Kräuter-App dienen soll. Überführen Sie das ER-Diagramm in das relationale Modell. Achten Sie darauf, dass Sie mögliche Verschmelzungen durchführen. Markieren Sie Primärschlüssel durch unterstreichen und Fremdschlüssel durch gestricheltes unterstreichen. Achten Sie bei Ihrer Lösung auf Redundanzfreiheit und Vermeidung von Nullwerten!



Lösung KRAUT (Name, Beschreibung, Standort, Pflege, Aussaat)

ANWENDUNGSART (Bezeichnung, Beschreibung)

KRANKHEIT (Bezeichnung, Beschreibung, Symptome)

INHALTSSTOFF (Bezeichnung, WissenschaftlName, Beschreibung)

VITAMIN (Bezeichnung, FettOderWasserlöslich)

MINERALSTOFF (Bezeichnung, MengenOderSpurenelement)

KRAUTANWENDUNGSART (Name, Bezeichnung)

KRAUTKRANKHEIT (Name, Bezeichnung)

KRAUTINHALTSSTOFF (Name, Bezeichnung)

VERTRAGENSICHT (Kraut1, Kraut2)

Kraut1 und *Kraut2* sind Fremdschlüssel auf KRAUT.Name

Die Ist-Beziehung hier ist partiell (es können Instanzen in der Oberklasse existieren, die in keiner Unterklasse vorkommen) und disjunkt (Subklassen sind bzgl. Instanzen zueinander disjunkt). Da wir in der Aufgabenstellung Vermeidung von Nullwerten fordern, kommt nur eine Auflösung der Ist-Beziehungen wie hier angegeben in Frage. Andere bekannte Lösungsmöglichkeiten verursachen Attribute mit Nullwertbelegung.

Aufgabe 3 *Anfragesprachen*

(5 + 5 + 6 + 4 = 20 Punkte)

Ein Online-Zoofachhandel verwaltet seine Kunden-, Produkt- und Verkaufsdaten mit folgender Datenbank:

KUNDE (Kundennr, Vorname, Nachname, Geburtsdatum, Kunde_seit, Email)

KAUFT (*Kundennr*, *ProduktID*, Kaufdatum, Menge, Gesamtpreis)

PRODUKT (ProduktID, Produktname, Preis)

GEEIGNET_FUER (*ProduktID*, *TierartID*)

HAUSTIER (Kundennr, *TierartID*)

TIERART (TierartID, Tierart)

Primärschlüssel sind unterstrichen, Fremdschlüssel sind *kursiv* dargestellt.

3.1 Formulieren Sie die folgenden Anfragen in der Relationenalgebra:

- (a) Welche Produkte (Produktname, Preis) kosten mehr als 100 Euro?

Lösung $\pi_{\text{Produktname, Preis}}(\sigma_{\text{Preis} > 100}(\text{PRODUKT}))$

- (b) Welche Kunden (Kundennr, Vorname, Nachname) haben eine Ratte als Haustier?

Lösung $\pi_{\text{Kundennr, Vorname, Nachname}}(\text{KUNDE} \bowtie \text{HAUSTIER} \bowtie (\sigma_{\text{Tierart} = \text{'Ratte'}}(\text{TIERART})))$
 $\pi_{\text{HID, HName}}(\sigma_{\text{PName} = \text{'Lasagne'}}(\text{HERSTELLER} \bowtie \text{PRODUKT}))$

- (c) Welche Produkte (Produktname, Preis) wurden noch nie verkauft?

Lösung $\pi_{\text{Produktname, Preis}}(\text{PRODUKT} \bowtie (\pi_{\text{ProduktID}}(\text{PRODUKT}) - (\pi_{\text{ProduktID}}(\text{KAUFT}))))$

Achtung: Für die Differenz muss das gleiche Relationenschema vorhanden sein!

- (d) Welche Produkte (Produktname, Preis) sind sowohl für Hunde als auch für Katzen geeignet?

Lösung Mittels Division:

$\pi_{\text{Produktname, Preis}}(\text{PRODUKT} \bowtie$
 $(\text{GEEIGNET_FUER} \div \pi_{\text{TierartID}}(\sigma_{\text{Tierart} = \text{'Hund'}} \vee \text{Tierart} = \text{'Katze'}}(\text{TIERART}))))$

Mittels Durchschnitt:

$\pi_{\text{Produktname, Preis}}(\text{PRODUKT} \bowtie$
 $(\pi_{\text{ProduktID}}(\text{GEEIGNET_FUER} \bowtie \sigma_{\text{Tierart} = \text{'Hund'}}(\text{TIERART})) \cap$
 $\pi_{\text{ProduktID}}(\text{GEEIGNET_FUER} \bowtie \sigma_{\text{Tierart} = \text{'Katze'}}(\text{TIERART}))))$

Achtung: Für den Durchschnitt muss das gleiche Relationenschema vorhanden sein! Außerdem ist die Projektion der „zu durchschneidenden“ Mengen auf ProduktID notwendig, sonst erhalten wir ein falsches Ergebnis.

3.2 Formulieren Sie die folgenden Anfragen mittels Tupelkalkül:

- (a) Welche Produkte (Produktname, Preis) sind für Meerschweinchen geeignet?

Lösung $\{p.\text{Produktname}, p.\text{Preis} \mid \text{PRODUKT}(p) \wedge \text{GEEIGNET_FUER}(gf) \wedge \text{TIERART}(t) \wedge$
 $p.\text{ProduktID} = gf.\text{ProduktID} \wedge gf.\text{TierartID} = t.\text{TierartID} \wedge$
 $t.\text{Tierart} = \text{'Meerschweinchen'}\}$

- (b) Welche Produkte (Produktname, Preis) wurden noch nie verkauft?

Lösung

$\{p.\text{Produktname}, p.\text{Preis} \mid \text{PRODUKT}(p) \wedge \neg \exists ka (\text{KAUFT}(ka) \wedge ka.\text{ProduktID} = p.\text{ProduktID})\}$

- (c) Für welche Tierarten wurden 2013 von Kunden, die sich vor 2007 angemeldet haben, schon Produkte gekauft?

Lösung

$\{t.\text{Tierart} \mid \text{TIERART}(t) \wedge \text{GEEIGNET_FUER}(g) \wedge \text{KUNDE}(ku) \wedge \text{KAUFT}(ka) \wedge$
 $t.\text{TierartID} = g.\text{TierartID} \wedge g.\text{ProduktID} = ka.\text{ProduktID} \wedge$
 $ka.\text{Kundenr} = ku.\text{Kundenr} \wedge ka.\text{Kaufdatum} \geq 1.1.2013 \wedge$
 $ku.\text{Kunde_seit} < 1.1.2007\}$

- (d) Für welche Tierarten gibt es Produkte, die auch für Lachtauben geeignet sind?

Lösung

$\{t_1.\text{Tierart} \mid \text{TIERART}(t_1) \wedge \text{TIERART}(t_2) \wedge t_2.\text{Tierart} = \text{„Lachtaube“} \wedge$
 $\text{GEEIGNET_FUER}(g_1) \wedge \text{GEEIGNET_FUER}(g_2) \wedge g_1.\text{ProduktID} = g_2.\text{ProduktID} \wedge$
 $g_1.\text{TierartID} = t_1.\text{TierartID} \wedge g_2.\text{TierartID} = t_2.\text{TierartID}\}$

Wiederholung des Schemas:

KUNDE (Kundennr, Vorname, Nachname, Geburtsdatum, Kunde_seit, Email)

KAUFT (Kundennr, ProduktID, Kaufdatum, Menge, Gesamtpreis)

PRODUKT (ProduktID, Produktname, Preis)

GEEIGNET_FUER (ProduktID, TierartID)

HAUSTIER (Kundennr, TierartID)

TIERART (TierartID, Tierart)

3.3 Formulieren Sie die folgenden Anfragen in SQL:

- (a) Für welche Tiere (Tierart) ist das Produkt „Plüschgeier Fred“ geeignet?

Lösung

```
SELECT t.Tierart
FROM TIERART t, GEEIGNET_FUER gf, PRODUKT p
WHERE t.TierartID = gf.TierartID AND
      gf.ProduktID = p.ProduktID AND
      p.Produktname = 'Plueschgeier Fred'
```

- (b) Welche Kunden (Kundennr, Vorname, Nachname) haben in diesem Jahr (2013) noch keine Einkäufe getätigt?

Lösung

```
SELECT Kundennr, Vorname, Nachname
FROM KUNDE
WHERE Kundennr NOT IN (SELECT Kundennr
                        FROM KAUF
                        WHERE Kaufdatum >= '2013-01-01' AND
                        Kaufdatum < '2014-01-01')
```

- (c) Geben Sie die Produkte (ProduktID, Produktname, Preis) mit ihren Gesamtverkaufszahlen (nach Gesamtverkaufszahlen absteigend sortiert) an. Beachten Sie hierbei auch die Menge des verkauften Produkts!

Lösung

```
SELECT p.ProduktID, p.Produktname, p.Preis, SUM(ka.Menge) AS
Gesamtverkaufszahl
FROM KAUF ka, PRODUKT p
WHERE ka.ProduktID = p.ProduktID
GROUP BY p.ProduktID, p.Produktname, p.Preis
ORDER BY Gesamtverkaufszahl DESC
```

- (d) Welche Kunden (Kundennr, Vorname, Nachname) haben mit allen ihren Einkäufen im Jahre 2012 einen Gesamtumsatz von mindestens 500 Euro gemacht? Geben Sie die Kunden und deren Gesamtumsatz (nach Gesamtumsatz absteigend sortiert) an.

Lösung

```
SELECT ku.Kundenrnr, ku.Vorname, ku.Nachname, SUM(ka.Gesamtpreis) AS  
Gesamtumsatz  
FROM KAUFTE ka, KUNDE ku  
WHERE ka.Kundenrnr = ku.Kundenrnr AND  
      ka.Kaufdatum >= '2012-01-01' AND  
      ka.Kaufdatum <= '2012-12-31'  
GROUP BY ku.Kundenrnr, ku.Vorname, ku.Nachname  
HAVING SUM(ka.Gesamtpreis) >= 500  
ORDER BY Gesamtumsatz DESC
```

3.4 Formulieren Sie die folgenden Änderungsoperationen in SQL:

- (a) Ein neuer Kunde *Mona Krümel* hat sich heute im Onlineshop registriert und die Daten *08.01.1973*, *mona@kruemel.de* hinterlassen, sowie die Haustiere *Fisch* und *Vogel* angegeben. Fügen Sie die neuen Daten in die Datenbank ein. Nehmen Sie an, dass die automatisch generierte Kundennummer für diese Kundin *20032013128* und die TierartIDs für *Fisch* und *Vogel* 4 und 5 lauten.

Lösung

```
INSERT INTO KUNDE (Vorname, Nachname, Geburtsdatum, Kunde_seit, Email)
VALUES ('Mona', 'Kruemel', '1973-08-01', '2016-02-12',
'mona@kruemel.de');
INSERT INTO HAUSTIER (Kundennummer, TierartID) VALUES (20032013128, 4);
INSERT INTO HAUSTIER (Kundennummer, TierartID) VALUES (20032013128, 5);
```

- (b) In dieser Woche soll ein Angebot für alle Hundeartikel gelten. Reduzieren Sie die Preise aller Produkte, die für Hunde geeignet sind, um 20 %.

Lösung

```
UPDATE PRODUKT
SET Preis = Preis*0.80
WHERE ProduktID IN (SELECT gf.ProduktID
FROM GEEIGNET_FUER gf, TIERART t
WHERE gf.TierartID = t.TierartID AND
t.Tierart = 'Hund')
```

- (c) Löschen Sie die Produkte, die noch nie verkauft worden sind, aus dem Sortiment.

Lösung

```
DELETE FROM PRODUKT
WHERE ProduktID NOT IN (SELECT ProduktID
FROM KAUF)
```


Aufgabe 4 Normalformen, Synthese-Algorithmus

(2 + 4 + 8 + 6 = 20 Punkte)

Gegeben sei die Relation $S(A, B, C, D, E, F, G, H)$, für die folgende funktionale Abhängigkeiten gelten:

D	$\rightarrow F$
AE	$\rightarrow CB$
CG	$\rightarrow B$
C	$\rightarrow G$
E	$\rightarrow GA$
FB	$\rightarrow H$

- (a) Bestimmen Sie für die Relation S *alle* Schlüsselkandidaten. Begründen Sie, warum es sich hier um Schlüsselkandidaten handelt.

Lösung Der einzige Schlüsselkandidat ist DE , da

$$\{C\}^+ = \{BCG\}$$

$$\{D\}^+ = \{DF\}$$

$$\{E\}^+ = \{ABCEG\}$$

$$\{DE\}^+ = \{ABCDEFGH\} = S \text{ und minimal}$$

Alle weiteren erfüllen nicht mehr die Minimalität, da D und E immer benötigt werden.

- (b) In welcher Normalform (bis einschließlich BCNF) befindet sich die Relation S ? Begründen Sie Ihre Antwort!

Lösung 1. NF: Ja.

2. NF: Nein, da es eine partielle Abhängigkeit $D \rightarrow F$ gibt (ein Teil des Schlüssels bestimmt ein Nicht-Prim-Attribut).

3. NF: Nein, da es eine transitive Abhängigkeit $AE \rightarrow C \rightarrow G$ gibt oder weil 2. NF schon nicht erfüllt wird.

BCNF : Nein, da nicht in 3. NF

- (c) Wenn die Relation S nicht in 3NF ist, überführen Sie diese in die 3. Normalform. Zeigen Sie, dass Ihre Lösung wirklich in 3NF ist.

Lösung Linksreduktion:

$D \rightarrow F$

$AE \rightarrow CB$ wird zu $E \rightarrow CB$, da $E \rightarrow A$

$CG \rightarrow B$ wird zu $C \rightarrow B$, da $C \rightarrow G$

$C \rightarrow G$

$E \rightarrow GA$

$FB \rightarrow H$

Rechtsreduktion:

$D \rightarrow F$

$E \rightarrow CB$ wird zu $E \rightarrow C$, da $C \rightarrow B$

$C \rightarrow B$

$C \rightarrow G$

$E \rightarrow GA$

$FB \rightarrow H$

Zusammenfassen:

$C \rightarrow BG$

$D \rightarrow F$

$E \rightarrow ACG$

$FB \rightarrow H$

Damit ergeben sich die Relationen:

$R1(\underline{CBG})$, $R2(\underline{DF})$, $R3(\underline{EACG})$, $R4(\underline{FBH})$ und die Schlüsselrelation $R5(\underline{DE})$

- (d) Überprüfen Sie, ob die folgenden Dekompositionen für die Relation S verbundtreu und abhängigkeitsbewahrend sind. Begründen Sie Ihre Antwort!

- (1) $S_1(CGB)$, $S_2(ACDEFH)$

Lösung Die Zerlegung ist verbundtreu, da $BCG \cap ACDEFH = C$ und $C^+ = \{BCG\} = S_1$

Die Zerlegung ist nicht abhängigkeitsbewahrend, da die Abhängigkeit $FB \rightarrow H$ in keiner Relation vorhanden ist.

- (2) $S_1(FBH)$, $S_2(ABCDEFH)$

Lösung Es sind nicht alle Attribute vorhanden (G fehlt), damit erübrigt sich die Frage nach Verbundtreue und Abhängigkeitsbewahrung.

- (3) $S_1(DF)$, $S_2(CGB)$, $S_3(EAC)$, $S_4(FBH)$

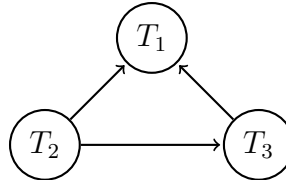
Lösung Die Zerlegung ist nicht verbundtreu, da S_1 und S_4 kein Attribut mit S_2 und S_3 teilen.

Die Zerlegung ist abhängigkeitsbewahrend, da alle Abhängigkeiten in einer eigenen Relation vorhanden sind.

- (a) Testen Sie die folgenden Schedules auf Ihre Konfliktserialisierbarkeit hin. Zeichnen Sie jeweils den Konfliktgraphen und geben Sie für jede Kante in jedem Graphen jeweils ein Konfliktpaar an.

- $S_1 := r_1(x), r_2(y), w_1(x), r_2(z), r_3(z), w_2(y), w_3(y), w_1(y)$

Lösung



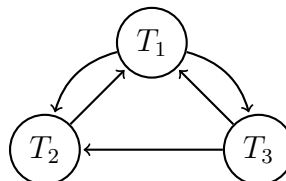
Konfliktpaare:

Konflikt	Paar
$T_2 \rightarrow T_1$	$r_2(y), w_1(y)$
$T_2 \rightarrow T_3$	$r_2(y), w_3(y)$
$T_3 \rightarrow T_1$	$w_3(y), w_1(y)$

Der Schedule ist serialisierbar, da der Konfliktgraph kreisfrei ist.

- $S_2 := r_1(z), r_1(x), r_3(z), w_3(z), r_2(x), r_2(z), w_1(x), r_3(y), w_1(z), w_2(y), w_2(z)$

Lösung



Konfliktpaare:

Konflikt	Paar
$T_1 \rightarrow T_2$	$r_1(z), w_2(z)$
$T_1 \rightarrow T_3$	$r_1(z), w_3(z)$
$T_2 \rightarrow T_1$	$r_2(x), w_1(x)$
$T_3 \rightarrow T_1$	$w_3(z), w_1(z)$
$T_3 \rightarrow T_2$	$w_3(z), r_2(z)$

Der Schedule ist nicht serialisierbar, da der Graph nicht kreisfrei ist: Es existieren die folgenden drei Kreise im Graphen: (T_1, T_2) , (T_1, T_3) , (T_1, T_3, T_2) .

- (b) Gegeben seien die Transaktionen $T_1 := r_1(x), r_1(y), w_1(x), r_1(x)$ und $T_2 := r_2(y), w_2(x), w_2(y)$. Geben Sie für die folgenden Ausdrücke an, ob es sich um Schedules zur Ausführung von T_1 und T_2 handelt. Begründen Sie Ihre Antwort, falls es sich nicht um einen Schedule handelt.

- (1) $S_1 := r_1(x), r_1(y), r_2(y), w_2(x), w_2(y)$

Lösung Es handelt sich nicht um einen Schedule, da nicht alle Operationen von T_1 und T_2 in S_1 vorkommen. So fehlt bspw. $w_1(x)$.

- (2) $S_2 := r_1(x), r_2(y), r_1(y), w_1(x), w_2(x), r_1(x), w_2(y)$

Lösung Es handelt sich bei S_2 um einen Schedule, da alle Operationen beider Transaktionen in der richtigen Reihenfolge bearbeitet werden.

(3) $S_3 := r_1(x), r_2(y), r_1(y), w_1(x), w_2(y), r_1(x), w_2(x)$

Lösung Es handelt sich hierbei nicht um einen Schedule, da die beiden letzten Operationen von T_2 in S_3 vertauscht sind: $w_2(y), w_2(x)$ in S_3 gegen $w_2(x), w_2(y)$ in T_2 .

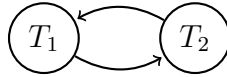
- (c) Geben Sie für den folgenden Schedule, wenn möglich, alle äquivalenten seriellen Schedules an:

$$T_1 := r_1(x), x = 10x, r_1(y), y = y + x, r_1(z), z = z + y, w_1(z), w_1(x)$$

$$T_2 := r_2(y), x = y/2, w_2(x), r_2(z), z = 2x, w_2(y)$$

$$S_1 := r_2(y), x = y/2 \text{ (zu } T_2), w_2(x), r_1(x), x = 10x \text{ (zu } T_1), r_2(z), r_1(y), y = y + x \text{ (zu } T_1), \\ z = 2x \text{ (zu } T_2), r_1(z), z = z + y \text{ (zu } T_1), w_1(z), w_1(x), w_2(y)$$

Lösung Aufgrund der Konfliktpaare $w_2(x), r_1(x)$ und $r_1(y), w_2(y)$ ergibt sich der folgende Konfliktgraph:



Dieser Graph ist nicht kreisfrei, daher ist S_1 nicht serialisierbar und es kann daher keinen äquivalenten seriellen Schedule geben.

- (d) Ergänzen Sie in folgenden Transaktionen Read-Lock (rl)-, Write-Lock (wl)- und Unlock (u)-Anweisungen nach dem konservativen Zwei-Phasen-Sperrprotokoll. Achten Sie darauf, dass Sperren möglichst früh wieder freigegeben werden.

$$T_1 := r_1(x), w_1(x), w_1(y)$$

$$T_2 := r_2(x), r_2(y), w_2(x)$$

Lösung

$$T_1 := wl_1(x), wl_1(y), r_1(x), w_1(x), u_1(x), w_1(y), u_1(y)$$

$$T_2 := wl_2(x), rl_2(y), r_2(x), r_2(y), u_2(y), w_2(x), u_2(x)$$

- (e) Konstruieren Sie aus zwei der folgenden drei Transaktionen einen Schedule, der unter dem Zwei-Phasen-Sperrprotokoll zu einem Deadlock führt. Geben Sie in diesem Schedule die Sperren an und begründen Sie, an welcher Stelle es wieso zu einem Deadlock kommt.

$$T_1 := r_1(x), r_1(y), r_1(x), w_1(z), w_1(y)$$

$$T_2 := r_2(x), r_2(y), w_2(x), r_2(z)$$

$$T_3 := r_3(z), r_3(x), r_3(y), w_3(y), w_3(x)$$

Lösung Wähle die Transaktionen T_1 und T_3 . Das Schedule

$$S := rl_1(x), r_1(x), rl_1(y), r_1(y), r_1(x), rl_3(z), r_3(z), rl_3(x), r_3(x), rl_3(y), r_3(y), wl_3(y), w_3(y), wl_3(x), \\ w_3(x), u_3(x), u_3(y), u_3(z), wl_1(z), w_1(z), wl_1(y), w_1(y), u_1(x), u_1(y), u_1(z)$$

führt zu einer Verklemmung, da T_3 an der Stelle $wl_3(y)$ versucht, ein Write-Lock auf y zu bekommen. Dieses kann aber nicht gesetzt werden, da T_1 schon ein Read-Lock auf y hat. An dieser Stelle kann also nur mit T_1 weiter gemacht werden. Auch T_1 kann aber nur bis zu der Stelle fortgeführt werden, zu der versucht wird, ein Write-Lock auf z für T_1 zu bekommen, obwohl T_3 schon ein Read-Lock auf z hat. An dieser Stelle tritt der Deadlock auf.