

1 Datenbanken

1. Allgemeines

- 3 (a) Ein Datenbank Management-System verwaltet eine Datenbank. Benötigt man mehrere Datenbank-Management Systeme, wenn mehrere Datenbanken verwendet werden?
- Ja
 Nein

Lösung:

Lösung:

- Ja
 Nein

- 3 (b) Was wird durch Normalisierung beseitigt? Was ist die Ursache davon? Geben Sie bitte ein kurzes Beispiel.

Lösung:

Durch Normalisierung werden Anomalien beseitigt. Es gibt die folgenden Arten von Anomalien: Update-; Einfügen-; Lösch-Anomalie. Die Ursache von Anomalien sind Redundanzen.

Beispiel: Zu jedem Artikel wird ein Lieferant mit Adresse gespeichert.

- 3 (c) Angenommen es soll eine Vereinigung der Relationen R und S durchgeführt werden. Welche Bedingung(en) muss(müssen) von R und S erfüllt werden? (Nur wenn alle richtigen und keine falschen Antworten angekreuzt wurden bekommen Sie 3 Punkte, ansonsten 0 Punkte.)
- Keine

- R und S sollten den gleichen Grad haben
- Alle Attribute von R und S müssen entsprechend den gleichen Namen haben
- Alle Attribute von R und S müssen entsprechend die gleichen Domänen

Lösung:

Lösung:

- Keine
- R und S sollten den gleichen Grad haben
- Alle Attribute von R und S müssen entsprechend den gleichen Namen haben
- Alle Attribute von R und S müssen entsprechend die gleichen Domänen

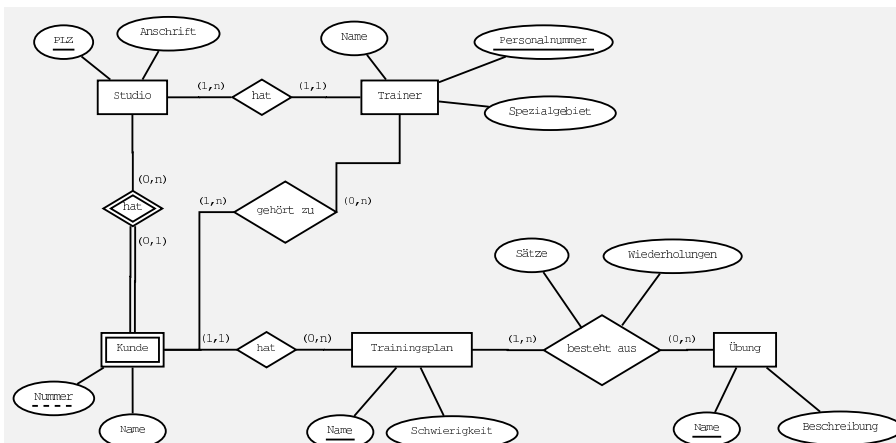
2. ERM

Die Fitnesskette besteht aus mehreren Studios. Um Konkurrenz zu vermeiden existiert nur ein Studio in jedem Postleitzahlenbereich. In den Studios arbeiten Trainer, die jedoch nur in einem Studio beschäftigt sein können. Trainer besitzen ein Spezialgebiet (z.B. Rückenschule). Unglücklicherweise arbeiten im Münchener Studio zwei Thomas Müller. Die Kunden der Filiale bekommen nur für ihr Studio eine Kundennummer und dürfen nicht in anderen Studios trainieren. Die Kunden bekommen Trainer zugewiesen, die sich um sie kümmern. In Studios mit wenigen Kunden kann es auch vorkommen, dass sich mehrere Trainer um einen Kunden kümmern. Um nicht planlos zu trainieren, bekommen die Kunden einen der Standardtrainingspläne zugewiesen. Diese Trainingspläne bestehen aus Übungen und beinhalten, wieviele Sätze und Wiederholungen gemacht werden sollten. Es gibt verschieden schwere Trainingspläne. Die Übungen der Trainingspläne werden aus einer Übungsliste zusammengestellt. Dort sind die Übungen beschrieben.

8

- (a) Modellieren Sie ein ERM für eine Fitness-Kette. Fügen Sie passende Komplexitäten hinzu.

Lösung:



Bemerkung: In dieser Aufgabenstellung sind die Typen relativ schwach spezifiziert, in dem Sinne, dass die Attribute nur Teilweise genau vorgeben werden. Das bedeutet, dass gültige Lösungen recht verschieden aussehen können. In einer echten Klausur werden meistens genauere Angaben gemacht, um die Lösung eindeutiger und damit leichter korregierbar zu machen.

4

(b) Erstellen Sie das zugehörige Datenbank-Schema.

PK					
FK					

PK					
FK					

PK					
FK					

PK					
FK					

PK					
FK					

PK					
FK					

PK					
FK					

PK					
FK					

Lösung:

Studio

	PLZ	Anschrift
PK	x	
FK		

Kunde

	Nummer	PLZ	Trainingsplan
PK	x	x	
FK		Studio.PLZ	Trainingsplan.Name

Trainer

	Personal#	Name	Fachgebiet	Studio
PK	x			
FK				Studio.PLZ

Trainingsplan

	Name	Schwierigkeit
PK	x	
FK		

Übung

	Name	Beschreibung
PK	x	
FK		

Trainer - Kunde

	Name	Studio	Personal#
PK	x	x	x
FK	Kunde.Nummer	Studio.PLZ	Trainer.Personal#

Trainingsplan - Übung

	TPName	ÜName	Sätze	Wiederholungen
PK	x	x		
FK	Trainingsplan.Name	Übung.Name		

3. SQL

Gegben ist das folgende Datenbank-Schema einer Kartbahn.

Runde

	Nummer	RennID	Zeit	Fahrer
PK	x	x		x
FK		Rennen.ID		Fahrer.ID

Fahrer

	ID	Name	Geburtsdatum
PK	x		
FK			

Rennen

	ID	Datum	Startzeit
PK	x		
FK			

Rennenergebnisse

	Rennen	Fahrer	Karthersteller	Platzierung
PK	x	x		
FK	Rennen.ID	Fahrer.ID	Karthersteller.Nummer	

Karthersteller

	Nummer	Hersteller
PK	x	
FK		

2

(a) Wer ist in Rennen 13 die schnellste Runde gefahren?

Lösung:

```

SELECT Fahrer.Name FROM Fahrer, Runde
WHERE Fahrer.ID = Runde.Fahrer AND Runde.Zeit =
  ( SELECT MIN(Zeit) FROM Runde WHERE RennID = 13 )
AND Runde.RennID = 13

```

- 3 (b) Listen Sie alle Karthersteller, die Rennen gewonnen haben, ausser "SlowKarts".

Lösung:

```
SELECT DISTINCT Hersteller FROM Kart, Rennergebnisse
WHERE Platzierung = 1
      AND Kart.Nummer = Rennergebnisse.Karthersteller
      AND NOT Kart.Hersteller = "SlowKarts"
```

Eine anderemögliche Lösung ist die folgende.

```
SELECT Hersteller FROM Kart, Rennergebnisse
WHERE Platzierung = 1
      AND Kart.Nummer = Rennergebnisse.Karthersteller
      AND NOT Kart.Hersteller = "SlowKarts"
GROUP BY Hersteller
```


- 3 (c) Fahrer mit Namen "Schumacher" werden nachträglich für alle Rennen disqualifiziert, da sie auffällig gut waren. Setzen Sie ihre bisherigen Platzierungen auf 100.

Lösung:

```
UPDATE Rennergebnisse SET Platzierung = 100
WHERE Fahrer IN
  ( SELECT ID FROM Fahrer WHERE Name = "Schumacher" )
```

- 3 (d) Listen Sie alle Fahrer und zeigen Sie alle vom Fahrer erreichte Platzierungen an. NULL falls der Fahrer bisher noch kein Rennen bestritten hat.

Lösung:

```
SELECT DISTINCT Fahrer.Name, Rennergebnisse.Platzierung
FROM Fahrer LEFT JOIN Rennergebnisse
  ON Fahrer.ID = Rennergebnisse.Fahrer
```

- 5 (e) Welcher Karthersteller hat die meisten Siege eingefahren?

Lösung:

```
SELECT Hersteller FROM Kart
WHERE Nummer =
  ( SELECT Karthersteller FROM
    ( SELECT MAX(Anzahl), Karthersteller FROM
      ( SELECT COUNT(*) as Anzahl, Karthersteller
        FROM Rennergebnisse WHERE Platzierung = 1
        GROUP BY Kart
      )
    )
  )
```

4. Relationalenalgebra

Ein Label ordnet seine erschienen Platten aktuell sehen die Tabellen wie folgt aus:

Platten

PlattenCode	Artist	Name	Genre
KTS01	Kollektiv Turmstrasse	Tristesse	Techno
RV03	Riccardo Villalobos	Sei Es Drum	Techno
RS06	Rammstein	Liebe ist für alle da	Tanzmetall
WTC01	Wu-Tang-Clan	Gravel Pit	Hip-Hop
DTK03	Dieter Thomas Kuhn	Best of	Schlager
PP01	Pan Pot	Dapayk - Sugar Remix	Minimal

Plattenrelease

ReleaseID	PlattenCode	Format	Jahr
1	KTS01	Vinyl	2008
2	RV03	Vinyl	2007
3	RS06	MP3	2010
4	WTC01	Vinyl	1999
5	DTK03	CD	2005
6	DTK03	Kassette	2005
7	WTC01	Kassette	1999
8	KTS01	MP3	2010
9	PP01	MP3	2009
10	PP01	CD	2008

Ein Onlineversandhaus bewertet die Alben und hat dazu folgende Datenbank erstellt.

Bewertungen

ReleaseID	PlattenCode	Sterne	Rezension
1	KTS01	5	Beste Single überhaupt.
2	RV03	4	Wie immer stabil.
3	RS06	5	Das Album des Jahres!
4	WTC01	4	Hip-Hop Klassiker
5	DTK03	3	Bestenfalls auf Volksfesten
6	DTK03	1	Schlechte Qualität auf Kassette
7	WTC01	5	Auf Kassette fast noch besser
8	KTS01	4	Nicht das selbe wie Vinyl.
9	PP01	4	Groß im Kommen!
10	PP01	4	Newcomer!

Verwenden Sie die Abkürzungen P für Platten, R für Plattenrelease, sowie die Abkürzungen für die jeweiligen Attribute P:(C, A, N, G) und L:(I, C, F, J).

- 2 (a) Formulieren Sie folgende Anfrage mittels Ausdrücken der Relationaler Algebra (RA). Wie heißen die 2008 erschienen Platten?

Lösung:

$$\pi_N(\sigma_{J=2008}(P \bowtie_{P.C=R.C} R))$$

- 2 (b) Drücken Sie folgenden Relationen-Tupel-Kalkül (RTK) Ausdruck in Relationaler Algebra aus:

$$\{t^{(2)} | (\exists q^{(4)})(\exists p^{(4)})(P(p) \wedge t[2] = q[3] \wedge R(q) \wedge t[1] = p[1] \wedge p[3] = \text{'Tristesse'} \wedge q[2] = p[1])\}$$

Lösung:

$$\pi_{C,F}(\sigma_{N=\text{'Tristesse'}}(P \bowtie_{P.C=R.C} R))$$

- 2 (c) Formulieren Sie eine Anfrage im Relationen-Wertebereich-Kalkül (RWK), die alle Genres von 2010 ausgibt.

Lösung:

$$\{t_1 | (\exists t_2)(\exists t_3)(\exists t_4)(\exists s_1)(\exists s_2)(\exists s_3)(\exists s_4)(P(t_2, s_1, s_2, t_1) \wedge R(s_3, t_3, s_4, t_4) \wedge t_2 = t_3 \wedge t_4 = 2010)\}$$

- 3 (d) Formulieren Sie eine Anfrage im RTK, die alle Bewertungen die Veröffentlichungen aus dem Genre 'Minimal' bekamen auflistet.

Lösung:

$$\{t^{(2)} | (\exists p^{(4)})(\exists b^{(4)})(P(p) \wedge B(b) \wedge t[1] = b[4] \wedge t[2] = b[3] \wedge p[1] = b[2] \wedge p[4] = \text{'Minimal'})\}$$

- 4 (e) Formulieren Sie eine Anfrage im RTK, die ausgibt, wieviele Sterne Dieter Thomas Kuhns Album auf Kasette bekam und wie es heißt?

Lösung:

$$\{t^{(2)} | (\exists p^{(4)})(\exists b^{(4)})(\exists r^{(4)})(P(p) \wedge B(b) \wedge R(r) \wedge t[1] = b[3] \wedge t[2] = P[3] \wedge p[2] = \text{'Dieter Thomas Kuhn'} \wedge p[1] = r[2] \wedge p[1] = b[2] \wedge r[3] = \text{'Kasette'})\}$$

2 Knowledge Engineering

5. Datalog

In einer Datalog-Datenbank, die einen Index auf das WWW speichert, seien folgende Relationen gegeben:

`hyperlink(P1, H, P2)` Es gibt auf der Seite `P1` einen Hyperlink `H`, der auf die Seite `P2` zeigt

`occurs(W, X, N)` Das Wort `W` kommt in `X N`-mal vor, wobei `X` entweder eine Seite, oder ein Hyperlink sein kann.

`page(P, Created, Expires, Crawled)` Die Seite `P` hat als Eigenschaften das Datum `Created`, an dem sie erstellt wurde, das Datum `Expires` bis zu dem sie gültig ist, und das Datum `Crawled`, an dem sie in gecrawlt wurde (d.h. in die Datenbank aufgenommen wurde).

Verwenden Sie diese Relationen zur Lösung folgender Aufgaben. Sie können dabei die Prädikate vorangehender Teilaufgaben zur Lösung der darauffolgenden verwenden (auch wenn Sie diese nicht selbst lösen konnten).

3

- (a) Definieren Sie ein Prädikat `must_crawl/2`, das alle Seiten `X` umfasst, die zu einem bestimmten Datum `D` gecrawlt werden müssen. Eine Seite muß gecrawlt werden, wenn Sie das Wort "News" enthält oder wenn Ihre Gültigkeit bereits abgelaufen ist.

Hinweis: Nehmen Sie an, daß Sie zwei Datumsangaben einfach durch `<` vergleichen können.

Lösung:

```

must_crawl(P,D) :- page(P,_,_,_),
                  occurs('News',P,N),
                  N >= 1.
must_crawl(P,D) :- page(P,_,E,_) , E < D.

```

4

- (b) Definieren Sie ein Prädikat `query_result/2` das für alle Seiten `P` erfüllt ist, auf denen ein Query-Wort `Q` mindestens zwei Mal auf der Seite vorkommt, oder auf die ein Hyperlink zeigt, auf dem das Query-Wort vorkommt.

Lösung:

```

query_result(Q,P) :- page(P,_,_,_),
                  occurs(Q,P,N),
                  N >= 2.
query_result(Q,P) :- hyperlink(_,H,P),
                  occurs(Q,H,N),
                  N > 0.

```

- 4 (c) Definieren Sie ein Prädikat `connected/2`, das angibt, ob zwei Seiten A und B miteinander verbunden sind, d.h. ob man durch eine Folge von Clicks auf Hyperlinks von der Seite A auf die Seite B gelangen kann.

Lösung:

```
connected(A,B) :- hyperlink(A,_,B) .  
connected(A,B) :- hyperlink(A,_,C),  
                  connected(C,B) .
```

6. Semantik von Prolog und Datalog

In einer Firma gilt folgende Befehlskette:

```
vorgesetzter(john,bill) .      vorgesetzter(adam, evan) .
vorgesetzter(evan, george) .  vorgesetzter(evan, frank) .
vorgesetzter(john, carl) .    vorgesetzter(evan, hans) .
vorgesetzter(adam, dave) .    vorgesetzter(rick, evan) .
vorgesetzter(john, adam) .    vorgesetzter(rick, carl) .
```

Weiters sei folgende rekursive Relation untergebener/2 definiert:

```
untergebener(X, Y) :- vorgesetzter(Y, X) .
untergebener(X, Y) :- untergebener(X, Z) , vorgesetzter(Y, Z) .
```

Beantworten Sie folgende Fragen:

Anmerkung: Sie können die Namen und Relationen mit ihren Anfangsbuchstaben abkürzen.

- 3 (a) Geben sie eine Query an, bei deren Abarbeitung PROLOG in eine unendliche Rekursion laufen würde.

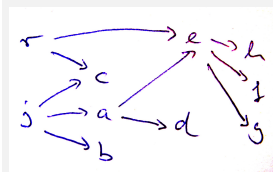
Lösung:

```
u(j, X)
```

- 4 (b) Wie viele Elemente enthält der Fixpunkt für diese Fakten- und Regelmenge? Nach wie vielen Iterationen der Fixpunkt-Semantik wird dieser gefunden? Begründen Sie beide Antworten.

Hinweis: Die Aufgabe kann auch gelöst werden, ohne alle Iterationen durchzuführen (was Ihnen aber natürlich freisteht). Eine Skizze kann hilfreich sein.

Lösung:



vorgesetzter liefert 10 gültige Literale, nämlich genau die, die im Programm gegeben sind. Zur Relation untergebener gibt es 21 Literale im Fixpunkt. Um sie zu zählen, stellen wir zunächst fest, dass u der (invertierte) transitive Abschluss von v ist. Dann zählen wir z.B. für jede Konstante x separat, wieviele y existieren, so dass $u(x, y)$ gilt, d.h. so dass ein v -Weg von y nach x existiert. Dann erhalten wir z.B. für $x=g$ vier mögliche y und insgesamt $1 + 1 + 2 + 2 + 3 + 4 + 4 + 4 + 0 + 0 = 21$ Fakten.

Es werden vier Iterationen benötigt. In der ersten kommen die Fakten des Programms hinzu, die in der Skizze den Wegen der Länge 1 entsprechen. In der i -ten Iteration kommen die hinzu, die einem Weg der Länge i entsprechen. Die längsten Wege haben Länge 4.

Hier nochmals die gegebenen Regeln und Fakten:

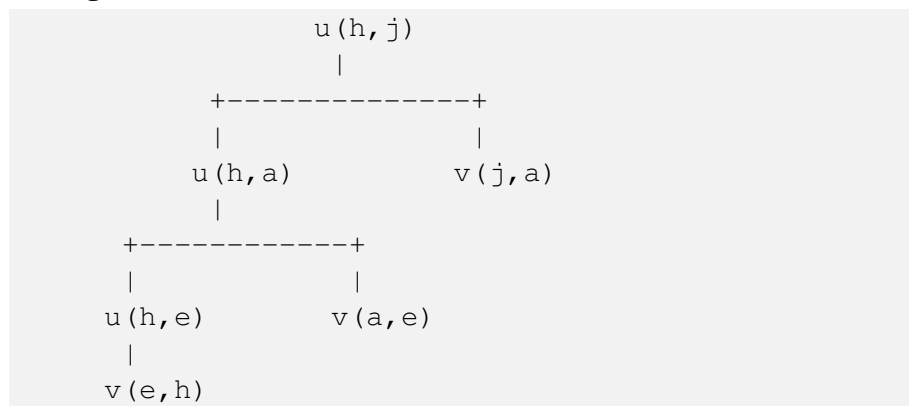
```
vorgesetzter(john, bill) .    vorgesetzter(adam, evan) .
vorgesetzter(evan, george) . vorgesetzter(evan, frank) .
vorgesetzter(john, carl) .   vorgesetzter(evan, hans) .
vorgesetzter(adam, dave) .   vorgesetzter(rick, evan) .
vorgesetzter(john, adam) .   vorgesetzter(rick, carl) .

untergebener(X, Y) :- vorgesetzter(Y, X) .
untergebener(X, Y) :- untergebener(X, Z), vorgesetzter(Y, Z) .
```

4

(c) Erstellen Sie einen Beweisbaum für die Query `untergebener(hans, john)`.

Lösung:



4

(d) Welche Regel(n) könnte(n) mit der Grundversion des EBL-Algorithmus gelernt werden. Geben Sie für jede Antwort entweder ein Beispiel an, aus dem die Regel mit EBL gelernt werden kann, oder eine Begründung, warum sie nicht gelernt werden kann.

- $\text{untergebener}(A, B) :- \text{vorgesetzter}(B, X), \text{vorgesetzter}(X, Y), \text{vorgesetzter}(Y, A).$
- $\text{untergebener}(A, B) :- \text{vorgesetzter}(B, X), \text{vorgesetzter}(A, X).$
- $\text{untergebener}(A, B) :- \text{vorgesetzter}(B, X), \text{untergebener}(A, X).$

Lösung:

Die jeweilige Regel kann gelernt werden, wenn wir die Variablen mit Konstanten belegen können, so dass es einen Beweisbaum gibt mit den Head in der Wurzel und den Literalen des Bodies an den Blättern. Dies ist bei der ersten Regel der Fall ($\{A/h, B/j, X/a, Y/e\}$). Bei der zweiten finden wir keinen solchen Beweisbaum. Bei der dritten finden wir zwar einen Beweisbaum, der die Body-Literale enthält ($\{A/e, B/j, X/a\}$), aber das $u(e, a)$ ist darin kein Blatt sondern ein Binnen-Knoten.

Da im Programm keine Fakten zu untergebener mitgeliefert werden, kann man für die dritte Regel auf den ersten Blick ausschließen, dass sie ein Ergebnis des EBL-Algorithmus ist. Denn die Body-Literale von EBL-Regeln verwenden stets die Relationen, die in den Blättern eines Beweisbaumes stehen und die Blätter wiederum sind immer Fakten aus dem Programm.

7. Induktion von Datalog Programmen

Gegeben sei folgende positive und negative Beispiele für eine Relation $a/1$.

\oplus a (v)	\ominus a (r)
\oplus a (w)	\ominus a (s)
\oplus a (x)	\ominus a (t)
\oplus a (y)	\ominus a (u)

Das Hintergrundwissen bestehe aus den Relationen $l/3$ und $m/1$, die wie folgt definiert sind:

$l(u, r, r)$	$l(v, v, v)$	$m(r)$
$l(v, s, s)$	$l(y, w, r)$	$m(s)$
$l(v, t, w)$	$l(x, y, s)$	$m(x)$
$l(x, u, y)$	$l(y, x, x)$	$m(y)$

Gelernt werden soll eine Regel(-menge) mit dem Head $a(A)$. Beantworten Sie folgende Fragen:

6

(a) Angenommen, ein Separate-and-Conquer Regel-Lerner beginnt mit der Regel

$$a(A) \text{ :- } l(X, A, Y)$$

und möchte diese nun weiter verfeinern. Evaluieren Sie die folgenden Bedingungen mit Hilfe des Maßes $p - n$, wobei p die abgedeckten positiven und n die abgedeckten negativen Beispiele sind. Welche Bedingung wird ausgewählt?

Literal	p	n	$p - n$
$m(A)$			
$m(X)$			
$m(Y)$			
$l(Y, A, A)$			
$l(X, Y, Z)$			

Lösung:

Literal	p	n	$p - n$
$m(A)$	2	2	0
$m(X)$	3	1	2
$m(Y)$	3	3	0
$l(Y, A, A)$	1	0	1
$l(X, Y, Z)$	2	3	-1

Es wird das Literal $m(X)$ ausgewählt.

- 4 (b) Nachdem Sie im letzten Schritt eine Bedingung ausgewählt haben, was passiert als nächstes?
- alle von dieser Regel abgedeckten Beispiele werden entfernt und eine weitere Regel wird gelernt
 - alle von dieser Regel abgedeckten Beispiele werden entfernt und der Lernprozess wird beendet
 - die Regel wird um ein determinates Literal erweitert
 - die Regel wird weiter spezialisiert
 - die Regel wird weiter generalisiert

Begründen Sie Ihre Wahl!

Lösung:

Die Regel deckt drei positive Beispiele und ein Negatives ab. Sie muß daher durch Anhängen einer weiteren Bedingung noch weiter spezialisiert werden.

- 3 (c) Welches der folgenden Literale ist für den Head $a(A)$ determinat? Warum?
- Hinweis:** Mehrfachantworten sind möglich, Sie erhalten aber nur Punkte, wenn alle Antworten richtig sind.
- $l(A, X, Y)$
 - $l(X, A, Y)$
 - $l(X, Y, A)$
 - keines

Lösung:

$l(X, A, Y)$, da es hier für jede Belegung von A genau eine Belegung der anderen Variablen gibt.

- 3 (d) Wie setzt Foil determinate Literale ein? Welches Problem wird dadurch zu lösen versucht?

Lösung:

Literale, die neue Variablen ins Spiel bringen, haben oft keine Gain, da Sie für jede Variablenbelegung eine gültige Belegung der freien Variablen bringen, und deshalb alle positiven und alle negativen Beispiele abdecken. Sie sind aber dennoch zum Lernen einer korrekten Definition notwendig. Um dieses Problem zu umgehen, fügt Foil *alle* determinaten Literale an eine Regel an, sobald es mit den bisherigen Variablen keine ausreichend gute Bedingung mehr finden kann (keine erreicht 80% des maximalen Gains). Die determinaten Literale blasen den Suchraum nicht unnötig auf, können den Lerner aber durch die neuen Variablen, die sie ins Spiel bringen, weiterhelfen.

8. Data Mining und Semantic Web

4

- (a) Erläutern Sie die geplante Rolle der Schichten *Logic* und *Proof* im Sieben-Schichten-Modell des Semantic Webs und beschreiben Sie kurz eine beispielhafte Anwendung.

Lösung:

Die Aufgabe dieser Schichten ist es, Sachverhalte, die in den darunterliegenden Schichten formuliert werden, zu verifizieren, und dadurch die Richtigkeit von Angaben zu gewährleisten. Z.B. können Angaben wie z.B., daß eine Person bei einer bestimmten Firma angestellt ist und berechtigt ist, die Firmenkreditkarte zu benutzen, durch entsprechende Semantic Web-Services verifiziert werden.

4

- (b) Was versteht man unter den Operationen *Slice* und *Dice* in einem Data Cube? Illustrieren Sie Ihre Antwort anhand eines Beispiels.

Lösung:

Beides sind Hilfsmittel zur Datenanalyse via OLAP (On-line Analytical Processing), die von einem Data Cube zur Verfügung gestellt werden.

Ein *Slice* erlaubt das Herausschneiden einer Scheibe aus dem Data Cube. Das entspricht der Selektion nach einem einzelnen Wert (also Z.B. $City = Darmstadt$).

Ein *Dice* erlaubt das Herausschneiden eines Teilwürfels. Das entspricht der Selektion nach mehreren Werten (also Z.B. $City = Darmstadt \vee Frankfurt \wedge (Sales = high)$).