FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
Professur für Didaktik der Informatik

Lösungs*vorschläge* zu den Staatsexamina:
Softwaresysteme und Datenbanken

Frühjahr 2018

1 Thema

2 Teilaufgabe 1

Aufgabe 1

(a) Selektivität ist ein Maß, das in der Informatik bei Datenbankabfragen auf Datenbanktabellen in relationalen Datenbankensystemen gebraucht wird; sie bestimmt den Anteil der Datensätze, die bei einer Abfrage nicht durch eine Selektionsbedingung aus der Ergebnismenge ausgefiltert werden, relativ zur Gesamtzahl der Datensätze des Datenbestandes, welcher der Abfrage zugrunde liegt. Bei der am weitesten verbreiteten Abfragesprache für relationale Datenbanken, SQL, werden Selektionsbedingungen in der WHERE-Klausel der Abfrage spezifiziert (WIKI)

(b) Structured Querry Language
Unterschied: Mengenorientiert, natürlichsprachlich
Vorteile: Spezifikation nur welche Daten benötigt werden, nicht wie diese geholt werden sollen (vgl. Liste, Array)

(c) Die atomare Ausführung der Transaktion verhindert, dass Inkonsistenzen im Datenbankbestand entstehen können. Die Transaktion wird entweder ganz oder gar nicht ausgeführt und hinterlässt so immer eine konsistente Datenbank.

(d) Wenn die Isolation von Transaktionen nicht durchgesetzt wird, dann kann es passieren, dass die Transaktionen ihre Arbeitsabläufe gegenseitig ßtören", indem z.B. zwei Transaktionen gleichzeitig auf gleiche Werte zugreifen und sich dann gegenseitig überspeichern. Dabei spricht man auch vom Lost Update-Problem.

(e) Nein, eine VIEW ist keine physische Tabelle. Abfragen gegen eine VIEW werden intern durch die Definition der VIEW ersetzt, damit wird die Abfrage letztendlich gegen die eigentlichen Tabellen ausgeführt.

(f) Nein, dies scheitert bei unveränderbaren Sichten und geht nur in veränderbaren. Sichten sind jedoch nur dann veränderbar, wenn sie weder Aggregatsfunktionen noch Anweisungen wie distinct, group by oder having enthalten, sowie die SELECT-Liste nur eindeutige Spaltennamen enthält und ein Schlüssel der Basisrelation enthalten ist sowie sie nur genau eine Tabelle verwenden, die ebenfalls veränderbar ist.

(g) Ja, zeit- und performanceaufwändige SQL-Anfragen können im Data Warehouse-Bereich durch kontrollierte, bewusste Redundanzen verbessert werden.

Aufgabe 2

Aufgabe 3

Aufgabe 4

(a)

(b)

(c)

(d)

(e)

(f)

(g)

Aufgabe 5

Aufgabe 6

(a)

(b)

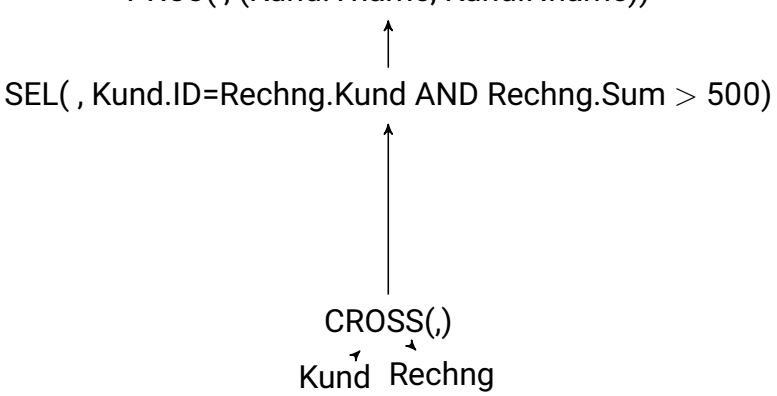
Aufgabe 7

(a) Logische Optimierung transformiert relational Algebraausdrücke in äquivalente Ausdrücke, die zu schnellerem Ausführungsplan führen (Fokus: Ausgaben der einzelnen Operatoren werden möglichst klein), dazu werden z.B. Selektionen und Kreuzprodukte zu Joins zusammengefasst, die Reihenfolge der Joins optimiert und Selektionen und Projektionen möglichst früh durchgeführt. Die physische Optimierung baut auf die logische Optimierung auf und ermittelt für die jeweiligen logischen Operatoren einen möglichst guten physischen. Zudem wird entschieden, ob Indices benutzt werden können, Zwischenergebnisse materiali-

siert etc. Hier tatsächlicher Zugriff af Tabellen und Joins, theoretische Optimierung vs. praktische Algorithmenwahl, mathematische Operationen der Relationalen Algebra vs. ausführbare Algorithmen, die logischen Operator implementieren.

(b)

Achtung: Andere Notation mit griechischen Buchstaben empfohlen !



(c) Join statt Kreuzprodukt (CROSS), Selektion (SEL) der Sum vorziehen vor Join

3 Teilaufgabe 2

Aufgabe 1

(a)

(b)

(c)

(d)

Aufgabe 2

(a)

(b)

(c)

Aufgabe 3

(a)

(b)

(c)

4 Thema

Teilaufgabe 1

Aufgabe 1

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Aufgabe 2

Aufgabe 3

Aufgabe 4

(a) $\{$s.vorname$|s \in$ Student $\wedge \forall v \in$ Vorlesung(v.Name = 'Datenbanksysteme I' $\Rightarrow \exists b \in$ besucht(v.VNR = b.VNR $\wedge$ s.MatrikelNr = b. MatrikelNr))$\}$

oder
$\{s.vorname|s \in Student \wedge s.MatrikelNr = b.Student \wedge b \in besucht \wedge b.vorlesung = v.VNR \wedge v \in Vorlesung \wedge v.titel =' DatenbanksystemeI'\}$

(b) $\{$s.MatrikelNr$|s \in$ Student $\wedge \forall b \in$ besucht $\Rightarrow \forall v \in Vorlesung(b.Vorlesung = v.VNR \wedge b.Student = s.Matrikelnummer \Rightarrow v.Klausurtermin \leq 31.12.17)\}$
oder
$\{s.Matrikelnummer|s \in Student \wedge s.MatrikelNr = b.Student \wedge b \in besucht \wedge b.Vorlesung = v.VNR \wedge v \in Vorlesung \wedge \nexists(v.Klausurtermin > 31.12.2017)\}$

(c) $\{s.m|s \in Student \wedge d.Nachname = Schulz \wedge d.titel = ProfDr \wedge d \in Dozent \wedge \forall v \in Vorlesungen(v.Dozent = d.DNR \Rightarrow \exists b \in besucht(b.Student = s.MatrikelNr \wedge b.Vorlesung = v.VNR))\}$

Aufgabe 5

(a)

(b)

(c)

Aufgabe 6

(a)

(b)

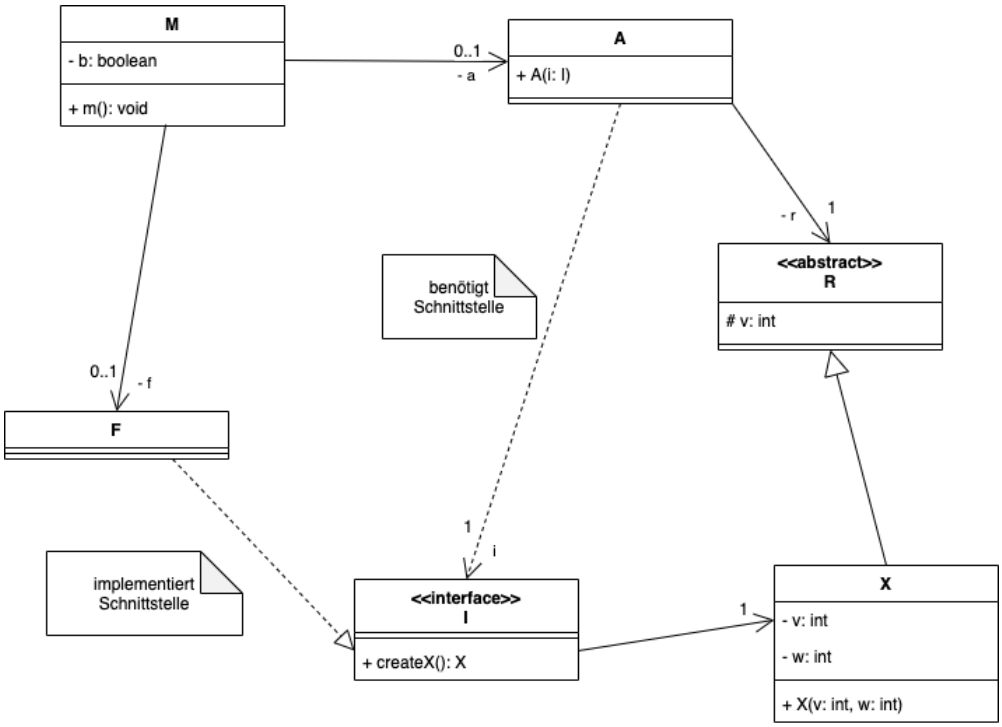
Teilaufgabe 2

Aufgabe 1

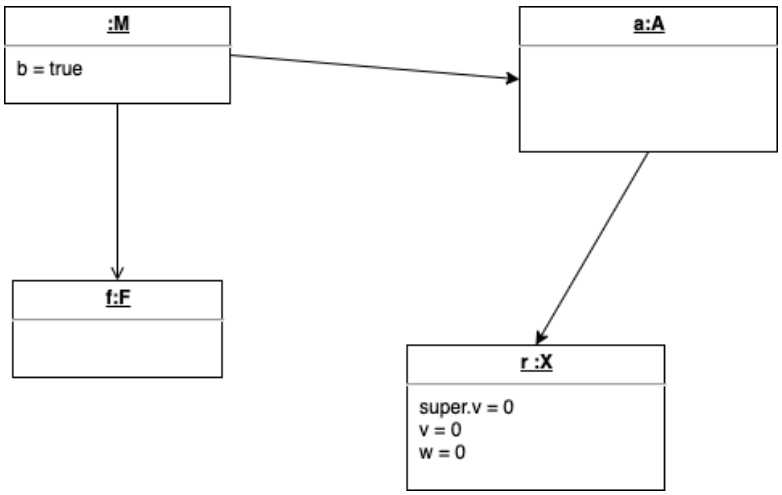
(a)

- Die Klasse F implementiert das Interface (die Schnittstelle) I. Sie hat somit alle Funktionalitäten, die als Schnittstelle von der Klasse I zur Verfügung gestellt werden.
- Die Klasse X erbt von der abstrakten Oberklasse R.

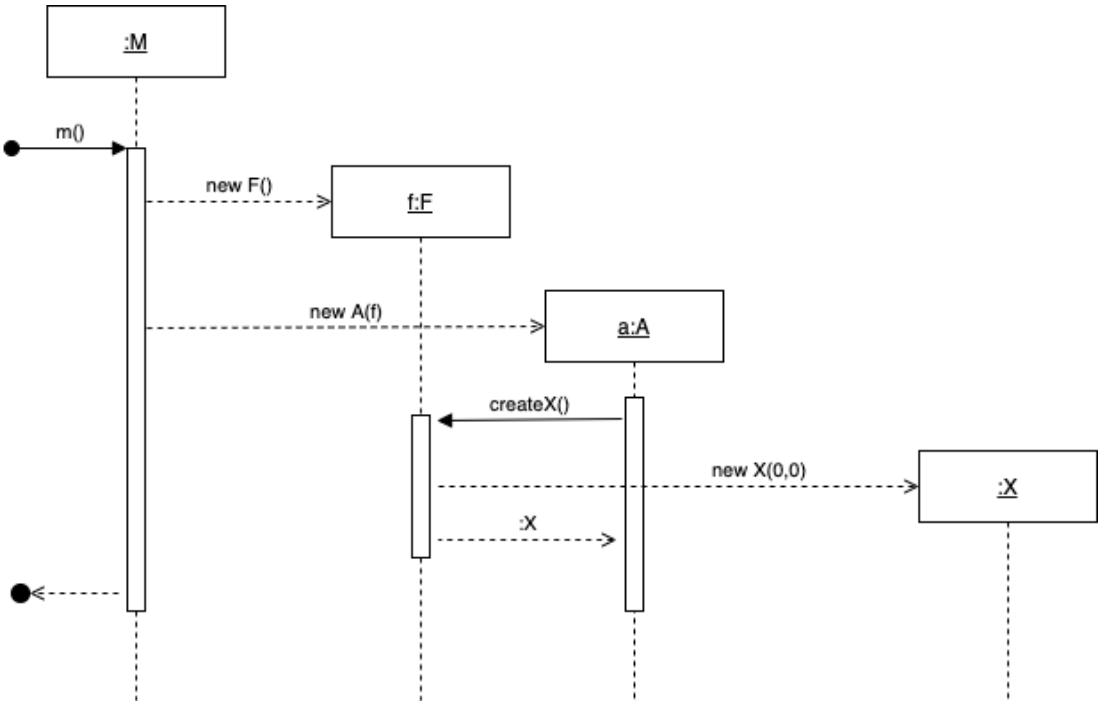
(b)



(c)



(d)



Aufgabe 2

(a)

(b)

(c)

(d)

Aufgabe 3

Aufgabe 4