

Lösungen zum Übungsblatt Nr. 6

Aufgabe 1 7/8

- a) Die Anfragen berechnen alle Ländercodes der Länder, in denen mindestens die selben Sprachen wie in den USA gesprochen werden.
Ausdrücke in Relationaler Algebra:

gilt für alle Anfragen : $\pi[\text{country}, \text{name}] \text{Language} \div \pi[\text{name}] (\sigma[\text{country} = 'us'] \text{Language})$

$Q_2: \pi[L1.\text{country}] (\sigma[L1.\text{name} = L2.\text{name}, L2.\text{country} = 'us'] (\delta[\text{country} \rightarrow L1.\text{country}, \text{name} \rightarrow L1.\text{name}] (\pi[\text{country}, \text{name}] \text{Language}) \bowtie \delta[\text{country} \rightarrow L2.\text{country}, \text{name} \rightarrow L2.\text{name}] (\pi[\text{country}, \text{name}] \text{Language})))$

$Q_3: \pi[\text{country}] \text{Language} - \pi[\text{country}] ((\pi[\text{country}] \text{Language} \bowtie \pi[\text{name}] (\sigma[\text{country} = 'us'] \text{Language}) - \pi[\text{country}, \text{name}] \text{Language}))$

- b) Q_1 : Diese Anfrage gibt dasselbe wie vorher aus, da MINUS Duplikate ignoriert.



Q_2 : Diese Anfrage würde nichts mehr zurückgeben, da $1 \leq \text{COUNT}(L1.\text{name}) \leq 2$, aber $\text{COUNT}(L3.\text{name}) = 3$ aufgrund des Duplikates. Die *HAVING* Klausel wäre also nie erfüllt.

Q_3 : Diese Anfrage hat exakt das selbe Ergebnis, wie ohne Duplikat. Die erste Subquery enthält Duplikate, da diese aber von den duplikatfreien Ländern abgezogen wird, macht das keinen Unterschied.

- c) Q_1 : Diese Anfrage würde alle einzigartigen Ländercodes ausgeben, da in der ersten Subquery *WHERE* *country* = 'us' nie wahr wird. Dadurch ist *WHERE NOT EXISTS* immer wahr.

Q_2 : Diese Anfrage gibt keine Zeilen zurück, da *WHERE* *L2.country* = 'us' nie erfüllt wird.

Q_3 : Diese Anfrage würde alle einzigartigen Ländercodes zurückgeben, da nichts davon abgezogen wird. Der *CROSS JOIN* ist leer, da die Bedingung *WHERE country* = 'us' nicht erfüllt wird und damit gibt die komplette Subquery keine Zeilen zurück.

Aufgabe 3 2/2

Wir zerlegen die gegebene Anfrage in 3 Teile :

SELECT * FROM R, WHERE R:A NOT IN und (SELECT A FROM S)

Der dritte Teil ergibt eine Tabelle $L : \{a, b, null, \dots, \dots\}$. Es gilt $R : A \text{ NOT } \in L \forall z \in L, R:A \neq z$.

Laut der Aussage hat $R:A \neq z$ den Wert **true**. Da L einen Nullwert enthält, ergibt sich

den Wert **unknown**. Daher die **Where** Klausel **R.A NOT IN (SELECT A FROM S)** nicht erfüllt.



Aufgabe 4

3.5/4

a) $COUNT(*) = 0$. $A \text{ NOT IN } B$ ergibt die Werte, die in A sind und nicht in B . Allerdings betrachtet man die Menge $B = \{2, NULL, 4, 1\}$, so sieht man, dass sie einen Nullwert enthält. Wenn man IN verwendet, weist man SQL an, einen Wert zu nehmen und ihn mit einem beliebigen Wert oder einer Gruppe von Werten in einer Liste zu vergleichen, indem man $=$ verwendet. Wenn $NULL$ -Werte vorhanden sind, wird keine Zeile zurückgegeben, auch wenn beide Werte $NULL$ sind. ✓

b) $COUNT(*) = 1$, analog zu a, allerdings enthält $B = \{2, 2, 4, 1\}$ dieses Mal keinen Nullwert.

$1 \notin \{2, 2, 4, 1\} = false$, $2 \notin \{2, 2, 4, 1\} = false$, $null \notin \{2, 2, 4, 1\} = false$ -liefert false (siehe a), $3 \notin \{2, 2, 4, 1\} = true$ und somit $COUNT(*) = 1$. ✓

unknown (-0.5)

Aufgabe 5

1/4

a)

```
CREATE VIEW NameOfCitiesAndCountries AS
(SELECT Country.name AS countryName, Stadt.name As CityName
From country, city
where country.lcode = city.lcode)
```

 ✓

b) Wenn wir die Leistung der materialisierten Sicht sehen, ist sie besser als die normale Sicht, da die Daten der materialisierten Sicht in Tabelle gespeichert werden und die Tabelle so indiziert sein kann, dass sie auch zum Zeitpunkt der Aktualisierung der materialisierten Sichten 'gejoint' wird.

aber würde sich das auch in diesem Fall anbeiten?

c) -1

d) -1