

Lösung zu Aufgabe 1

Fragmentierung: Daten einer Relation können fragmentiert, d.h. auf mehrere Rechner verteilt sein.

Replikation: Daten können auf mehreren Rechnern gleichzeitig gehalten werden.

Unabhängigkeit bedeutet in beiden Fällen, dass sich das System an der Schnittstelle nicht anders verhält, ob nun mit oder ohne Fragmentierung bzw. Replikation gearbeitet wird.

Lösung zu Aufgabe 2

Eine Datenbank heißt verteilt, wenn die zusammengehörigen Daten dieser Datenbank auf mindestens zwei Rechnern aufgeteilt sind und von einem gemeinsamen Datenbankverwaltungssystem verwaltet werden.

Lösung zu Aufgabe 3

In einem verteilten Datenbanksystem können selbstverständlich auch lokale Deadlocks auftreten. Denken wir nur an eine Transaktion, die nur lokale Zugriffe durchführt. Es können also jederzeit sowohl globale als auch lokale Deadlocks vorkommen.

Lösung zu Aufgabe 4

Die Regel 2 ist dann erfüllbar, wenn der Koordinator nicht zentral auf einem Rechner gehalten wird. Übernimmt beispielsweise immer derjenige Rechner die Koordination, auf dem eine globale Transaktion gestartet wurde, so ist Regel 2 nicht verletzt.

Lösung zu Aufgabe 5

Individuell zu beantworten. In der Praxis verbreitet ist das Zwei-Phasen-Commit-Protokoll. Hier werden mehrere Regeln von Date verletzt: 1, 2, 4, 5, 6, 8, 12. Die Verteilung ist sehr eingeschränkt, dafür wird sehr hoher Wert auf die Konsistenz gelegt. Es liegt ein CA-System vor.

Lösung zu Aufgabe 6

Das CAP-Theorem sagt aus, dass es nur entweder CA-, CP- oder PA-Datenbanken gibt. Im Falle von PA-Datenbanken wird Konsistenz „vernachlässigt“. Das Base-Modell zeigt Möglichkeiten auf, um die Konsistenz zwar nicht unmittelbar, aber im Endeffekt doch noch sicherzustellen. Die letztendliche Konsistenz ist auch in PA-Datenbanken extrem wichtig.

Lösung zu Aufgabe 7

Mittels der Einführung von NF^2 -Relationen und des Recursive-Union-Operators.

Lösung zu Aufgabe 8

Zunächst sind Datentypen (insbesondere variable Felder und eingebettete Relationen) mittels des Create-Type-Befehls zu definieren. Hier sind als Datentypen auch Unterobjekte zugelassen. Das Erzeugen einer Relation dieses Typs erfolgt dann mit einem syntaktisch erweiterten Create-Table-Befehl.

Lösung zu Aufgabe 9

```
CREATE OR REPLACE TYPE TPerson AS OBJECT
  ( Name CHARACTER (20), Adresse TAdresse, GebDatum DATE );
CREATE TABLE PersonalNeu
(   Persnr INTEGER, Person TPerson, Stand CHARACTER(6),
    Vorgesetzt INTEGER REFERENCES PersonalNeu, Gehalt NUMERIC(10,2),
    Beurteilung SMALLINT, Aufgabe CHARACTER(10) );
```

Lösung zu Aufgabe 10

```
SELECT * FROM THE (   SELECT Einzelposten FROM AuftragNeu
                    WHERE AuftrNr = 4 ) T
  WHERE T.Anzahl > 1 ;
```

Lösung zu Aufgabe 11

In relationalen Systemen ist dies notwendig, sobald eine Transaktion auf mehr als eine Datenbank schreibend zugreift, für die kein gemeinsames Log geführt wird.

Lösung zu Aufgabe 12

Je Datenbank muss ein Log geführt werden, zusätzlich ist ein globales Log erforderlich. Der Hauptaufwand ist in der Regel jedoch der hohe Kommunikationsoverhead.

Lösung zu Aufgabe 13

```
INSERT INTO AuftragNeu ( Auftrnr, Datum, Einzelposten, Kundnr, Persnr )
SELECT Auftrnr, Datum,
       CAST( MULTISET( SELECT Teilnr, Anzahl, Gesamtpreis
                       FROM Auftragsposten
                       WHERE AuftrNr = A.AuftrNr )
           AS ER_Einzelposten ) ,      Kundnr, Persnr
FROM Auftrag A ;
```