

Übungsblatt 6 – Concurrency Control

Aufgabe 1

Gegeben sind die Transaktionen T1 und T2, sowie die Objekte A und B. Eine Leseoperation wird mit r bezeichnet, eine Schreiboperation mit w. Gegeben sind die folgenden Schedules:

S1: (r₁(B), r₁(A), r₂(A), w₂(A), w₁(B), w₁(A))
S2: (r₂(A), r₂(B), r₁(B), r₁(A), w₁(A), w₁(B))
S3: (r₁(B), r₁(A), r₂(A), r₂(B), w₁(A), w₁(B))
S4: (r₂(A), r₁(B), r₁(A), w₁(A), w₁(B), w₂(A))

- Geben Sie für jeden dieser Schedules den Serialisierbarkeitsgraphen an.
- Welche dieser Schedules sind seriell oder serialisierbar?
- Geben Sie bei allen serialisierbaren Schedules den äquivalenten seriellen Schedule an.
- In welchen Schedules tritt ein Lost-Update-Problem auf? Geben Sie gegebenenfalls an, welcher Änderungsbefehl auf welchem Objekt verloren geht.
- Zeigen Sie bei allen nicht-seriellen Schedules, wie ein sperrbasiertes Verfahren die einzelnen Befehle synchronisieren würde. Geben Sie an, falls ein Deadlock auftreten würde.

$$\textcircled{7} \quad a) \quad S1: \begin{array}{l} r_1(A) \rightarrow w_2(A) : T_1 \rightarrow T_2 \\ r_2(A) \rightarrow w_1(A) : T_2 \rightarrow T_1 \end{array} = T_1 \leftrightarrow T_2$$

$$S2: \begin{array}{l} r_2(A) \rightarrow w_1(A) : T_2 \rightarrow T_1 \\ r_2(B) \rightarrow w_1(B) : T_2 \rightarrow T_1 \end{array} = T_2 \rightarrow T_1$$

$$S3: \begin{array}{l} r_2(A) \rightarrow w_1(A) : T_2 \rightarrow T_1 \\ r_2(B) \rightarrow w_1(B) : T_2 \rightarrow T_1 \end{array} = T_2 \rightarrow T_1$$

$$S4: \begin{array}{l} r_2(A) \rightarrow w_1(A) : T_2 \rightarrow T_1 \\ r_1(A) \rightarrow w_2(A) : T_1 \rightarrow T_2 \end{array} = T_1 \leftrightarrow T_2$$

b) seriell: S2

serialisierbar: S3

c) S3 seriell ist gleich wie S2:

$(r_2(A), r_2(B), r_1(B), r_1(A), w_1(A), w_1(B))$

d) In S1 wird $w_2(A)$ durch $w_1(A)$ am Ende überschrieben
Lost Update: $w_2(A)$

In S4 wird $w_1(A)$ durch $w_2(A)$ am Ende überschrieben
Lost Update: $w_1(A)$

e) S1: T_1 T_2

Lock R(B)
Lock R(A)

Lock X(B)
Lock X(A)

Lock R(A)
Lock X(A)

T_2 wartet

Da T_2 wartet, wartet T_1 auf T_2 und T_2 auf $T_1 \rightarrow$ Deadlock

S3: T_1 T_2

Lock R(B)
Lock R(A)

Lock X(A)

write(A)
Lock X(B)
write(B)
Commit

Lock R(A)
Lock R(B)

commit

T_1 wartet
hält Locks und
 T_1 kann fortfahren

S4:

T1

T2

Lock R(B)
Lock R(A)
Lock X(A)

Lock R(A)

Lock X(A)

T1 wartet, da T2 Lock R(A) hält
T2 wartet, da T1 Lock R(A) hält und
auf T2 wartet → Deadlock

Aufgabe 2

Das Problem der inkonsistenten Analyse soll in Oracle rekonstruiert werden. Dazu können Sie den folgenden, vereinfachten SQL-Code übernehmen. Für die zweite Transaktion (Überweisung) benötigen Sie einen zweiten Account, hier dbsysXY.

```
CREATE TABLE girokonto (  
  name VARCHAR2(20) primary key,  
  kontostand INT,  
  land VARCHAR2(20)  
);  
  
GRANT INSERT, SELECT, UPDATE ON girokonto TO dbsysXY;  
  
INSERT INTO girokonto VALUES ('A', 1000, 'D');  
INSERT INTO girokonto VALUES ('B', 1000, 'D');  
INSERT INTO girokonto VALUES ('C', 1000, 'D');  
INSERT INTO girokonto VALUES ('D', 1000, 'D');  
INSERT INTO girokonto VALUES ('E', 1000, 'D');  
INSERT INTO girokonto VALUES ('F', 1000, 'CH');  
INSERT INTO girokonto VALUES ('G', 1000, 'CH');  
INSERT INTO girokonto VALUES ('H', 1000, 'CH');  
INSERT INTO girokonto VALUES ('I', 1000, 'CH');  
INSERT INTO girokonto VALUES ('J', 1000, 'CH');  
COMMIT;
```

Es soll nun die Summe aller Kontostände berechnet werden. Um die Gleichzeitigkeit mit einer anderen Transaktion zu erzwingen werden zuerst die Kontostände aller deutschen Girokonten und danach aller schweizer Konten addiert.

```
SELECT SUM(kontostand) FROM girokonto  
WHERE land = 'D';  
  
SELECT SUM(kontostand) FROM girokonto  
WHERE land = 'CH';  
COMMIT;
```

Zwischen der Berechnung der beiden Teilsummen kann man nun eine Überweisung von 500 Euro durch eine andere Transaktion durchführen.

```
UPDATE eck.girokonto SET kontostand = kontostand - 500 WHERE  
  name = 'A';  
UPDATE eck.girokonto SET kontostand = kontostand + 500 WHERE  
  name = 'F';  
COMMIT;
```

Aufgaben:

- Testen Sie, ob das Problem der inkonsistenten Analyse eintritt.
- Ändern Sie das Isolation Level so, dass die Summenberechnung korrekt erfolgt.
- Vergleichen Sie das Verhalten von Oracle mit der in der Vorlesung vorgestellten klassischen Sperrverwaltung.