DES3UE Datenbanksysteme WS 2018 Übung 10

Abgabetermin: 12.12.2018, 13:30 Uhr

☐ **DES31UE Niklas** **Name** Niklas Veit **Aufwand in h** 10
☐ **DES32UE Niklas**
☒ **DES33UE Traxler** **Punkte** _____ **Kurzzeichen Tutor** _____

Hinweis:

Bilden Sie für die Ausarbeitung der Beispiele, insbesondere Beispiel 2.1, Gruppen und geben Sie in der Lösung den entsprechenden Verweis auf Ihren Partner/Ihre Partnerin an.

1. Zeitstempelverfahren (2,5 Punkte)

Überprüfen Sie die Serialisierbarkeit der in Abbildung 1 angegebenen Ausführung der Transaktionen T1, T2 und T3 mit Hilfe des (*nicht* optimierten) Zeitstempel-Verfahrens.

T1	T2	T3
		READ a
READ a		
WRITE b		
	WRITE d	
		WRITE d
READ c		
WRITE a		
	WRITE c	
		WRITE c
		READ b

Abbildung 1: Ausführung T1, T2 und T3

Welche Transaktionen werden abgebrochen, wenn den Transaktionen T1, T2 und T3 die Zeitstempel 75, 85 und 100 zugewiesen werden? Stellen Sie eine entsprechende Tabelle auf.

2. Vergabe und Entzug von Rechten (2 Punkte)

1. Gewähren Sie einem anderen Benutzer lesenden Zugriff auf Ihre Tabelle `DEPARTMENTS`. Lassen Sie sich von diesem Benutzer das Privileg zur Abfrage seiner Tabelle `DEPARTMENTS` erteilen und testen Sie den Zugriff.
2. Fragen Sie die View `USER_TABLES` und die View `ALL_TABLES` im Data Dictionary ab, um Informationen über die Tabellen anzuzeigen, die Ihnen gehören und auf die Sie zugreifen können. Bei der Abfrage auf die View `ALL_TABLES` schließen Sie die Tabellen aus, die Ihnen gehören.

3. COMMIT, SAVEPOINT und ROLLBACK (2,5 Punkte)

1. Erstellen Sie eine Tabelle MY_EMPLOYEE als Kopie der Tabelle employees, verwenden Sie dazu einen CREATE … AS SELECT … Befehl.
2. Erhöhen Sie das Gehalt aller Manager um 5 % (MGR und MAN). Schreiben Sie die veränderten Daten dauerhaft fest.

3. Ändern Sie das Gehalt aller Angestellten, deren Gehalt unter $ 5000 liegt, in $ 6100. Markieren Sie einen Zwischenpunkt in der Verarbeitung der Transaktion.
4. Leeren Sie die gesamte Tabelle und prüfen Sie ob die Tabelle leer ist.
5. Verwerfen Sie die letzte DELETE-Operation, ohne die vorherige UPDATE-Operationen zu verwerfen; prüfen Sie, ob die Änderungen aus Punkt 3.4 weiterhin vorhanden sind und schreiben Sie die Änderungen fest.

4. Isolationsstufen in Oracle I (Read Committed) (8 Punkte – 3+3+2)

1. In der Vorlesung wurde das Problem der verlorengegangenen Änderung (*lost update*) vorgestellt. Untersuchen Sie anhand eines selbstgewählten Schedules, ob dieses Problem auch in Oracle mit der Default-Einstellung der Isolationsstufe `READ COMMITTED` möglich ist. Verwenden Sie dazu das SQL-Skript ue10_schema.sql (enthält Beispielschemadefinition und Beispieldatensätze). Was kann ein Anwendungs- bzw. SQL-Programmierer (mit dieser Einstellung der Isolationsstufe) tun, um *lost updates* zu vermeiden? Begründen Sie Ihre Antwort.

2. Re-Initialisieren Sie Ihre Datenbank auf Basis des SQL-Skripts ue10_schema.sql und betrachten Sie die beiden folgenden Transaktionen T1 und T2 (mit der Default Isolationsstufe `READ COMMITTED`):

```
T1: Select * from emp where sal > 2800;
T1: Select * from dept where dname = 'RESEARCH';
T2: Select * from dept where dname = 'RESEARCH';
T2: Select * from emp where sal > 2800;
T1: Update emp set sal = sal*1.05 where sal > 2800;
T2: Update dept set dname = 'MARKETING' where dname = 'RESEARCH';
T2: Update emp set sal = sal*1.05 where sal > 2800;
T1: Update dept set dname = 'MARKETING' where dname = 'RESEARCH';
```

   Was stellen Sie bei diesem Ablauf fest? Was würde der Einsatz eines `SELECT ... FOR UPDATE` bei allen Select-Statements bewirken? Unter welchen Umständen können dadurch Deadlocks vermieden werden? Konstruieren Sie einen Fall (auf Basis des gegebenen Schedules) mit `SELECT ... FOR UPDATE`, in dem die Vermeidung des Deadlocks nicht möglich ist!

   Gibt es Möglichkeiten, Deadlocks in *jedem* Fall zu verhindern? Wenn ja, zählen Sie diese auf!

3. Ein weiteres Problem stellt das Non-Repeatable Read dar. Kann in Oracle bei der Isolationsstufe `READ COMMITTED` ein Non-Repeatable Read auftreten? Falls ja, geben Sie einen entsprechenden Beispiel-Schedule an und zeigen Sie Möglichkeiten auf, wie dies verhindert werden kann.

5. Isolationsstufen in Oracle II (Serializable) (9 Punkte – 3+3+3)

In der Isolationsstufe `SERIALIZABLE` verwendet Oracle das in der Vorlesung vorgestellte Mehrversionen-Concurrency-Control-Protokoll (mit der Bezeichnung Snapshot Isolation). Jede Transaktion arbeitet auf einer eigenen Version der Datenbank (snapshot), die den konsistenten Zustand aller committeter Daten zum Zeitpunkt des Transaktionsbeginns reflektiert. Leseaktionen einer Transaktion lesen also Werte so wie sie zu Beginn der Transaktion vorlagen (außer natürlich Werte, welche von dieser Transaktion selbst geschrieben wurden). Beim Schreiben wird eine neue Version des jeweiligen Objekts erzeugt, das bis zum erfolgreichen Commit nur für die eigene

Transaktion sichtbar ist.

1. Wiederholen Sie das Beispiel mit dem *lost update* Problem aus Aufgabe 4.1 mit der Isolationsstufe `SERIALIZABLE`. Kann hier das Problem des *lost updates* ebenso auftreten? Begründen Sie die sich ergebende Ausgabe!

2. Setzen Sie die Isolationsstufe auf `SERIALIZABLE` und testen Sie einen *nicht* serialisierbaren Ablauf, bei dem zwei parallele Transaktionen schreibend auf *dasselbe Objekt* zugreifen. Geben Sie die Ausgabe von Oracle an und begründen Sie diese!

3. Die Isolationsstufe `SERIALIZABLE` entspricht NICHT der Isolationsstufe `SERIALIZABLE` des SQL-2003 Standards, d. h. es können unter gewissen Bedingungen nicht serialisierbare Transaktionen fehlerfrei ausgeführt werden. Führen Sie eine Suche im Web durch, um entsprechende Anomalien herauszufinden, die in Snapshot Isolation auftreten können und beschreiben Sie diese kurz (mit jeweils einem kleinen Beispiel).