ETH Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Dr. G. Alonso *C. Hagen, B. Kemme*

AR 1: Transaktionsverwaltung

Praktische Übung 2 Abgabe: 15. 5. 1997

Einleitung

Ziel dieser Aufgabe ist es, sich mit der Wirkung von Sperren in der Transaktionsverwaltung in einem Datenbanksystem auseinanderzusetzen. Dies soll am Beispiel kleiner Anwendungszenarios durchgeführt werden. Die Arbeitsumgebung ist ORACLE mit der interaktiven Benutzerschnittstelle SQL*Plus und SQL*DBA.

Grundlage ist das *ORACLE7 Server Concepts Manual* (speziell Kapitel 10). Für weitere Informationen können folgende Manuals nützlich sein: *ORACLE7 Utilities User's Guide* für eine Beschreibung von SQL*DBA und *ORACLE7 Application Developer's Guide* für praktische Hinweise zum Einsatz von SQL-Statements, welche das Verhalten von Transaktionen beeinflussen. Die Manuals sind auf rif/raf online abrufbar mit `oradocm&`. Dafür muss allerdings das Oracle-Modul (siehe unten) geladen sein. Wählen Sie z.B. im Oradocm-Hauptfenster unter der Überschrift **Oracle7 Server Documentation, Release 7.2** *ORACLE7 Server Concepts Manual* aus (Doppelklick).

Verwendete Daten

Die Basis der Aufgaben ist eine kleine relationale Datenbank aus dem Verwaltungsbereich, welche folgende Relationen enthält:

Auto(Kennzeichen, Fahrgestellnummer, Farbe, *Halter:AHVNummer*)
Halter(AHVNummer, Name, Vorname, Wohnort)

(Primärschlüssel sind unterstrichen, Fremdschlüssel in *italic* gesetzt)

Starten Sie in drei Fenstern SQL*Plus, um parallele Benutzer zu simulieren. Experimentieren Sie mit den verschiedenen Möglichkeiten, welche Ihnen ORACLE in bezug auf die Transaktionsverwaltung bietet, um folgende Fragen zu beantworten. Ziehen Sie die oben erwähnten Handbücher zu Rate.

Aufgabe 1: Konsistenz in Oracle

a) Geben Sie eine Definition der *Konsistenz*, wie sie von Oracle aus gesehen wird, und vergleichen Sie diese mit der in der Vorlesung gegebenen Definition. Was wird unter *Lese-Konsistenz* verstanden, wie wird es in ORACLE eingesetzt und was wird dadurch vom System garantiert? Wie wirkt sich das konkret aus? Welche Effekte erwarten Sie aufgrund der Beschreibung in den Handbüchern?

b) Welches sind die Vorteile der Konsistenz, so wie sie in ORACLE definiert sind? Wo sind die Schwachstellen? Zeigen Sie anhand eines Beispiels, wie diese Art der Konsistenz zu Problemen führen kann. Zeigen Sie, welche Mittel dem Datenbankadministrator resp. dem Applikationsprogrammierer zur Verfügung stehen, um mit ORACLE Korrektheit im Sinne der Konfiliktserialisierbarkeit zu garantieren.

Um mit ORACLE arbeiten zu können, muss ein entsprechendes Environment in der aktuellen Shell gesetzt sein. Dies wird durch

module add oracle/stud

erreicht. Login und Passwort für Oracle haben Sie in der Übung erhalten. In SQL*Plus können Sie mittels

@<filename (mit Pfad, aber ohne .sql)>

sogenannte *Skripten* ausführen. Dies sind Befehlsdateien, deren Name auf .sql endet.

Legen Sie sich eine eigene Datenbank an, welche die eingangs erwähnten Relationen enthält und füllen Sie sie mit Tupeln. Benutzen Sie hierzu das Script

/home/ar0/pub/createAndFill.sql

c) Zeigen Sie die Auswirkungen der Option `SET TRANSACTION READ ONLY` bei Änderungsoperationen. Konstruieren Sie einen Fall, in welchem zu einer Zeit ein und dieselbe Query in den drei Fenstern unterschiedliche Resultate liefern.

Was sagt das Handbuch zu der Option `SET TRANSACTION READ ONLY`? Ordnen Sie sie in die Vorlesung ein. Zeigen Sie insbesondere die Zusammenhänge zur Mehrversionenverwaltung.

d) Im Zusammenhang mit Lesekonsistenz wird im ORACLE-Handbuch von einem möglichen Fehler geredet (`ORA-1555: snapshot too old`). Wann und weshalb tritt er auf? Welche Massnahmen stehen dem Datenbankadministrator und dem Applikationsprogrammierer zur Verfügung, um den Fehler zu vermeiden? Erklären Sie den Fehler im Umfeld der theoretischen Grundlagen der Mehrversionenverwaltung.

Aufgabe 2: Sperrverwaltung in Oracle

a) Welche Kategorien von Locks verwendet ORACLE? Wozu werden Sie verwendet? Zeigen Sie insbesondere den Zusammenhang zwischen den von ORACLE verwendeten Tablelocks (`RS, RX, S, SRX und X`) und den in der Vorlesung besprochenen Lockmodi auf.

b) Starten Sie in einem vierten Fenster den *SQL-DBA Monitor*. Dazu müssen Sie auf der Servermaschine (mittels `rlogin dblab1`; gleiches Passwort wie auf apollo und rif/raf) eingeloggt sein.

Nach dem Setzen der ORACLE-Umgebung mit *module add oracle/stud* rufen Sie das Programm `sqldba` auf. Mittels `connect <userid/password>` stellen Sie die Verbindung zur Datenbank her. Zum Anzeigen der Sperrtabelle dient der Befehl `monitor lock`. Geben Sie als `Username Filter` Ihr Oracle-Login an, und starten Sie den Monitor mittels `(Start)` (mit `<TAB>` Modus wechseln). Mit ESC-q verlassen Sie den Monitor wieder.

Notieren Sie, welche Sperren bei folgenden Operationen angefordert werden.

1. TA 1: Selektieren aller blauen Autos
2. TA 2: Ermitteln der Halter und Wohnorte aller blauen Autos
3. TA 1: Umzug von *Weikum, Gerhard* nach *Saarbruecken*
4. TA 2: Einführen einer neuen Konsistenzbedingung: (1) Alle blauen Autos werden hellblau gestrichen. (2) Alle Autos, deren Halter in Zürich wohnen, werden blau angemalt (somit sind nur noch jene Autos blau, deren Halter in Zürch wohnen...).
5. TA 1: COMMIT
6. TA 2: Ermitteln der Halter und Wohnorte aller blauen Autos
7. TA 2: COMMIT

Warum sind die erhaltenen Ergebnisse nicht zufriedenstellend? Gibt es eine Möglichkeit, befriedigendere Ergebnisse zu erhalten? Wenn ja, welche?

c) In Teilaufgabe 1c wurde die Wirkungsweise der Option `SET TRANSACTION READ ONLY` untersucht und gezeigt, dass damit *repeatable read* ermöglicht wird. Betrachten sie nun mit dem Lockmonitor die angeforderten Sperren für folgendes Szenario:

1. TA 1: SET TRANSACTION READ ONLY
2. TA 1: SELECT * FROM HALTER WHERE NAME='WEIKUM'
3. TA 2: UPDATE HALTER SET NAME='SCHEK' WHERE NAME='WEIKUM'
4. TA 2: COMMIT
5. TA 1: SELECT * FROM HALTER WHERE NAME='WEIKUM'

Das beobachtbare Ergebnis ist für diesen Fall korrekt. Beim genauen Betrachten der angeforderten Sperren wird ersichtlich, dass *repeatable read* für TA1 trotzdem nicht garantiert werden kann. Ändern Sie die Anweisung der TA2 so, dass die zweite Query der TA1 ein anderes Ergebnis liefert als die erste.

d) Reinitialisieren Sie Ihre Datenbank (mit `createAndFill`) und betrachten sie folgende Transaktionen:

1. TA 1: SELECT * FROM AUTO WHERE FARBE='BLAU'
2. TA 1: SELECT * FROM HALTER WHERE NAME='WEIKUM'
3. TA 2: SELECT * FROM HALTER WHERE NAME='WEIKUM'
4. TA 2: SELECT * FROM AUTO WHERE FARBE='BLAU'
5. TA 1: UPDATE AUTO SET FARBE='VIOLETT' WHERE FARBE='BLAU'
6. TA 2: UPDATE HALTER SET NAME='SCHEK' WHERE NAME='WEIKUM'
7. TA 2: UPDATE AUTO SET FARBE='GRAUGRUEN' WHERE FARBE='BLAU'
8. TA 1: UPDATE HALTER SET NAME='SCHEK' WHERE NAME='WEIKUM'

Was bewirkt `SELECT FOR UPDATE`? Welche Sperren werden mit dieser Anweisung angefordert? Unter welchen Umständen können dadurch Deadlocks vermieden werden? Konstruieren Sie einen Fall, in dem dies nicht möglich ist!

Gibt es eine Möglichkeit, Deadlocks in jedem Fall zu verhindern? Wenn ja, was sind die Konsequenzen?