

## Praktikum 6

### I. JDBC

#### Realisierung einer Java-Applikation mit Zugriff auf eine relationale Datenbank

**Vorbereitung zum Praktikum:** Teilaufgabe 1. – 5. lauffähig implementiert.

Im Rahmen dieser Praktikumsaufgabe sollen auf der Tabelle *Autor* des **Datenmodells Bibliothek** des dritten und vierten Praktikums lesende und schreibende Zugriffe aus einer Java-Applikation erfolgen:

1. Schreiben Sie eine Java-Applikation, die eine **Connection** zur Oracle-Datenbank aufbaut. Erfragen Sie *User* und *Passwort* über die Console.
2. Setzen Sie in der Java-Applikation **AutoCommit** für diese Session auf „false“ und erfragen Sie einen gewünschten **Isolationlevel** für die aktuelle Session über die Console, der dann von der Java-Applikation gesetzt wird.
3. Geben Sie an der Console die **aktuelle Liste der Autoren** in der Datenbank aus.
4. Fügen Sie ein **PreparedStatement** über die Java-Applikation aus, über das ein **Insert** eines neuen Autors in der Datenbank durchgeführt werden können.  
Die hierfür erforderlichen Daten werden wieder über die Console eingelesen.
5. Fragen Sie anschließend den Benutzer über die Console, ob erneut eine **Liste der Autoren** an der Console ausgegeben werden soll. Falls der Benutzer dies wünscht, zeigen Sie die Liste an der Console an. Andernfalls wird die Applikation beendet.

*Hinweis:* Die Teilaufgaben 1. – 5. können zu Hause oder im DB-Labor (D14/111) auf einem beliebigen relationalen Datenbanksystem vorbereitet und vollständig getestet werden!

#### Hinweise zur Arbeit mit Eclipse:

##### 1. Projekt mit Package und Klasse anlegen

Anlegen eines neuen Projekts:

File -> New -> Project

innerhalb des Projekts: Anlegen eines neuen Packages:

File -> New -> Package

innerhalb des Packages: Anlegen einer neuen Klasse:

File -> New -> Class

##### 2. Programmieren

##### 3. Programm Ausführen

- beim ersten Mal:

Run -> Run (Projekt und Klasse auswählen)

- beim jeder weiteren Ausführung:

Run -> Run Last Launched (o. CTRL + F11)

II. Transaktionsmanagement – Isolationlevel  
im konkurrierenden Zugriff einer SQL\*PLUS-Session  
mit einer Java-Applikation

**Diese Aufgabe wird während des Praktikums durchgeführt, nachdem der I. Teil von den Betreuern abgenommen wurde.**

6. Starten Sie zu Beginn eine SQL\*PLUS-Session auf Ihrem Arbeitsplatzrechner im Labor mit dem gewohnten Login und starten Sie über Ihre Java-Applikation ebenfalls eine Oracle-Session.

Ändern Sie den **Autocommit-Status** der aktuellen Session auf OFF wie folgt:

SET AUTOCOMMIT OFF (Anzeige des aktuellen Status: SHOW AUTOCOMMIT)

Mit der folgenden Anweisung können Sie den ISOLATION\_LEVEL der aktuellen Session ändern:

ALTER SESSION SET ISOLATION\_LEVEL = <Serializable/Read Committed>;

Führen Sie die Operationen innerhalb der beiden nun offenen Oracle-Sessions (SQL\*PLUS und Java-Applikation) auf der Datenbank *Bibliothek* entsprechend der nachfolgenden Tabelle interaktiv aus:

1	<b>BOT T1 (SQL*PLUS)</b>	
2	T1 liest Autor	
3	T1 schreibt Autor	
5	T1 liest Autor	
6		<b>BOT T2 (Java-Applikation)</b>
7		T2 liest Autor
8		T2 schreibt Autor
9	T1 liest Autor	
10	<b>Commit T1</b>	
11		T2 liest Autor
12		<b>Commit T2</b>

Protokollieren Sie die Ergebnisse der beiden Transaktionen auf der Oracle Datenbank unter folgenden Isolationleveln: **Serializable** und **Read Committed**.

Erklären Sie das beobachtete, unterschiedliche Verhalten.

7. Führen Sie die konkurrierende Transaktion und die Java-Applikation so aus, dass ein **Dead Lock** entsteht.

Praktikum	Freitag 3y	Freitag 3x
P6	16.06.2006 30.06.2006	23.06.2006