



Prof. Dr. Bernhard Seeger
Jana Holznigenkemper, M.Sc.
Andreas Morgen, M.Sc.

Übungen zur Vorlesung
Datenbanksysteme I

Abgabe: 05.07.2019,
bis **spätestens** 10:00 Uhr
über die ILIAS Plattform

Übung 9

Hinweis: Auf diesem Übungszettel ist es Ihr Ziel verschiedene Aufgaben mit Java, JDBC und PostgreSQL zu lösen. Für diese ist es nötig PostgreSQL lokal zu installieren (<https://www.postgresql.org/>). Um aus Java auf die Datenbank zugreifen zu können, müssen sie außerdem den korrekten JDBC-Treiber (<https://jdbc.postgresql.org/>) installieren.

Aufgabe 9.1: JDBC und Metadaten

(6 Punkte)

Schreiben Sie ein Java-Programm, das eine SQL-Select-Anfrage vom Nutzer entgegennimmt und das Ergebnis auf dem Bildschirm (Konsole) in Tabellenform darstellt! Nutzen Sie die Metadaten des Ergebnisobjekts, um den Tabellenkopf zu beschriften.

Aufgabe 9.2: JDBC und Batch Inserts (1+2+2+2+2)

(9 Punkte)

In dieser Aufgabe untersuchen Sie, wie schnell die Daten einer Tabelle über JDBC geladen werden können. Im Code-Ordner im ILIAS finden Sie eine Datei points.csv. Diese Datei enthält 250.000 Zeilen. Jede Zeile enthält eine Komma-separierte Liste von drei Zahlen, welche einem Eintrag in der folgenden Tabelle entspricht:

```
CREATE TABLE Points (  
    point_id int primary key,  
    point_x int not null,  
    point_y int not null  
);
```

a) Implementieren Sie zwei Methoden:

public static void createTablePoints() – legt die Tabelle Points an.

public static void dropTablePoints() – löscht die Tabelle Points an.

Hinweis: PostgreSQL bietet ab Version 9.1 die Option **create table if not exists ...**

b) Schreiben Sie ein JDBC-Programm für das Laden der Daten in die Tabelle Points.

public static void loadTuples(String inputPath)

Verwenden Sie dabei die Klasse Statement.

c) Implementieren Sie das Ladeprogramm unter der Verwendung von PreparedStatements.

public static void loadTuplesPrepared(String inputPath)

d) Implementieren Sie das Ladeprogramm mit Hilfe von Batch-Inserts (siehe Folie 485ff).

```
public static void loadTuplesBatch(String inputPath, int batchSize)
```

Der Parameter batchSize gibt hierbei die Größe der Batches an.

Hinweis: Betrachten Sie die Methoden `PreparedStatement.addBatch(...)` und `executeBatch()`.

e) Vergleichen Sie die Laufzeiten der Methoden `loadTuples`, `loadTuplesPrepared` und `loadTuplesBatch`. Verwenden Sie dabei Batch-Größen von 10, 100 und 1000 SQL-Insert-Befehlen, um die Tabelle `Points` mit Daten aus `points.csv` zu befüllen. Geben Sie jeweils die absoluten Laufzeiten und die relative Geschwindigkeitsänderung aus.

Aufgabe 9.3: JDBC mit Transaktionen

(5 Punkte)

Betrachten Sie die ERP-Datenbank aus der Vorlesung. Verwenden Sie das *Skript zum Erzeugen der Datenbank* aus dem Code-Order im Ilias.

Implementieren Sie eine Funktion `youAreFired(pnr)` für die ERP-Datenbank. Diese löscht die Person mit der angegebenen Personalnummer aus der Datenbank genau dann, wenn es für jede Maschine, die diese Person bedienen kann, mindestens eine weitere Person gibt, die diese Maschine ebenfalls bedienen kann.

Löschen Sie dazu zunächst die Einträge zu der übergebenen `pnr` aus der Datenbank. Überprüfen sie anschließend, ob die gerade beschriebene Bedingung gilt und führen Sie gegebenenfalls einen Rollback durch. Orientieren Sie sich bei dieser Aufgabe an den Folien 490/491 im Skript.