



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

Übungen zur Vorlesung
Datenbanksysteme I

Abgabe: 12. 07. 2019,
bis **spätestens** 10:00 Uhr
über die ILIAS Plattform

Übung 10

Aufgabe 10.1: Table Functions (4+4+2)

(10 Punkte)

Schreiben Sie eine *Table Function* `rtable(n)` (vgl. Gaussumme auf Folie 501), die für eine ganze Zahl n eine Tabelle `RandomTable(i, j, diff)` mit n zufällig generierten Einträgen erzeugt. Die Einträge der erzeugten Tabelle sollen dabei aus jeweils zufällig generierten Integeren i und j , sowie deren Differenz $diff$ bestehen.

Schreiben Sie nun in Java unter Verwendung von JDBC eine Methode, die analog in der Datenbank zwei zufällig generierte Tabellen mit n Einträgen erzeugt. Verwenden Sie bei der ersten Tabelle einzelne Inserts und bei der zweiten Tabelle Batch-Inserts mit der Größe 1000. Messen Sie jeweils die Laufzeiten und geben Sie diese auf der Konsole aus.

Vergleichen Sie abschließend die Laufzeiten der verschiedenen Ansätze für ein hinreichend großes n (je nach Hardware bspw. 500.000). Was fällt auf? Begründen Sie die Beobachtungen kurz.

Aufgabe 10.2: B⁺-Baum (4+1+2+3)

(10 Punkte)

a) Fügen Sie folgende Zahlen in der angegebenen Reihenfolge in einen leeren B⁺-Baum vom Grad (3, 2) ein:

11, 4, 17, 20, 1, 2, 5, 9, 16, 6, 34, 12, 23, 42, 8, 13

Zeichnen Sie den Baum nach jedem Überlauf und das Endresultat.

- b) Geben Sie den Pfad (Knoten) an, um das Element mit der Nummer 20 zu suchen.
- c) Geben Sie den Pfad (Knoten) an, um eine Bereichsanfrage 11 bis 15 durchzuführen.
- d) Löschen Sie die Elemente 4, 2 und 5 und stellen Sie sicher, dass die Invarianten des Baums anschließend wieder erfüllt sind.

Aufgabe 10.3: Table Functions (Bonus-Aufgabe)

(+5 Punkte)

Schreiben Sie analog zur Gaussumme (Folie 501) eine *Table Function* `fib(n)`, die die ersten n Fibonacci Zahlen ausgibt. Das Schema der Tabelle soll wie folgt aussehen:

`Fibonacci(i, fib)`