

Datenbank-Praktikum_4-Bericht

Was wurde gemacht?

Wir haben am Anfang alle Daten von der Datenbank gelöscht, das Schema jedoch erhalten.

Dann hat schließlich auch unsere Massendatengenerierung begonnen. Wir haben insgesamt 1000 User erstellt, die jeweils bis zu 100 Spiele gespielt haben. Jedes Spiel hat 20 Fragen und diese Fragen gehören zu 2 – 6 Kategorien.

Mit Was haben wir gearbeitet?

Wir haben mit unserem Heimnotebook gearbeitet und haben eine lokale Datenbank benutzt anstatt eine Datenbank mit VPN zu nutzen, da die Massendaten sehr langsam über VPN verarbeitet werden können. Als IDE wurde IntelliJ verwendet.

Wie lange dauert die Massendatengenerierung bei Ihrer Anwendung?

25 min und 40 Sekunden

Wie haben Sie eine schnelle Erzeugung der Daten bewirkt?

- durch das batch-writing (hat am meisten gebracht)
- durch flush() und clear()
- durch Deaktivierung von Log-files

Wie war denn die Laufzeit vor Ihrer Änderung?

Die Laufzeit hat bei uns ohne Optimierung über 40 min gedauert. Wir haben jedoch viele Sachen nicht verwendet bzw. gemacht, wie zum Beispiel Batch-writing, Log files deaktivieren, flush and clear nicht ordentlich verwendet.

```
<property name="eclipselink.jdbc.batch-writing" value="JDBC"/>
<property name="eclipselink.jdbc.batch-writing.size" value="1000"/>
<property name="eclipselink.logging.level" value="off"/>
```

Wie benutzen Sie Transactions und warum?

- die Transaktionen werden zum Schluss des Spiel geöffnet und geschlossen, nachdem alle Daten, die persistiert werden sollen, in die Klassen geschrieben worden sind
- die Daten für die Statistik werden ganz zum Schluss persistiert, da man dadurch ständiges redundantes Persistieren vermeiden kann.

Wie verwenden Sie flush(), clear(), etc. und warum?

- bei jedem 100 Spieler, wird flush und clear aufgerufen, um den Persistenzkontext nicht groß werden zu lassen.
- Es optimiert dadurch die Laufzeit
- flush(): sorgt für die Synchronisation mit der Datenbank
- clear(): leert den Persistenzkontext und detached die aktuell gemanagten Entities