

Fragen, Antworten, Kommentare

Die Online-Befragung zur gewählten alternativen Veranstaltungsform ist online. Bitte bis zum 12.1 ausfüllen: <https://forms.gle/FSv4BCM6YdhHhneG6>. Danke an alle bisherigen teilnehmenden Personen, wer noch nicht mitgemacht hat, bitte mitmachen.

Frage: Da wir in der Vorlesung noch nicht auf NoSQL-Datenbankmanagementsystemen eingegangen sind, wollte ich nochmal nachfragen wie es dort mit der Bewertung aussieht. Da wir uns selbständig in dieses Thema einarbeiten müssten wollte ich nachfragen ob dies auch in der Bewertung berücksichtigt wird oder wird die selbständige Einarbeitungszeit in das Thema nicht berücksichtigt und beides wird komplett gleich gewertet.

Klar ist, dass eine NoSQL-Datenbank eine höhere Einarbeitungszeit hat, da erst das generelle Konzept verstanden werden muss. Dies wird auch im Bericht deutlich, da das Kapitel zu den „Grundlagen“ deutlich länger wird, da das Konzept vorgestellt werden muss. Die nachfolgenden Kapitel sind dann teilweise kürzer oder nur ein Satz, der begründet, warum das Thema nicht zum gewählten Datenbanktyp passt, die Analyse der Anfragemöglichkeiten ist ähnlich aufwändig. Zu beachten ist aber, dass auch relationale Datenbanken einiges an Einarbeitung kosten, sei es um zusätzliche Werkzeuge zu studieren, genau auf die Datentypen zu schauen, die Mächtigkeit der unterstützen SQL-Anfragen zu analysieren, die Transaktionssteuerung zu betrachten und ob und wie Trigger umgesetzt werden zu studieren. Für sehr gute Ergebnisse schätze ich den Arbeitsaufwand ähnlich ein.

Frage: Wie funktioniert das mit dem rs.next() genau?

Etwas abstrakter handelt es sich bei rs um ein Iterator-Objekt, wie es aus Java und C++ bekannt ist. Dort funktionieren Schleifen typischerweise mit folgender Struktur.

```
while (it.hasNext()) {
    PassendeKlasse tmp = it.next(); // damit wird Iterator weitergesetzt
    ...
}
```

Eine vergleichbare Struktur sieht beim ResultSet anders aus.

```
while (rs.next()){
    String nurEinBeispiel = rs.getString(1);
    ...
}
```

Man erkennt, das next() zwar in beiden Fällen den Iterator weitersetzt, es aber sonst unterschiedliche Funktionen hat. Beim ResultSet wird auf hasNext() verzichtet, dafür muss der Wert über andere Methoden gelesen werden. Weiterhin muss beim ResultSet einmal next() gemacht werden, damit der erste Wert gelesen werden kann, was die Schleifenstruktur recht einfach macht. Beide Ideen zur Umsetzung von Iteratoren sind funktional gleichwertig. Sollte man selbst einen Iterator schreiben, sollte man abhängig von der Programmiersprache eine der Varianten nehmen. Da es das Konzept des ResultSet länger als Java gibt, wurde hier die Idee aus der DB-Welt übernommen.

Da das Thema SQL-Injection in der Vorlesung etwas zu kurz kommt, möchte ich auf folgenden Klassiker hinweisen: <https://xkcd.com/327/>

In der letzten Praktikumsaufgabe von Blatt 10 wurde der Transaktionsbegriff an praktischen Beispielen experimentell kennengelernt. Das Thema wird am Ende der Vorlesung aufgegriffen. Es sei angemerkt, dass es sich bei *Transaktionen* um einen der wichtigsten, wohl den wichtigsten Begriff der Datenbankveranstaltung handelt. Das Thema, dass mehrere Prozesse sich fachlich überlappen, wird auch in anderen Veranstaltungen behandelt. „Spoiler Alert“: Es gibt nicht die immer passende Lösung sondern man muss wissen warum eine Lösung die passendste zu jeweiligen Aufgabenstellung ist.

Werbung: Die deklarative Programmierung verfolgt den Ansatz aufzuschreiben was man haben möchte und nicht iterativ alle Befehle zu schreiben, wie die Berechnung durchzuführen ist. SQL ist ein Beispiel dafür. Diese Idee ist Grundlage der Programmiersprache Prolog, die auf klassischer Logik basiert und ein größerer Bestandteil des Wahlfachs Logik ist. Ein kleines einführendes Beispiel sieht wie folgt aus.

```
note(42,'Logik', 1.3).           % Fakten
note(42,'Mathe1', 1.7).
note(43,'Logik', 5.0).
studi(42, 'Leila').
studi(43, 'Heinz').

?- studi(X,Name), note(X,'Logik',Wert). % Anfrage, X, Name und Wert
                                         % (Großbuchstaben am Anfang)
                                         % sind Variablen

X = 42,                               % Ergebnisse
Name = 'Leila',
Wert = 1.3 ;
X = 43,
Name = 'Heinz',
Wert = 5.0.
```

In der Veranstaltung werden zunächst die Grundlagen der Logik (Aussagenlogik und Prädikatenlogik) geschaffen und in den Praktika u. a. in Programmieraufgaben eingeübt. Danach steht die Programmierung mit Prolog im Mittelpunkt. Ein weiterer zentraler Veranstaltungsteil beschäftigt sich mit der Übertragung der Ideen auf praktische Anwendungen in Business Rules Engines. Als Beispiel wird Drools genommen, das auch Bestandteil zweier von mir betreuter Bachelorarbeiten war. Die Veranstaltung findet im 2+2-Format statt, die Prüfungsform ist eine Hausarbeit. Es ist geplant die Vorlesung über Videos und Praktika online abzuhalten; dies kann sich aber noch ändern.