



Technische  
Universität  
Braunschweig



ifis

Institut für Informationssysteme  
Technische Universität Braunschweig



# SQL Alchemist - Teamprojekt

Tobias Grünhagen, Philip Holzhüter, Tobias Runge, 06.08.2015

# Motivation

- Hohe Durchfallquote bei RDB1
  - ⇒ Zurückzuführen auf mangelnde Kenntnisse im Umgang mit SQL
  - ⇒ SEP zur Entwicklung des Lernspiels „SQL Alchemist“
- Grundlage für Überprüfung von formal korrektem SQL
- SEP hauptsächlich Frontend, Nutzerinteraktion



# Inhalt

- Themenüberblick
- Allgemeines
- Parsen
- Taskmanagement
- Codegeneration
- Zusammenarbeit mit dem SEP
- Fazit

- Themenüberblick
- Allgemeines
- Parsen
- Taskmanagement
- Codegeneration
- Zusammenarbeit mit dem SEP
- Fazit

# Themenüberblick

- Einlesen und validieren von XML-Aufgaben
- Überprüfen von SQL in Aufgaben
- Anlegen von Datenbanken, Tabellen und Datensätzen
- Erzeugen von neuen Datensätzen
  - Nutzereingaben
  - Aufgabenstellungen
- Überprüfen von Nutzerstatements

# Inhalt

- Themenüberblick
- **Allgemeines**
- Parsen
- Taskmanagement
- Codegeneration
- Zusammenarbeit mit dem SEP
- Fazit

# Allgemeines

- Programmiersprache: Java 8
- Entwicklungsumgebung: NetBeans IDE 8.0.2
- Genutzte Bibliotheken/Tools
  - Git zur Versionsverwaltung
  - SBT zur Projektverwaltung
  - H2 Datenbank
  - JUnit für Tests
  - Typesafe für die Pfadverwaltung
  - Fluttercode.datafactory für die Erstellung zufälliger Werte

# Inhalt

- Themenüberblick
- Allgemeines
- **Parsen**
- Taskmanagement
- Codegeneration
- Zusammenarbeit mit dem SEP
- Fazit



# Parsen

- Validierung der XML Struktur
- XML-Aufgabe mit Hilfe der Tags in Java Objekte überführen
  - Header
  - Relation
    - Extraktion der einzelnen Informationen durch einen eigenen Parser
  - Exercise
- Validierung der SQL-Syntax mit Hilfe einer Memory-Instanz DB

# Inhalt

- Themenüberblick
- Allgemeines
- Parsen
- **Taskmanagement**
- Codegeneration
- Zusammenarbeit mit dem SEP
- Fazit

# Taskmanagement

- Einlesen der XML-Aufgaben
- Erstellung einer Task pro Teilaufgabe
- Initialisierung einer Datenbank beim Starten einer Task
  - Falls sie noch nicht existiert
  - Mehrere Nutzer können die gleiche DB verwenden (keine neuen DBs)
- Überprüfung der Userstatements

# Taskmanagement - Ablauf

- Sämtliche Inhalte der XML-Files in Listen speichern
  - Task erstellen mit Angabe des DB-Typs (local, memory, server)
    - Überprüfen, ob Task bereits existiert
    - DB erstellen, falls nicht existent (CREATE TABLE bzw. INSERT)
    - Ansonsten: Task laden, Spielerzahl erhöhen
  - Task beenden
    - Spielerzahl erniedrigen
    - DB löschen, falls kein Spieler mehr spielt

# Inhalt

- Themenüberblick
- Allgemeines
- Parsen
- Taskmanagement
- **Codegeneration**
- Zusammenarbeit mit dem SEP
- Fazit

# Codegeneration

- Codegenerierungsmöglichkeit nach dem Erstellen der DB-Tabellen und des Tasks
  - Nutzerspezifisch
  - Zufällig
  - Aufgabenspezifisch
- Erzeugung/Nutzung von Generierungstupeln
  - Auslesen aus dem XML-Dokument
  - Generierung anhand der Tabellendaten
  - Parsen des SELECT-Reference-Statements

# Codegeneration - Generierungstupel (1)

- Aufbau: Anzahl;Ref;Spalte1;Spalte2;...
  - Anzahl
    - Bedeutung: Anzahl der erzeugten Datensätze
    - Aufbau: Zahl oder Funktion „span“ mit zwei Parametern
  - Ref
    - Bedeutung: Generierte Daten sollen auf andere Tabelle referenzieren (none, refAll, refRandom)
    - Aufbau: RefType, referenzierter Tabellename, referenzierter Spaltenname

# Codegeneration - Generierungstupel (2)

- Spalte
  - Spaltenspezifische Daten (Generierungsfunktion)
  - Aufbau: Funktionsname,Parameter1,Parameter2
  - Funktionen (Auswahl)
    - random: Zufälliger Eintrag
    - Min: mindestens größer als Parameter
    - Max: maximal so groß wie Parameter
    - Gauss: Werte nach gausscher Normalverteilung
    - ...



# Codegeneration - Implementierung

- Tupelinformationen auslesen
- Spalten und Referenzierungsinformationen bestimmen und Daten auslesen
- Spaltenfunktionen durchgehen
  - Referenzierung beachten
  - Wert mit Funktion und Parametern generieren
- Primary Keys zum Schluss gesondert bestimmen
- INSERT-Statement mit allen Daten erstellen und ausführen

# Inhalt

- Themenüberblick
- Allgemeines
- Parsen
- Taskmanagement
- Codegeneration
- Zusammenarbeit mit dem SEP
- Fazit

# Zusammenarbeit mit SEP

- E-Mail-Verkehr über Probleme und Fragen bezüglich unserer API
- Persönliche Treffen bei größeren Problemen
- Über viele Probleme erst sehr spät diskutiert
  - Trotzdem in den meisten Fällen eine gute Lösung erarbeitet

# Inhalt

- Themenüberblick
- Allgemeines
- Parsen
- Taskmanagement
- Codegeneration
- Zusammenarbeit mit dem SEP
- **Fazit**

# Fazit (1)

- Erweiterbare API zur Bearbeitung von SQL-Aufgaben
  - Liest XML-Aufgaben ein
  - Erstellt eine Datenbank mit zufälligen Daten
  - Überprüft Nutzer-Statements
- Interaktives Spiel regt zum SQL-Lernen an

⇒ Sieg beim TDSE 😊

## Fazit (2)

**Vielen Dank für die Aufmerksamkeit!**



# Quellen und Verweise

- Verschiedene bereits vorgestellte Java-Bibliotheken und Versionsverwaltungssoftwares
- Bildquelle: David Wille, Tim Wesemeyer:  
<https://www.tu-braunschweig.de/isf/news/tdse2015>