

Rückwärtssalto

JDBC 2.Aufgabe



January 7, 2015

TGM XX

Reiländer Manuel, Hüseyin Bozkurt

[1. Aufgabenstellung 2](#_Toc411449744)

[2. Zeitaufwand/-abschätzung 3](#_Toc411449745)

[3. Diagramme 3](#_Toc411449746)

[Das erstellte Diagramm 3](#_Toc411449747)

[4. Things I did 3](#_Toc411449748)

[5. Lessons Learned 3](#_Toc411449749)

[6. Quellenangabe 3](#_Toc411449750)

# Aufgabenstellung

Erstelle ein Java-Programm, dass Connection-Parameter und einen Datenbanknamen auf der Kommandozeile entgegennimmt und die Struktur der Datenbank als EER-Diagramm und Relationenmodell ausgibt (in Dateien geeigneten Formats, also z.B. PNG für das EER und TXT für das RM)

Verwende dazu u.A. das ResultSetMetaData-Interface, das Methoden zur Bestimmung von Metadaten zur Verfügung stellt.

Zum Zeichnen des EER-Diagramms kann eine beliebige Technik eingesetzt werden für die Java-Bibliotheken zur Verfügung stehen: Swing, HTML5, eine WebAPI, ... . Externe Programme dürfen nur soweit verwendet werden, als sich diese plattformunabhängig auf gleiche Weise ohne Aufwand (sowohl technisch als auch lizenzrechtlich!) einfach nutzen lassen. (also z.B. ein Visio-File generieren ist nicht ok, SVG ist ok, da für alle Plattformen geeignete Werkzeuge zur Verfügung stehen)

Recherchiere dafür im Internet nach geeigneten Werkzeugen.

Die Extraktion der Metadaten aus der DB muss mit Java und JDBC erfolgen.

Im EER müssen zumindest vorhanden sein:

korrekte Syntax nach Chen, MinMax oder IDEFIX

alle Tabellen der Datenbank als Entitäten

alle Datenfelder der Tabellen als Attribute

Primärschlüssel der Datenbanken entsprechend gekennzeichnet

Beziehungen zwischen den Tabellen inklusive Kardinalitäten soweit durch Fremdschlüssel nachvollziehbar. Sind mehrere Interpretationen möglich, so ist nur ein (beliebiger) Fall umzusetzen: 1:n, 1:n schwach, 1:1

Kardinalitäten

Fortgeschritten (auch einzelne Punkte davon für Bonuspunkte umsetzbar)

Zusatzattribute wie UNIQUE oder NOT NULL werden beim Attributnamen dazugeschrieben, sofern diese nicht schon durch eine andere Darstellung ableitbar sind (1:1 resultiert ja in einem UNIQUE)

optimierte Beziehungen z.B. zwei schwache Beziehungen zu einer m:n zusammenfassen (ev. mit Attributen)

Erkennung von Sub/Supertyp-Beziehungen

# Zeitaufwand/-abschätzung

Versuch 1: Gegebene Zeit 32h

Zeitaufwand🡺 2h 07.01.2015(Unterricht): Extraktionen der Metadaten

2h: 14.01.2015(Unterricht): Verarbeitung, parsen der args

15.01.2015 fertigstellung des RM (4h)

Umsetzung des Codes mittels Decorator-Pattern(4h, noch in arbeit)

# Diagramme

# Das erstellte Diagramm

# Things I did

*Siehe Protokoll.docx* | *Protokoll.pdf*

# Things left

* Relationen werden falsch ausgelesen, foreign keys zeigen auf die Tabelle in der sie sich befinden -> siehe output -> korrigieren
* ERD zeichnen

# Lessons Learned

Die Variante zum *Speichern* der Informationen (Primary Key, Foreign Key, …) war nicht wirklich gut überlegt und führte im Nachhinein zu einer komplizierten String Bearbeitung.

Besser wäre gewesen die gespeicherten Informationen mit Hilfe des Decorator-Patterns abzubilden.

*To be continued …*

# Quellenangabe

JDBC – Folien (Moodle)

<http://www.java2s.com/Code/Java/Database-SQL-JDBC/GetPrimaryKeyColumnFromATable.htm>

<http://visjs.org/docs/index.html>