Rückwärtssalto

Hackenberger, Hollander

**4AHIT**

Informationssysteme

14. Jänner 2015

Inhaltsverzeichnis

1. Aufgabenstellung 3

2. Requirements Analyse 4

3. Aufwandsschätzung 4

4. Zeitaufzeichnung 5

5. Design-Überlegung 6

6. UTA 6

7. Ausführen 7

# 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

# Requirements Analyse

* Programm muss über CLI Options steuerbar sein
* Programm muss Fehlermeldungen bei falschen oder fehlenden CLI Options ausgeben
* --help Option muss verfügbar sein
* Es soll gewählt werden können ob nur das Relationenmodell, ER-Diagramm oder beides generiert werden soll
* Programm muss zumindest mit MySQL funktionieren
* Output-Files müssen plattformunabhängig sein
* ER Diagramm nach Chen Notation
* Alle Tabellen als Entitäten
* Alle Datenfelder der Tabellen als Attribute
* Primärschlüssel gekennzeichnet
* Beziehungen
* Kardinalitäten
* Zusatzattribute Unique, Not Null

# Aufwandsschätzung

|  |  |  |
| --- | --- | --- |
| **Arbeitspaket** | **geschätzte Dauer** | **Person/en** |
| Designüberlegung | 60 Minuten | Hackenberger, Hollander |
| Sources generieren und Error frei machen | 10 Minuten | Hackenberger |
| Implementierung Objekt Struktur | 45 Minuten | Hollander |
| Implementierung der JDBC Parsers | 120 Minuten | Hollander |
| Implementierung RMExporter | 100 Minuten | Hackenberer |
| Implementierung ERExporter | 150 Minuten | Hackenberger |
| Testen | 90 Minuten | Hackenberger, Hollander |
| Dokumentation | 120 Minuten | Hackenberger, Hollander |
| **Gesamt** | **11 Stunden, 35 Minuten** | **-** |

# Zeitaufzeichnung

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Arbeitspaket** | **Datum** | **von** | **bis** | **Dauer** | **Person/en** |
| Designüberlegung und Source generiert | 12/29/14 | 14:00 | 18:00 | 4 Stunden | Hackenberger, Hollander |
| Objektstruktur implementiert | 01/05/15 | 13:00 | 16:00 | 3 Stunden | Hollander |
| RMExporter Implementierung begonnen | 01/05/15 | 13:00 | 15:00 | 2 Stunden | Hackenberger |
| Protokoll angefangen | 01/07/15 | 8:00 | 9:40 | 100 Minuten | Hackenberger |
| JDBC Parser angefangen | 01/14/15 | 8:20 | 9:40 | 80 Minuten | Hollander |
| Protokoll fortgesetzt | 01/14/15 | 8:20 | 9:40 | 80 Minuten | Hackenberger |
| Fertigstellung ER und RM Exporter, Bugfixes | 03/12/15 | 20:30 | 22:15 | 105 Minuten | Hackenberger, Hollander |
| Protokoll fortgesetzt | 03/13/15 | 8:00 | 9:00 | 60 Minuten | Hackenberger |

# Design-Überlegung

# UTA

|  |  |
| --- | --- |
| X | Programm muss über CLI Options steuerbar sein |
| X | Programm muss Fehlermeldungen bei falschen oder fehlenden CLI Options ausgeben |
| X | --help Option muss verfügbar sein |
| X | Es soll gewählt werden können ob nur das Relationenmodell, ER-Diagramm oder beides generiert werden soll |
| X | Programm muss zumindest mit MySQL funktionieren |
| ~ | Output-Files müssen plattformunabhängig sein (Probleme mit dot Version auf OSX PKs werden nicht gekennzeichnet im ER) |
| X | ER Diagramm nach Chen Notation |
| X | Alle Tabellen als Entitäten |
| X | Alle Datenfelder der Tabellen als Attribute |
| X | Primärschlüssel gekennzeichnet |
| X | Beziehungen |
| X | Kardinalitäten |
| X | Zusatzattribute Unique, Not Null |

# Ausführen

java -jar Rueckwaertssalto.jar [options]

--help : show the help menu

-d WERT : database name

-h WERT : hostname of the dbms (Only MySQL supported)

-o DATEI : filename of the file to write the output to

-p WERT : password for the dbms

-t WERT : type of export file

-ER: Chen ER diagramm in form of a dot file

-RM: Relationemodell in form of a RTF File

-u WERT : username for the dbms

Als Output des Programms kommt je nach Auswahl des Types (RM oder ER) entsteht entweder ein fertiges RTF File wo das RM abgebildet wird bzw. es entsteht ein dot File welches mittels dem Tool dot (inkludiert im graphviz) dann in ein Bildformat Ihrer Wahl umgewandelt werden als Bsp.:

*dot -Tpng out.dot -o dia.png*  Wandelt das File out.dot in ein PNG um (andere möglichen Formate wären: PostScript, SVG, XFIG, PNG, GIF, Imagemap, CMAPX)