

# **Rückwärtssalto**

Tombor, Kisbedo 4CHITM

# Inhaltsverzeichnis

Zeitaufzeichnung	3
Aufgabenstellung	3
Requirementsanalyse	4
Diagramme	4
Things we've Done	5
Lessons Learned	5
Quellen	5

Version	Beschreibung	Autor	Datum
0.1	Draft	Tombor	26-01-2015

# Zeitaufzeichnung

Aufgabe	Schätzung	Tombor	Kisbedo

## 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

# Requirementsanalyse

- **Eine CLI Verarbeitung**

- Datenbankurl, User, PW, Datenbankname, EER/RM, Filename können Übergeben werden.
- Erstellt neue Verbindung, bekommt Datenbank Objekt und erstellt damit neues RM bzw. EER Diagram

- **Eine Datenstruktur welche die Metadaten einer Datenbank speichert.**

- Erstellen eines Datenbank Objektes
  - Name und Tabellen werden gespeichert.
  - Setter und Getter Methoden.
  - toRMFile() erstellt ein File mit dem RM der Datenbank.
  - toRMConsole() gibt das RM in der Console aus.
  - toEER() erstellt ein PNG File mit dem EER Diagramm.
- Erstellen eines Tabellen Objektes
  - Name und Attribute werden gespeichert.
  - Setter und Getter Methoden.
- Erstellen eines Attributes
  - Name, Typ, istFK, FKtabelle, FKentfernt und istPK wird gespeichert
  - Setter und Getter Methoden
  - FKtabelle als Relation zur Tabelle
  - FKentfert als Relation zum Attribut.

- **Eine Verarbeitung welche die Metadaten aus der Datenbank ausliest und in die Datenstruktur speichert.**

- Auslesen der Tabellen Information (name, alle Attributnamen bzw. Typen)
- Auslesen des FK Namen in der Tabelle, der Quellen Tabelle und den Namen in der Quellen Tabelle für alle FK'S
- Auslesen der PK's
- Speichern der Daten in die Datenstruktur und Rückgabe eines Database Objektes an die CLI Verarbeitung

- **Eine passenden Library für die grafische Verarbeitung (damit toEER() umgesetzt werden kann).**

## Diagramme

Datenbank
- tables : Tabelle[] - name : String
+ setTabellen(tables : Tabelle []) : void + getTabellen() : Tabelle[] + addTabelle(table : Tabelle) : void + setName(name : String) : void + toRM() : void + toEER() : void

Tabelle
- attributes : Attribute[] - name : String
+ setAttribute(attributes : Attribute []) : void + getAttribute() : Attribute [] + addAttribut (attribut : Attribut) : void + setName(name : String) : void + getName() : String

Attribut
- name : String - typ : String - istFK : boolean - FKtabelle: Tabelle - FKattr : Attribut - istPK : boolean
+ setName(name : String) : void + getName() : String + setTyp(typ : String) : void + getTyp() : String + setFK(arg : boolean) : void + getFK() : boolean + setFKtabelle (tabel : Tabelle) : void + getFKtabelle () : Tabelle + setFKattr(attr: Attribut) : void + get FKattr() : Attribut

**Things we've Done**

**Lessons Learned**

**Quellen**

