

Rückwärts- salto

Velkoski, Wichert

Patrik



Contents

1. Aufgabenstellung.....	2
2. Auslesen von Meta-Daten	3
2.1 Auslesen von Metadaten.....	4
3. Requirements-Analyse	5
4. Graphische Darstellung: GraphViz.....	6
4.1 Graphische Darstellung: GraphViz.....	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

Auslesen von Meta-Daten

Um Meta-Daten aus der Datenbank auszulesen, kann man `DatabaseMetaData` importieren und mit deren Methoden arbeiten. Man kann unterscheiden zwischen dem `ResultSetMetaData`-Interface und dem `DatabaseMetaData`.

Mit der `DatabaseMetaData` kann man generell Informationen beschaffen aus der Datenbank.

`ResultSetDataBase` kann detailliertere Informationen von einer Query-Resultset holen mit der Verwendung eines `ResultSetMetaData`-Objekts.

DatabaseMetaData:

```
DatabaseMetaData metadata=conn.getMetaData();
```

Man erzeugt ein Objekt von `DatabaseMetaData` und holt mittels der `getMetaData`-Funktion

alle Metadaten aus der Datenbank. Durch Anwendung von weiteren Funktionen können dann die gewünschten Informationen in ein `ResultSet` gespeichert werden zb. `PrimaryKey`

```
ResultSet ps=metadata.getPrimaryKeys(null,null,tabellenname);
```

ResultSetDataBase:

Nach einer Query, die in ein Resultset gespeichert wurde, kann man wieder mittels der `getMetaData`-Funktion die MetaDaten holen. Diese werden in ein Objekt von `ResultSetMetaData` gespeichert.

```
ResultSet rs = stmt.executeQuery(sql);
```

```
ResultSetMetaData md = rs.getMetaData();
```

Man verwendet dann eine jeweilige Funktion, für die gewünschte Information.

Um zb. Einen Spaltennamen auszugeben wird wie folgt vorgegangen:

```
for( int i = 1; i <= md.getColumnCount(); i++ ){  
        outp += md.getColumnLabel(i) + " " ;  
        outp += "\n";  
}
```

Auslesen von Metadaten

Verwendete Methoden in dem Programm:

`getPrimaryKeys` (für PrimaryKeys)

`getImportedKeys` (für ForeignKeys)

`getTables(null,null,null,null)` (Alle Parameter enthalten null,in diesem Fall werden alle Tabellen besorgt)

`getString(x)` besorgt Namen von den gewünschten Sachen(x= zb. „COLUMN_NAME“ , oder „FKTABLE_NAME“, je nachdem was man braucht)

Requirements-Analyse

Paket	Nr.
EER-Diagramm in PNG	1
RM in TXT	2
ResultSetMetaData	3
Extraktion der Metadaten mit JDBC und Java	4
Korrekte Syntax nach Chen,MinMax oder IDEFIX	5
Alle Tabellen der Datenbank als Entitäten	6
Alle Datenfelder der Tabellen als Attribute	7
Primärschlüssel der Datenbanken entsprechend gekennzeichnet	8
Kardinalitäten	9

Nr.	1	2	3	4	5	6	7	8	9
Zeit	1,5	0,2	7	1	1	1	1	1	1

Graphische Darstellung: GraphViz

Für die Darstellung des EER-Diagramms wird das Programm GraphViz verwendet. Dies verwendet ein DOT-File, welches wir selber schreiben werden, und macht anhand dieses Files automatisch ein Diagramm. Die Wahl auf dieses Programms fiel wegen mehreren Gründen: Viel Dokumentation, viele Beispiele, viele Tutorials, plattformunabhängig.

DOT-Files sind nichts anderes als normale txt-Files(die trotzdem mit .dot gespeichert werden) die die DOT-Syntax aufweist mit welcher GraphViz das Diagramm zeichnen kann.

In Java wird alles, was benötigt wird (Tabellen, Primarykeys, Foreignkeys, Columns) in entsprechender DOT-Syntax in einen String gespeichert und die dann in ein File geschrieben. (Filewriter, BuffetedWriter)

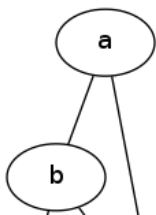
Syntax:

Im Dot-File beginnt man mit „graph graphname{ }“. Zwischen den geschwungenen Klammern befindet sich alles um das Diagramm zu zeichnen.

Um Objekte(nodes) miteinander zu verbinden wird ein - - verwendet.

Die Verbindung wird Edge genannt.

Zb ein a - - b; gibt folgendes aus :

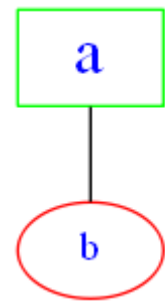


Inhalte von Objekten können Eigenschaften haben zB. Farbe oder Dicke .

Graphische Darstellung: GraphViz

Mittels Label können wir es benennen und dann weitere Eigenschaften vermitteln.

```
graph a{  
a[label="a" color=Green, fontcolor=Blue, fontsize=25, shape=box]  
b[label="b" color=Red, fontcolor=Blue, fontsize=15, shape=ellipse]  
a --b  
}
```



In unserem DOT-File werden PrimaryKeys unterstrichen. Foreignkeys werden rot gemacht. Sollte beides der Fall sein, sind sie rot und unterstrichen.

Folgende Webseiten bringen große Hilfe:

<http://www.graphviz.org/pdf/dotguide.pdf>

<http://www.tonyballantyne.com/graphs.html>

<http://www.graphviz.org/Documentation.php>

<http://graphs.grevian.org/example>