

visualDependencies for Databases

Visualisierung der Abhängigkeiten von Datenbankobjekten

Andre Kasper ,
Jan Philipp
Marc Kastleiner

Dr. Andreas Behrend
Prof. Dr. Birgit Bertelsmeier
Prof. Dr. Heide Faeskorn-Woyke

**FH Köln, Institut für Informatik
Uni Bonn**

Schlüsselworte:

Visuelle Datenanalyse, Views, Trigger, Mutating Table, Prozeduren , referentielle Integrität

Einleitung

In den meisten Unternehmen sind die geschäftsrelevanten Daten nachwievor in relationalen Datenbanken gespeichert. Erstaunlicherweise existiert nach Recherche der Autoren bisher kein visuelles Werkzeug, das verschiedenartige Beziehungen zwischen relationalen Tabellen, Views, Prozeduren und Datenbanktriggern visualisiert. Aus dieser Idee entstand das Programm visualDependencies for Databases, das als Open Source-Projekt unter <http://www.visualdependencies.de/> als Download für Windows, Mac OS X und Linux zu finden ist. VisualDependencies wurde als Diplomarbeit von Jan Philipp und Andre Kasper entwickelt und in diesem Jahr in einer nach-folgenden Arbeit von Marc Kastleiner erweitert. Die Arbeit erhielt den Univention Preis (1.Platz) für die beste innovative und praxisnahe Open-Source-Software-Lösung (<http://www.univention.de/univention/univention-preis/die-preistraeger-2010/>). Das Werkzeug richtet sich an Datenbankentwickler und Datenbankadministratoren, die sich einen visuellen Überblick über die in ihrer Datenbank (Oracle oder MySQL) vorhandenen verschiedenartigen Beziehungen machen möchten.

VisualDependencies: Ein Werkzeug für Datenbankentwickler und Administratoren

Es gibt zwar eine Reihe von visuellen Werkzeugen für Datenbanksysteme oder Anwendungen, die grafische Zusatzfunktionen anbieten. Nahezu alle beschränken sich dabei auf die Darstellung der Fremdschlüsselbeziehungen und zielen daher auf die referentielle Integrität. Für die Entwicklung eines Datenbankmodells und auch für das Verständnis des Modells ist eine solche Darstellung hilfreich und deshalb auch sinnvoll und ausreichend. Hingegen verschafft es dem Datenbankadministrator oder einem interessierten Anwender nicht die gewünschte Transparenz, um die komplexen, wechselseitigen Abhängigkeiten der Views und Tabellen nachvollziehen zu können.

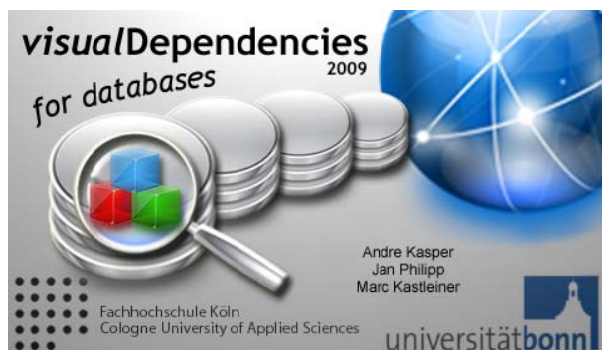


Abbildung 1: VisualDependencies

visualDependencies for Databases stellt die Abhängigkeiten von Objekten einer Datenbank transparent dar. Datenänderungen können nicht vorgenommen werden. Die Tabellen, Trigger, Prozeduren, Funktionen und Views werden dazu zunächst analysiert und anschließend mittels Graphen visualisiert. Die Anwendung stellt dabei fünf Funktionalitäten zur Verfügung: Verbindungsübersicht, View-Hierarchie, Trigger, Einzelsicht und Fremdschlüsselbeziehungen. Die Verbindungsübersicht zeigt die alle angelegten Datenbankverbindungen an. Weiterhin werden die Views, die jeweils Abhängigkeiten zu andere Tabellen und Views besitzen, in einem Graphen dargestellt. Die vorhandenen Trigger eines Datenbankschemas mit ihren Abhängigkeiten werden ebenfalls gesondert dargestellt. Als vierte Funktionalität werden die Tabellenabhängigkeiten und Fremdschlüssel in einem einfachen ERD veranschaulicht. Die neu aufgenommene Einzelansicht zeigt auch alle Prozeduren und Funktionen an, die eine vorgegebene Tabelle benutzen.

visualDependencies unterstützt Oracle 9i/10g/11g-Datenbanken und MySQL-Datenbanken. Der OR-Wrapper Hibernate mit einer HSQL-Datenbank speichert die bereits analysierten Verbindungsobjekte. Die Benutzerschnittstelle wurde mit Java und Eclipse realisiert.

Die Verbindungsübersicht

Beim Starten von visualDependencies wird die Verbindungsübersicht direkt geöffnet und zeigt dem Benutzer einen Willkommensbildschirm, der eine Übersicht über die Funktionalitäten der Anwendung gewährt. Die Verbindungsübersicht besteht aus zwei verschiedenen Bereichen. Im linken Bereich werden die bereits angelegten Datenbankverbindungen sowie schon in der Datenbank abgelegte Objekte (Tabellen, Views und Trigger eines Datenbankschemas) angezeigt. Es können neue Verbindungen erstellt werden, aber keine sonstigen Datenbankobjekte. Der rechte Bereich enthält eine Beschreibung der Grundfunktionalitäten der Anwendung mit einem kurzen Hilfetext.



Abbildung 2: Verbindungsübersicht

Die View-Hierarchie

Zwischengespeicherte Sichten für oft angeforderte Anfragen dienen in einem DBMS zur Optimierung der Abfrageperformanz und des Datenaufkommens. Diese Sichten erreichen oft einen hohen Grad der Komplexität, da sie auch in einer View-Hierarchie oder sogar rekursiv aufeinander aufbauen können. Um die Visualisierung von Views geht es in der zweiten grafischen Ansicht, in der Views und Tabellen angezeigt werden, die Abhängigkeiten zu anderen Tabellen oder Views in der Datenbank besitzen. Dabei werden diese Abhängigkeiten weiter differenziert. Positive Abhängigkeiten sind solche, in welchen der WHERE-Bedingungsteil der SELECT-Anfrage eine Enthaltungsmenge durch Operatoren wie „=“ oder „LIKE“ spezifiziert. Neutrale SELECT-Anweisungen ohne WHERE-Bedingung sind immer positiv. Negative Abhängigkeiten sind solche, in welchen der WHERE-Bedingungsteil explizit Daten ausschließt. Hier wird die äußere SELECT-Anweisung mittels einer Suchbedingung mit dem

Quantor NOT EXISTS mit einem Subselect oder einem MINUS verbunden. (siehe http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Sicht). Die Farben der Kanten im Graphen sind farblich durch den Kontext ihrer Abhängigkeit bestimmt. Während grüne Kanten positive Abhängigkeiten repräsentieren, sind negative Abhängigkeiten durch rote Kanten hervorgehoben. Die Kantenobjekte zeigen in einer kleinen Box die genauen Details der Abhängigkeit, wenn der Mauszeiger über sie gehalten wird.

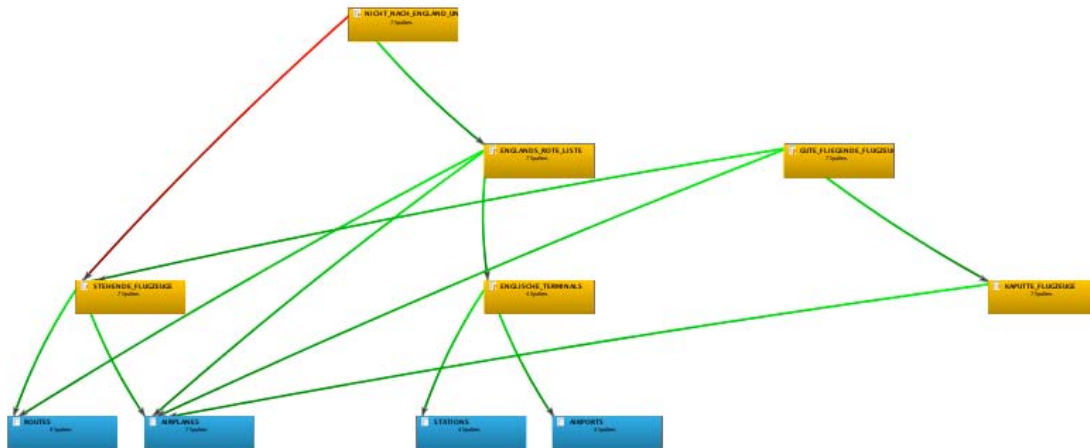


Abbildung 3: Abhängigkeiten von Views in Datenbanken

Datenbanktriggersicht

Um der steigenden Komplexität der Datenbanken gerecht zu werden, helfen oft automatische, interne Aufträge (Trigger) im Datenbanksystem. Die Trigger definieren ein Stück Programmcode und führen ihn zu einem bestimmten Zeitpunkt und unter einer bestimmten Bedingung aus. Der Zweck eines Triggers ist es, die Datenintegrität aufrecht zu erhalten oder in Verknüpfung mit materialisierten Views für konsistente Daten zu sorgen. Allerdings ist es nicht so ohne Weiteres ersichtlich, ob die Ausführung eines Triggers erfolgreich sein kann, wenn dadurch weitere Trigger angestoßen werden. Erst zur Laufzeit, also nach dem statischen Analysieren und gegebenenfalls auch Parsen des Triggers, kann ein Fehler auftreten. Damit stößt man jedoch schnell an die Grenzen des Machbaren. Dies ist auch ein Grund, warum beispielsweise Oracle trotz der beschriebenen Problematik anstandslos das Anlegen von Triggern akzeptiert, was zu rekursiven Abläufen führen oder das Mutating Table-Problem (siehe auch: http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Datenbanktrigger) auslösen kann. In der dritten Ansicht werden die vorhandenen Trigger der Datenbank analysiert, die auf den Tabellen liegen. Mit dieser Funktionalität wird sowohl eine Übersicht der zu einer Tabelle verfügbaren Trigger erzeugt als auch die Beziehungen der Triggern untereinander visualisiert. Dabei werden durch die Abhängigkeiten die impliziten, potenziellen Aufrufe anderer Trigger sichtbar gemacht. Die Informationen über zyklische, rekursive Abfolgen oder eines möglichen Triggers mit Mutating-Table-Problem werden im Graphen dargestellt.

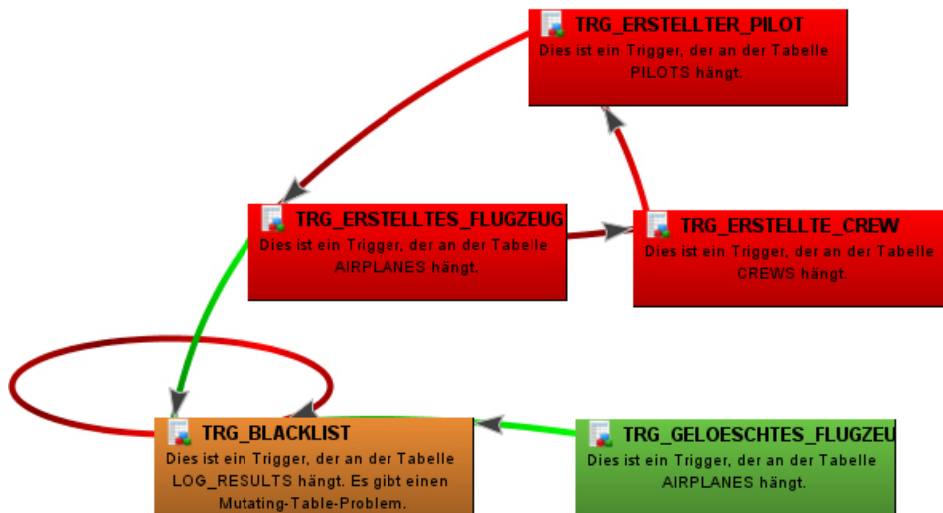


Abbildung 4: Abhängigkeiten zwischen Triggern

Die Triggerobjekte und die Verbindungen (Kanten) werden in dieser Darstellung grün angezeigt, solange es zu diesem Objekt keine besondere Auffälligkeiten gibt. Ein Trigger, der ein Mutating-Table-Problem hervorrufen kann, wird in der Ansicht orange dargestellt und es gibt einen Hinweistext zu diesem Objekt. Weiterhin wird eine rote Kante von diesem Trigger auf sich selbst angezeigt. Die zweite Auffälligkeit, die gesondert dargestellt wird, ist die Rekursion unter den Triggern. Die Farbe der Triggerobjekte ist rot und wie die rekursiven Kanten, die diese Trigger verbinden.

Entity-Relationship-Diagramm

Eine weitere Ansicht stellt das ER-Diagramm in Form von Fremdschlüsselbeziehungen dar. Im Gegensatz zu anderen Produkten beschränkt sich die Anzeige jedoch auf die lose Darstellung der Abhängigkeiten ohne weitere Details wie Kardinalität oder Art der Assoziation. Einzig und allein die Richtung der Abhängigkeit wird dargestellt, weshalb die Visualisierung auch durch einen Graphen realisiert wird.

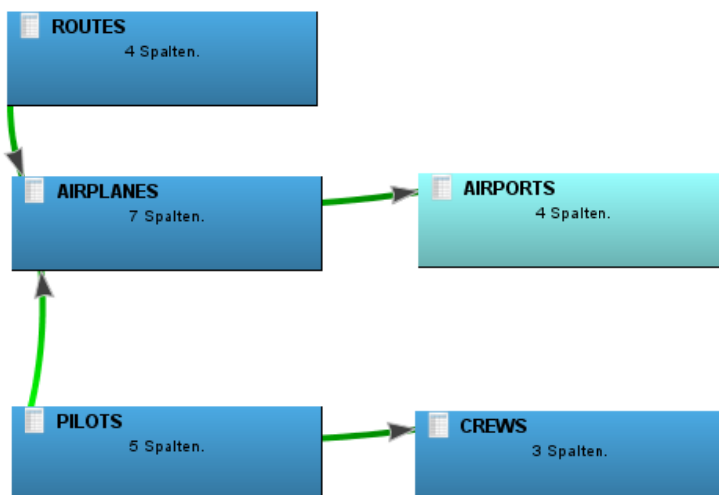


Abbildung 5: Fremdschlüsselbeziehungen in Datenbanken

Einzelansicht

Die in der Software „Einzelansicht“ genannte grafische Aufbereitung stellt die Tabellen mehr in das Zentrum der Betrachtung. Diese Ansicht beinhaltet eine Maske zur Auswahl einer Tabelle mit allen darauf aufbauenden Abhängigkeiten. Hierzu zählen neben den bereits in den anderen Ansichten erwähnten Objekten auch die Prozeduren und Funktionen. Jeder Objekttyp wird in einer eigenen Farbe dargestellt. Bei einem Rechtsklick auf die Trigger, Prozeduren und Funktionen kann man in einem Kontextmenü die betroffenen Tabellen abfragen.

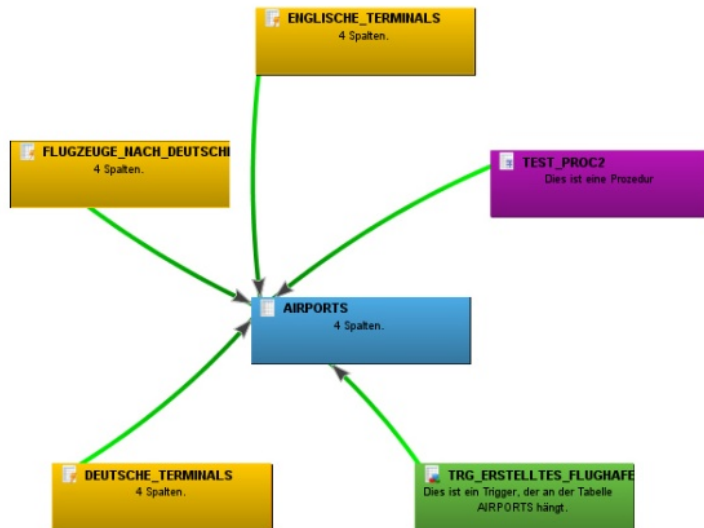


Abbildung 6: Einzelansicht mit der Tabelle im Zentrum der Betrachtung

Ausblick

Mit diesen Funktionalitäten wurde ein hilfreiches Werkzeug bei der visuellen Betrachtung und Untersuchung von Datenbankschemata erstellt. Es gibt noch zahlreiche weitere Ideen, um die Darstellung zu verbessern oder weitere Funktionalitäten aufzunehmen. So kann die Aufschlüsselung einer objektrelationalen Typhierarchie in Erwägung gezogen werden. Weitere Verbesserungsmöglichkeiten sind im Bereich der Graphen angesiedelt. Diese Ansichten sind noch nicht optimal aufbereitet. Hier wäre es denkbar, weitere grafische Algorithmen zur Layout-Gestaltung zu implementieren. Dies könnten beispielsweise Erweiterungen im dreidimensionalen Raum sein. Insbesondere die Problematik der sich überschneidenden Kanten kann durch effizientere Algorithmen noch besser gelöst werden.

Kontaktadressen:

Prof. Dr. Heide Faeskorn-Woyke
FH Köln, Campus Gummersbach
Steinmüllerallee 1
D-51643 Gummersbach

Telefon: +492261 8196 6379

Fax: +492261 8196 76379

E-Mail: Heide.Faeskorn-Woyke@fh-koeln.de

Andre.Kasper@fh-koeln.de

Jan@philipp-online.de

M.Kastleiner@gmx.de

Andreas Behrend <behrend@cs.uni-bonn.de>

Birgit.Bertelsmeier@fh-koeln.de

Internet: <http://www.visualdependencies.de/>