



Studienarbeit

Filtern von Fremdschlüsseln aus Inklusionsbeziehungen

Oliver Albrecht

oliver.albrecht@informatik.hu-berlin.de

Betreuerin:
Jana Bauckmann

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Studienarbeit selbstständig und nur unter Zuhilfenahme der angegebenen Quellen angefertigt habe.

Berlin, 17. Dezember 2007

Oliver Albrecht

Inhaltsverzeichnis

1	Einleitung.....	5
1.1	Motivation.....	5
1.2	Zielsetzung.....	5
2	Von der Inklusionsabhängigkeit zum Fremdschlüssel.....	5
2.1	Begriffsklärung.....	5
2.2	Eigenschaften von Fremdschlüsseln.....	6
2.2.1	Abdeckung.....	6
2.2.2	Verteilung.....	7
2.2.3	Wertebereich im abhängigen Attribut.....	7
2.2.4	Anzahl der Referenzen.....	8
2.2.5	Gleiche oder ähnliche Bezeichner.....	8
3	Heuristiken.....	9
3.1	Klassifizierung von Heuristiken.....	9
3.2	Entwickelte Heuristiken.....	9
3.2.1	Minimale Anzahl abhängiger Werte.....	10
3.2.2	Abdeckung.....	10
3.2.3	Verhältnis der begrenzten Wertemengen.....	10
3.2.4	Verteilung über referenziertem Attribut.....	11
3.2.5	Verhältnis der Wertlängen.....	11
3.2.6	Anzahl referenzierter Attribute.....	12
3.2.7	Attributnamen.....	12
3.2.8	Kleine Tabellen.....	12
3.3	Evaluierung der Heuristiken.....	13
3.3.1	Testumgebung.....	13
3.3.2	Evaluierung der einzelnen Heuristiken.....	14
3.3.3	Kombination der Heuristiken.....	18
3.3.4	Ergebnisse für einzelne Schemata.....	21
4	Implementation.....	24
4.1	Benutzerschnittstellen.....	24
4.2	Struktur der Implementation.....	26
4.2.1	Konfiguration.....	27
4.2.2	Start des Programms.....	28
4.2.3	Implementation einzelner Heuristiken.....	28
4.3	Feature der GUI-Variante.....	30
4.4	Performance.....	31
4.4.1	Statistikdaten.....	31
4.4.2	Laufzeiten.....	31
4.4.3	Verbesserung der Laufzeiten.....	32
5	Fazit.....	32
6	Anhang.....	35
6.1	Klassendiagramm.....	35
6.2	Ergebnisse für einzelne Schemata im Detail.....	37
6.2.1	Filmdienst.....	37
6.2.2	Movielens.....	38
6.2.3	SCOP.....	38
6.2.4	UNIPROT.....	39
6.2.5	TPC-H.....	40
6.2.6	IMDB.....	41

Tabellenverzeichnis

Tabelle 1: Beschreibung der Testdatenbanken.....	13
Tabelle 2: Bewertungskriterien Konfigurationen.....	14
Tabelle 3: Rückgabewerte von Heuristiken und deren Auswirkung.....	18
Tabelle 4: Verschiedene Gewichtungen von Heuristiken.....	20
Tabelle 5: Gewichtung der Heuristiken in den Testergebnissen.....	21
Tabelle 6: Konfiguration der Heuristiken in den Testergebnissen.....	21
Tabelle 7: Ergebnisse für Schema UNIPROT.....	22
Tabelle 8: Ergebnisse für Schema SCOP.....	22
Tabelle 9: Ergebnisse Schema TPC-H.....	23
Tabelle 10: Ergebnisse im Schema Filmdienst.....	23
Tabelle 11: Ergebnisse im Schema IMDB.....	24
Tabelle 12: Ergebnisse im Schema MovieLens.....	24
Tabelle 13: Performance von FKFinder auf verschiedenen Schemata.....	31
Tabelle 14: Laufzeit einzelner Heuristiken.....	32

Abbildungsverzeichnis

Abbildung 1: Abdeckung in einer Inklusionsabhängigkeit,	6
Abbildung 2: Verteilung von Verknüpfungen in einer IND,	7
Abbildung 3: Anzahl von Werten einer IND.....	8
Abbildung 4: Anzahl von Referenzen auf ein Attribut.....	8
Abbildung 5: Grenzwert der Heuristik Minimale Anzahl abhängiger Werte.....	15
Abbildung 6: Konfiguration für die Heuristik Abdeckung (oberer Grenzwert).....	16
Abbildung 7: Konfiguration für die Heuristik Abdeckung (unterer Grenzwert).....	16
Abbildung 8: Konfiguration der Verteilung über referenziertem Attribut.....	17
Abbildung 9: Ausgabe eines Rankings von IND's.....	19
Abbildung 10: Ergebnisausgabe FKFinder (Shell).....	25
Abbildung 11: Ergebnisausgabe FKFinder (GUI).....	26
Abbildung 12: Ordnerstruktur FKFinder.....	26
Abbildung 13: Konfiguration FKFinder über GUI.....	27
Abbildung 14: Statusausgabe FKFinder.....	30

1 Einleitung

1.1 Motivation

Im Projekt Aladin [1] am Lehrstuhl für Wissensmanagement in der Bioinformatik¹ werden Profiling-Verfahren untersucht, mit deren Hilfe sich Aussagen zur Datenstruktur existierender Datenbanken treffen lassen. In der Regel handelt es sich dabei um sehr große biowissenschaftliche Datenbanken. Die Struktur der Datenbank ist häufig nur eingeschränkt oder gar nicht bekannt, das heißt häufig fehlen Informationen zu Beziehungen zwischen den Relationen.

Fremdschlüsselbeziehungen zu finden, ermöglicht es die Struktur der Datenbank besser zu verstehen und somit sinnvolle und ressourcenschonende Anfragen an die Datenbank zu stellen.

Ziel dieser Studienarbeit ist es, Verfahren zu evaluieren und umzusetzen, mit deren Hilfe Fremdschlüsselbeziehungen auf Grundlage von Inklusionsabhängigkeiten in Datenbanken gefunden werden.

1.2 Zielsetzung

Grundlage für diese Studienarbeit ist der SPIDER Algorithmus [2]. Dieser wurde im Rahmen des Projekts Aladin am Lehrstuhl entwickelt und findet Inklusionsabhängigkeiten in Datenbanken.

Ziel dieser Studienarbeit ist es, auf Basis der von SPIDER gefundenen Inklusionsabhängigkeiten diejenigen heraus zu filtern, die als Fremdschlüssel in Betracht kommen. Um diese zu finden, ist es nötig, Eigenschaften von Fremdschlüsseln mit Hilfe von zu erstellenden Heuristiken auf den gefundenen Inklusionsabhängigkeiten zu untersuchen.

Ideen für Heuristiken existierten bereits vor dieser Studienarbeit. Ziel ist es, darüber hinaus weitere Heuristiken zu finden, mit deren Hilfe sich eine zuverlässige Aussage treffen lässt ob eine Inklusionsabhängigkeit als Fremdschlüssel in Frage kommt.

Um die verschiedenen Heuristiken testen zu können und eine Aussage über ihre Zuverlässigkeit treffen zu können, kommen verschiedene Testdatenbanken zum Einsatz, deren Struktur bekannt ist. Die technische Umsetzung der Studienarbeit soll in Form eines Java Programms erfolgen, das gegebene Inklusionsabhängigkeiten untersucht und ausgibt, welche der Inklusionsabhängigkeiten als Fremdschlüssel in Frage kommen.

2 Von der Inklusionsabhängigkeit zum Fremdschlüssel

2.1 Begriffsklärung

Inklusionsabhängigkeit (IND)

Eine Inklusionsabhängigkeit $A \subseteq B$ zwischen zwei Attributen verschiedener Relationen liegt dann vor, wenn die Menge der Werte des Attributs A komplett in der Menge der Werte des Attributs B enthalten ist.

Dabei heißt A das abhängige Attribut und B das referenzierte Attribut.

¹ <http://www.informatik.hu-berlin.de/forschung/gebiete/wbi>

Fremdschlüssel (FK)

Ein Fremdschlüssel verweist von einer Relation auf eine andere Relation. Die andere Relation enthält den Fremdschlüssel als Primärschlüssel. Fremdschlüssel können auch aus mehreren Attributen bestehen.

Ein Fremdschlüssel ist immer auch eine Inklusionsabhängigkeit. Jedoch ist nicht jede Inklusionsabhängigkeit auch ein Fremdschlüssel. Die Unterscheidung liegt in der Semantik der Daten. Die über einen Fremdschlüssel verknüpften Tupel haben einen semantischen Bezug zueinander.

2.2 Eigenschaften von Fremdschlüsseln

Um von der Inklusionsabhängigkeit, die SPIDER findet, zu der Entscheidung zu kommen, ob diese ein Fremdschlüssel sein kann, muss man sich bestimmter Eigenschaften von Fremdschlüsseln bedienen. Diese Eigenschaften sind keinem Dokument entnommen, sondern basieren auf eigenen Beobachtungen und Erfahrungen im Umgang mit Datenbanken. Sinn dieser Auflistung ist es, Eigenschaften aufzuzeigen die häufig bei Fremdschlüsseln zutreffen und einfach von einem Algorithmus geprüft werden können.

Die einzelnen Eigenschaften können in zwei Klassen eingeteilt werden. Solche die einen Fremdschlüssel als sehr wahrscheinlich erscheinen lassen, ihn also bestätigen, und solche die einen Fremdschlüssel eher ausschließen.

2.2.1 Abdeckung

Als Abdeckung werden die Anzahl der Verknüpfungen zwischen dem abhängigen und referenzierten Attribut bezeichnet. Abdeckung deswegen, weil die Werte des abhängigen Attributs einem gewissen Prozentsatz der Werte des referenzierten Attributs entsprechen, also eine bestimmte Menge der Werte „abdecken“. Abbildung 1 zeigt zwei verschiedene Arten der Abdeckung. In der oberen Variante ist ein Fremdschlüssel wahrscheinlich, in der unteren eher unwahrscheinlich.

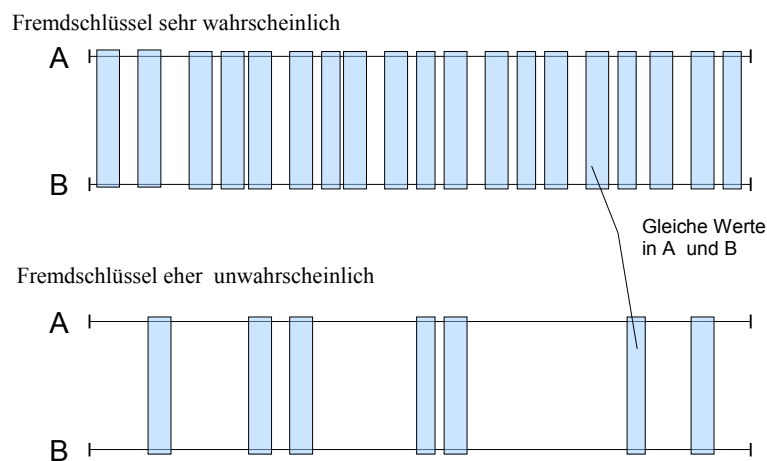


Abbildung 1: Abdeckung in einer Inklusionsabhängigkeit, $A \subseteq B$

Beispiel: In einer Mitarbeiterdatenbank existieren die Tabellen *Mitarbeiter* und *Abteilung*, wobei jeder Mitarbeiter einer Abteilung zugeordnet ist. Vermutlich ist die Abdeckung in diesem Fall bei nahezu 100%, da es kaum eine Abteilung ohne Mitarbeiter gibt.

Diese Eigenschaft ist einfach zu überprüfen und bietet eine sehr gute Möglichkeit Fremdschlüssel, die sich durch eine hohe Abdeckung auszeichnen, zu finden. Wenn diese Eigenschaft in einer IND nachgewiesen werden kann, ist die Wahrscheinlichkeit, dass sie ein Fremdschlüssel ist, sehr hoch.

2.2.2 Verteilung

Mit dieser Eigenschaft wird untersucht, wie die Verknüpfungen zwischen den Tabellen auf dem Wertebereich verteilt sind. Ist die Abdeckung in einer IND nur sehr gering, die einzelnen Verknüpfungen hingegen sehr gleichmäßig über den gesamten Wertebereich verteilt, ist ein Fremdschlüssel sehr wahrscheinlich. Abbildung 2 zeigt im oberen Bild eine gleichmäßige Verteilung, im unteren Bild eine ungleichmäßige Verteilung. Um die Gleichmäßigkeit festzustellen, wird eine lexikographische Sortierung beider Attribute angenommen.

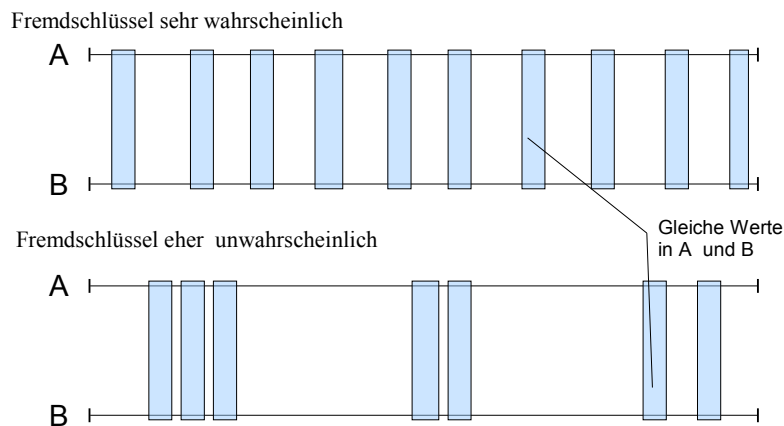


Abbildung 2: Verteilung von Verknüpfungen in einer IND, $A \subseteq B$

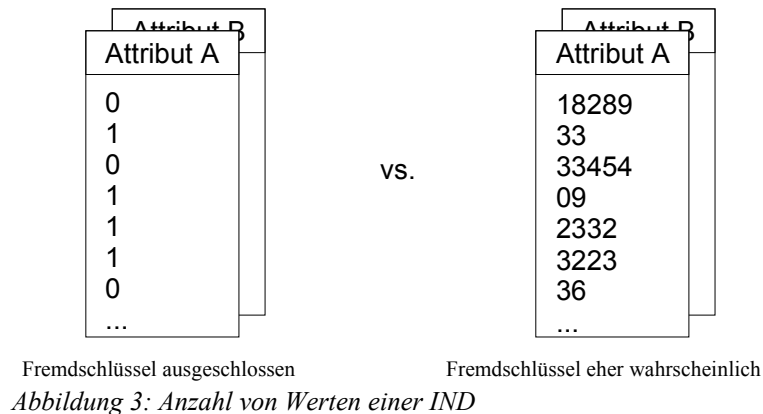
Beispiel: In einer Kundendatenbank eines Onlineversandhandels gibt es Kunden und Premiumkunden. Die Stammdaten der Kunden werden in der Tabelle *Stammdaten* gespeichert, die nötigen Daten der Premiumkunden in der Tabelle *Premiumkunden*. Nur ein sehr geringer Anteil der Kunden lässt sich zusätzlich als Premiumkunde registrieren, die Abdeckung zwischen beiden Tabellen ist also sehr gering. Dennoch ist die Verteilung sehr gleichmäßig, da sich zum Beispiel nur jeder 100. Kunde registrieren lässt.

Beispiel für ungleichmäßige Verteilung: In der oben genannten Tabelle *Stammdaten* werden fortlaufende Nummern als Primärschlüssel verwendet. Bei 1.000.000 Kunden sind das ebenso viele einzelne Zahlen. Zusätzlich gibt es in der Datenbank noch eine weitere Tabelle *Bücher* mit etwa 100.000 Einträgen. Auch in dieser Tabelle werden fortlaufende Nummern als Primärschlüssel verwendet. Die jeweiligen Primärschlüssel der Tabellen *Bücher* und *Stammdaten* würden also als IND gefunden. Jedoch ist die Verteilung sehr ungleichmäßig, da vermutlich nur die ersten 100.000 Werte aus der Tabelle *Stammdaten* auch in *Bücher* vorkommen. Eine Fremdschlüsselbeziehung kann also mit sehr hoher Wahrscheinlichkeit ausgeschlossen werden.

Diese Eigenschaft kann eine IND als Fremdschlüssel sowohl bestätigen als auch ausschließen.

2.2.3 Wertebereich im abhängigen Attribut

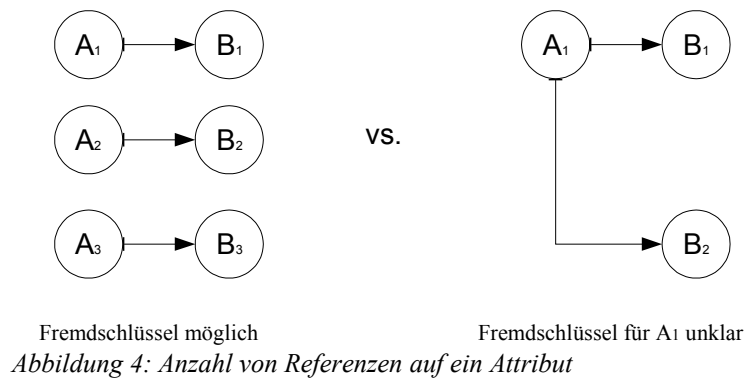
Diese Eigenschaft untersucht die Anzahl der verschiedenen Werte im abhängigen Attribut. Sind nur sehr wenig verschiedene Werte im Attribut vorhanden, ist ein Fremdschlüssel eher unwahrscheinlich (siehe Abbildung 3). Wichtig dabei: Trifft das Gegenteil zu, ist dies kein Hinweis auf einen Fremdschlüssel. Die Beobachtung hinter dieser Eigenschaft ist, dass häufig Binärwerte in Datenbanken verwendet werden um Tupel zu klassifizieren. Das einfachste Beispiel ist in einer Kundendatenbank die Speicherung des Geschlechts eines Kunden durch zwei verschiedene Zahlen. Diese beiden Zahlen sind auch in vielen anderen Attributen vorhanden und es würden daher unzählige IND's gefunden werden.



Mit dieser Eigenschaft können IND's als Fremdschlüssel ausgeschlossen werden. Die Bestätigung eines Fremdschlüssels durch Untersuchung dieser Eigenschaft ist nicht zu erbringen.

2.2.4 Anzahl der Referenzen

Für jedes abhängige Attribut in der Menge der gefundenen Inklusionsabhängigkeiten kann gezählt werden in wie vielen IND's es als abhängiges Attribut auftaucht. Ist es in mehreren IND's vorhanden, ist es sehr unwahrscheinlich, dass dieses Attribut Teil jeder dieser indizierten Fremdschlüsselbeziehungen ist (siehe Abbildung 4). Die Erfahrung zeigt, dass die Tupel einer Tabelle fast nie über ein und das selbe Attribut zu mehreren anderen Tabellen in Beziehung gebracht werden.



Mit dieser Eigenschaft können IND's als Fremdschlüssel mit hoher Wahrscheinlichkeit ausgeschlossen werden.

2.2.5 Gleiche oder ähnliche Bezeichner

Wenn Menschen die Struktur einer Datenbank anlegen benutzen sie oft für Dinge, die mit einander in Beziehung stehen, auch gleiche oder ähnliche Bezeichnungen. Angenommen in einer Kundendatenbank ist eine Tabelle *Anrede* vorhanden, in der verschiedene Anredeformen („Frau“, „Herr“, „Frau Dr.“ u.ä.) gespeichert werden.

Dann macht es Sinn in der Tabelle *Kunden*, in der die Daten der Kunden gespeichert werden, die Verknüpfung zur Tabelle *Anrede* über ein gleichnamiges Attribut vorzunehmen. Es können also folgende Merkmale einer IND untersucht werden:

- Ähnlichkeiten zwischen den Attributnamen
- Ähnlichkeiten zwischen Attributnamen und Relationsnamen

Durch diese Eigenschaft können IND's als Fremdschlüssel bestätigt werden. Jedoch ist dies eine sehr unzuverlässige Eigenschaft, da nicht jeder Datenbankentwickler immer einen semantischen Zusammenhang durch die Ähnlichkeit der Bezeichner darstellt.

3 Heuristiken

Als Heuristiken werden hier Algorithmen bezeichnet, die an Abschnitt 2.2 Eigenschaften von Fremdschlüsseln auf den gefundenen IND's untersuchen.

3.1 Klassifizierung von Heuristiken

Wie schon im Abschnitt 2.2 beschrieben, gibt es Eigenschaften die einen Fremdschlüssel bestätigen und solche die einen Fremdschlüssel ausschließen. Einige der Eigenschaften, zum Beispiel die Eigenschaft *Wertebereich im abhängigen Attribut* (2.2.3), können einen Fremdschlüssel zwar ausschließen, ihn aber nicht bestätigen. Sie können also das eine ODER das andere.

Besonders deutlich wird diese Unterscheidung bei der Untersuchung der *gleichen oder ähnlichen Bezeichner* (2.2.5): Wenn sich die Namen der zwei Attribute einer IND nicht ähneln, sagt dies nichts über die Wahrscheinlichkeit einer Fremdschlüsselbeziehung aus.

Aus diesen Überlegungen ist ersichtlich, dass Heuristiken, die einzelne Eigenschaften untersuchen, drei mögliche Entscheidungen treffen können:

- IND ist ein Fremdschlüssel, Rückgabewert *TRUE*
- IND ist kein Fremdschlüssel, Rückgabewert *FALSE*
- Keine Entscheidung, Rückgabewert *UNKNOWN*

Durch die verschiedenen Rückgabewerte werden die Heuristiken in zwei Klassen eingeteilt:

1. Ausschließende Heuristiken (Rückgabewert *FALSE*)
2. Bestätigende Heuristiken (Rückgabewert *TRUE*)

3.2 Entwickelte Heuristiken

Im Folgenden werden alle Heuristiken beschrieben, die im Rahmen dieser Studienarbeit entstanden sind. Die Idee hinter jeder Heuristik wird aufgezeigt, ihre Klassifizierung und ihre möglichen Rückgabewerte angegeben. Zusätzlich werden die Schwellwerte für die Einzelnen Heuristiken empirisch ermittelt.

3.2.1 Minimale Anzahl abhängiger Werte

Die Heuristik zählt den Wertebereich des abhängigen Attributs und bezieht sich auf die gleichnamige Eigenschaft in Abschnitt 2.2.3. Wenn die Anzahl der verschiedenen Werte einen vorher konfigurierten Wert unterschreitet, wird angenommen, dass die IND kein Fremdschlüssel ist. Liegt die Anzahl der Werte jedoch über dem konfigurierten Wert, wird keine Aussage getroffen.

Hintergrund dieser Heuristik ist die Annahme, dass Attribute mit einem sehr geringen Wertebereich in der Regel keine echt Fremdschlüsselbeziehung haben. Als Beispiel kann hier ein binäres Attribut angenommen werden, in dem eine Klassifizierung gespeichert wird (z.B. Ist Mann/Frau bei Personendaten).

Dies ist eine ausschließende Heuristik. Informationen aus der Tabelle *column_infos* (s. 4.4.1) werden bei der Berechnung zu Hilfe genommen.

Mögliche Rückgabewerte: *FALSE*, *UNKNOWN*

3.2.2 Abdeckung

Die Heuristik untersucht die Eigenschaft Abdeckung (2.2.1) zwischen dem abhängigen (A) und dem referenzierten (B) Attribut. Die Abdeckung wird dabei in Prozent angegeben, also wie viel Prozent der Werte in B auch in A vorhanden sind.

Bei dieser Heuristik kommen zwei Grenzwerte zum Tragen. Eine untere und eine obere Grenze. Beim Überschreiten der oberen Grenze ist die Wahrscheinlichkeit eines Fremdschlüssels sehr hoch und die Heuristik gibt eine positive Rückmeldung. Beim Unterschreiten der unteren Grenze ist die Wahrscheinlichkeit eines Fremdschlüssels sehr gering und es wird eine Entscheidung gegen einen Fremdschlüssel getroffen. Liegt der prozentuale Wert zwischen den beiden Grenzen, erfolgt keine Bewertung.

Idee dieser Heuristik ist, dass Fremdschlüsselbeziehungen zwischen zwei Attributen auch eine sehr hohe Anzahl an Verknüpfungen nach sich ziehen. Eine Abdeckung von 100%, also eine 1:1 Verknüpfung, ist dabei der Idealfall.

Im Gegensatz dazu steht eine sehr geringe Abdeckung, bei der nur ein Prozentsatz von unter 1% der Werte im referenzierten Attribut auch im abhängigen Attribut vorkommen. Solch eine geringe Abdeckung schließt einen Fremdschlüssel in der Regel aus.

Diese Heuristik ist sowohl eine ausschließende als auch eine bestätigende Heuristik.

Mögliche Rückgabewerte: *TRUE*, *FALSE*, *UNKNOWN*

3.2.3 Verhältnis der begrenzten Wertemengen

Diese Heuristik betrachtet eine Verfeinerung der Eigenschaft Abdeckung (2.2.1). Sie zählt die unterschiedlichen Werte des abhängigen Attributs (A) und die des referenzierten Attributs (B). Als obere und untere Wertgrenze für die Zählung in B wird dabei der maximale und minimale Wert des abhängigen Attributs betrachtet. Verglichen wird also die Anzahl der Werte in A mit der Anzahl der Werte in B die in der min/max-Grenze von A liegen. Übersteigt dabei die Anzahl der Werte in B einen errechneten Grenzwert, wird eine negative Bewertung getroffen. Sonst erfolgt eine positive Bewertung. Der Grenzwert wird dabei durch einen prozentualen Anteil der Anzahl der Werte in A angegeben.

Beispiel: Wenn in A 100 verschiedene Werte vorliegen und der Grenzwert bei 10 Prozent liegt, dürfen in B nicht mehr als 110 Werte im definierten Grenzbereich liegen, um eine positive Bewertung zu erhalten.

Die Idee dieser Heuristik ist ähnlich der *Abdeckung* - Heuristik. Bei einer Fremdschlüsselbeziehung wird davon ausgegangen, dass es eine sehr hohe Anzahl von Verknüpfungen zwischen den Attributen gibt.

Diese Heuristik ist sowohl eine ausschließende als auch eine bestätigende Heuristik.

Mögliche Rückgabewerte: *FALSE, TRUE*

3.2.4 Verteilung über referenziertem Attribut

Diese Heuristik untersucht, die Eigenschaft Verteilung (2.2.2). Als Maß dient die Anzahl der Werte des referenzierten Attributs außerhalb der Grenzen liegen, die durch den minimalen und maximalen Wert des abhängigen Attributs gesetzt werden.

Die Idee der Heuristik ist, dass die Verknüpfungen in einem Fremdschlüssel oft gleichmäßig verteilt sind. Beispiel: In einer Filmdatenbank sind mehrere Millionen Filme gespeichert. Zu einigen (wenigen) Filmen sind Daten zu Auszeichnungen gespeichert (Oscar, Bambi etc.). Es handelt sich also nur um eine sehr geringe Anzahl der Verknüpfungen. Dennoch ist die Verteilung über die Gesamtzahl der Filme ungefähr gleichmäßig. Es kann also angenommen werden, dass nur eine sehr geringe Anzahl von Filmen außerhalb der Grenzen der prämierten Filme liegt. Bei dieser Heuristik spielt die Abdeckung keine Rolle.

Der Unterschied zur *Verhältnis der begrenzten Wertemengen* - Heuristik ist, dass sich der Prozentwert der Abweichung auf die Gesamtanzahl aller Werte im referenzierten Attribut bezieht. Die Anzahl der Werte im abhängigen Attribut spielt bei dieser Heuristik keine Rolle.

Die Heuristik ist sowohl eine ausschließende als auch eine bestätigende Heuristik.

Mögliche Rückgabewerte: *FALSE, TRUE*

3.2.5 Verhältnis der Wertlängen

Die Heuristik vergleicht die durchschnittliche Länge der Werte der Attribute einer IND als vereinfachtes Maß der Abdeckung (2.2.1). Weichen diese beiden Länge von einer vorher definierten Grenze ab, wird eine negative Bewertung getroffen. Liegt die Abweichung jedoch im konfigurierten Toleranzbereich, erfolgt eine positive Bewertung.

Die Idee dieser Heuristik ist, dass sich die Werte der Attribute in einer IND sehr ähnlich sind. Eine ähnliche Länge der Werte der Attribute weist oft auf eine Fremdschlüsselbeziehung hin. Allerdings ist die Genauigkeit bei Zahlenwerten deutlich höher als bei Zeichenketten.

Diese Heuristik ist sowohl eine bestätigende als auch eine ausschließende Heuristik.

Mögliche Rückgabewerte: *FALSE, TRUE*

3.2.6 Anzahl referenzierter Attribute

Die Heuristik bezieht sich auf die Eigenschaft Anzahl der Referenzen (2.2.4). Sie zählt die Anzahl der verschiedenen Attribute, die von einem abhängigen Attribut referenziert werden. Wenn dieser Wert eine vorher konfigurierte Grenze übersteigt, wird eine negative Bewertung getroffen.

Idee der Heuristik ist, dass ein abhängiges Attribut in der Regel nur ein anderes Attribut referenziert. Ist das abhängige Attribut auch in anderen IND's als abhängiges zu finden, ist davon auszugehen, dass es sich nur bei einer der IND's wirklich um einen Fremdschlüssel handelt.

Wird die Grenze hingegen nicht überschritten, erfolgt keine Bewertung. Daher handelt es sich hierbei um eine ausschließende Heuristik.

Mögliche Rückgabewerte: *FALSE*, *UNKNOWN*

3.2.7 Attributnamen

Die Heuristik nutzt die Eigenschaft Gleiche oder ähnliche Bezeichner (2.2.5), d.h. sie untersucht die Attribut- und Relationsnamen der beiden Attribute einer IND. Stimmen diese in bestimmten Punkten überein oder sind sich ähnlich, wird eine positive Bewertung getroffen. Stimmen diese nicht überein, wird keine Bewertung getroffen. Beim Vergleich der Namen wird die Groß- und Kleinschreibung nicht beachtet. Untersucht werden:

- A.Spaltenname == B.Spaltenname?
- A.Spaltenname == B.Tabellenname?
- A.Spaltenname enthält B.Spaltenname?
- A.Spaltenname enthält B.Tabellenname?
- B.Spaltenname enthält A.Spaltenname?

Idee der Heuristik ist, dass beim Anlegen von Datenbanken oft der Name der referenzierten Tabelle oder des referenzierten Attributs im Namen des abhängigen Attribut benutzt wird.

Bei dieser Heuristik handelt es sich um eine bestätigende Heuristik.

Mögliche Rückgabewerte: *TRUE*, *UNKNOWN*

3.2.8 Kleine Tabellen

Die Heuristik zählt die Anzahl der Tupel im abhängigen und referenzierten Attribut. Wird ein konfigurierter Grenzwert dabei unterschritten, wird eine negative Bewertung getroffen.

Die Idee dieser Heuristik ist, Verknüpfungen zwischen sehr kurzen Tabellen auszuschließen. Solche IND's kommen oft bei Tabellen vor, die einen Zahlenwert als Schlüssel benutzen. Beispiel: In einer (fiktiven) Filmdatenbank gibt es eine Tabelle in der Anredearten von Personen beschreiben sind (1- „Frau“, 2 - „Herr“, 3 - „Frau Dr“, 4 - „Herr Dr“) und eine Tabelle in der Genre von Filmen gespeichert sind (1- „Horror“, 2 - „Drama“, 3 - „Sci-Fi“, 4 - „Komödie“). Die Schlüssel beider Tabellen würden als IND erkannt werden. Sie stehen aber semantisch in keinem Zusammenhang und müssten daher als Fremdschlüssel ausgeschlossen werden.

Diese Heuristik ist eine ausschließende Heuristik.

Mögliche Rückgabewerte: *FALSE*, *UNKNOWN*

3.3 Evaluierung der Heuristiken

3.3.1 Testumgebung

Um die Wirksamkeit der Heuristiken sinnvoll zu testen und ihre individuellen Parameter zu ermitteln, wurden einige Testdatenbanken ausgesucht, deren Struktur bekannt ist und mit deren Hilfe eine Aussage über die Korrektheit des Ergebnisses getroffen werden kann. Die Testdatenbanken in der Übersicht:

Datenbank	Beschreibung	Tabellen	Tupel	IND's (davon bekannte FK's)
SCOP ²	Life Science Datenbank deren Struktur nur in Teilen erschlossen ist. Nur wenige Fremdschlüssel vorhanden.	4	~340.000	11 (5)
TPC-H ³	Benchmarkdatenbank mit sehr wenigen Tabellen, dafür einer sehr großen Datenmenge. Es liegen neun Fremdschlüssel vor.	8	8,6 Mio	33 (9)
UNIPROT ⁴	Große Life Science Datenbank mit 33 Fremdschlüsselbeziehungen, von denen einige schwer zu erfassen sind, da sie nur auf einem geringen Wertebereich von ein oder zwei Werten basieren.	29	~ 8,4 Mio	36 (31)
MOVIELENS ⁵	Kleinere Filmdatenbank deren Hauptinhalt Userbewertungen sind. Sechs echte Fremdschlüssel sind in der Datenbank vorhanden.	7	> 1 Mio	19 (6)
IMDB ⁶	Internationale Filmdatenbank, mit einem sehr großen Archiv an Filmen, deren Mitwirkenden und deren Auszeichnungen. ~390.000 Filme liegen in der Datenbank vor, dazu über 3 Mio Schauspieler. 14 Fremdschlüsselbeziehungen sind in dieser Datenbank bekannt.	22	> 8,3 Mio	34 (14)
FILMDIENST ⁷	Filmdatenbank eines deutschen Anbieters, die in etwa 55.000 Filme enthält. Darüber hinaus Informationen zu Genres, Personen und Auszeichnungen. Es existieren 15 Fremdschlüsseln in dieser Datenbank.	14	> 1,3 Mio	79 (15)
PDB ⁸	Sehr große Life Science Datenbank mit unbekannter Struktur. Grund diese Datenbank mit in den Test einzubeziehen war es, die Performance auf sehr großen Datenbanken zu testen.	176	~ 266 Mio	4972 (unbekannt)

Tabelle 1: Beschreibung der Testdatenbanken

Diese Testdatenbanken sind frei verfügbar und lagen bereits auf den Rechnern des Lehrstuhls vor. Einige mussten neu importiert werden.

2 <http://scop.mrc-lmb.cam.ac.uk/scop/>

3 <http://www.tpc.org/tpch/>

4 <http://www.pir.uniprot.org/>

5 <http://movielens.umn.edu/login> (registrierung notwendig)

6 <http://german.imdb.com/>

7 <http://film-dienst.kim-info.de/>

8 <http://www.rcsb.org/pdb/home/home.do>

3.3.1.1 Testläufe

Nach der Umsetzung der beschriebenen Heuristiken, wurde eine Vielzahl von Testdurchläufen auf den Testdatenbanken durchgeführt, um Antworten auf drei Fragen zu finden:

1. Welche Heuristiken sind für ein maximal korrektes Ergebnis sinnvoll, welche können ignoriert werden?
2. Welche Konfigurationsparameter sind für einzelne Heuristiken sinnvoller Weise zu benutzen, um eine maximale Anzahl von Fremdschlüsseln zu finden?
3. Welche Gewichtung der einzelnen Heuristiken ist sinnvoll, um möglichst wenige IND's falsch zu bewerten.

3.3.1.2 Bewertungskriterien

Um eine *Konfiguration*⁹ zu bewerten, spielen vier Werte eine Rolle. Ziel ist es, möglichst viele Fremdschlüssel als solche zu erkennen und parallel möglichst viele IND's als mögliche Fremdschlüssel auszuschließen. Die Werte im Einzelnen:

Richtig Positiv	Anzahl der IND's, die von der Heuristik als FK erkannt werden und ein Fremdschlüssel sind.
Falsch Positiv	Anzahl der IND's, die von der Heuristik als FK erkannt werden, jedoch kein Fremdschlüssel sind.
Richtig Negativ	Anzahl der IND's, die von der Heuristik als FK ausgeschlossen werden und kein Fremdschlüssel sind.
Falsch Negativ	Anzahl der IND's, die von der Heuristik als FK ausgeschlossen werden, jedoch ein Fremdschlüssel sind.

Tabelle 2: Bewertungskriterien Konfigurationen

Diese Werte können zusammengefasst werden zu den Maßen Precision und Recall:

1. Precision: $Precision = \frac{(Richtig\ Positiv)}{(Richtig\ Positiv + Falsch\ Positiv)}$
2. Recall: $Recall = \frac{(Richtig\ Positiv)}{(Richtig\ Positiv + Falsch\ Negativ)}$
3. F-Measure: $F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)}$

Ziel ist es sowohl Precision als auch Recall zu maximieren.

3.3.2 Evaluierung der einzelnen Heuristiken

Schon nach wenigen Testdurchläufen wurde klar, dass sich die Ergebnisse einzelner Heuristiken überdecken bzw. einige Heuristiken für sich genommen keine aussagekräftigen Ergebnisse liefern und daher nur eine Untermenge der vorgestellten Heuristiken nötig ist, um ein gutes Ergebnis zu erzielen.

⁹ Eine *Konfiguration* besteht aus der Gewichtung der benutzten Heuristiken und der Konfigurationsparameter für diese Heuristiken.

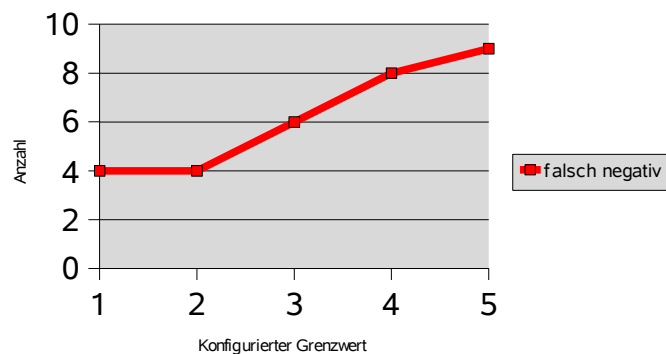
Für ausschließende Heuristiken werden nur richtig/falsch negative betrachtet da diese nur negative Entscheidungen treffen. Für bestätigende Heuristiken werden nur richtig/falsch positive betrachtet, da sie nur positive Bewertungen treffen.

Basis für die Bewertungen der einzelnen Heuristiken sind Testläufe mit verschiedenen Konfigurationen auf allen benutzten Testdatenbanken. Dabei wurden die Ergebnisse einzelner Datenbanken zu einem Gesamtergebnis addiert.

3.3.2.1 Minimale Anzahl abhängiger Werte

Die Heuristik liefert auf allen DB's lediglich vier IND's Falsch Negative und 12 Richtig Negative auf den insgesamt 212 IND's. Die Falsch Negativen sind Verknüpfungen in denen eine Klassifizierung eines Objekts vorgenommen wird, jedoch alle Objekte die selbe Klasse haben und daher auch den gleichen Attributwert. Betroffen ist eine Verknüpfung in der Datenbank *Filmdienst*. In dieser werden Filmtexte gespeichert, die von einem bestimmten Typ sind. Zwar sind vier verschiedene Texttypen in der Tabelle *Filmtexttypen* vorhanden, benutzt wird davon jedoch nur einer.

Aus dem Diagramm in Abbildung 5 geht hervor, dass auf Basis der Testdatenbanken kein Unterschied besteht, ob die konfigurierte Grenze auf eins oder zwei gesetzt wird. Die Grenze von zwei erscheint jedoch sinnvoller, um auch Binärattribute auszuschließen. Zur Erinnerung: Der Grenzwert gibt an, wie viele verschiedene Werte im abhängigen Attribut vorkommen dürfen, ohne dass die Heuristik *FALSE* zurück gibt.



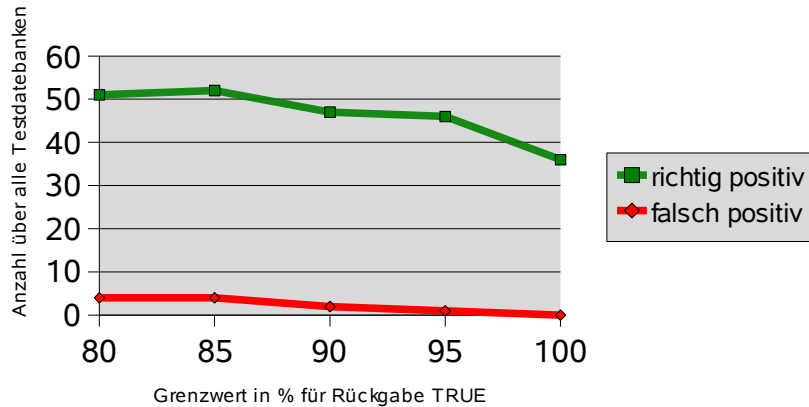
3.3.2.2 Abdeckung

Die *Abdeckung* - Heuristik lieferte in nahezu allen Testdurchläufen sehr gute Ergebnisse. In allen Testdatenbanken haben ca. 60% der vorhandenen Fremdschlüssel eine Abdeckung von fast 100%. Die restlichen Fremdschlüssel haben eine Abdeckung von unter ~90% und über ~10%.

Um auf sinnvolle Grenzwerte zu kommen, wurden sehr viele Testdurchläufe durchgeführt. Zur Erinnerung: Die Heuristik hat zwei Grenzwerte, eine obere und eine untere Grenze. Liegt die Abdeckung unter der unteren Grenze, gibt die Heuristik *FALSE* zurück, liegt sie hingegen über der oberen Grenze, gibt sie *TRUE* zurück. Sonst *UNKNOWN*.

Die folgenden zwei Diagramme (Abbildungen 6 und 7) zeigen, wie viele IND's Falsch Positiv/Negativ bzw. Richtig Positiv/Negativ bei verschiedenen Grenzwerten sind.

Für den oberen Grenzwert:



Für den unteren Grenzwert:

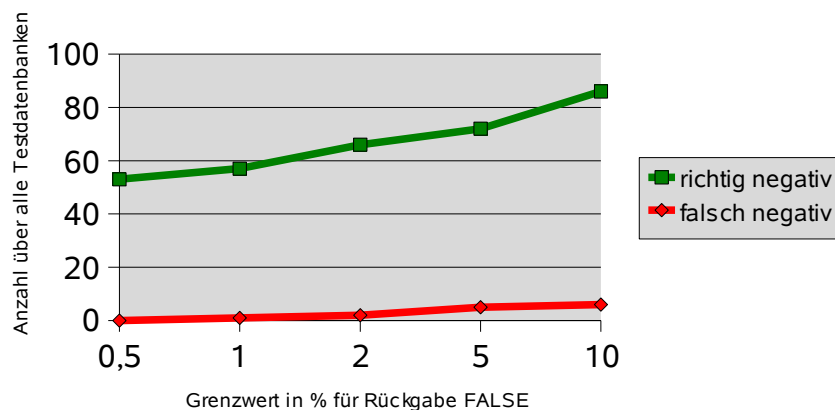


Abbildung 7: Konfiguration für die Heuristik Abdeckung (unterer Grenzwert)

Daher werden die Grenzwerte wie folgt empfohlen:

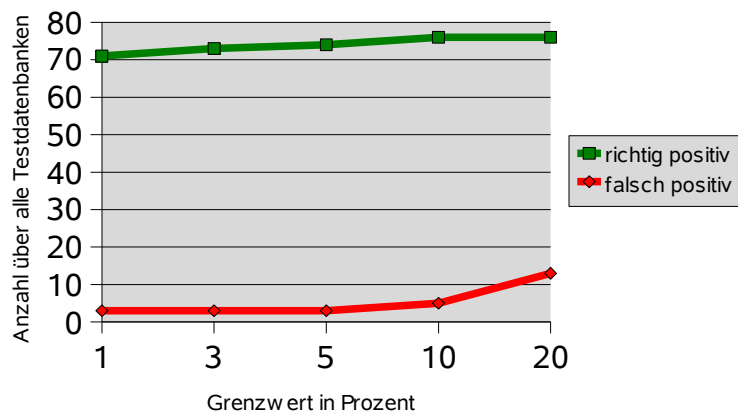
- Oberer Grenzwert: 90%
- Unterer Grenzwert: 1%

3.3.2.3 Verhältnis der begrenzten Wertemengen

Die Ergebnisse dieser Heuristik ähneln den Ergebnissen der Heuristik *Abdeckung* sehr, wenn auch mit einem Unterschied: Fällt bei *Abdeckung* die Entscheidung auf *UNKNOWN*, wird bei dieser Heuristik *FALSE* zurück gegeben, da in dieser Heuristik der Rückgabewert *UNKNOWN* nicht vorgesehen ist.. Dies ist jedoch häufig die falsche Entscheidung und der Rückgabewert der Heuristik *Verteilung über referenziertem Attribut* bietet eine bessere Bewertung. Daher ist die Kombination aus *Abdeckung* und *Verteilung über referenziertem Attribut* dieser Heuristik vorzuziehen und diese wird nicht weiter betrachtet.

3.3.2.4 Verteilung über referenziertem Attribut

Diese Heuristik hat die Fremdschlüssel der Testdatenbanken sehr zuverlässig gefunden. Auch hier waren einige Testläufe notwendig, um einen Grenzwert zu finden, der eine gute Balance zwischen möglichst vielen richtig-positiv und möglichst wenigen falsch-positiv Entscheidungen bietet. Das folgende Diagramm (Abbildung 8) zeigt eine Auswahl einiger Testdurchläufe mit verschiedenen Konfigurationswerten:



Besonders das Zusammenspiel zwischen *Abdeckung* und *Verteilung über referenziertem Attribut* Heuristik macht sich in den Ergebnissen bemerkbar, da diese beiden Heuristiken die beiden wesentlichen Eigenschaften von Fremdschlüsseln abdecken und sich daher sehr gut ergänzen.

3.3.2.5 Verhältnis der Wertlängen

Im Zuge der Testdurchläufe waren die Ergebnisse dieser Heuristik in der Regel ohne große Aussagekraft. Zu oft waren die Längen der Attribute sehr ähnlich, ohne jedoch in einem semantischen Zusammenhang zu stehen. Besonders sehr kurze Durchschnittslängen waren problematisch. Die Einführung einer Mindestlänge¹⁰ brachte dabei auch keine wesentliche Verbesserung der Ergebnisse.

Da andere Heuristiken, für sich genommen, wesentlich bessere Ergebnisse liefern, wird diese Heuristik im Weiteren nicht mehr betrachtet.

3.3.2.6 Anzahl referenzierter Attribute

Die Heuristik lieferte in keiner der untersuchten IND's der Testdatenbanken den Wert FALSE zurück. Dennoch erscheint es sinnvoll, dass sie in der Menge der zu benutzenden Heuristiken verbleibt, da ein Attribut in der Regel nur zu einem anderen in Beziehung stehen sollte und mit dieser Heuristik IND's, bei denen das nicht der Fall ist, ausgeschlossen werden können.

Dies ist natürlich nur ein theoretischer Gedanke, da in den durchgeführten Tests dieser Fall nicht aufgetreten ist.

Als Grenzwert bei dieser Heuristik ist der Wert eins sinnvoll. Wenn ein abhängiges Attribut mehr als ein anderes Attribut referenziert, gibt die Heuristik FALSE zurück.

¹⁰ Wenn die Durchschnittslänge beider Attribute unter dieser Grenze liegt, wird *UNKNOWN* zurück gegeben.

3.3.2.7 *Attributnamen*

Wie zu erwarten, liefert die Heuristik sehr gute Ergebnisse bei Datenbanken in denen die Attribute eines Fremdschlüssels auch eine ähnliche Bezeichnung haben. Da dies jedoch nicht in jeder Datenbank gegeben ist, fällt die Gewichtung im Endergebnis nur gering aus.

Von den untersuchten Testdatenbanken hat nur eine keine Namensgleichheit oder -ähnlichkeit in den beteiligten Attributen einer IND.

3.3.2.8 *Kleine Tabellen*

In den Testdatenbanken gab es nahezu keine IND's zwischen zwei sehr kurzen Tabellen. Daher und auf Grund der sehr geringen Aussagekraft dieser Heuristik, wird sie im Weiteren nicht mehr betrachtet.

3.3.3 *Kombination der Heuristiken*

3.3.3.1 *Gesamtergebnis aller Heuristiken für eine Inklusionsabhängigkeit*

Die Bewertungen der verschiedenen Heuristiken werden für eine IND zusammengerechnet und bilden ein Gesamtergebnis. Dabei spielt die Gewichtung der Einzelergebnisse eine sehr große Rolle. So hat eine Heuristik, welche die Abdeckung (s. 2.2.1) untersucht, nicht die gleiche Aussagekraft wie eine Heuristik, welche die Ähnlichkeit der Attributnamen vergleicht.

Die übliche Vorgehensweise für ein gewichtetes Gesamtergebnis wäre die Zuordnung eines festen Gewichts zu jeder Heuristik wobei die Summe der gewichte 100% ergäbe. Dieses Vorgehen ist jedoch im vorliegenden Fall ungeeignet, da eine Heuristik auch keine Aussage zu einer IND treffen kann (Rückgabewert *UNKNOWN*). In diesem Fall soll die Heuristik überhaupt nicht zum Gesamtergebnis beitragen. Aufgrund dieser Beobachtung wird der folgende Gewichtungsansatz gewählt: Es werden zwei Werte für jede IND gespeichert:

1. Die Anzahl der erreichten Punkte
2. Die Anzahl der möglichen Punkte

Jede Heuristik hat eine bestimmte Anzahl von Punkten, die in Abhängigkeit von ihrem Rückgabewert auf diese beiden Werte aufaddiert werden. Heuristiken mit großer Aussagekraft bekommen einen höheren Punktwert als solche mit niedriger Aussagekraft. Die Abhängigkeit zwischen dem Rückgabewert einer Heuristik und der Aufsummierung zeigt die folgende Tabelle:

Rückgabewert	Anzahl der erreichten Punkte	Anzahl der möglichen Punkte
TRUE	Addierung	Addierung
FALSE	keine Addierung	Addierung
UNKNOWN	keine Addierung	keine Addierung

Tabelle 3: Rückgabewerte von Heuristiken und deren Auswirkung

Aus diesen beiden Zahlenwerten, lässt sich ein Ranking aller untersuchten IND's erstellen. Dieses Ranking gibt die Wahrscheinlichkeit wieder, ob eine IND ein Fremdschlüssel ist.

Beispiel für die Ausgabe eines Rankings von IND's (Abbildung 9):

MOVIE_GENRES.GENRE_ID	-> GENRES.ID	130/130 = 100.0%
RATINGS.MOVIE_ID	-> MOVIES.ID	130/130 = 100.0%
MOVIE_GENRES.MOVIE_ID	-> MOVIES.ID	130/130 = 100.0%
RATINGS.USER_ID	-> USERS.ID	130/130 = 100.0%
USERS.AGE	-> AGES.ID	100/100 = 100.0%
USERS.OCCUPATION	-> OCCUPATIONS.ID	100/100 = 100.0%

Abbildung 9: Ausgabe eines Rankings von IND's

Die Bewertung „100%“ sagt dabei aus, dass diese IND auf Grundlage der untersuchten Eigenschaften ein Fremdschlüssel ist.

Die Reihenfolge der einzelnen IND's ergibt sich aus zwei Kriterien in dieser Reihenfolge:

1. Gesamtprozentzahl (= erreichte Punkte / mögliche Punkte * 100)
2. Erreichte Punkte

3.3.3.2 Empfehlung für Gewichtung

Das Endergebnis eines Testdurchlaufs ist das beschriebene Ranking, in dem tabellarisch dargestellt wird, welche IND's vermutlich Fremdschlüssel sind und welche nicht. Dies geschieht durch eine Prozentangabe. Diese Prozentangabe kann durch zwei Parameter beeinflusst werden. Zum einen die Konfigurationsparameter der einzelnen Heuristiken, wie im vorherigen Absatz beschrieben, zum anderen durch die Gewichtung der einzelnen Heuristiken im Endergebnis. Ziel war es, einen möglichst eindeutigen Grenzwert zu finden, anhand dessen im Endergebnis abgelesen werden kann, welche IND's Fremdschlüssel sind und welche nicht.

Ein Beispiel, wie Gewichtungen auf das Endergebnis Einfluss haben können: Die Heuristik *Minimale Anzahl abhängiger Werte* kann einen Fremdschlüssel sehr zuverlässig ausschließen. Wenn sie eine IND als Fremdschlüssel ausschließt, soll keine andere Heuristik diese Entscheidung revidieren können. Daher bekommt die Heuristik *Minimale Anzahl abhängiger Werte* in der Gewichtung einen so hohen Wert, dass dieser von anderen Heuristiken nicht mehr ausgeglichen werden kann.

Die folgende Tabelle zeigt exemplarisch drei mögliche Gewichtungen und welcher Grenzwert benutzt werden muss, um einen hohen Recall bei gleichzeitiger hoher Precision zu bekommen.

Beschrieben sind für jede Konfiguration die Anzahl der Punkte, mit denen eine Heuristik in die Gesamtwertung eingeht und welcher Grenzwert nötig ist, um ein maximales F-Measure zu erhalten. Am Ende der Tabelle findet sich die Angabe, wie viele IND's in allen Testdatenbanken falsch bewertet wurden.

	Gewichtung 1	Gewichtung 2	Gewichtung 3
Minimale Anzahl abhängiger Werte	30	30	100
Abdeckung	30	30	50
Anzahl referenzierter Attribute	30	30	100
Attributnamen	10	10	30
Verteilung über referenziertem Attribut	20	50	50
Grenzwert für FK	85%	88%	100%
Falsch Positive	34	2	3
Falsch Negative	8	6	5

Tabelle 4: Verschiedene Gewichtungen von Heuristiken

Die Konfiguration, wie sie in Gewichtung 3 beschrieben ist, ist auch die Empfehlung für die weitere Benutzung des Tools. Sie zeigt, durch den eindeutigen Grenzwert, am deutlichsten, welche IND's vermutlich ein Fremdschlüssel sind. Darüber hinaus wird nur eine geringe Anzahl an IND's falsch bewertet.

Generell bleibt anzumerken, dass sich der Nutzer des Tools bei der Überlegung zur Gewichtung entscheiden muss: Ob möglichst viele Fremdschlüssel gefunden werden sollen und dabei auch der ein oder andere falsch gefunden wird oder die Gewichtung so gewählt wird, dass nur sehr wenige IND's falsch bewertet werden und vielleicht ein paar wenige Fremdschlüssel durch das Raster fallen. Für die Qualität des Gesamtergebnisses würde sprechen, möglichst wenige IND's falsch-positiv zu bewerten.

3.3.3.3 Alternative Methode

Zu der im Absatz 3.3.3.1 beschriebenen Methode die Ergebnisse der Heuristiken durch eine Gewichtung miteinander zu einem Endergebnis zu kombinieren, gibt es eine Alternative. Diese besteht darin, die Heuristiken in Form einer Reihe nacheinander zu durchlaufen. Sobald eine der Heuristiken eine Entscheidung für oder gegen einen Fremdschlüssel trifft, wird der Durchlauf für die IND beendet. Folgende Heuristiken finden in dieser Methode Anwendung. Wichtig ist hierbei die Reihenfolge:

1. *Minimale Anzahl abhängiger Werte* (kann einen FK ausschließen)
2. *Abdeckung1* (bestätigt einen FK bei überschreiten der oberen Grenze)
3. *Abdeckung2* (schließt einen FK bei unterschreiten der unteren Grenze aus)
4. *Verteilung über referenziertem Attribut* (bestätigt einen FK oder schließt ihn aus)

Aus der Reihenfolge ist zu sehen, dass es immer eine Entscheidung für oder gegen einen Fremdschlüssel gibt.

Mit dieser Methode lässt sich das selbe Ergebnis auf den Testdatenbanken erzielen, wie mit der Methode in 3.3.3.1. So werden im Schema UNIPROT alle falsch-positiven IND's ausgeschlossen und lediglich drei echte Fremdschlüssel. Diese drei ausgeschlossenen Fremdschlüssel stellen jedoch generell einen Sonderfall in den Ergebnissen dar, da sie lediglich einen Wertebereich der Kardinalität eins im abhängigen Attribut besitzen.

Im TPC-H Schema werden 23 der 24 falsch-positiven IND's als solche erkannt. Die verbliebene IND hat eine Abdeckung von 99% und wird daher als Fremdschlüssel eingeordnet.

Diese Alternative ist starr in Ihrer Bewertung. Die flexiblere Methode aus 3.3.3.1 wird daher bevorzugt, da mit ihr besser auf die Erkenntnisse in anderen Datenbanken reagiert werden kann. So können durch die Änderung der Gewichtung Spezialisierungen besser aufgefangen werden.

3.3.4 Ergebnisse für einzelne Schemata

In den folgenden Abschnitten werden die Ausgaben des Tools für die einzelnen Testdatenbanken beschrieben. Die benutzte Konfiguration ist in den nachstehenden Tabellen aufgezeigt:

Heuristik	Gewichtung
Minimale Anzahl abhängiger Werte	100
Abdeckung	50
Anzahl referenzierter Attribute	100
Attributnamen	30
Verteilung über referenziertem Attribut.	50

Tabelle 5: Gewichtung der Heuristiken in den Testergebnissen

Heuristik	Konfigurationsparameter	Wert
Minimale Anzahl abhängiger Werte	Minimale Anzahl an Werten	2
Abdeckung	Obere Grenze	95%
Abdeckung	Untere Grenze	1%
Verteilung über referenziertem Attribut	Anzahl Werte kleiner als der minimale Wert in A ¹¹ oder größer als der maximale Wert in A	5%
Anzahl referenzierter Attribute	Maximale Anzahl referenzierter Attribute	1

Tabelle 6: Konfiguration der Heuristiken in den Testergebnissen

¹¹ A = abhängiges Attribut

Als Grenzwert des Gesamtergebnisses, dass eine IND als Fremdschlüssel angesehen wird, kommt ein Wert von 100% zum Tragen.

Die detaillierten Ergebnisse sind im Anhang im Abschnitt 6.2 angefügt.

3.3.4.1 UNIPROT

Von den vorhandenen 33 Fremdschlüsselbeziehungen werden 29 richtig-positiv mit 100% bewertet. Drei weitere bekommen eine Bewertung von ~56% bzw. ~16% und sind damit falsch-negativ eingeordnet. Hierbei handelt es sich um Verknüpfungen, in denen nur ein Wert referenziert wird.

Ein Fremdschlüssel (*SG_BIOENTRY_QUALIFIER_ASSOC.TRM_OID* -> *SG_TERM.OID*) bekommt eine Bewertung von ~37%. Hierbei handelt es sich um eine Verknüpfung, in der das abhängige Attribut 9% der Werte im referenzierten Attribut abdeckt. Darüber hinaus befinden sich 65% der Werte vom referenzierten Attribut außerhalb der Grenzen, die durch den minimalen und maximalen Wert im abhängigen Attribut vorgegeben werden. Erschwerend für die Bewertung kommt hinzu, dass der Wertebereich mit 44 Werten im referenzierten Attribut sehr gering ist.

Die Ergebnisse in der Übersicht

Bewertung	Anzahl
Richtig Positiv	29 (von 33 echten FK)
Falsch Positiv	0
Falsch Negativ	4

Tabelle 7: Ergebnisse für Schema UNIPROT

Damit ergibt sich eine Precision von 1 bei einem Recall von 0,87 für diese Datenbank.

3.3.4.2 SCOP

Alle vorhandenen Fremdschlüssel werden mit 100% bewertet. Keine IND wird falsch bewertet.

Bewertung	Anzahl
Richtig Positiv	5 (von 5 echten FK)
Falsch Positiv	0
Falsch Negativ	0

Tabelle 8: Ergebnisse für Schema SCOP

Damit ergibt sich eine Precision von 1 bei einem Recall von 1 für diese Datenbank.

3.3.4.3 TPC-H

Von den neun vorhandenen Fremdschlüsseln werden alle mit 100% bewertet. Die anderen IND's bekommen eine Bewertung von 0%.

Eine IND (*PARTSUPP.PS_AVAILQTY* -> *SUPPLIER.S_SUPPKEY*) wird mit 100% bewertet, die kein Fremdschlüssel ist. Hierbei handelt es sich um zwei Attribute, die aus Zahlenwerten bestehen und bei denen die Tabellen annähernd gleich groß sind. Darüber hinaus ist der Wertebereich dieser zwei Spalten mit ~10.000 Werten recht hoch.

Bewertung	Anzahl
Richtig Positiv	9 (von 9 echten FK)
Falsch Positiv	1
Falsch Negativ	0

Tabelle 9: Ergebnisse Schema TPC-H

Damit ergibt sich eine Precision von 0,9 bei einem Recall von 1 für diese Datenbank.

3.3.4.4 Filmdienst

Zwölf der vorhandenen 15 Fremdschlüssel werden mit 100% bewertet. Eine IND (*FILMTEXTTYPEN.FILMTEXTTYPNR* -> *FILMBEWERTUNGEN.FILMBEWERTUNGNR*) bekommt darüber hinaus eine Bewertung von 100%, ist aber kein Fremdschlüssel. Hierbei handelt es sich um eine Verknüpfung, bei denen in beiden Spalten die Zahlen eins bis vier als Schlüssel benutzt werden, ohne dass ein semantischer Zusammenhang besteht.

Drei Fremdschlüssel¹² bekommen eine Bewertung von unter 50%. Davon ist der Wertebereich des abhängigen Attributs gleich eins. Die anderen beiden Verknüpfungen wären mit einer leicht erhöhten Grenze (~7,5% statt der benutzten 5%) bei der *Verteilung über referenziertem Attribut* - Heuristik Richtig Positiv. Allerdings hätte dieses Auswirkungen auf die Datenbank IMDB, da dort dann drei Falsch Positive entstehen würden.

Bewertung	Anzahl
Richtig Positiv	12 (von 15 echten FK)
Falsch Positiv	1
Falsch Negativ	3

Tabelle 10: Ergebnisse im Schema Filmdienst

Damit ergibt sich eine Precision von 0,92 bei einem Recall von 0,8 für diese Datenbank.

3.3.4.5 IMDB

Von den vorhandenen 14 Fremdschlüsseln werden alle mit 100% bewertet. Die anderen IND's bekommen zum Großteil eine Bewertung von 37%.

Eine IND (*MOVIE_LINKS.ID* -> *DIRECTORS.ID*) wird mit 100% bewertet, die jedoch kein Fremdschlüssel ist. Dort liegt eine Abdeckung von 92% vor und die Werte im referenzierten Attribut, die außerhalb der Grenzen des minimalen und maximalen Werts des abhängigen Attributs liegen, betragen lediglich 1%. Dies ist erneut eine Verknüpfung basierend auf Zahlen, deren Anzahl mit etwa 388.000 sehr hoch ist.

¹² AUSZEICHNUNGEN.FILMNR -> FILME.FILMNR, AUSZEICHNUNGEN.PERSONNR -> PERSONEN.PERSONNR und FILMTEXTE.FILMTEXTTYPNR -> FILMTEXTTYPEN.FILMTEXTTYPNR

Bewertung	Anzahl
Richtig Positiv	14 (von 14 echten FK)
Falsch Positiv	1
Falsch Negativ	0

Tabelle 11: Ergebnisse im Schema IMDB

Damit ergibt sich eine Precision von 0,93 bei einem Recall von 1 für diese Datenbank.

3.3.4.6 MOVIELENS

Von den vorhandenen sechs Fremdschlüsseln werden sechs mit 100% bewertet. Die restlichen IND's bekommen zur Hälfte eine Bewertung von ~37% bzw. 0%.

Bewertung	Anzahl
Richtig Positiv	6 (von 6 echten FK)
Falsch Positiv	0
Falsch Negativ	0

Tabelle 12: Ergebnisse im Schema MovieLens

Damit ergibt sich eine Precision von 1 bei einem Recall von 1 für diese Datenbank.

4 Implementation

Das Ergebnis dieser Studienarbeit ist neben der Definition und Evaluierung der Heuristiken auch ein Javaprogramm mit dem Namen: *FKFinder*. Das Programm wendet die beschriebenen Heuristiken auf ein gegebenes Schema an und gibt das Gesamtergebnis aus bzw. speichert es in einer Datenbank.

Grundlage für dieses Programm ist die existierende Klassen- und Paketstruktur von SPIDER. Einige wenige Klassen wurden erweitert, der Großteil neu hinzugefügt.

4.1 Benutzerschnittstellen

Das Programm kann sowohl als reine Konsolenapplikation benutzt werden, als auch über eine GUI. Die Ausgabe erfolgt sortiert nach dem Ranking der IND's zusammen mit den Einzelergebnissen einzelner Heuristiken.

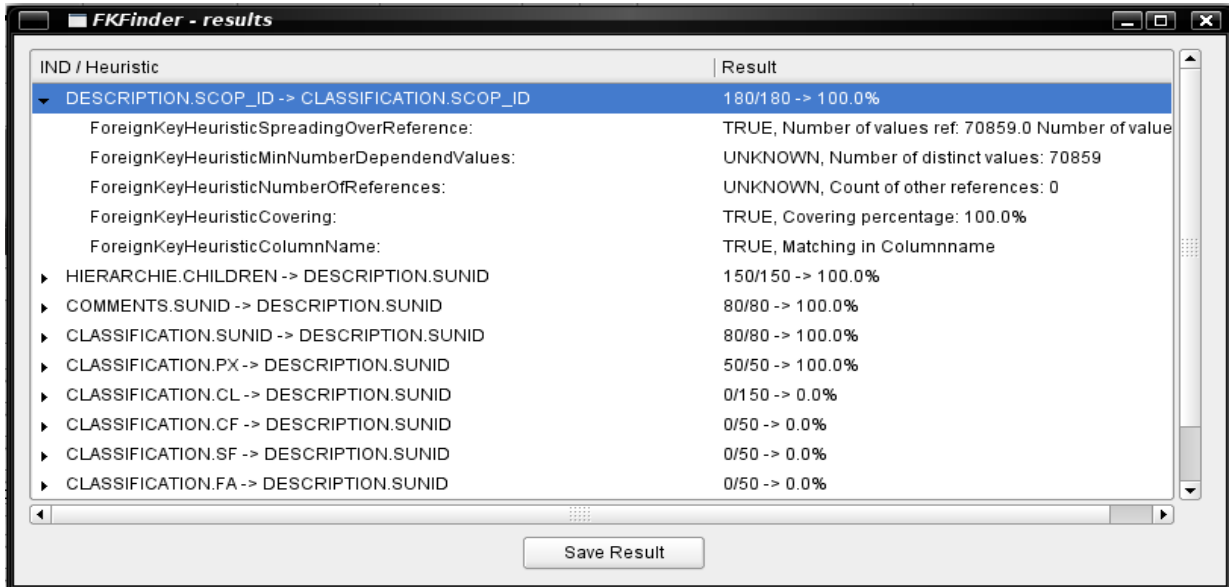
Beispielausgabe auf einer Konsole (Abbildung 10):

```
Results for Schema 'SCOP'
Time: 19s
DESCRIPTION.SCOP_ID -> CLASSIFICATION.SCOP_ID
Result: 180/180 ->100.0%
Heuristics: 3/3
  ForeignKeyHeuristicSpreadingOverReference --> TRUE, Number of values ref:
70859.0 Number of values outer: 0.0(0.0%) max allowed: 4960.1304
  ForeignKeyHeuristicMinNumberDependendValues --> UNKNOWN, Number of distinct
values: 70859
  ForeignKeyHeuristicNumberOfReferences --> UNKNOWN, Count of other references:
0
  ForeignKeyHeuristicCovering --> TRUE, Covering percentage: 100.0%
  ForeignKeyHeuristicColumnName --> TRUE, Matching in Columnname

HIERARCHIE.CHILDREN -> DESCRIPTION.SUNID
Result: 150/150 ->100.0%
Heuristics: 2/2
  ForeignKeyHeuristicSpreadingOverReference --> TRUE, Number of values ref:
94440.0 Number of values outer: 0.0(0.0%) max allowed: 6610.8003
  ForeignKeyHeuristicMinNumberDependendValues --> UNKNOWN, Number of distinct
values: 94440
  ForeignKeyHeuristicNumberOfReferences --> UNKNOWN, Count of other references:
0
  ForeignKeyHeuristicCovering --> TRUE, Covering percentage: 100.0%
  ForeignKeyHeuristicColumnName --> UNKNOWN, No Matching in Names
...
```

Abbildung 10: Ergebnisausgabe FKFinder (Shell)

Wenn das Programm mit der GUI gestartet wird, erfolgt die Ausgabe des Ergebnis in Form einer Tabelle (Abbildung 11):



IND / Heuristic	Result
DESCRIPTION.SCOP_ID -> CLASSIFICATION.SCOP_ID	180/180 -> 100.0%
ForeignKeyHeuristicSpreadingOverReference:	TRUE, Number of values ref: 70859.0 Number of value
ForeignKeyHeuristicMinNumberDependentValues:	UNKNOWN, Number of distinct values: 70859
ForeignKeyHeuristicNumberOfReferences:	UNKNOWN, Count of other references: 0
ForeignKeyHeuristicCovering:	TRUE, Covering percentage: 100.0%
ForeignKeyHeuristicColumnName:	TRUE, Matching in Columnname
▶ HIERARCHIE.CHILDREN -> DESCRIPTION.SUNID	150/150 -> 100.0%
▶ COMMENTS.SUNID -> DESCRIPTION.SUNID	80/80 -> 100.0%
▶ CLASSIFICATION.SUNID -> DESCRIPTION.SUNID	80/80 -> 100.0%
▶ CLASSIFICATION.PX -> DESCRIPTION.SUNID	50/50 -> 100.0%
▶ CLASSIFICATION.CL -> DESCRIPTION.SUNID	0/150 -> 0.0%
▶ CLASSIFICATION.CF -> DESCRIPTION.SUNID	0/50 -> 0.0%
▶ CLASSIFICATION.SF -> DESCRIPTION.SUNID	0/50 -> 0.0%
▶ CLASSIFICATION.FA -> DESCRIPTION.SUNID	0/50 -> 0.0%

Save Result

4.2 Struktur der Implementation

Dieser Abschnitt beschreibt die Struktur und Bedienung des Programms „FKFinder“. Das Programm liegt in folgender Ordnerstruktur (Abbildung 12):

```

FKFinder
|-- src
    |-- de
        |-- ...
|-- lib
    |-- ...
|-- cfg
    |-- ForeignKeyFinder.properties
|-- bin
    |-- de
        |-- ...

```

Abbildung 12: Ordnerstruktur FKFinder

Das Verzeichnis *src* enthält die Quellcodedateien, in *lib* befinden sich die nötigen zusätzlichen Bibliotheken, in *cfg* befindet sich eine oder mehrere Konfigurationsdateien. Unter *bin* werden die kompilierten Dateien des Programms gespeichert.

4.2.1 Konfiguration

Innerhalb des Programms können verschiedenen Parameter konfiguriert werden. Die Speicherung der Konfiguration erfolgt über eine java-properties Datei. Diese enthält sowohl generelle Parameter zur Datenbank (zu untersuchendes Schema, Username und Passwort), als auch die Konfigurationsparameter für die einzelnen Heuristiken.

In der Konfigurationsdatei können einzelne Heuristiken ausgeschlossen werden, um nur eine Untergruppe der verfügbaren Heuristiken zu benutzen.

Der konfigurierbare Debuglevel ist ausschlaggebend dafür, wie viele Ausgaben auf der Konsole ausgegeben werden. Sinnvolle Werte sind *INFO*, um nur das nötigste zu sehen, oder *DEBUG*, um alle Ausgaben zu sehen.

Die GUI-Variante bietet die Möglichkeit alle verfügbaren Konfigurationsparameter anzuzeigen und einfach zu ändern (Beispiel in Abbildung 13). Darüber hinaus können vorhandene Konfigurationen importiert und in einer Datei gespeichert werden.

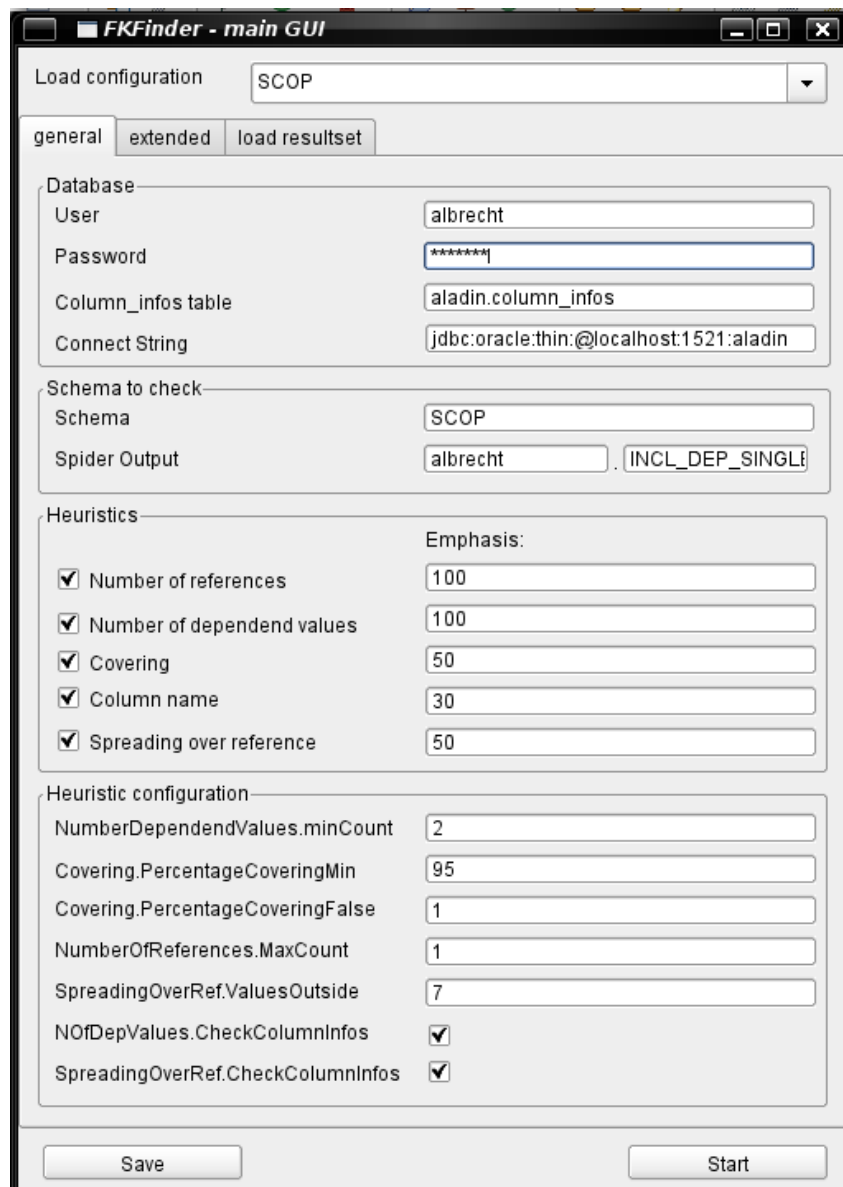


Abbildung 13: Konfiguration FKFinder über GUI

4.2.2 Start des Programms

Das Programm FKFinder wird in der Konsolenvariante durch den Aufruf der Hauptklasse (`de.huberlin.informatik.wbi.aladin.intraschema.ForeignKeyFinderExec`) gestartet. Voraussetzungen für das Programm sind:

- JDBC Treiber im CLASSPATH
- Logger¹³ library im CLASSPATH

Der Aufruf der GUI erfolgt über die Klasse des Konfigurationsfensters mit der Klasse: `de.huberlin.informatik.wbi.aladin.intraschema.fkfinderGui.FkfinderGui`. In diesem Fenster wird die Konfiguration des Tools durchgeführt und im Anschluss der Suchlauf gestartet.

4.2.3 Implementation einzelner Heuristiken

In diesem Abschnitt wird kurz auf die Implementation einzelner Heuristiken eingegangen um die Länge der Laufzeiten verständlich zu machen. Außerdem wird der jeweilige Klassenname genannt, um die Übersicht über die einzelnen Heuristiken zu verbessern.

Alle Heuristiken liegen im Paket

```
de.huberlin.informatik.wbi.aladin.intraschema.heuristics
```

und erweitern die Klasse `ForeignKeyHeuristic`. Im Anhang ist ein ausführliches Klassendiagramm angefügt.

4.2.3.1 Minimale Anzahl abhängiger Werte

Klassenname: `ForeignKeyHeuristicMinNumberDependendValues`

Die Berechnung erfolgt durch eine Datenbankabfrage oder direkt über `column_infos`, basierend auf der Konfiguration des Tools:

```
Select count(DISTINCT <abhängiges_attribut>) from <Tabellenname>;
```

oder

```
Select NUM_DISTINCT_COLUMN FROM <column_info_table> where COLUMN_NAME = '<abhängiges_attribut>' AND TABLE_NAME = '<abhängige_tabelle>';
```

4.2.3.2 Abdeckung

Klassenname: `ForeignKeyHeuristicCovering`

Die Berechnung erfolgt durch zwei Datenbankabfragen:

```
countDependend = Select count(DISTINCT <abhängiges_attribut>) from <tabellenname>;
```

```
countReferenced = Select count(DISTINCT <referenziertes_attribut>) from <tabellenname>;
```

Ergebnis: $(\text{countDependend} / \text{countReferenced}) * 100$

13 <http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Logger.html>

4.2.3.3 Verhältnis der begrenzten Wertemengen

Klassenname: ForeignKeyHeuristicProportionalValueAmount

Die Berechnung erfolgt mit Hilfe von zwei Datenbankabfragen, wobei die zweite Abfrage auch mit Unterstützung der Tabelle *column_infos* erfolgen kann.:

```
countReferenced = Select COUNT(DISTINCT <referenziertes_attribut>) FROM
<tabellenname> WHERE <referenziertes_attribut> < (Select
max(abhängiges_attribut) from <tabellenname2>) AND
<referenziertes_attribut> > (Select min(abhängiges_attribut)
from <tabellenname2>);
```

```
countDependend = Select count(DISTINCT <abhängiges_attribut>) from
<Tabellenname>;
```

oder

```
Select NUM_DISTINCT_COLUMN FROM <column_info_table> where
COLUMN_NAME = '<abhängiges_attribut>' AND TABLE_NAME =
'<abhängige_tabelle>';
```

Ergebnis: $\text{abs}(\text{countReferenced} - \text{countDependend}) < (\text{countDependend} / 100 * X)$

Wobei X der konfigurierte Grenzwert ist.

4.2.3.4 Verteilung über referenziertem Attributen

Klassenname: ForeignKeyHeuristicSpreadingOverReference

Die Berechnung erfolgt durch zwei Datenbankabfragen:

```
countDependend = Select count(DISTINCT <abhängiges_attribut>) from
<tabellenname>;
```

```
countReferenced = Select count(DISTINCT <referenziertes_attribut>) from
<tabellenname>;
```

Ergebnis: $(\text{countDependend} / \text{countReferenced}) * 100$

4.2.3.5 Verhältnis der Wertelängen

Klassenname: ForeignKeyHeuristicProportionalValueLength

Die Berechnung erfolgt mit Hilfe von zwei Datenbankabfragen:

```
lengthDependend = Select AVG(LENGTH(<abhängiges_attribut>)) from
<tabellenname>;
```

```
lengthReferenced = Select AVG(LENGTH(<referenziertes_attribut>)) from
<tabellenname>;
```

Ergebnis = $\text{abs}(\text{lengthDependend} - \text{lengthReferenced})$

4.2.3.6 Anzahl der referenzierten Attribute

Klassenname: ForeignKeyHeuristicNumberOfReferences

Die Berechnung erfolgt durch die Iteration aller IND's und den Vergleich von abhängiger Tabelle und abhängigem Attribut. Es erfolgt kein Datenbankzugriff.

4.2.3.7 Attributnamen

Klassenname: ForeignKeyHeuristicColumnName

Der Vergleich erfolgt mit den Java String Methoden `equalsIgnoreCase` und `indexOf`. Es erfolgt kein Datenbankzugriff.

4.2.3.8 Kleine Tabellen

Klassenname: ForeignKeyHeuristicShortTables

Die Berechnung erfolgt über zwei Datenbankabfragen, entweder über die Tabelle `column_infos` oder direkt auf die zu untersuchenden Attribute.

```
countDependent = Select count(DISTINCT <abhängiges_attribut>) from
                  <tabellenname>;
```

```
countReferenced = Select count(DISTINCT <referenziertes_attribut>) from
                  <tabellenname>;
```

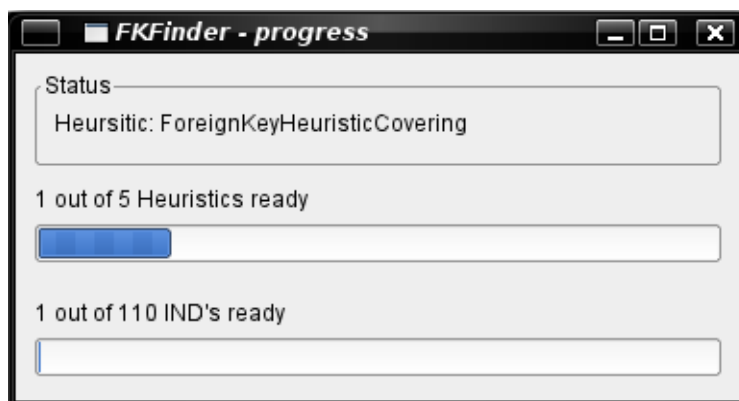
Ergebnis: `countDependent < X || countReferenced < X`

Wobei X der konfigurierte Grenzwert ist.

4.3 Feature der GUI-Variante

Für die erstellte GUI-Variante wurden folgende Funktionen umgesetzt:

- Importieren einer vorhandenen Konfiguration
- Speichern einer Konfiguration als Datei
- Start des Programms mit Anzeige des aktuellen Fortschritts (abgeschlossene Heuristiken und IND's)
- Ausgabe der Ergebnisse eines Suchlaufes und Anzeige der Detailergebnisse für jede einzelne IND
- Speichern eines Ergebnisses
- Laden und Anzeige eines früheren Laufes



Theoretisch können mit Hilfe der GUI mehrere Suchläufe (Beispiel für einen Suchlauf in Abbildung 14) parallel gestartet werden. Dies verlangsamt jedoch jeden einzelnen Durchlauf auf Grund des parallelen Datenbankzugriffs.

4.4 Performance

4.4.1 Statistikdaten

Da es sich bei den zu untersuchenden Datenbanken um sehr große Datenbestände mit vielen Relationen und IND's handelt, ist eine der wichtigsten Anforderungen an jede Heuristik die Zeit, die sie braucht, um alle IND's zu untersuchen.

Ein Hilfsmittel um die IND's möglichst schnell zu bearbeiten, bietet dabei eine von SPIDER angelegte Tabelle: *column_infos*. Diese wird aus den Statistikdaten der Datenbank gewonnen und enthält die folgenden Informationen über die einzelnen Attribute der gefundenen IND's:

- Schema- und Tabellennamen
- Datentypen
- Anzahl der verschiedenen („DISTINCT“) Werte
- Anzahl der NULL Werte

Besonders die Angaben der DISTINCT-Werte helfen viel Rechen- und I/O-Zeit der Datenbank zu sparen, da diese in verschiedenen Heuristiken benutzt werden.

4.4.2 Laufzeiten

Die folgenden zwei Tabellen zeigen die Laufzeit für das Tool, wenn es direkt auf dem Institutsrechner¹⁴ ausgeführt wird, auf dem auch die Datenbankinstanz mit den benutzten Testdatenbanken läuft. Dabei wurden die von SPIDER erzeugten Statistikdaten benutzt. Anwendung fand das Verfahren wie in 3.3.3.1 beschreiben.

Schema	Laufzeit (in Sekunden)	Anzahl untersuchter IND's	Als FK verworfene IND's ¹⁵
SCOP	31	11	6
MOVIELENS	45	19	13
FILMDIENST	228	79	66
BIOSQL_SP	127	36	7
TPCD	373	33	23
IMDB	85	34	19
PDB¹⁶	5565 (~92min)	4972	4444

Tabelle 13: Performance von FKFinder auf verschiedenen Schemata

Exemplarisch die Laufzeiten einzelner Heuristiken auf zwei Schemata:

¹⁴ brunhilde.informatik.hu-berlin.de, 4x Intel Xeon 2.8 Ghz/512KB cache, 12 GB RAM, Oracle 10.2.0

¹⁵ basierend auf der empfohlenen Gewichtung wie in 3.3.3.2 beschreiben. Enthalten sind auch Falsch Negative

¹⁶ PDB wurde lediglich zur Performancemessung benutzt, keine Auswertung der Ergebnisse.

Heuristik	UniProt (110 IND)	TPC-H (86 IND)
Minimale Anzahl abhängiger Werte	74s	65s
Abdeckung	93s	563s
Anzahl referenzierter Werte	<1s	<1s
Attributnamen	<1s	<1s
Verhältnis der Wertlängen	105s	301s
Verhältnis der begrenzten Wertmengen	110s	415s
Verteilung über referenziertem Attribut	99s	220s

Tabelle 14: Laufzeit einzelner Heuristiken

4.4.3 Verbesserung der Laufzeiten

Eine minimale Laufzeit des Tools zu erreichen, war nicht Aufgabe dieser Studienarbeit. Ziel war es lediglich, zufriedenstellende Laufzeiten zu erreichen. Dies ist auch gelungen.

Um die Performance weiter zu verbessern und damit kürzere Laufzeiten zu erreichen, sind folgende zwei Verbesserungen denkbar:

1. Berechnungen einzelner Heuristiken, die auch in anderen Heuristiken benötigt werden, können zwischengespeichert werden und müssen daher nicht mehrfach berechnet werden. Gemeint sind hierbei zeitintensive Datenbankabfragen wie zum Beispiel die Suche nach Werten, die außerhalb einer bestimmten Grenze liegen.
2. Die Kombination von SPIDER und *FKFinder* zu einem Tool. Nahezu alle Heuristiken könnten bereits in SPIDER integriert werden, um jede IND nur einmal untersuchen zu müssen und damit massiv Rechenzeit zu sparen.

Im SPIDER Algorithmus wird jede IND Zeile für Zeile verglichen. Wenn man diesen Schritt ausbauen würde und um die Untersuchungen der einzelnen Heuristiken erweitert, wäre eine weitere Untersuchung kaum mehr nötig. Außerdem wäre es mit dieser Methode möglich die Abstände zwischen den Werte im referenzierten Attribut zu messen die auch im abhängigen Attribut vorliegen. Dadurch kann eine genaue Verteilung der Abdeckung berechnet werden um eine sehr hohe *Precision* zu erreichen, da Fremdschlüssel häufig sehr gleichmäßig verteilt sind. (wie in Abschnitt 2.2 beschrieben)

3. Eventuell ist eine Ergebnisverbesserung durch die Benutzung von Histogrammen der einzelnen Attribute einer IND möglich, da diese einfacher mit einander verglichen werden können.

5 Fazit

Ziel dieser Studienarbeit war es, aus gegebenen Inklusionsabhängigkeiten einer Datenbank jene heraus zu filtern, die vermutlich eine Fremdschlüsselbeziehung zwischen zwei Tabellen darstellen. Dies ist, wie anhand der Testergebnisse gezeigt wurde, gelungen. Von den 82 Fremdschlüsseln in den Testdatenbanken wurden 75 eindeutig gefunden. Dies ergibt eine *Precision* von 91,5 %. Der *Recall* auf den untersuchten IND's liegt bei 91,4%. Eine höhere *Precision* zu erreichen erscheint auf Grund der durchgeführten Tests sehr schwierig.

Einfach zu identifizierende Fremdschlüssel sind solche bei denen keine fortlaufenden Zahlenwerte benutzt werden und die Anzahl der verschiedenen Werte sehr hoch ist. Schwierigkeiten, besonders beim Ausschluss als Fremdschlüssel, bereiten solche IND's bei denen fortlaufende Zahlen benutzt werden und deren Kardinalität sehr groß ist. In solchen Fällen kann nur noch der Benutzer einen Fremdschlüssel ausschließen. Darüber hinaus bereiten nur einige Sonderfälle Probleme. Zum Beispiel Fremdschlüssel, bei denen nur einer von vier vorhandenen Werten im abhängigen Attribut vorliegt.

Eventuell ließen sich einige Grenzfälle noch mit weiteren Heuristiken untersuchen. Zum Beispiel wäre denkbar bei fortlaufenden Zahlen zu prüfen an welcher Position in der Tabelle das abhängige Attribut vorkommt. Hintergrund dieses Tests ist die Idee, dass Fremdschlüssel vermutlich nicht an erster Stelle einer Tabelle stehen. Außerdem könnte man in solchen Fällen die Heuristik zur Untersuchung der Namensähnlichkeit anders gewichten. Dies wären jedoch nur Hinweise die das Programm geben kann, die endgültige Entscheidung kann nur der Mensch vor dem Rechner treffen.

Verbesserungsmöglichkeiten sind am ehesten durch eine Kombination des erstellten Programms *FKFinder* und dem aus dem Projekt ALADIN hervorgegangenen Programm SPIDER zu erreichen. Vermutlich lassen sich dadurch Geschwindigkeiten erreichen, die nur leicht über der eigentlichen Ausführungszeit von SPIDER liegen.

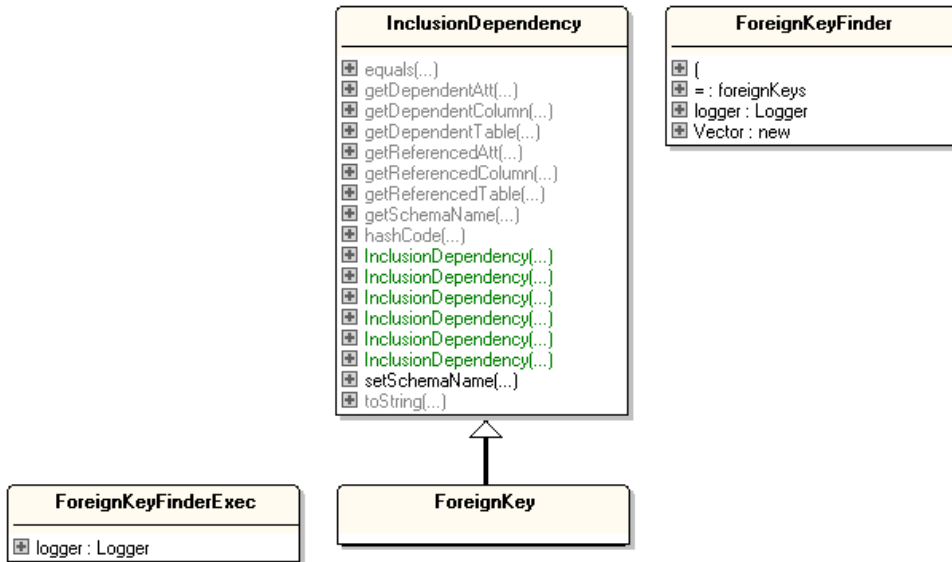
Literaturverzeichnis

- [1] Jana Bauckmann, Automatically Integrating Life Science Data Sources, VLDB '07, September 23-28, 2007, Vienna, Austria., 2007
- [2] J. Bauckmann, U. Leser, F. Naumann, V. Tietz, Efficiently Detecting Inclusion Dependencies, 18. Workshop über Grundlagen von Datenbanken (GI-Workshop), Wittenberg, Juni, 2006

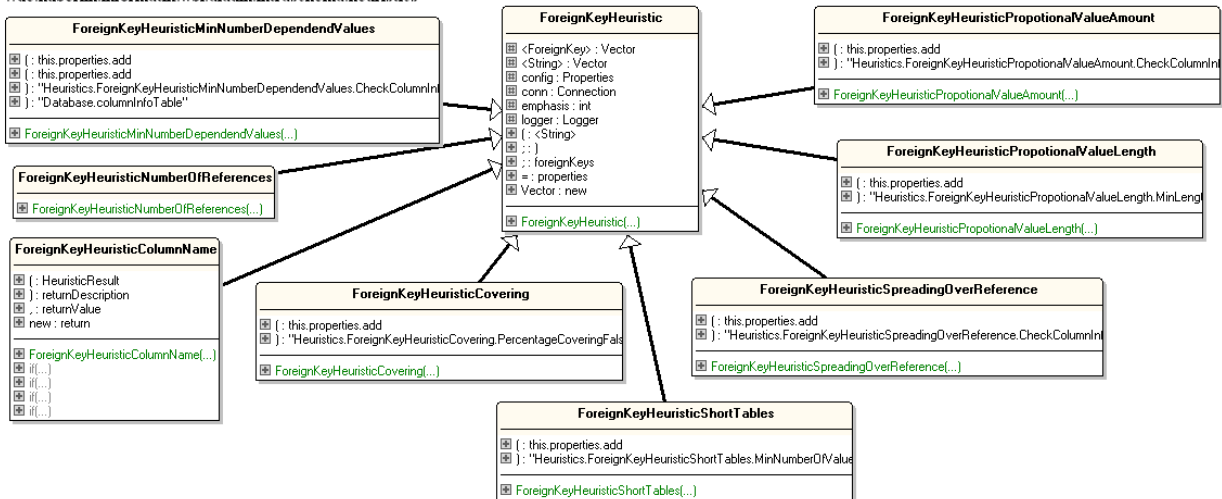
6 Anhang

6.1 Klassendiagramm

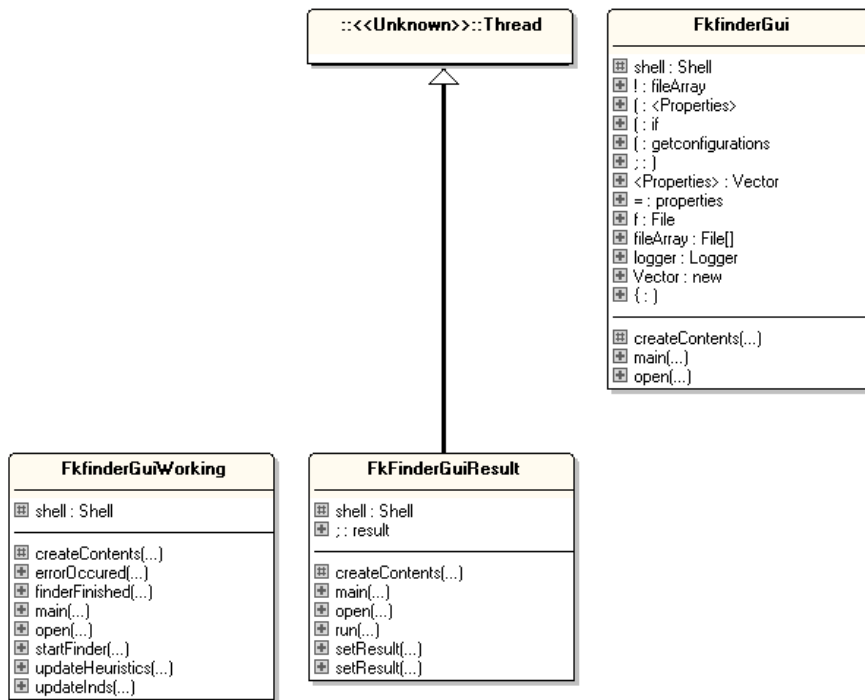
::de.huberlin.informatik.wbi.aladin.intraschema



::de.huberlin.informatik.wbi.aladin.intraschema.heuristics



::de.huberlin.informatik.wbi.aladin.intrascema.fkfinderGui



6.2 Ergebnisse für einzelne Schemata im Detail

6.2.1 Filmdienst

DEPENDEND	REFERENCED	RESULT	PERCENT	FK?
PERSONENNAMEN.PERSONNR	PERSONEN.PERSONNR	130/130	100,00%	y
FILMTEXTE.FILMNR	FILME.FILMNR	130/130	100,00%	y
CREDITS.FILMNR	FILME.FILMNR	130/130	100,00%	y
FILMTITEL.FILMNR	FILME.FILMNR	130/130	100,00%	y
FILME.FILMSERIENR	FILMSERIEN.FILMSERIENR	130/130	100,00%	y
CREDITS.FUNKTIONNR	FUNKTIONEN.FUNKTIONNR	130/130	100,00%	y
FILME.FILMBEWERTUNGNR	FILMBEWERTUNGEN.FILMBEWERTUNGNR	130/130	100,00%	y
FILMEUNDGENRES.GENRENR	GENRES.GENRENR	130/130	100,00%	y
FILMTITEL.FILMTITELTYPNR	FILMTITELTYPEN.FILMTITELTYPNR	130/130	100,00%	y
FILMEUNDGENRES.FILMNR	FILME.FILMNR	80/80	100,00%	y
CREDITS.PERSONENNAMENR	PERSONENNAMEN.PERSONENNAMENR	80/80	100,00%	y
CREDITS.PERSONNR	PERSONEN.PERSONNR	80/80	100,00%	y
FILMTEXTYPEN.FILMTEXTYPNR	FILMBEWERTUNGEN.FILMBEWERTUNGNR	50/50	100,00%	y
AUSZEICHNUNGEN.FILMNR	FILME.FILMNR	30/80	37,50%	y
AUSZEICHNUNGEN.PERSONNR	PERSONEN.PERSONNR	30/130	23,08%	y
FILMTEXTE.FILMTEXTYPNR	FILMTEXTYPEN.FILMTEXTYPNR	30/180	16,67%	y
FILME.FILMBEWERTUNGNR	CREDITS.CREDITNR	0/100	0,00%	
FILME.FILMBEWERTUNGNR	GENRES.GENRENR	0/50	0,00%	
FILME.FILMBEWERTUNGNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FILME.FILMBEWERTUNGNR	FUNKTIONEN.FUNKTIONNR	0/50	0,00%	
FILME.FILMBEWERTUNGNR	FILMSERIEN.FILMSERIENR	0/100	0,00%	
FUNKTIONEN.FUNKTIONNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FUNKTIONEN.FUNKTIONNR	CREDITS.CREDITNR	0/100	0,00%	
FUNKTIONEN.FUNKTIONNR	GENRES.GENRENR	0/50	0,00%	
FUNKTIONEN.FUNKTIONNR	FILMSERIEN.FILMSERIENR	0/50	0,00%	
FILME.GEANDERTVON	CREDITS.CREDITNR	0/100	0,00%	
FILME.GEANDERTVON	GENRES.GENRENR	0/50	0,00%	
FILME.GEANDERTVON	FILMTITEL.FILMTITELNR	0/100	0,00%	
FILME.GEANDERTVON	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FILME.GEANDERTVON	FILMSERIEN.FILMSERIENR	0/50	0,00%	
GENRES.GENRENR	CREDITS.CREDITNR	0/100	0,00%	
GENRES.GENRENR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/50	0,00%	
GENRES.GENRENR	FILMSERIEN.FILMSERIENR	0/50	0,00%	
FILMEUNDGENRES.GENRENR	CREDITS.CREDITNR	0/100	0,00%	
FILMEUNDGENRES.GENRENR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/50	0,00%	
FILMEUNDGENRES.GENRENR	FILMSERIEN.FILMSERIENR	0/50	0,00%	
FILME.ANGELEGTVON	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FILME.ANGELEGTVON	CREDITS.CREDITNR	0/100	0,00%	
FILME.ANGELEGTVON	GENRES.GENRENR	0/50	0,00%	
FILME.ANGELEGTVON	FILMTITEL.FILMTITELNR	0/100	0,00%	
FILME.ANGELEGTVON	FILMSERIEN.FILMSERIENR	0/50	0,00%	
CREDITS.FUNKTIONNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
CREDITS.FUNKTIONNR	GENRES.GENRENR	0/50	0,00%	
CREDITS.FUNKTIONNR	FILMSERIEN.FILMSERIENR	0/50	0,00%	
FILMSERIEN.FILMSERIENR	CREDITS.CREDITNR	0/100	0,00%	
FILME.FILMSERIENR	CREDITS.CREDITNR	0/100	0,00%	
FILMTEXTYPEN.FILMTEXTYPNR	GENRES.GENRENR	0/50	0,00%	
FILMTEXTYPEN.FILMTEXTYPNR	FILMSERIEN.FILMSERIENR	0/100	0,00%	
FILMTEXTE.FILMTEXTYPNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/200	0,00%	
FILMTEXTE.FILMTEXTYPNR	FILMTITELTYPEN.FILMTITELTYPNR	0/150	0,00%	
FILMTEXTE.FILMTEXTYPNR	FUNKTIONEN.FUNKTIONNR	0/150	0,00%	
FILMTEXTE.FILMTEXTYPNR	FILMBEWERTUNGEN.FILMBEWERTUNGNR	0/150	0,00%	
FILMTEXTE.FILMTEXTYPNR	CREDITS.CREDITNR	0/200	0,00%	
FILMTEXTE.FILMTEXTYPNR	GENRES.GENRENR	0/150	0,00%	
FILMTEXTE.FILMTEXTYPNR	FILMSERIEN.FILMSERIENR	0/200	0,00%	
AUSZEICHNUNGEN.JAHR	CREDITS.CREDITNR	0/100	0,00%	
AUSZEICHNUNGEN.JAHR	FILMTITEL.FILMTITELNR	0/100	0,00%	
FILMTITEL.FILMTITELTYPNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FILMTITEL.FILMTITELTYPNR	FUNKTIONEN.FUNKTIONNR	0/50	0,00%	
FILMTITEL.FILMTITELTYPNR	FILMBEWERTUNGEN.FILMBEWERTUNGNR	0/50	0,00%	
FILMTITEL.FILMTITELTYPNR	CREDITS.CREDITNR	0/100	0,00%	
FILMTITEL.FILMTITELTYPNR	GENRES.GENRENR	0/50	0,00%	
FILMTITEL.FILMTITELTYPNR	FILMSERIEN.FILMSERIENR	0/100	0,00%	
FILMTITEL.FILMTITELTYPNR	FILMTEXTE.FILMTEXTNR	0/100	0,00%	
FILMTITELTYPEN.FILMTITELTYPNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FILMTITELTYPEN.FILMTITELTYPNR	FUNKTIONEN.FUNKTIONNR	0/50	0,00%	
FILMTITELTYPEN.FILMTITELTYPNR	FILMBEWERTUNGEN.FILMBEWERTUNGNR	0/50	0,00%	
FILMTITELTYPEN.FILMTITELTYPNR	CREDITS.CREDITNR	0/100	0,00%	
FILMTITELTYPEN.FILMTITELTYPNR	GENRES.GENRENR	0/50	0,00%	
FILMTITELTYPEN.FILMTITELTYPNR	FILMSERIEN.FILMSERIENR	0/100	0,00%	
FILMTITELTYPEN.FILMTITELTYPNR	FILMTEXTE.FILMTEXTNR	0/100	0,00%	
FILMBEWERTUNGEN.FILMBEWERTUNGNR	CREDITS.CREDITNR	0/100	0,00%	
FILMBEWERTUNGEN.FILMBEWERTUNGNR	GENRES.GENRENR	0/50	0,00%	
FILMBEWERTUNGEN.FILMBEWERTUNGNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FILMBEWERTUNGEN.FILMBEWERTUNGNR	FUNKTIONEN.FUNKTIONNR	0/50	0,00%	
FILMBEWERTUNGEN.FILMBEWERTUNGNR	FILMSERIEN.FILMSERIENR	0/100	0,00%	
FILMTEXTYPEN.FILMTEXTYPNR	AUSZEICHNUNGEN.AUSZEICHNUNGNR	0/100	0,00%	
FILMTEXTYPEN.FILMTEXTYPNR	FUNKTIONEN.FUNKTIONNR	0/50	0,00%	
FILMTEXTYPEN.FILMTEXTYPNR	CREDITS.CREDITNR	0/100	0,00%	

6.2.2 MovieLens

SCHEMA	DEPENDEND	REFERENCED	RESULT	PERCENT	FK?
MOVIELENS	MOVIE_GENRES.GENRE_ID	GENRES.ID	130/130	100,00%	y
MOVIELENS	RATINGS.USER_ID	USERS.ID	130/130	100,00%	y
MOVIELENS	MOVIE_GENRES.MOVIE_ID	MOVIES.ID	130/130	100,00%	y
MOVIELENS	RATINGS.MOVIE_ID	MOVIES.ID	130/130	100,00%	y
MOVIELENS	USERS.AGE	AGES.ID	100/100	100,00%	y
MOVIELENS	USERS.OCCUPATION	OCCUPATIONS.ID	100/100	100,00%	y
MOVIELENS	RATINGS.MOVIE_ID	USERS.ID	30/80	37,50%	
MOVIELENS	MOVIE_GENRES.MOVIE_ID	USERS.ID	30/80	37,50%	
MOVIELENS	GENRES.ID	OCCUPATIONS.ID	30/80	37,50%	
MOVIELENS	MOVIES.ID	USERS.ID	30/80	37,50%	
MOVIELENS	MOVIE_GENRES.GENRE_ID	OCCUPATIONS.ID	30/80	37,50%	
MOVIELENS	AGES.ID	MOVIES.ID	30/130	23,08%	
MOVIELENS	AGES.ID	USERS.ID	30/130	23,08%	
MOVIELENS	RATINGS.RATING	MOVIES.ID	0/100	0,00%	
MOVIELENS	RATINGS.RATING	GENRES.ID	0/50	0,00%	
MOVIELENS	RATINGS.RATING	OCCUPATIONS.ID	0/50	0,00%	
MOVIELENS	MOVIES.YEAR	USERS.ID	0/50	0,00%	
MOVIELENS	RATINGS.RATING	USERS.ID	0/100	0,00%	
MOVIELENS	USERS.AGE	MOVIES.ID	0/100	0,00%	

6.2.3 SCOP

DEPENDEND	REFERENCED	RESULT	PERCENT	FK?
DESCRIPTION.SCOP_ID	CLASSIFICATION.SCOP_ID	130/130	100,00%	y
HIERARCHIE.CHILDREN	DESCRIPTION.SUNID	100/100	100,00%	y
CLASSIFICATION.SUNID	DESCRIPTION.SUNID	80/80	100,00%	y
COMMENTS.SUNID	DESCRIPTION.SUNID	80/80	100,00%	y
CLASSIFICATION.PX	DESCRIPTION.SUNID	50/50	100,00%	y
CLASSIFICATION.DM	DESCRIPTION.SUNID	0/50	0,00%	
CLASSIFICATION.FA	DESCRIPTION.SUNID	0/50	0,00%	
CLASSIFICATION.SF	DESCRIPTION.SUNID	0/50	0,00%	
CLASSIFICATION.CF	DESCRIPTION.SUNID	0/50	0,00%	
CLASSIFICATION.CL	DESCRIPTION.SUNID	0/100	0,00%	
CLASSIFICATION.SP	DESCRIPTION.SUNID	0/50	0,00%	

6.2.4 UNIPROT

DEPENDEND	REFERENCED	RESULT	PERCENT	FK?
SG_BIOSEQUENCE.ENT_OID	SG_COMMENT.ENT_OID	130/130	100,00%	y
SG_COMMENT.ENT_OID	SG_BIOENTRY.OID	130/130	100,00%	y
SG_BIOENTRY_REF_ASSOC.ENT_OID	SG_BIOSEQUENCE.ENT_OID	130/130	100,00%	y
SG_BIOENTRY_DBXREF_ASSOC.ENT_OID	SG_BIOENTRY.OID	130/130	100,00%	y
SG_BIOENTRY_DBXREF_ASSOC.ENT_OID	SG_COMMENT.ENT_OID	130/130	100,00%	y
SG_BIOENTRY.OID	SG_COMMENT.ENT_OID	130/130	100,00%	y
SG_BIOENTRY.OID	SG_BIOSEQUENCE.ENT_OID	130/130	100,00%	y
SG_LOCATION.FEA_OID	SG_SEQFEATURE.OID	130/130	100,00%	y
SG_BIOENTRY_REF_ASSOC.ENT_OID	SG_COMMENT.ENT_OID	130/130	100,00%	y
SG_COMMENT.ENT_OID	SG_BIOSEQUENCE.ENT_OID	130/130	100,00%	y
SG_BIOENTRY_QUALIFIER_ASSOC.ENT_OID	SG_COMMENT.ENT_OID	130/130	100,00%	y
SG_BIOENTRY_QUALIFIER_ASSOC.ENT_OID	SG_BIOSEQUENCE.ENT_OID	130/130	100,00%	y
SG_TAXON_NAME.TAX_OID	SG_TAXON.OID	130/130	100,00%	y
SG_TERM.ONT_OID	SG_ONTOLOGY.OID	130/130	100,00%	y
SG_BIOENTRY_REF_ASSOC.ENT_OID	SG_BIOENTRY.OID	130/130	100,00%	y
SG_BIOENTRY_DBXREF_ASSOC.ENT_OID	SG_BIOSEQUENCE.ENT_OID	130/130	100,00%	y
SG_BIOENTRY_QUALIFIER_ASSOC.ENT_OID	SG_BIOENTRY.OID	130/130	100,00%	y
SG_BIOSEQUENCE.ENT_OID	SG_BIOENTRY.OID	130/130	100,00%	y
SG_SEQFEATURE.OID	SG_LOCATION.FEA_OID	130/130	100,00%	y
SG_BIOENTRY_REF_ASSOC.REF_OID	SG_REFERENCE.OID	130/130	100,00%	y
SG_SEQFEATURE.ENT_OID	SG_COMMENT.ENT_OID	80/80	100,00%	y
SG_SEQFEATURE.TYPE_TRM_OID	SG_TERM.OID	80/80	100,00%	y
SG_BIOENTRY_DBXREF_ASSOC.DBX_OID	SG_DBXREF.OID	80/80	100,00%	y
SG_BIOENTRY.TAX_OID	SG_TAXON.OID	80/80	100,00%	y
SG_SEQFEATURE.ENT_OID	SG_BIOENTRY.OID	80/80	100,00%	y
SG_REFERENCE.DBX_OID	SG_DBXREF.OID	80/80	100,00%	y
SG_SEQFEATURE.ENT_OID	SG_BIOSEQUENCE.ENT_OID	80/80	100,00%	y
SG_SEQFEATURE_QUALIFIER_ASSOC.FEA_OID	SG_SEQFEATURE.OID	80/80	100,00%	y
SG_SEQFEATURE_QUALIFIER_ASSOC.FEA_OID	SG_LOCATION.FEA_OID	80/80	100,00%	y
SG_BIOENTRY.DB_OID	SG_BIODATABASE.OID	130/230	56,52%	y
SG_BIOENTRY_QUALIFIER_ASSOC.TRM_OID	SG_TERM.OID	30/80	37,50%	y
SG_SEQFEATURE_QUALIFIER_ASSOC.TRM_OID	SG_TERM.OID	30/180	16,67%	?
SG_SEQFEATURE.SOURCE_TRM_OID	SG_TERM.OID	30/180	16,67%	?
SG_SEQFEATURE_QUALIFIER_ASSOC.RANK	SG_TAXON.LEFT_VALUE	0/200	0,00%	
SG_COMMENT.RANK	SG_TAXON.LEFT_VALUE	0/200	0,00%	
SG_LOCATION.RANK	SG_TAXON.LEFT_VALUE	0/200	0,00%	

6.2.5 TPC-H

SCHEMA	DEPENDEND	REFERENCED	RESULT	PERCENT	FK?
TPCD	CUSTOMER.C_NATIONKEY	NATION.N_NATIONKEY	130/130	100,00%	y
TPCD	NATION.N_REGIONKEY	REGION.R_REGIONKEY	130/130	100,00%	y
TPCD	SUPPLIER.S_NATIONKEY	NATION.N_NATIONKEY	130/130	100,00%	y
TPCD	PARTSUPP.PS_SUPPKEY	SUPPLIER.S_SUPPKEY	130/130	100,00%	y
TPCD	LINEITEM.L_PARTKEY	PART.P_PARTKEY	130/130	100,00%	y
TPCD	PARTSUPP.PS_PARTKEY	PART.P_PARTKEY	130/130	100,00%	y
TPCD	LINEITEM.L_SUPPKEY	SUPPLIER.S_SUPPKEY	100/100	100,00%	y
TPCD	PARTSUPP.PS_AVAILQTY	SUPPLIER.S_SUPPKEY	100/100	100,00%	
TPCD	LINEITEM.L_ORDERKEY	ORDERS.O_ORDERKEY	100/100	100,00%	y
TPCD	ORDERS.O_CUSTKEY	CUSTOMER.C_CUSTKEY	50/50	100,00%	y
TPCD	LINEITEM.L_QUANTITY	CUSTOMER.C_CUSTKEY	0/100	0,00%	
TPCD	LINEITEM.L_QUANTITY	PART.P_PARTKEY	0/100	0,00%	
TPCD	LINEITEM.L_QUANTITY	SUPPLIER.S_SUPPKEY	0/100	0,00%	
TPCD	PARTSUPP.PS_AVAILQTY	CUSTOMER.C_CUSTKEY	0/50	0,00%	
TPCD	PARTSUPP.PS_AVAILQTY	PART.P_PARTKEY	0/50	0,00%	
TPCD	SUPPLIER.S_SUPPKEY	CUSTOMER.C_CUSTKEY	0/50	0,00%	
TPCD	SUPPLIER.S_SUPPKEY	PART.P_PARTKEY	0/50	0,00%	
TPCD	PARTSUPP.PS_SUPPKEY	CUSTOMER.C_CUSTKEY	0/50	0,00%	
TPCD	PARTSUPP.PS_SUPPKEY	PART.P_PARTKEY	0/50	0,00%	
TPCD	LINEITEM.L_SUPPKEY	CUSTOMER.C_CUSTKEY	0/50	0,00%	
TPCD	LINEITEM.L_SUPPKEY	PART.P_PARTKEY	0/50	0,00%	
TPCD	ORDERS.O_CUSTKEY	PART.P_PARTKEY	0/50	0,00%	
TPCD	PART.P_SIZE	SUPPLIER.S_SUPPKEY	0/100	0,00%	
TPCD	PART.P_SIZE	CUSTOMER.C_CUSTKEY	0/100	0,00%	
TPCD	LINEITEM.L_LINENUMBER	SUPPLIER.S_SUPPKEY	0/100	0,00%	
TPCD	LINEITEM.L_LINENUMBER	ORDERS.O_ORDERKEY	0/100	0,00%	
TPCD	LINEITEM.L_LINENUMBER	PART.P_PARTKEY	0/100	0,00%	
TPCD	LINEITEM.L_LINENUMBER	CUSTOMER.C_CUSTKEY	0/100	0,00%	
TPCD	LINEITEM.L_LINENUMBER	NATION.N_NATIONKEY	0/50	0,00%	
TPCD	REGION.R_REGIONKEY	NATION.N_NATIONKEY	0/50	0,00%	
TPCD	CUSTOMER.C_CUSTKEY	PART.P_PARTKEY	0/50	0,00%	
TPCD	ORDERS.O_SHIPPRIORITY	NATION.N_NATIONKEY	0/150	0,00%	
TPCD	ORDERS.O_SHIPPRIORITY	REGION.R_REGIONKEY	0/150	0,00%	

6.2.6 IMDB

SCHEMA	DEPENDEND	REFERENCED	RESULT	PERCENT	FK?
IMDB_NO_FK	RELEASE-DATES.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	COUNTRIES.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	KEYWORDS.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	CERTIFICATES.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	PRODUCERS.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	PROD_COM.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	DISTRIBUTORS.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	QUOTES.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	MOVIE_LINKS.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	MOVIE_LINKS.ID	DIRECTORS.ID	80/80	100,00%	
IMDB_NO_FK	ALTERNATE_VERSIONS.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	ACTOR.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	AKA_TITLES.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	ACTRESS.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	GENRES.MOVIE_ID	MOVIE.MID	80/80	100,00%	y
IMDB_NO_FK	RELEASE-DATES.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	LANGUAGE.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	EDITORS.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	DISTRIBUTORS.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	QUOTES.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	QUOTES.ID	EDITORS.ID	30/80	37,50%	
IMDB_NO_FK	PRODUCERS.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	PROD_COM.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	PRODUCTION_DESIGNERS.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	PRODUCTION_DESIGNERS.ID	EDITORS.ID	30/80	37,50%	
IMDB_NO_FK	ALTERNATE_VERSIONS.ID	EDITORS.ID	30/80	37,50%	
IMDB_NO_FK	ALTERNATE_VERSIONS.ID	PRODUCTION_DESIGNERS.ID	30/80	37,50%	
IMDB_NO_FK	ALTERNATE_VERSIONS.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	CERTIFICATES.ID	DIRECTORS.ID	30/80	37,50%	
IMDB_NO_FK	ACTRESS.ORDER	PRODUCTION_DESIGNERS.ID	0/100	0,00%	
IMDB_NO_FK	ACTRESS.ORDER	EDITORS.ID	0/100	0,00%	
IMDB_NO_FK	ACTRESS.ORDER	DIRECTORS.ID	0/100	0,00%	
IMDB_NO_FK	ACTOR.ORDER	DIRECTORS.ID	0/100	0,00%	
IMDB_NO_FK	ACTOR.ORDER	EDITORS.ID	0/100	0,00%	