Schritt-für-Schritt: Konvertieren einer Verbindung

Folgende Schritte sind für die Definition einer Konvertierung zu erledigen:

1. Die beteiligten Klassen der Konvertierung auswählen.
2. Die Breite der beiden Klassentabellen für die Konvertierung festlegen.
3. (optional) einen Filter definieren.
4. Die Konvertierung möglichst genau beschreiben.
5. Konvertierungsregeln für fromJoin (obj1) und toJoin (obj2) definieren.
6. Die Konvertierung durchführen und das Ergebnis kontrollieren.

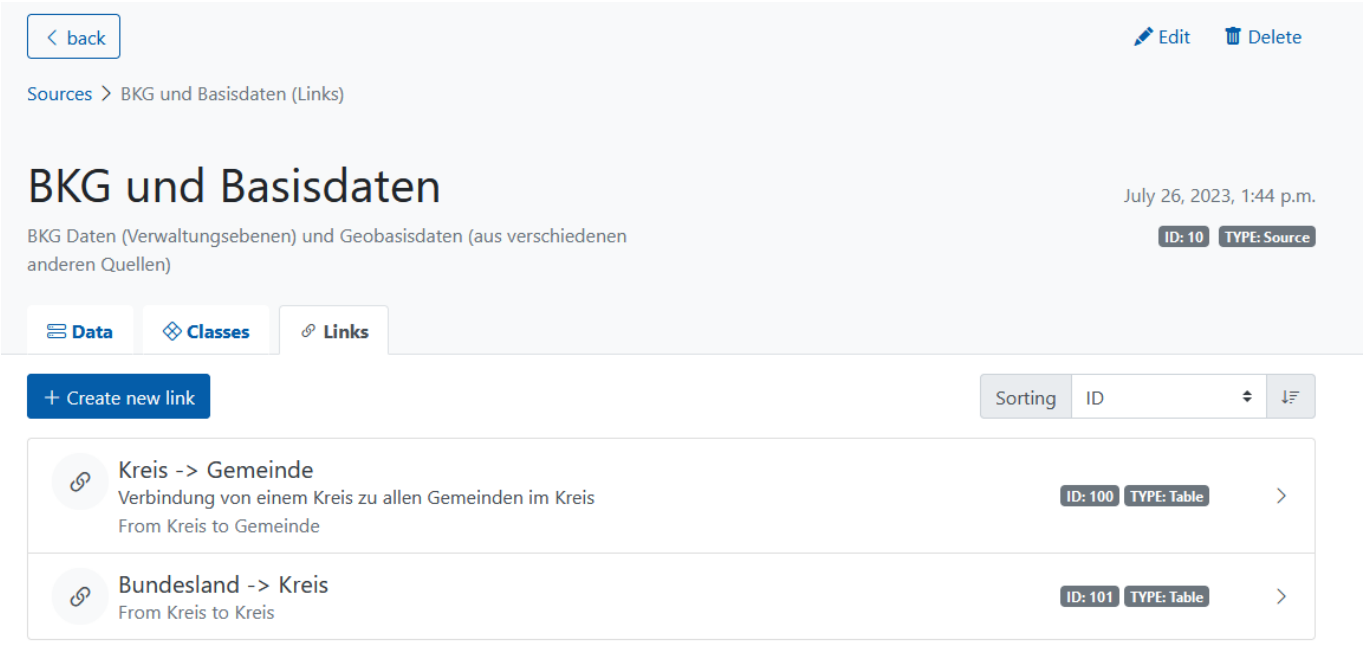
Diese Schritte werden im Folgenden anhand des Beispiels durchgegangen. Am Ende des Kapitels findet sich ein Schema des Ablaufs der Konvertierung.

Voraussetzungen

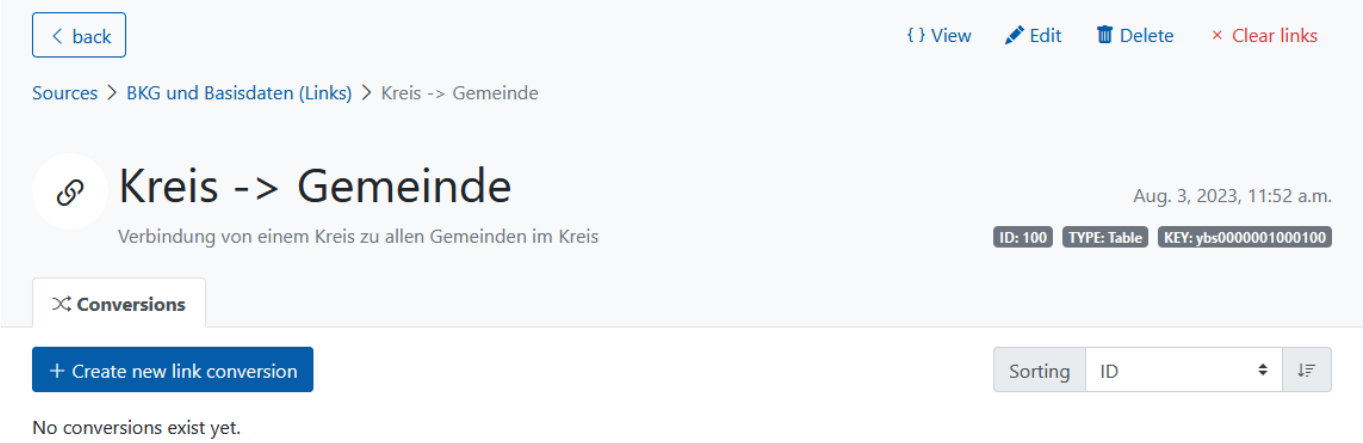
- Es muss sichergestellt sein, dass die Klassen, deren Objekte Sie verbinden wollen, bereits gefüllt sind. Beim Konvertieren einer Verbindung können nur bereits vorhandene Objekte der (2-N) Klassen, zwischen denen die Verbindung angelegt wurde, verbunden werden. Im vorliegenden Beispiel - der Verbindung zwischen den Klassen Gemeinden und Kreise - ist das bereits passiert, wie im Tutorial zur Klassenkonvertierung für die Klasse Gemeinden beschrieben.
- Verbindungskonvertierungen können als unabhängige Konvertierungen definiert werden (wie im folgenden Tutorial) oder mit einer Objektkonvertierung verbunden werden.

Start

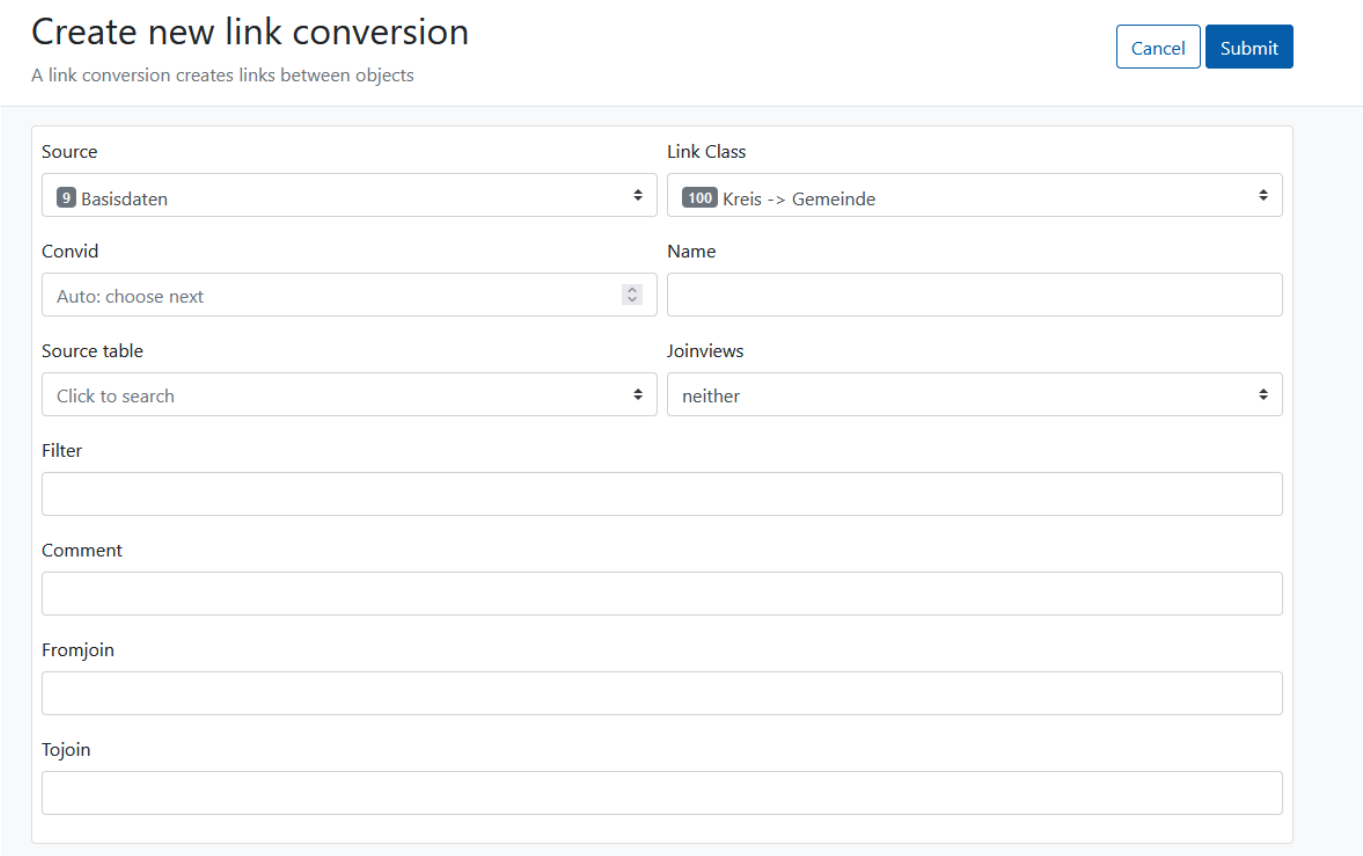
Um die Konvertierung einer Verbindung neu anzulegen, muss die Verbindung anvisiert werden. Sie ist über die Liste aller Verbindungen zu erreichen.



Im Menü der Verbindung werden alle bereits vorhandenen Konvertierungen für diese Verbindung angezeigt. Wie die Abbildung zeigt, ist noch keine Konvertierung definiert. Mittels des blauen Knopfes "Create new link conversion" öffnet sich ein Formular, in dem eine neue Konvertierung definiert werden kann.



Es öffnet sich ein Formular für eine neue Verbindungskonvertierung. Es enthält eine ganze Reihe von Felder, die in den folgenden Schritten nach und nach ausgefüllt werden.



1. Die beteiligten Klassen auswählen

Die Konvertierung einer Verbindung enthält vier Tabellen, die berücksichtigt werden müssen. Zwei davon werden ausgewählt:

- Die Verbindung selbst: Sie ist das Ziel der Konvertierung. Die Verbindungstabelle soll mit Einträgen gefüllt werden, die jeweils etwas darüber aussagen, welcher Kreis mit welcher Gemeinde verbunden ist.

- Die Quelltabelle: Sie ist die Datenlieferantin der Konvertierung. Spalten dieser Tabelle können innerhalb der gesamten Definition der Konvertierung mittels "q.[spaltenname]" referenziert werden (vgl. z.B. den Filter).

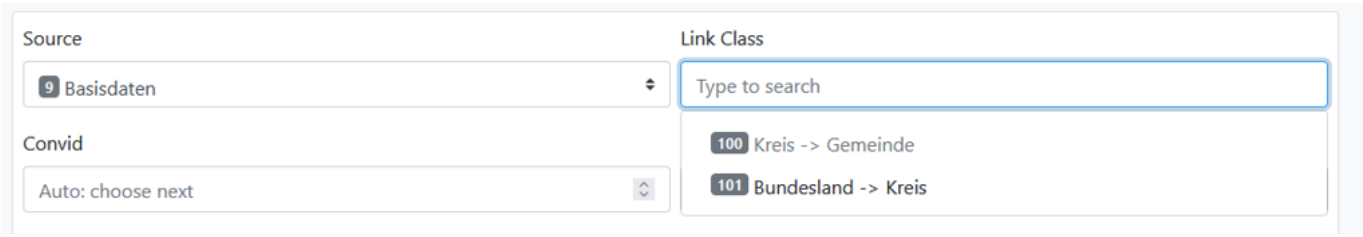
Neben diesen beiden Tabellen, die es auch in Objektkonvertierungen gibt, existieren zwei weitere Tabellen, die berücksichtigt werden müssen, nämlich die beiden Klassen, zwischen deren Objekten nun Verbindungen hergestellt werden müssen:

- Objekt 1 / die FROM-Klasse / linke Seite: Spalten dieser Tabelle können innerhalb der gesamten Definition der Konvertierung mittels "obj1.[spaltenname]" referenziert werden (vgl. z.B. fromJoin).
- Objekt 2 / die TO-Klasse / rechte Seite: Spalten dieser Tabelle können innerhalb der gesamten Definition der Konvertierung mittels "ob21.[spaltenname]" referenziert werden (vgl. z.B. toJoin).

Diese beiden Klassen müssen mit der Quelltabelle zusammengebracht werden, um Verbindungsinformationen zu generieren. Das erfolgt mittels der Felder "fromJoin" und "toJoin" . Die Klassen werden nicht separat ausgewählt, sondern ergeben sich aus der Wahl der Verbindungstabelle.

1.1. Die Zielverbindung der Konvertierung auswählen

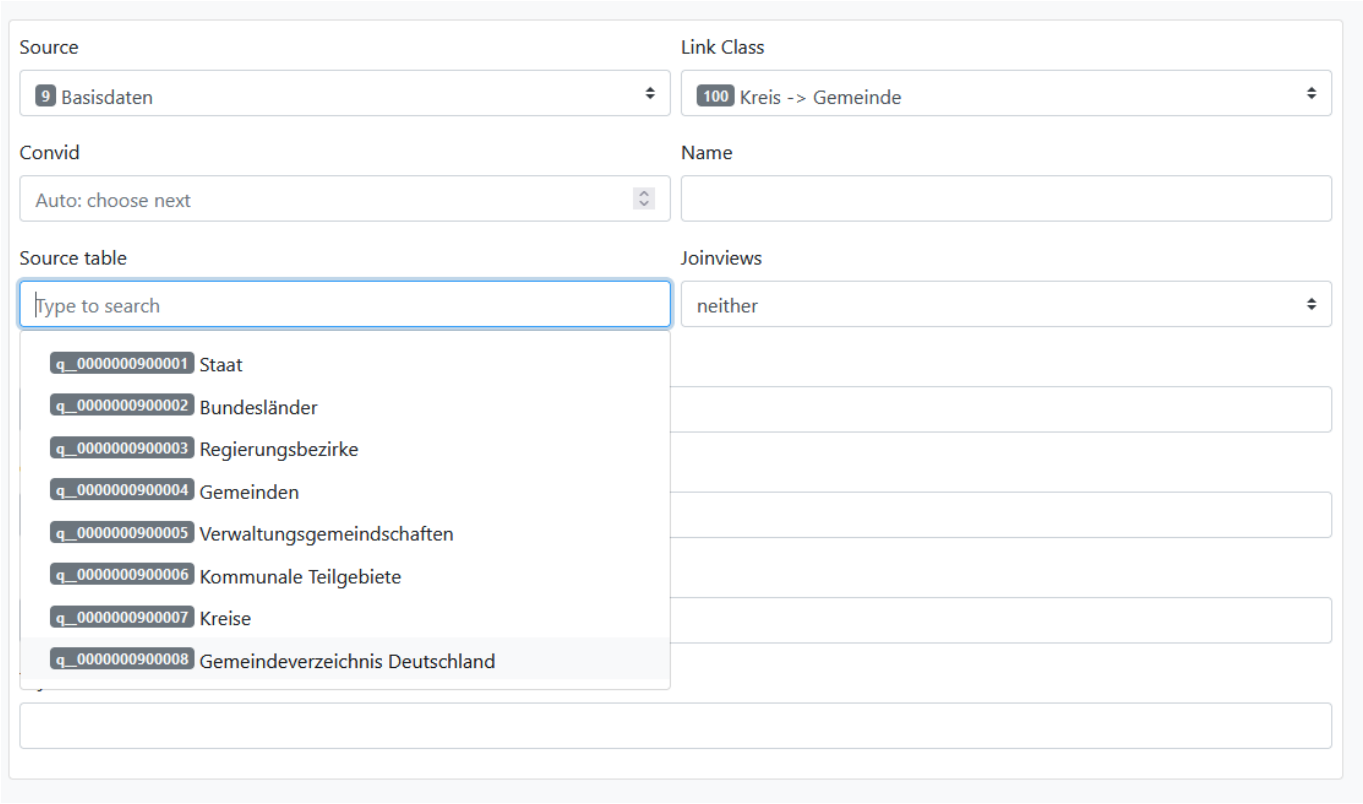
Zunächst muss in "**Link Class**" das Ziel der Konvertierung, d.h. die Verbindungsklasse eingetragen werden. Da die Navigation über die Klasse "Kreis -> Gemeinde" erfolgte, ist diese vorausgewählt. Dieser Zustand wird beibehalten.



1.2. (optional) eine Quelltabelle (q), deren Daten für die Konvertierung benutzt werden sollen, auswählen

Als nächstes wird eine Quelltabelle für die Konvertierung ausgewählt. Dies geschieht durch ein Dropdown-Menü im Feld "**Source Table**". Im Dropdown sind sämtliche in Simplex4Data verfügbaren Quelltabellen aufgelistet. Für dieses Beispiel wird die schon im Tutorial zur Objektkonvertierung benutzte und als Beispieldatensatz vorgestellte Quelltabelle q_00000000900008 ausgewählt. Aus ihr wurden bereits die Objekte beider Klassen konvertiert.

Verbindungskonvertierungen können es - im Gegensatz zu anderen Konvertierungen - auch ohne Angabe einer Quelltabelle definiert werden. Siehe dazu den Exkurs unten.



Mit den ersten beiden Schritten sind somit folgende vier Tabellen als Bestandteile dieser Konvertierung festgelegt:

- die Quelltabelle,
- das Ziel der Konvertierung, die Verbindungstabelle,
- die Tabelle der from-Klasse,
- die Tabelle der to-Klasse.

Obj 1 (FROM-Klasse) Kreis

o	ndx	key	typ	Attribut A	Attribut B

Obj 2 (TO-Klasse) Gemeinde

o	ndx	key	typ	Attribut A	Attribut B

Verbindungstabelle Kreis → Gemeinde
yb_0000000900100

o1	o2

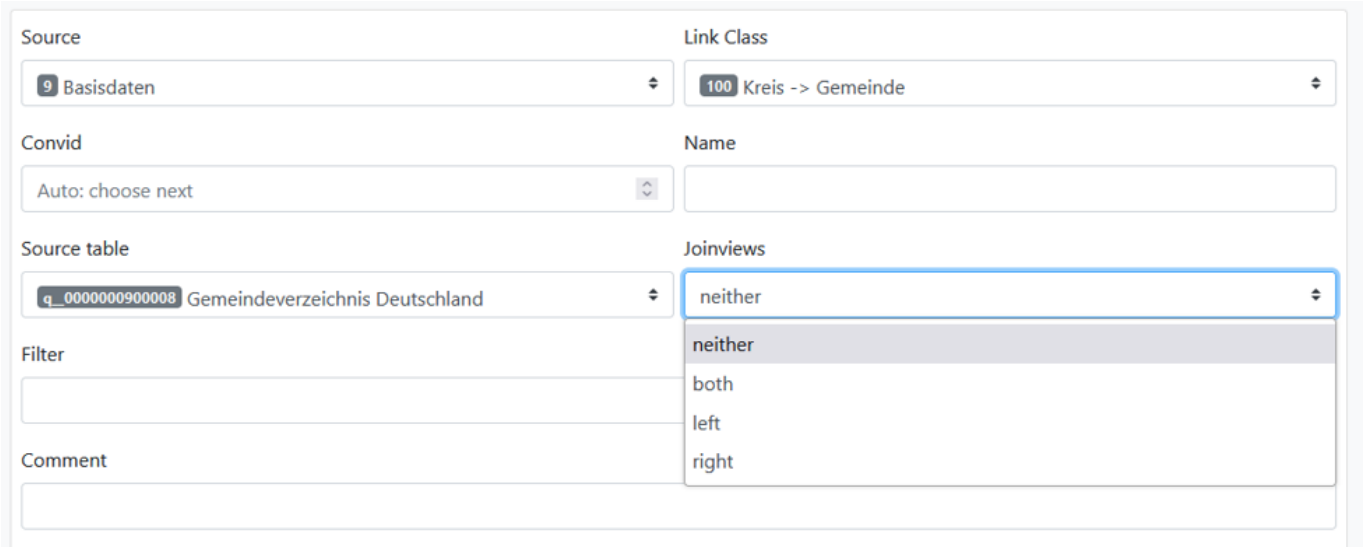
Quelltabelle Q_00000000900008

satz-art	text-kenn-zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...

2. Die Breite der beiden Klassentabellen für die Konvertierung festlegen

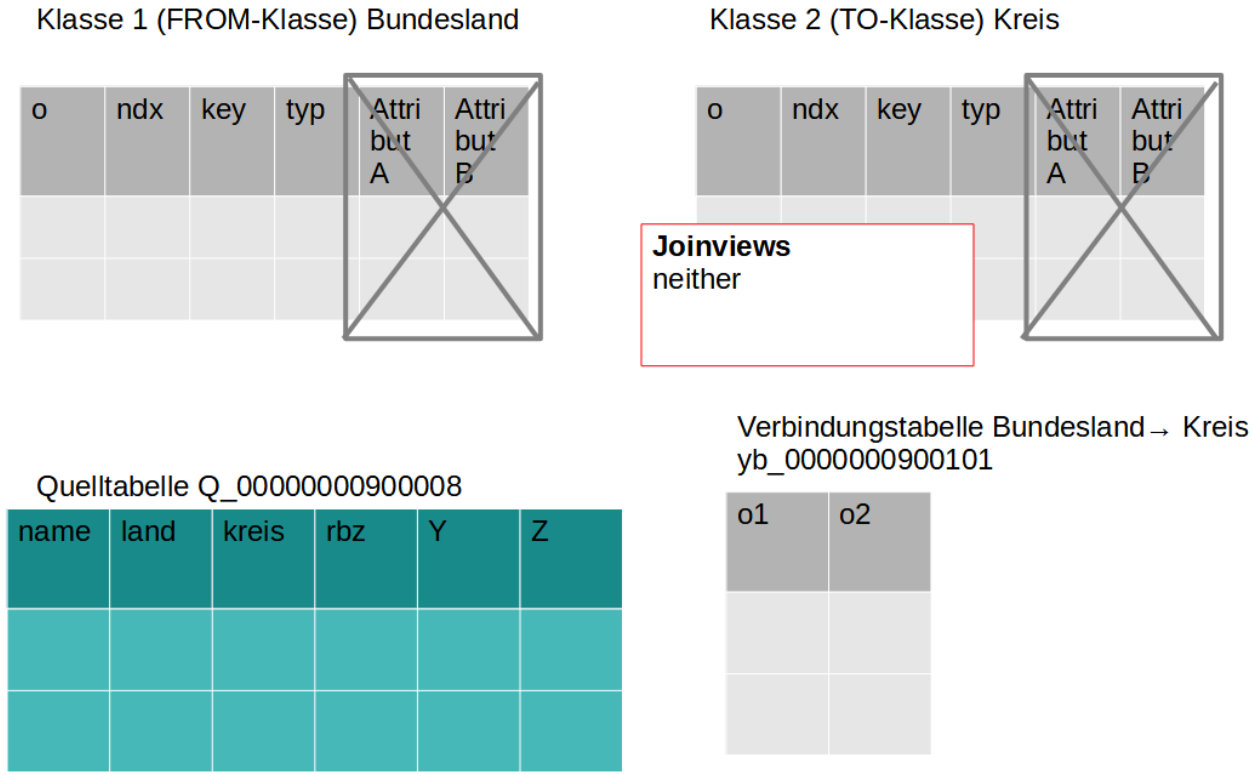
Mittels des Feldes "**Joinviews**" wird festgelegt, ob für die Konvertierung die Klassen mit ihren Standardfeldern ausreichen, oder ob ihre Sachattribute hinzugezogen werden sollen. Es existieren vier

Möglichkeiten.



joinviews neither: keine Sachattribute zur Konvertierung heranziehen

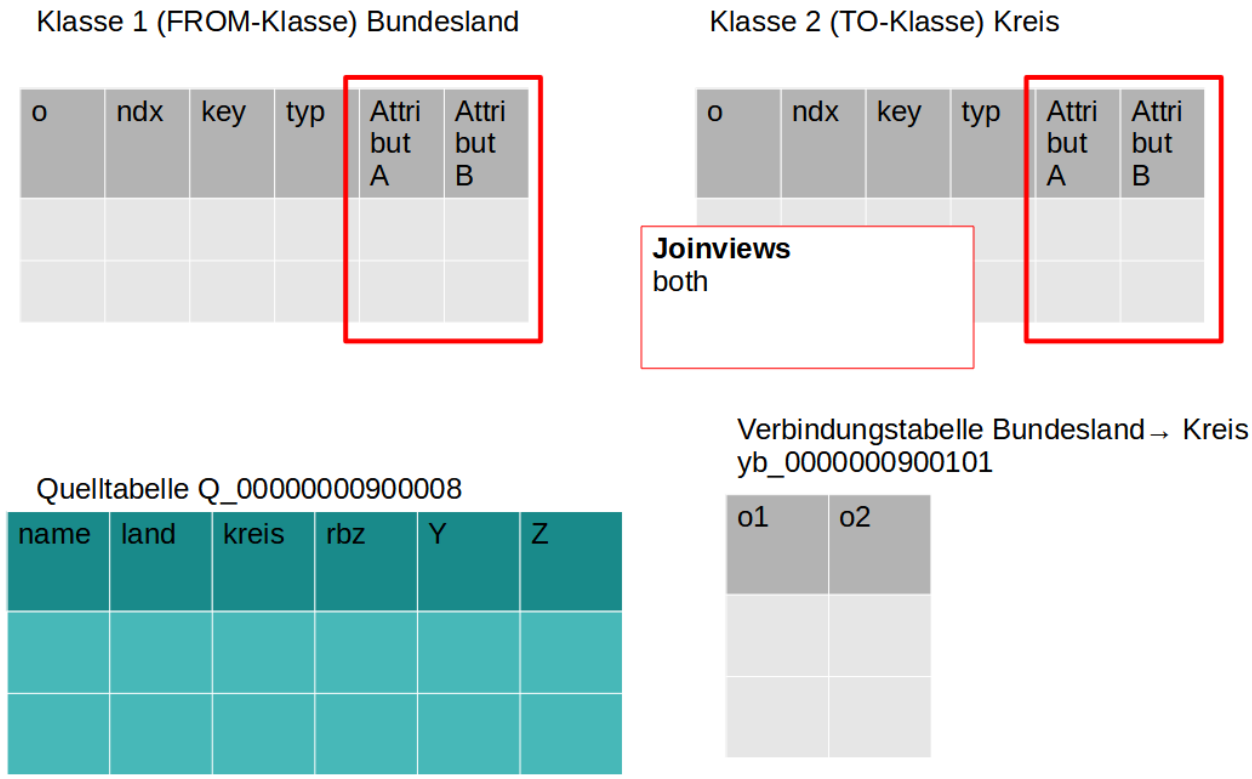
Im vorliegenden Beispiel wird die Konvertierung ausschließlich an dem ARS orientieren, welcher in beiden Klassen im Standardfeld ndx angelegt ist. Somit werden Sachattribute nicht benötigt. Die Auswahl dieser Option kann die Ausführung der Konvertierung beschleunigen, sodass sie, wenn möglich, gewählt werden sollte.



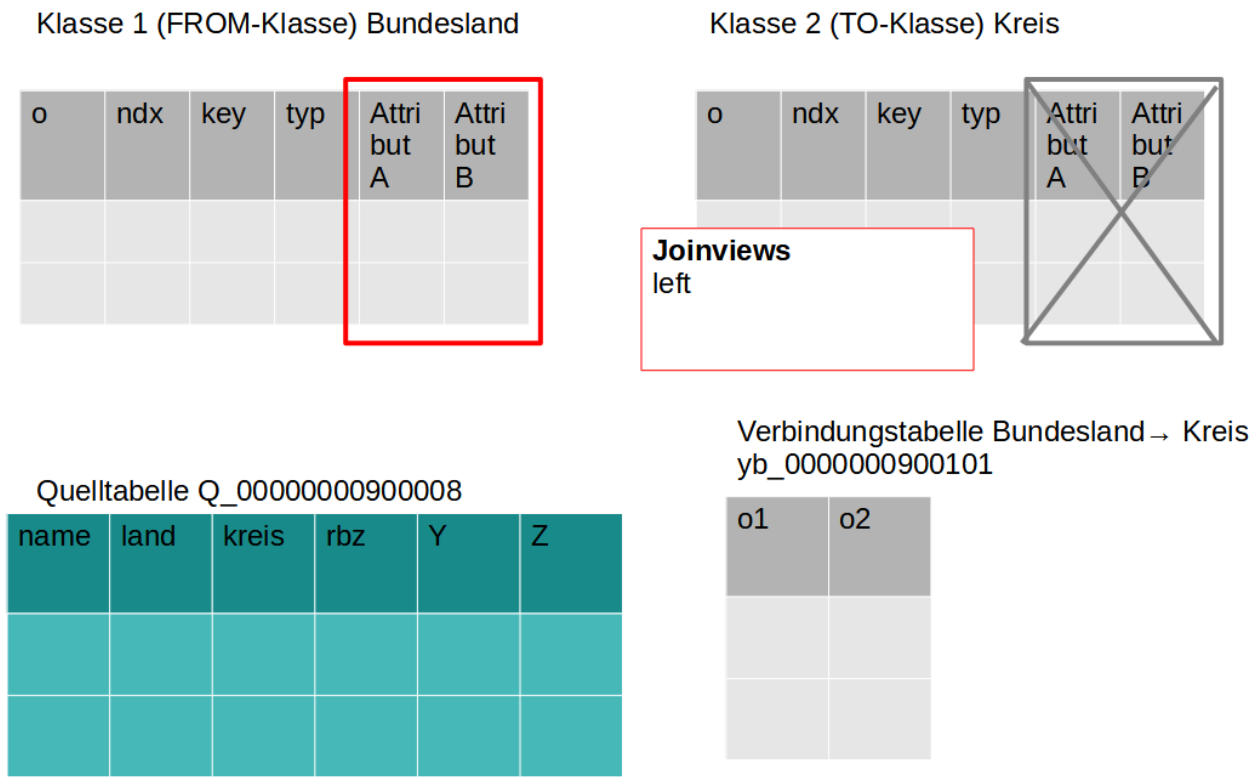
joinviews both: Sachattribute beider Klassen zur Konvertierung heranziehen

Das Gegenstück ist die Option "both": Hier werden die Sachattribute beider Klassen für die Konvertierung

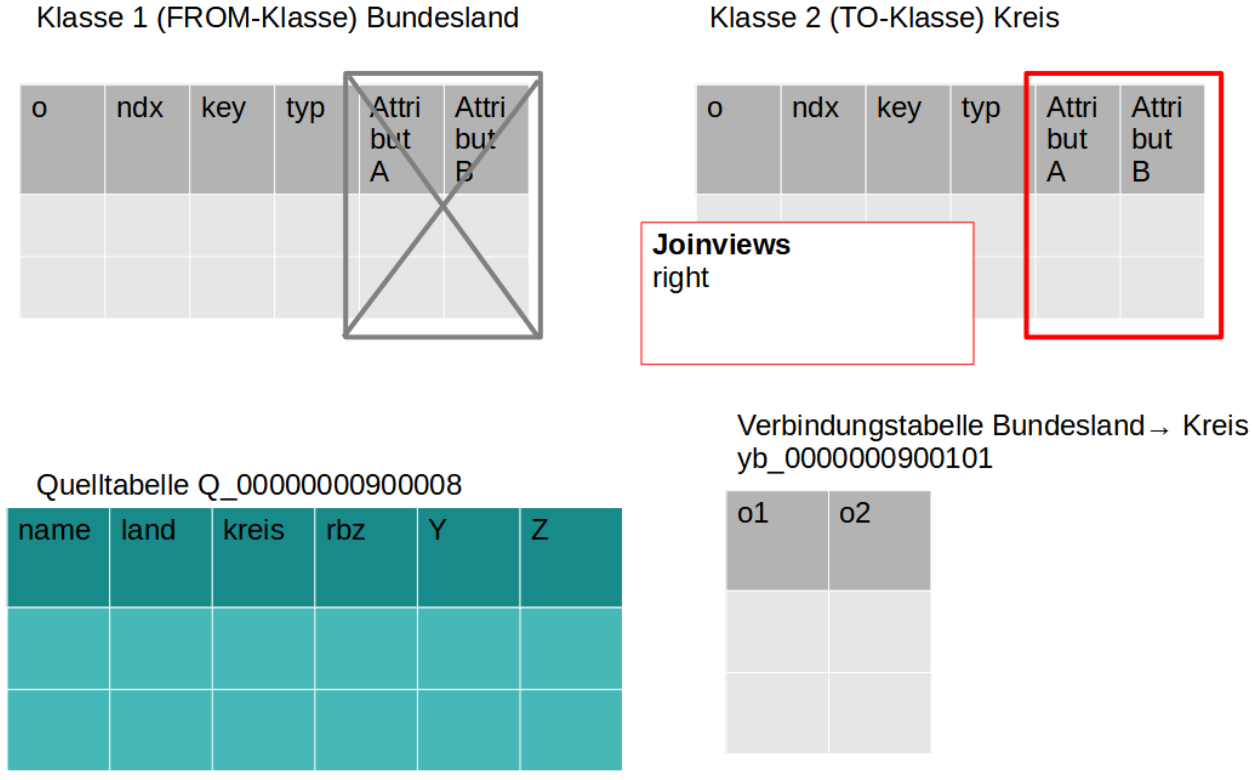
herangezogen. Diese Option sorgt für die langsamste Ausführung der Konvertierung, doch dies ist nur bei größeren Datenmengen oder komplexen Rechenoperationen relevant.



joinviews left: nur Sachattribute der From-Klasse zur Konvertierung heranziehen



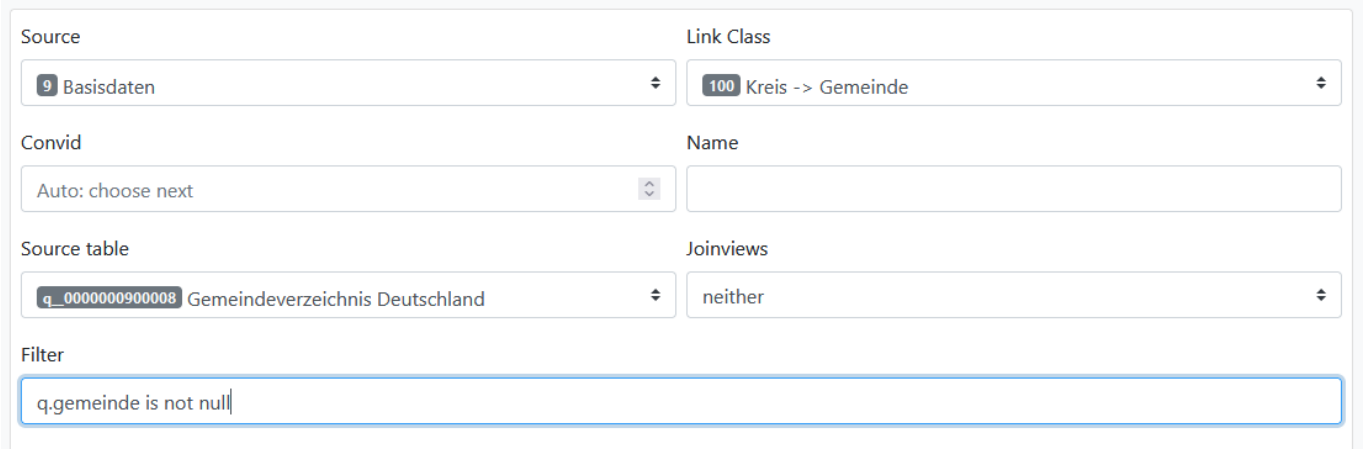
joinviews right: nur Sachattribute der To-Klasse zur Konvertierung heranziehen



3. (optional) einen Filter definieren

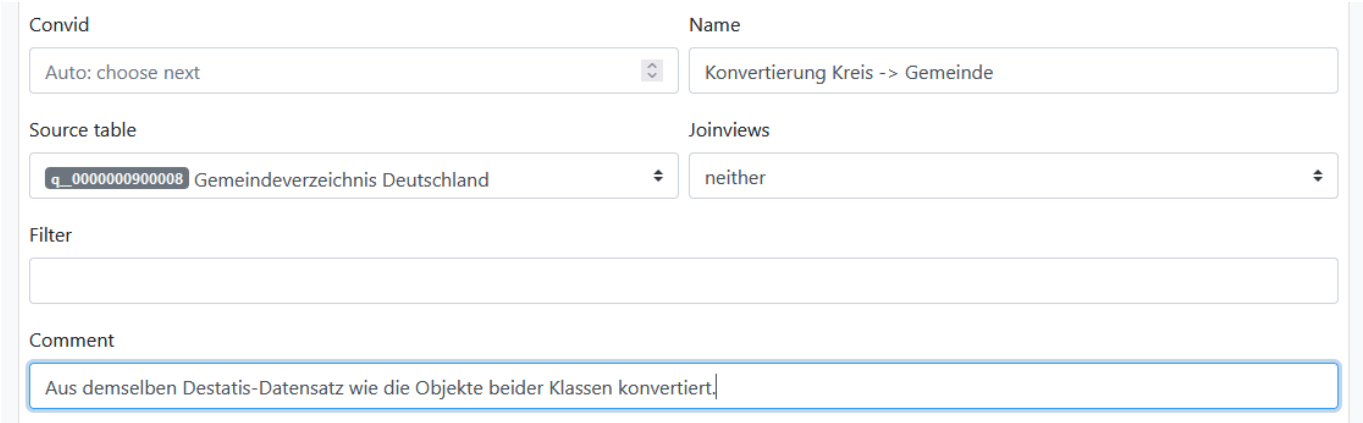
Bei der folgenden Konvertierung werden die beteiligten Tabellen (die Quelltabelle, die From-Klasse und die To-Klasse) eingelesen und - gemäß der Konvertierungsregeln für fromJoin und toJoin (vgl. 6) - miteinander gematcht. Mittels des Feldes "**Filter**" können Regeln festgelegt werden, um bestimmte Zeilen in diesen drei Tabellen zu überspringen. Das geschieht mittels Angaben mit "q", "obj1" und "obj2" im Filter.

Hier wird der in der Objektkonvertierung für Gemeinden verwendete Filter für die Quelltabelle übernommen. Das bedeutet, dass exakt dieselben Zeilen der Quelltabelle, die zur Erzeugung der Gemeindeobjekte verwendet wurden, auch für die Erzeugung der Verbindungstabelle verwendet werden. Filter auf die From- und die To-Klasse sind nicht nötig, da es keine Objekte in diesen Klassen (d.h. Gemeinden oder Kreise) gibt, die nicht verbunden werden sollen.



4. Die Konvertierung möglichst genau beschreiben

Es stehen die Felder "**Name**" und "**Comment**" zur Verfügung, um die Konvertierung zu beschreiben. Um langfristig den Überblick zu behalten, lohnt sich eine aussagekräftige Beschreibung. Diese beiden Felder sind u.a. in der Übersicht aller Konvertierungen zu einer Verbindung (vgl. oben, 0.) zu sehen. Gerade wenn mehrere Konvertierungen definiert werden (was aus verschiedenen Gründen sinnvoll ist und häufig passiert), sind diese beiden Felder wertvoll, um die Konvertierungen auseinander halten zu können.



5. Konvertierungsregeln für fromJoin (obj1) und ToJoin (obj2) definieren

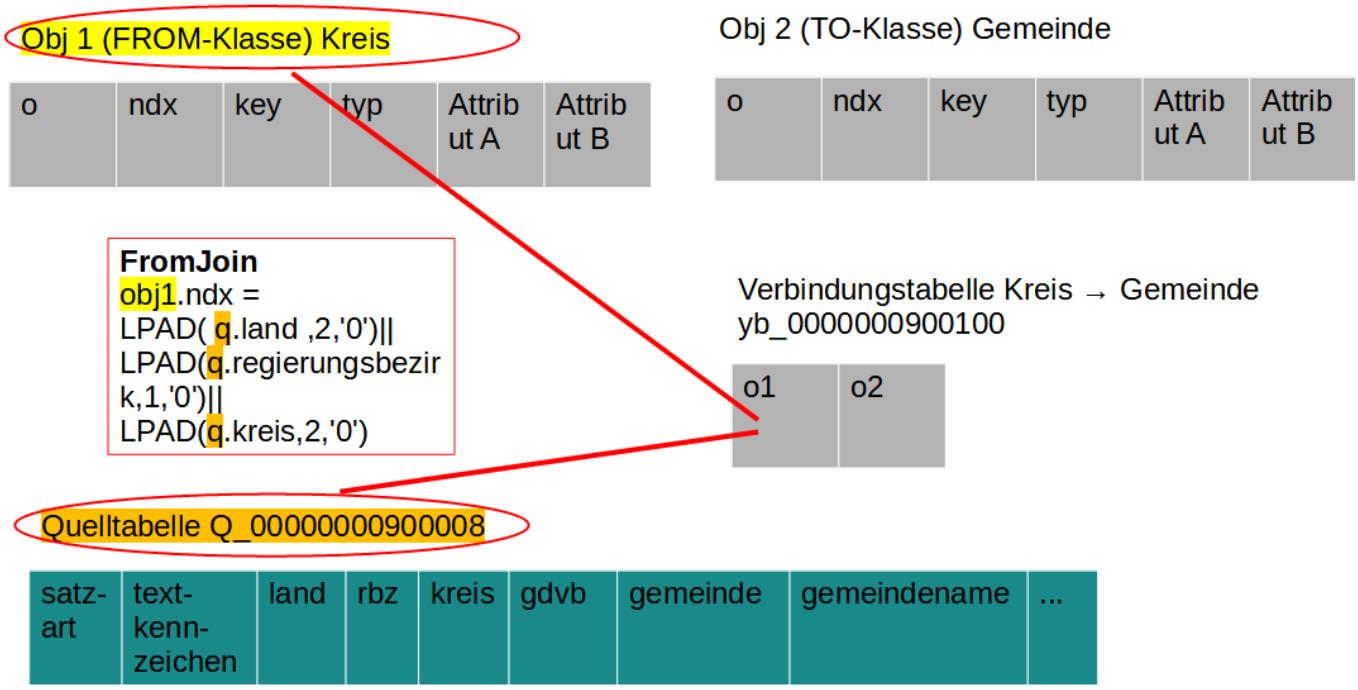
Bei einer Objektkonvertierung war der wichtigste Schritt die Befüllung des Feldes "ndx" der Objekttabelle, um eindeutig identifizierbare und aktualisierbare Objekte zu erzeugen. Bei der Konvertierung einer Verbindung geht es dagegen darum, Regeln zu finden, welche Objekte aus beiden

Seiten der Verbindung (obj1 und obj2) verbunden werden sollen.

Dazu stehen zwei Felder zur Verfügung: "**fromJoin**" und "**toJoin**". Nur wenn die Regeln beider Felder erfüllt sind, wird eine Verbindung eingetragen.

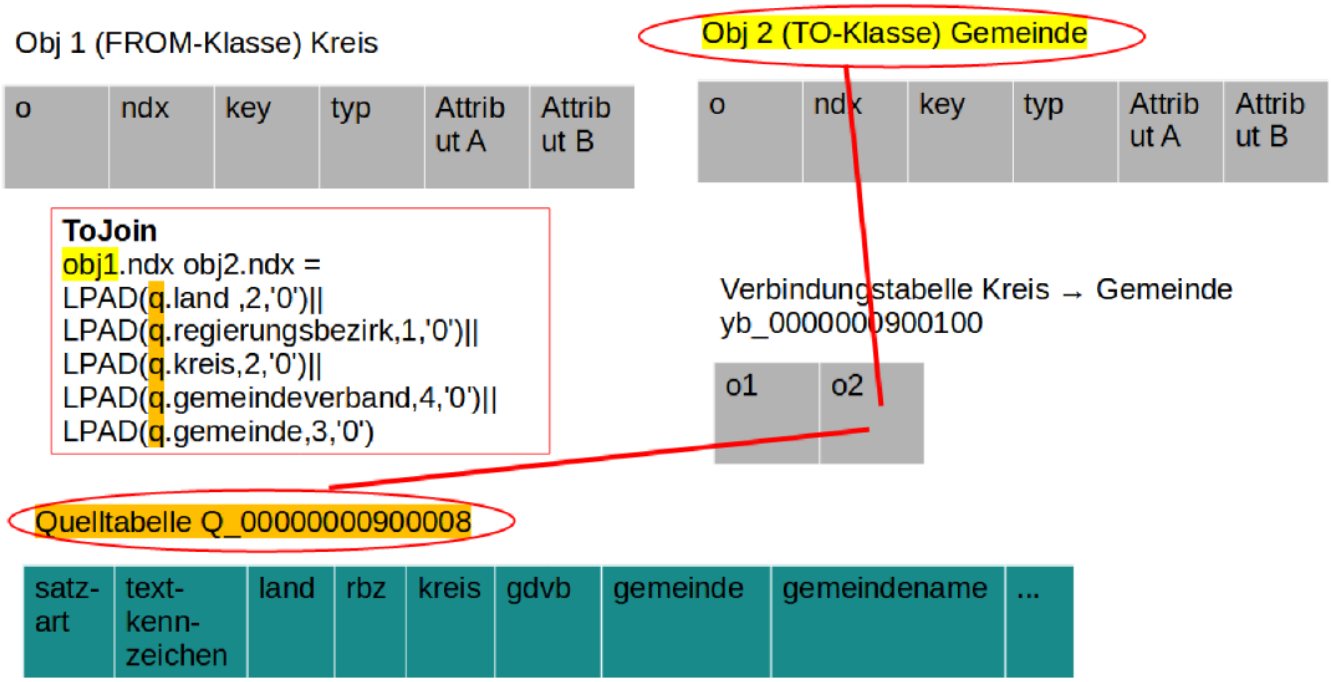
5.1. fromJoin

Für jede Zeile in der Quelltabelle, die den Filter passiert, wird mittels der in "fromJoin" definierten Regel versucht, ein Objekt in der From-Klasse zu identifizieren.

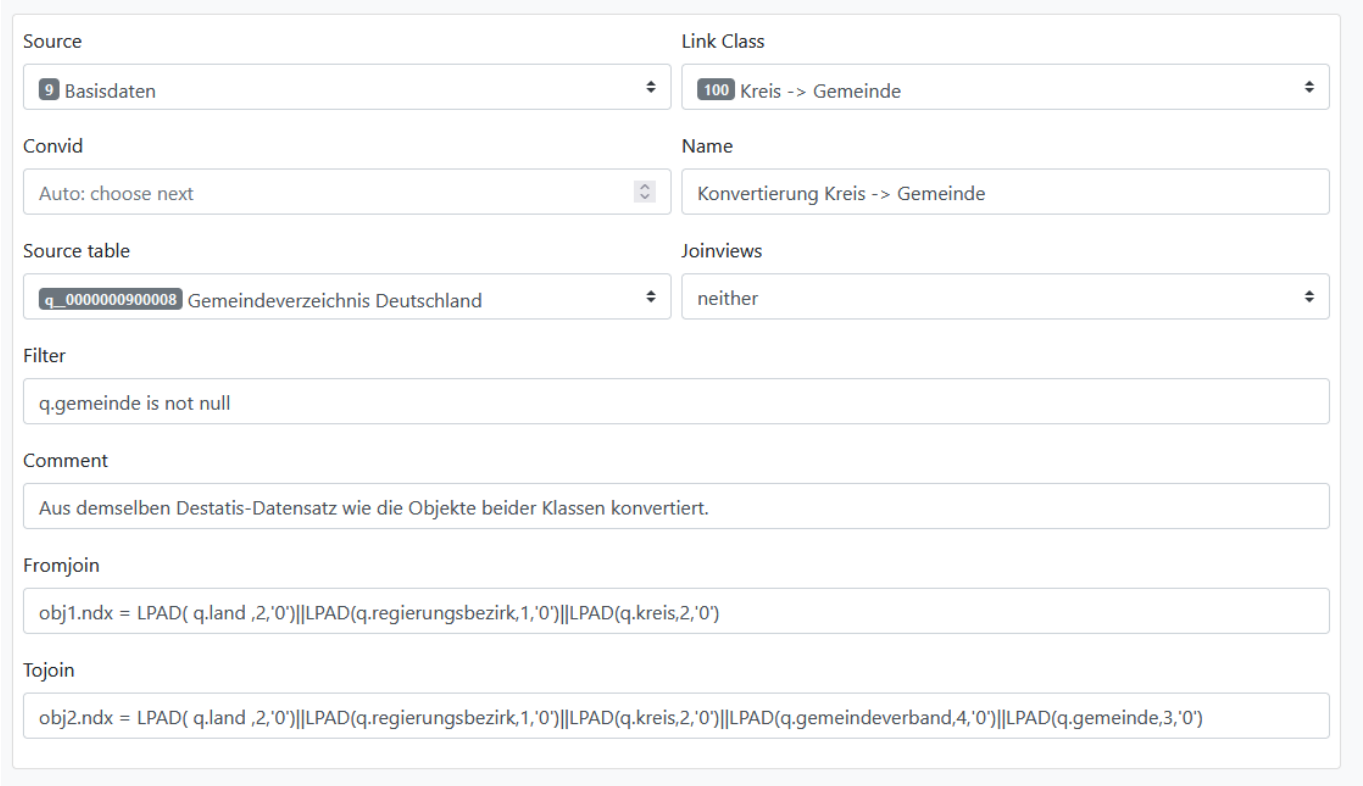


5.2. toJoin

Für jede Zeile in der Quelltabelle, die den Filter passiert, wird mittels der in "toJoin" definierten Regel versucht, ein Objekt in der To-Klasse zu identifizieren.



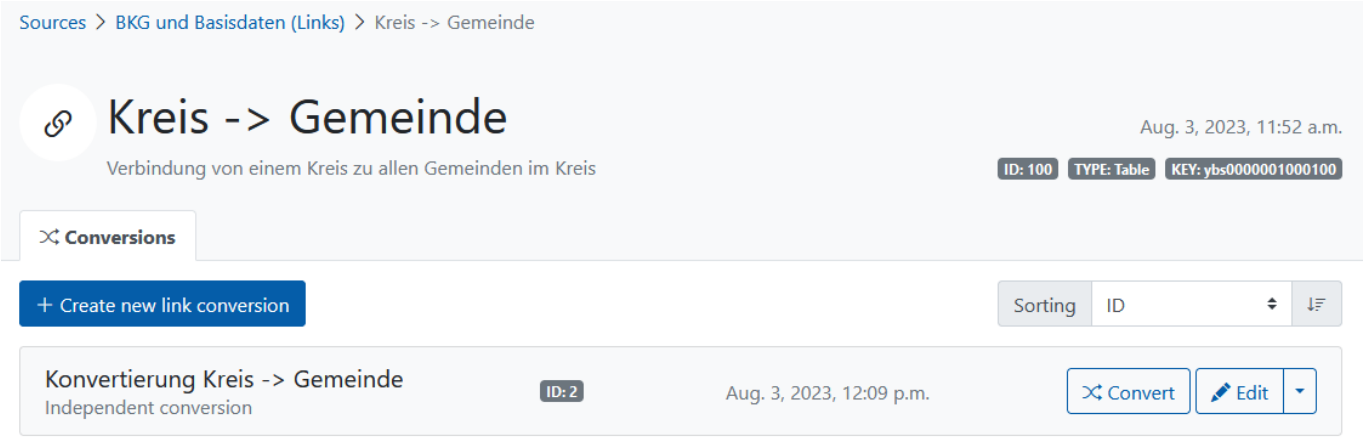
Im Ergebnis ist das Konvertierungsformular nun folgendermaßen befüllt und damit die Definition der Konvertierung vollständig:



6. Die Konvertierung durchführen und das Ergebnis kontrollieren

6.1 Durchführen, Prozessergebnis der Konversion

Die fertige Konvertierung kann nun abgespeichert werden und taucht in der Liste der Konvertierungen dieser Verbindung auf, die zu Beginn dieses Tutorials noch leer war. Mittels des Kopfes "Convert" kann die Konvertierung ausgeführt werden.



Das Interface leitet auf eine neue Seite um, die das Ergebnis der durchgeführten Verbindungskonvertierung anzeigt:

Die Angaben haben folgende Bedeutung:

- source count: So viele Zeilen hat die Quelltabelle.
- target count: So viele Zeilen hat die Verbindungstabelle, nachdem die Konversion gelaufen ist, d.h. so viele Verbindungen zwischen Kreisobjekten und Gemeindeobjekten gibt es nun im Simplex4Data.
- converted objects: So viele Elemente der Zielstruktur (d.h. der Verbindungstabelle) wurden in dieser Konvertierung neu eingelesen ("insert") bzw. aktualisiert oder neu eingelesen ("upsert"), durch Aktualisierungen überschrieben und als altes ergebnis archiviert ("archive")

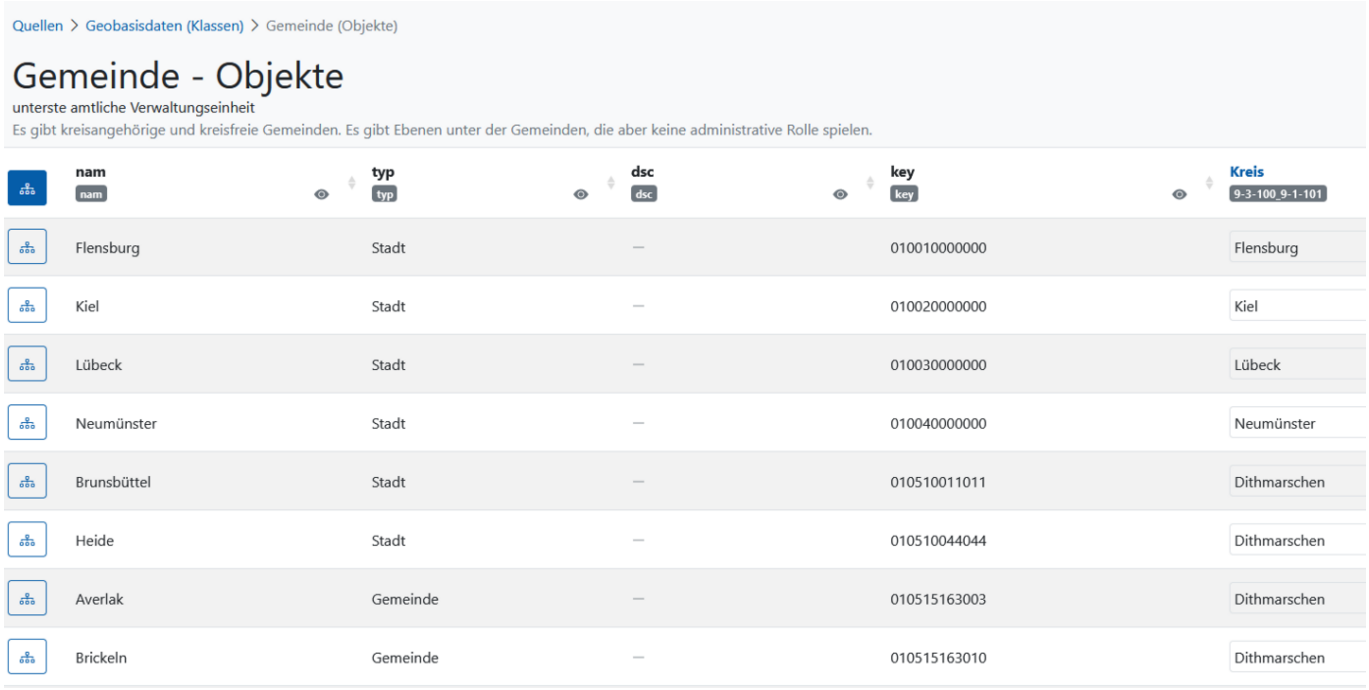
Daraus lassen sich u.a. folgende Sachverhalte ablesen:

- Der source count entspricht exakt demjenigen aus dem vorherigen Tutorial zur Objektkonvertierung. Da dieselbe Quelltabelle mit demselben Filter verwendet wurde, war dies zu erwarten.
- Es wurden 0 existierende Verbindungen archiviert, was nicht überrascht, da es bisher keine Verbindungen gab.

Eine tiefergreifende Qualitätsanalyse kann von diesen Ergebnissen ausgehen und sie weiter vertiefen.

6.2 fertige Verbindungen

Im Klassenmenü kann über den Knopf "Betrachten" eine Tabellenansicht der aktuellen Objekte der Klasse angezeigt werden. Dabei können neben den Sachattributen eines Objekt auch all seine verbundenen Objekte angezeigt werden. Das folgende Bild zeigt eine Ansicht der Klasse "Gemeinden", in der die meisten Standardfelder ausgeblendet sind und für jede Gemeinde der verbundene Kreis zu sehen ist.



Quellen > Geobasisdaten (Klassen) > Gemeinde (Objekte)

Gemeinde - Objekte

unterste amtliche Verwaltungseinheit

Es gibt kreisangehörige und kreisfreie Gemeinden. Es gibt Ebenen unter der Gemeinden, die aber keine administrative Rolle spielen.

nam (nam)	typ (typ)	dsc (dsc)	key (key)	Kreis (9-3-100_9-1-101)
Flensburg	Stadt	–	010010000000	Flensburg
Kiel	Stadt	–	010020000000	Kiel
Lübeck	Stadt	–	010030000000	Lübeck
Neumünster	Stadt	–	010040000000	Neumünster
Brunsbüttel	Stadt	–	010510011011	Dithmarschen
Heide	Stadt	–	010510044044	Dithmarschen
Averlak	Gemeinde	–	010515163003	Dithmarschen
Brickeln	Gemeinde	–	010515163010	Dithmarschen

Beispiel: Detaillierter Ablauf der Konvertierung

Um das Beschriebene noch einmal zu veranschaulichen, hier ein schematischer Ablauf, wie die Konvertierung der ersten Elemente der Quelltabelle mit der oben definierten Konvertierung abläuft.

1. Start

Die Objekttabelle ist zu Beginn leer, denn die Verbindung ist neu definiert und es wurde noch keine Konvertierung durchgeführt. Die beiden Klassentabellen sind mit Objekten gefüllt, die nun verbunden werden können. Die Quelltabelle steht zur Verfügung, im folgenden wird je Iteration eine Zeile der Quelltabelle herangezogen.

Obj 1 (FROM-Klasse) Kreis

o	ndx	nam
1	01051	Dithmarschen

Obj 2 (TO-Klasse) Gemeinde

o	ndx	nam
32998	010515163003	Averlak

o1	o2

Verbindungstabelle Kreis → Gemeinde yb_0000000900100

Quelltabelle Q_00000000900008

satz-art	text-kenn-zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...
40	43	1	0	51	null	null	Dithmarschen	
60	64	1	0	51	5163	3	Averlak	
60	64	1	0	51	5163	10	Brickeln	

Start

2. Iteration 1

Die erste Zeile der Quelltabelle passiert den Filter nicht.

Obj 1 (FROM-Klasse) Kreis

o	ndx	nam
1	01051	Dithmarschen

Filter
q.gemeinde is not null

Obj 2 (TO-Klasse) Gemeinde

o	ndx	nam
32998	010515163003	Averlak

o1	o2

Verbindungstabelle
Kreis → Gemeinde
yb_0000000900100

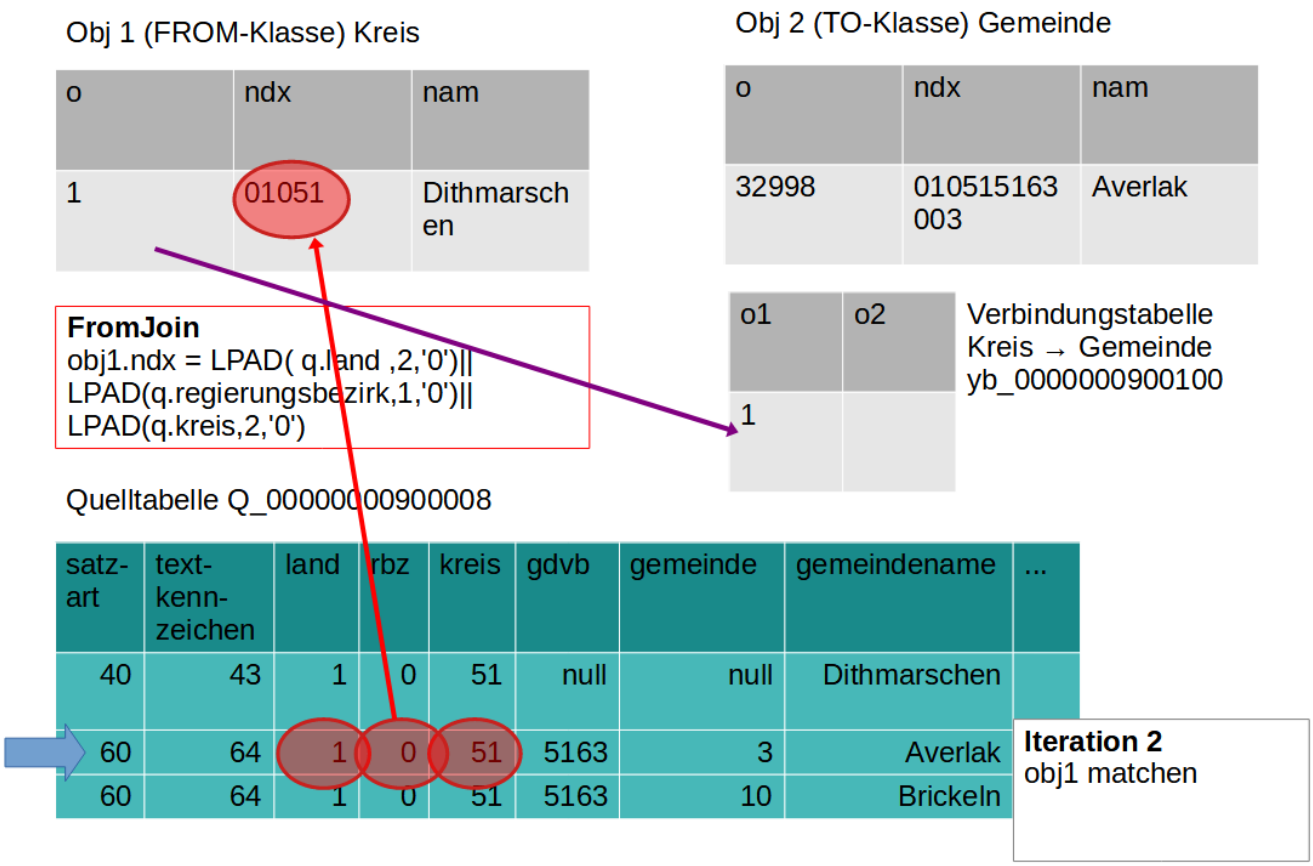
Quelltabelle Q_00000000900008

satz-art	text-kenn-zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...
40	43	1	0	51	null	null	Dithmarschen	
60	64	1	0	51	5163	3	Averlak	
60	64	1	0	51	5163	10	Brickeln	

Iteration 1
Kein Eintrag, da Filter nicht erfüllt

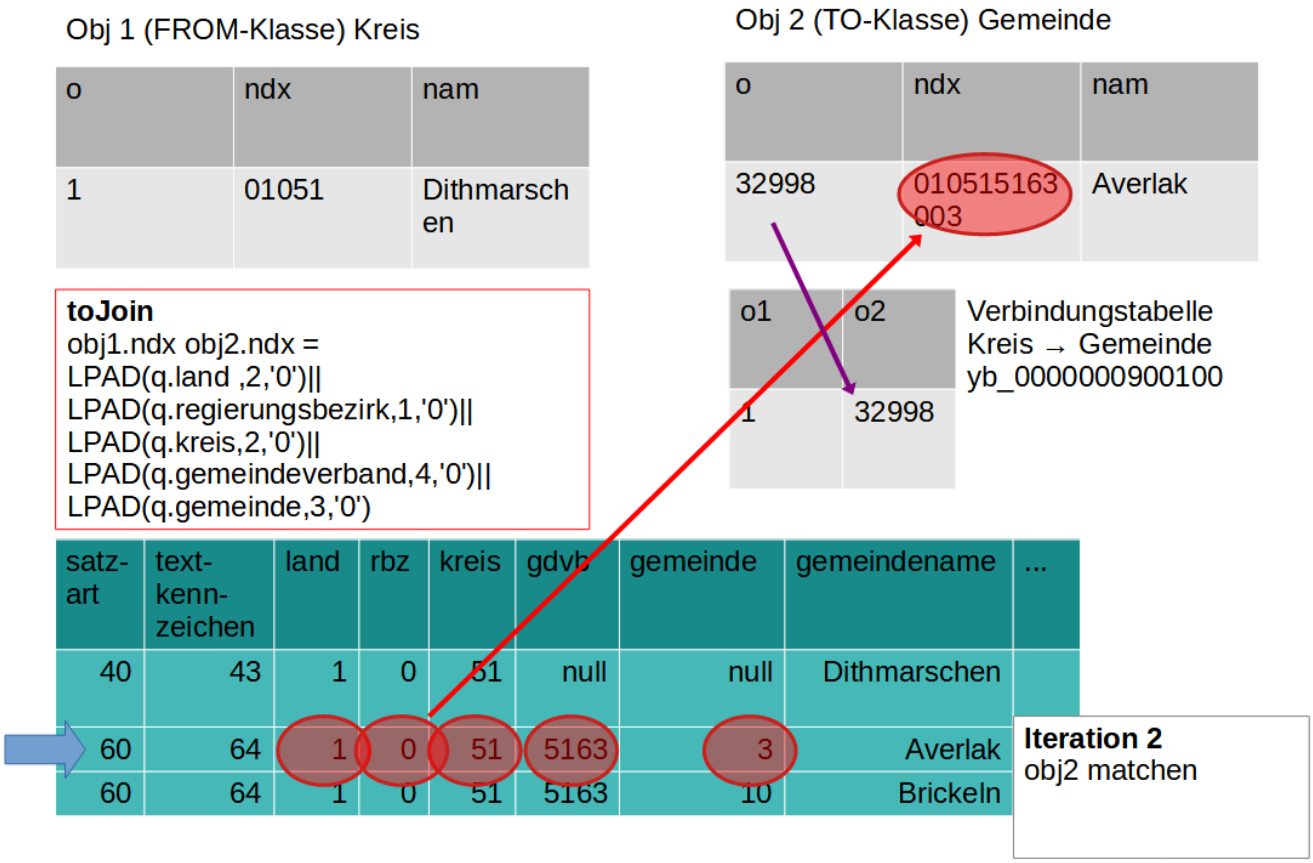
3. Iteration 2: fromJoin

Die zweite Zeile passiert den Filter. Nun kann ein Matching mit den beiden Klassen versucht werden. In der From-Klasse findet sich tatsächlich ein Objekt (d.h. ein Kreis), auf den die Bedingung des fromJoin zutrifft.



4. Iteration 2: toJoin

Um eine Verbindung erfolgreich herzustellen, muss in derselben Iteration auch das Matching mit der to-Klasse erfolgreich sein. Auch dort findet sich tatsächlich ein Objekt (d.h. eine Gemeinde) auf den die Bedingung des toJoin zutrifft.



5. bisheriges Ergebnis

Als Ergebnis der ersten beiden Iterationen ist ein Eintrag in der Verbindungstabelle entstanden, d.h. es wurde eine Verbindung angelegt, zwischen dem Kreis Dittmarschen und der Gemeinde Averlak. Es schließen sich so viele Iterationen an, wie die Quelltabelle weitere Zeilen hat.

Variante: Konvertierungen ohne Quelltabellen

Wie bereits angedeutet, kann bei Verbindungskonvertierungen auch auf eine Quelltabelle verzichtet werden, wenn die für eine Konvertierung nötigen Informationen in den beiden Klassen zu finden sind. Dazu ist folgendes zu beachten:

- fromJoin, toJoin und filter müssen ohne Bezug zu "q" formuliert werden.
- fromJoin und toJoin müssen beide "obj1" und "obj" aufeinander beziehen.

Klasse 1 (FROM-Klasse) Bundesland

o	ndx	key	typ	Attribut A	Attribut B

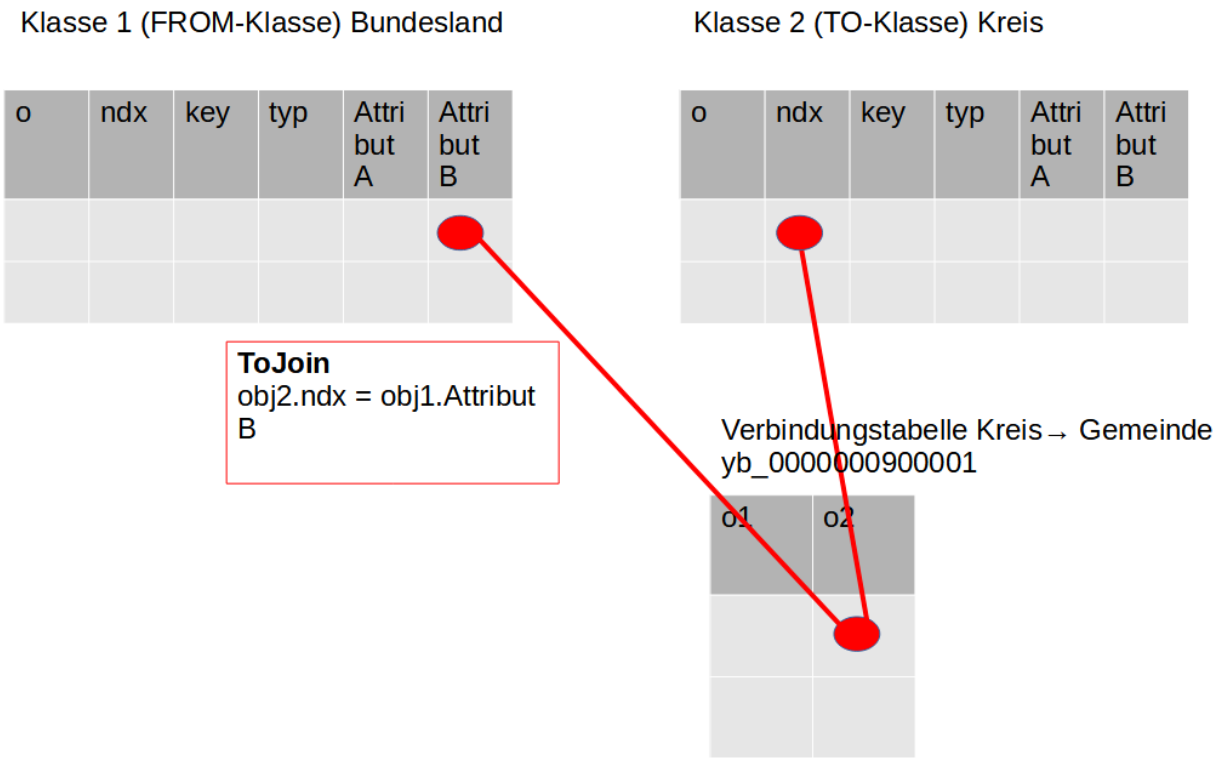
Klasse 2 (TO-Klasse) Kreis

o	ndx	key	typ	Attribut A	Attribut B

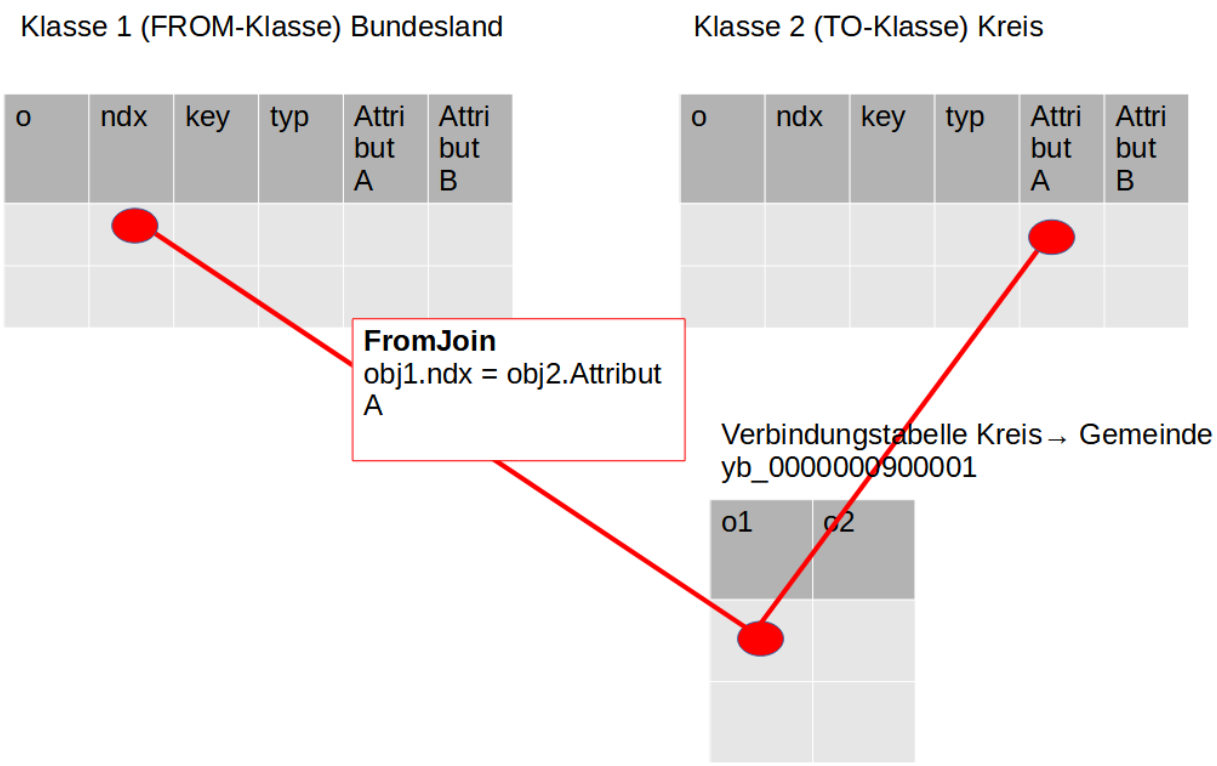
Verbindungstabelle Kreis → Gemeinde
yb_0000000900001

o1	o2

Eine toJoin-Regel könnte beispielsweise folgendermaßen aussehen:



Und eine fromJoin-Regel beispielsweise so:



- Tutorial: ein Sachattribut konvertieren
- Nach oben
- Datenpool des Simplex verwalten