

Schritt-für-Schritt: Konvertieren einer Klasse

Folgende Schritte sind für die Definition einer Konvertierung zu erledigen:

1. Die **Zielklasse** der Konvertierung (obj) **auswählen**.
2. Eine **Quelltabelle** (q), deren Daten für die Konvertierung benutzt werden sollen, **auswählen**.
3. (optional) einen **Filter definieren**.
4. Eine **Aktualisierungsstrategie** für die Konvertierung **festlegen**.
5. Die Konvertierung möglichst genau **beschreiben**.
6. Konvertierungen für die Standardfelder **definieren**.
7. 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


- Als Grundlage einer Objektkonvertierung muss eine Quelltabelle existieren, die mit Daten gefüllt ist. Im vorliegenden Beispiel liegt eine solche Quelltabelle vor, wie im [Tutorial zum Rohdatenimport](#) beschrieben.
- Außerdem muss eine Klasse definiert worden sein.

Start

Ausgangspunkt ist das Menü der Klasse "Gemeinden", für welche die Konvertierung angelegt werden soll.

[< back](#) [Ansehen](#) [Edit](#) [Löschen](#)

Sources > Basisdaten (Classes) > Gemeinde (Attributes)



Gemeinde

unterste amtliche Verwaltungseinheit

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

3. Mai 2023 08:50

ID: 100 TYPE: table KEY: BSR0000000900100

Attributes
Conversions
Links

[+ Create new attribute](#) Sorting ID ID ↕ ⌵


123	Fläche Fläche in km2. Stand: 31.12.2021	ID: 133 TYPE: ar Edit Löschen
☰	Bevölkerung to be deleted	ID: 134 TYPE: ac Edit Löschen
123	Bevölkerung menschliche Population innerhalb Gemeinde	ID: 135 TYPE: ai Edit Löschen
123	Männliche Bevölkerung männliche menschliche Population innerhalb Gemeinde	ID: 136 TYPE: ai Edit Löschen
123	Weibliche Bevölkerung weibliche menschliche Population innerhalb Gemeinde	ID: 137 TYPE: ai Edit Löschen

Es folgt eine Übersicht über bereits definierte Konvertierungen. Dabei werden neben Objektkonvertierungen auch Konvertierungen für Sachattribute und Verbindungen angezeigt, die mit dieser Klasse zusammenhängen. Die Übersicht ist aktuell leer.

Mittels "create new object conversion" kann eine neue Objektkonvertierung gestartet werden.

[< back](#) [Ansehen](#) [Edit](#) [Löschen](#)

Sources > Basisdaten (Classes) > Gemeinde (Conversions)



Gemeinde

unterste amtliche Verwaltungseinheit

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

3. Mai 2023 08:50

ID: 100 TYPE: table KEY: BSR0000000900100

Attributes
Conversions
Links

[+ Create new object conversion](#) Sorting ID ID ↕ ⌵

[Convert](#)

Attribute conversions (0) [+ New attribute conversion](#)

All attribute conversions are associated with an object conversions.

Link conversions (0) [+ New link conversion](#)

All link conversions are associated with an object conversions.

1. Die Zielklasse der Konvertierung (obj) auswählen

Wie in 0. bereits deutlich wurde, ist die Zielklasse dieser Konvertierung die Klasse Gemeinden, d.h. es wird nun darum gehen, einzelne Klassenobjekte aus einer noch festzulegenden Quelltable zu erzeugen.

Sie ist im Formular als "Class" bereits gesetzt, ebenso ihre Quelle (hier als "Source").

Create new object conversion

An object conversion creates objects with attributes and links

Cancel Submit

General information Attribute conversions Link conversions

Source: 9 Basisdaten Class: 100 Gemeinde Convid: Auto: choose next

Source table: Click to search Name:

Comment:

Filter: Update mode: archive

Zur Klasse Gemeinden mit der ClassId 100 gehört datenbankseitig die Objekttable bsr0000000900100. Einige der wichtigsten Standardfelder, sind im folgenden schematisch abgebildet. Aktuell ist die Tabelle leer, da noch keine Konvertierung ausgeführt, ja noch nicht mal eine definiert wurde.

ndx	key	typ	nam	...

Die Objekttable

bsr0000000900100, schematisch dargestellt

2. Eine Quelltable (q), deren Daten für die Konvertierung benutzt werden sollen, auswählen

Als nächstes muss eine Quelltable für die Konvertierung ausgewählt werden. Dies geschieht durch ein Dropdown-Menü im Feld "Source Table". Im Dropdown sind sämtliche in Simplex4Data verfügbaren Quelltabellen aufgelistet.

Die Quelltablette q_00000000900008 mit ihren ersten Zeilen, schematisch dargestellt

3. (optional) einen Filter definieren

Bei der Objektkonvertierung wird die Quelltablette zeilenweise eingelesen und versucht, aus jeder Zeile ein Objekt der Zielklasse zu erzeugen. Mittels des Filters auf "q" können Regeln festgelegt werden, um bestimmte Zeilen zu überspringen.

Für die Objektkonvertierung zu Gemeinden sollen nur die Zeilen herangezogen werden, die auch tatsächlich Daten zu Gemeinden enthalten (und nicht etwa zu Landkreisen oder Bundesländern). Deswegen wird der Filter auf "q.gemeinde is not null" gesetzt. Damit werden alle Zeilen, die für die Spalte "gemeinde" keinen Wert aufweisen, von der Konvertierung ausgenommen. Im der abgebildeten Tabellenausschnitt sind das die Zeilen Nr.1, 2, 4 und 6.

4. Eine Aktualisierungsstrategie für die Konvertierung festlegen

Zuletzt muss auch für Verbindungen eine Aktualisierungsstrategie ausgewählt werden, die für die zukünftige Verwaltung der Daten relevant ist, siehe dazu den [Artikel zum Datenmodell, in dem die Aktualisierungsstrategien beschrieben und erklärt werden.](#)

Für diese Konvertierung wird der default **B1, C2, D2 and archived A1, C1** gewählt, aus folgenden Gründen:

- Es kann davon ausgegangen werden, dass künftige Aktualisierungen einen vollständigen neuen Datensatz enthalten (d.h. den dann gültigen Stand aller dann existierenden Landkreise bzw. Gemeinden abbilden). Deswegen sind nicht im neuen Stand enthaltene Objekte sicher nicht mehr aktuell.
- Da es sich um sehr basale Objekte mit potentiell vielen Verbindungen handelt, ist eine Archivierung nicht mehr aktueller Objekte dem Weglöschen vorzuziehen, da dabei wichtige Informationen verloren gehen können.

5. 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 einem Objekt (vgl. oben, Abschnitt 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.

Ausgefüllt sieht die Oberfläche nun folgendermaßen aus:

General information
Attribute conversions
Link conversions

Source 9 Basisdaten	Class 100 Gemeinde	Convid Auto: choose next
Source table q_0000000900008 Gemeindeverzeichnis Deutschland	Name Konvertierung Gemeinden	
Comment Deutschlandweit alle Gemeinden aus dem DESTATIS-Datensatz zu Verwaltungsebenen		
Filter q.gemeinde is not null	Updatemode archive	

6. Konvertierungen für die Standardfelder definieren

In der Klassenkonvertierung werden in der [Klassendefinition beschriebenen Standardfelder](#) aus den Quellstrukturen befüllt. Dazu wird nun - in derselben Syntax wie für "Filter" - für jedes Standardfeld eine Regel aufgestellt, wie es zu befüllen ist.







Es müssen nicht alle Standardfelder befüllt werden, ebenfalls können Standardfelder auch nur für manche Objekte befüllt werden und für andere nicht.

Für Felder, die nicht befüllt werden, weil es (a) keine Regel zur Befüllung definiert wird oder (b) die Regel für die aktuelle Zeile der Quelltable kein Ergebnis liefert, wird automatisch der PostgreSQL-Datentyp null eingetragen, um zu markieren, dass zu ihnen keine Informationen vorliegen.

Einzig und allein das Standardfeld "ndx" muss für jedes einzelne Objekt gesetzt werden, da es eine eindeutige Identifizierung des Objekts ermöglichen muss.

Im vorliegenden Beispiel werden die Felder folgendermaßen befüllt:

Object information

Key	<input type="text" value="LPAD(q.land ,2,'0') LPAD(q.regierungsbezirk,1,'0') LPAD(q.kreis,2,'0') LPAD(q.gemeindeverband,4,'0') LPAD(q.gemeinde,3,'0')"/>	 ▾
Internal key	<input type="text" value="LPAD(q.land ,2,'0') LPAD(q.regierungsbezirk,1,'0') LPAD(q.kreis,2,'0') LPAD(q.gemeindeverband,4,'0') LPAD(q.gemeinde,3,'0')"/>	 ▾
Type	<input type="text" value="CASE WHEN q.textkennzeichen = '60' THEN 'Markt' WHEN q.textkennzeichen = '61' THEN 'kreisfreie Stadt' WHEN q.textkennzeichen = '62'"/>	 ▾
Name	<input type="text" value="q.gemeindename"/>	 ▾
Description	<input type="text"/>	 ▾
Comment	<input type="text"/>	 ▾
Begin	<input type="text"/>	End <input type="text"/>

Die Konvertierung wird gespeichert und erscheint nun in der bisher leeren Liste der Konvertierungen für die Klasse Gemeinden. Sie ist an ihrem Namen und der ausgewählten Quelltable zu erkennen. Wie die Übersicht ebenfalls zeigt, sind mit ihr keine Attributs- oder Verbindungskonvertierungen verbunden.

Gemeinde

unterste amtliche Verwaltungseinheit

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

July 31, 2023, 1:32 p.m.

ID: 100
TYPE: table
KEY: bs_0000001000100

Attributes

Conversions

Links

+ Create new object conversion

Sorting ID ⌵ ⌴

Independent conversions
↻ Convert

Attribute conversions (0)
+ New attribute conversion

All attribute conversions are associated with an object conversions.

Link conversions (0)
+ New link conversion

All link conversions are associated with an object conversions.

Konvertierung Gemeinden q_0000000900008 1

↻ Convert all
✎ Edit

Attribute conversions (0)
+ New attribute conversion

This object conversion does not include attribute conversions.

Link conversions (0)
+ New link conversion

This object conversion does not include link conversions.

7. Die Konvertierung durchführen und das Ergebnis kontrollieren

7.1. Durchführen, Prozessergebnis der Konversion

Durch Anklicken der Auswahl neben dem Button "Convert all" öffnet sich die Möglichkeit, "Convert objects only". Diese wird hier genutzt. Für den vorliegenden Fall wäre sie streng genommen nicht nötig, da keine Attributs- oder Verbindungskonvertierungen mit der Objektkonvertierung verbunden sind.

Das Interface leitet auf eine neue Seite um, die das Ergebnis der durchgeführten Objektkonvertierung

anzeigt:

Conversions for source 10 conversion

	Source table	Name	Source count	Converted objects	Target table	Target count
Edit conversion	q_0000000900008	Konvertierung Gemeinden	16072	insert 0 upsert 10997 archive 0	O_0000001000000	10997
Convert again						
Show instances						

Die Angaben haben folgende Bedeutung:

- source count: So viele Zeilen hat die Quelltable.
- target count: So viele Zeilen hat die Objekttable, nachdem die Konversion gelaufen ist, d.h. so viele Gemeindeobjekte gibt es nun im Simplex4Data.
- converted objects: So viele Objekte der Zielstruktur wurden neu eingelesen ("insert"), aktualisiert oder neu eingelesen ("upsert"), durch Aktualisierungen überschrieben und als altes Ergebnis archiviert ("archive")

Daraus lassen sich u.a. folgende Sachverhalte ablesen:

- Da der target count exakt dem "upsert"-Wert entspricht, war die Objekttable vor der Ausführung der Konvertierung leer. Alle nun im Simplex4Data enthaltenen Gemeindeobjekte stammen aus dieser einen Konvertierungsausführung.
- Für $16072 - 10997 = 5075$ Zeilen der Quelltable wurden keine Objekte angelegt. Das liegt im vorliegenden Fall daran, dass sie durch den gesetzten Filter aussortiert wurden.
- Es wurden 0 existierende Gemeindeobjekte archiviert, was nicht überrascht, da es bisher keine Gemeindeobjekte gab.

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

7.2. fertige Gemeindeobjekte

Im Klassenmenü kann über den Knopf "Betrachten" eine Tabellenansicht der aktuellen Objekte der Klasse angezeigt werden.









[< zurück](#)

[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	typ	dsc	key	cmt	ndx	beg	fin	Letzte Bearbeitungszeit
	Flensburg	Stadt	—	010010000000	—	010010000000	2008-01-01T00:00:00Z	—	19. Oktober 2023 11:28
	Kiel	Stadt	—	010020000000	—	010020000000	2006-01-01T00:00:00Z	—	19. Oktober 2023 11:28
	Lübeck	Stadt	—	010030000000	—	010030000000	2006-02-01T00:00:00Z	—	19. Oktober 2023 11:28
	Neumünster	Stadt	—	010040000000	—	010040000000	1970-04-26T00:00:00Z	—	19. Oktober 2023 11:28
	Brunsbüttel	Stadt	—	010510011011	—	010510011011	2009-01-01T00:00:00Z	—	19. Oktober 2023 11:28
	Heide	Stadt	—	010510044044	—	010510044044	2009-01-01T00:00:00Z	—	19. Oktober 2023 11:28
	Averlak	Gemeinde	—	010515163003	—	010515163003	2008-01-01T00:00:00Z	—	19. Oktober 2023 11:28
	Brickeln	Gemeinde	—	010515163010	—	010515163010	2008-01-01T00:00:00Z	—	19. Oktober 2023 11:28

Beispiel: Detaillierter Ablauf der Konvertierung

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

1. Start

Die Objekttable ist zu Beginn leer, denn die Klasse ist neu definiert und es wurde noch keine Konvertierung durchgeführt. Die Quelltable steht zur Verfügung, im folgenden wird je Iteration eine Zeile der Quelltable herangezogen.

ndx	key	typ	nam	...

satz- art	text- kenn- zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...
40	43	1	0	51	null	null	Dithmarschen	
50	50	1	0	51	11	null	Brunsbüttel, Stadt	
60	63	1	0	51	11	11	Brunsbüttel, Stadt	
50	50	1	0	51	44	null	Heide, Stadt	
60	63	1	0	51	44	44	Heide, Stadt	
50	51	1	0	51	5163	null	Burg-St. Michaelisdonn	
60	64	1	0	51	5163	3	Averlak	Start
60	64	1	0	51	5163	10	Brickeln	
...								

2. Iteration 1

Die erste Zeile der Quelltable passiert den Filter nicht. Dementsprechend wird kein Objekt der Zielklasse angelegt.

Filter
q.gemeinde is not null

ndx	key	typ	nam	...

satz- art	text- kenn- zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...
40	43	1	0	51	null	null	Dithmarschen	
50	50	1	0	51	11	null	Brunsbüttel, Stadt	
60	63	1	0	51	11	11	Brunsbüttel, Stadt	
50	50	1	0	51	44	null	Heide, Stadt	
60	63	1	0	51	44	44	Heide, Stadt	
50	51	1	0	51	5163	null	Burg-St. Michaelisdonn	
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

Auch die zweite Zeile der Quelltable passiert den Filter nicht. Dementsprechend wird kein Objekt der Zielklasse angelegt.

Filter
q.gemeinde is not null

ndx	key	typ	nam	...

satz-art	text-kenn-zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...
40	43	1	0	51	null	null	Dithmarschen	
50	50	1	0	51	11	null	Brunsbüttel, Stadt	
60	63	1	0	51	11	11	Brunsbüttel, Stadt	
50	50	1	0	51	44	null	Heide, Stadt	
60	63	1	0	51	44	44	Heide, Stadt	
50	51	1	0	51	5163	null	Burg-St. Michaelisdonn	
60	64	1	0	51	5163	3	Averlak	
60	64	1	0	51	5163	10	Brickeln	
...								

Iteration 2
Kein Eintrag, da Filter nicht erfüllt

4. Iteration 3: ndx

Die dritte Zeile passiert den Filter, nun wird ein Gemeindeobjekt angelegt.

Das Objekt erhält als ndx einen Wert, der nach genau definierten Regeln aus fünf Spalten der Quelltable zusammengesetzt wird.

internal key

```

LPAD(q.land,2,'0')||
LPAD(q.regierungsbezirk,1,'0')||
LPAD(q.kreis,2,'0')||
LPAD(q.gemeindeverband,4,'0')||
LPAD(q.gemeinde,3,'0')

```

ndx	key	typ	nam	...
0105				
10011				
011				

satz-art	text-kenn-zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...
40	43	1	0	51	null	null	Dithmarschen	
50	50	1	0	51	11	null	Brunsbüttel, Stadt	
60	63	1	0	51	11	11	Brunsbüttel, Stadt	
50	50	1	0	51	44	null	Heide, Stadt	
60	63	1	0	51	44	44	Heide, Stadt	
50	51	1	0	51	5163	null	Burg-St. Michaelisdonn	
60	64	1	0	51	5163	3	Averlak	
60	64	1	0	51	5163	10	Brickeln	
...								

Iteration 3
ndx (=ars) bilden

5. Iteration 3: nam

Weiter erhält das entstehende Objekt als nam einen Wert, der direkt aus einer Spalte der Quelltable übernommen wird.

name
q.gemeindenname

ndx	key	typ	nam	...
0105 10011 011			Brunsbüttel, Stadt	

satz-art	text-kenn-zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindenname	...
40	43	1	0	51	null	null	Dithmarschen	
50	50	1	0	51	11	null	Brunsbüttel, Stadt	
60	63	1	0	51	11	11	Brunsbüttel, Stadt	
50	50	1	0	51	44	null	Heide, Stadt	
60	63	1	0	51	44	44	Heide, Stadt	
50	51	1	0	51	5163	null	Burg-St. Michaelisdonn	
60	64	1	0	51	5163	3	Averlak	
60	64	1	0	51	5163	10	Brickeln	
...								

Iteration 3
nam bilden

6. Iteration 3: typ

Weiter erhält das entstehende Objekt als typ den Wert "Stadt". Dieser wird nicht direkt aus einer Spalte der Quelltable übernommen, stattdessen sind in der Konvertierung des Feldes "typ" mehrere Konstanten (unter anderem "Stadt") eingetragen, aus denen abhängig von der Spalte "textkennzeichen" der Quelltable eine ausgewählt wird.

type

```

CASE WHEN q.textkennzeichen = '60'
THEN 'Markt' WHEN q.textkennzeichen
= '61' THEN 'kreisfreie Stadt' [...]

```

ndx	key	typ	nam	...
0105 10011 011		Stadt	Brunsbüttel, Stadt	

satz-art	text-kenn-zeichen	land	rbz	kreis	gdvb	gemeinde	gemeindename	...
40	43	1	0	51	null	null	Dithmarschen	
50	50	1	0	51	11	null	Brunsbüttel, Stadt	
60	63	1	0	51	11	11	Brunsbüttel, Stadt	
50	50	1	0	51	44	null	Heide, Stadt	
60	63	1	0	51	44	44	Heide, Stadt	
50	51	1	0	51	5163	null	Burg-St. Michaelisdonn	
60	64	1	0	51	5163	3	Averlak	
60	64	1	0	51	5163	10	Brickeln	
...								

Iteration 3
typ bilden

7. bisheriges Ergebnis

Als Ergebnis der ersten drei Iterationen ist ein Eintrag in die Objekttable entstanden, d.h. es wurde ein Objekt angelegt, die Gemeinde "Brunsbüttel, Stadt". Es schließen sich so viele Iterationen an, wie die Quelltable weitere Zeilen hat.

- [Konvertieren](#)
- [Nach oben](#)
- [Tutorial: ein Sachattribut konvertieren](#)