

Datenbankobjekte

🌐 Sprache: **Deutsch** · [English](#)

🔧 Datenbankobjekte

Eigene Views, Funktionen und Prozeduren als Teil eines Nuclets – inkl. Reihenfolge/Löschanweisung und CA_-Konvention.

HOW-TO

ANWENDUNGSENTWICKLER

STAND: JUL 2026

GILT FÜR NUCLOS 4.2026.X

Auf dieser Seite

- [Was sind Datenbankobjekte?](#)
- [Objekt anlegen](#)
- [Quelltext eintragen](#)
- [Verwandte Seiten](#)

Was sind Datenbankobjekte?

Datenbankobjekte konfigurieren **benutzerdefinierte Views, Funktionen, Prozeduren usw. als Teil eines Nuclets** – bei einem Nuclel-Transfer werden sie mitübernommen. Typische Anwendungen: DB-Funktionen für [berechnete Attribute](#), Views für virtuelle Businessobjekte oder Packages für Jobs.

Menüaufruf: *Konfiguration Datenbank Datenbankobjekte*



Abhängigkeiten unter PostgreSQL

Über das Feld **Reihenfolge** (seit Nucleos 3.10) legst du Abhängigkeiten fest: Beim Ändern/Transfer werden alle Objekte in absteigender Reihenfolge gelöscht und in aufsteigender wieder angelegt. Voraussetzung ist ein **Löschstatement** je Objekt.

Objekt anlegen

Zuerst ein **Datenbankobjekt** mit passendem **Typ** anlegen: *function, index, package, package body, procedure, view*. Für jedes unterstützte Datenbanksystem kann ein eigenes Objekt hinterlegt werden; herangezogen wird das des aktiven Systems.



Namenskonvention

Soll eine Function für ein berechnetes Attribut dienen, muss ihr Name mit CA_ beginnen – nur solche Objekte erscheinen im BO-Assistenten zur Auswahl.

The screenshot shows a web-based application interface for creating a database object. The top navigation bar includes a gear icon and a menu with items: Datei, Bearbeiten, Administration, Konfiguration, Artikel, Auftragswesen, Bestellwesen, Lager, and Stammdaten. Below this is a sub-menu with 'Datenbankobjekt - Neueingabe' (active) and 'Datenbankobjekt-Quelltext'. The main form area has two tabs: 'Liste' and 'Details' (selected). The form fields are: 'Datenbankobjekt' with the value 'DEF_' and a yellow highlight; 'Beschreibung' with an empty text box; 'Typ' with a dropdown menu open showing a list of database object types: function, index, package, package body, procedure, and view; and 'Reihenfolge' with an empty text box. The dropdown menu for 'Typ' is highlighted with a yellow border and a mouse cursor is pointing at it.

Datenbankobjekt anlegen und Typ zuweisen.

Quelltext eintragen

Im **Datenbankobjekt-Quelltext** (Konfiguration Datenbank Datenbankobjekt-Quelltexte) trägst du den DB-spezifischen Code ein. Ist **Aktiv** gesetzt, wird das Objekt beim Speichern erzeugt/aktualisiert. Den Schema-/DB-Namen setzt Nuclos automatisch davor – nicht selbst angeben. Die **Löschanweisung** wird beim Löschen des Datensatzes ausgeführt.

Suche Liste Details

Extras

Datenbankobjekt

BAK2_CA_ARTIKEL_BESTAND

Datenbanktyp

POSTGRESQL

Aktiv?



Quelltext

```
CREATE OR REPLACE FUNCTION BAK2_CA_ARTIKEL_BESTAND(id numeric)
RETURNS numeric AS
$BODY$
DECLARE
    menge numeric;
BEGIN
    SELECT SUM(dblbestand) INTO menge
    FROM BAK2_LAGERBESTAND
    WHERE intid_strartikel = id;

    RETURN menge;
END;
$BODY$
LANGUAGE 'plpgsql' VOLATILE
COST 100;
```

Löschanweisung

```
DROP FUNCTION BAK2_CA_ARTIKEL_BESTAND(id numeric);
```

Maske „Datenbankobjekt-Quelltext“.

Funktions-Rumpf (PostgreSQL)

```
CREATE OR REPLACE FUNCTION <name>(imoduleid numeric)
  RETURNS numeric AS
$BODY$

$BODY$
  LANGUAGE 'plpgsql' VOLATILE
  COST 100;v
```

Beispiel

```
CREATE OR REPLACE FUNCTION GET_NEXT_NUMBER(imoduleid numeric)
  RETURNS numeric AS
$BODY$
DECLARE
  iresult NUMERIC (20, 0);
BEGIN
  SELECT intnaechstenummer INTO iresult
  FROM t_eo_nummern
  WHERE strbezeichnung = 'Kundennummer';

  RETURN iresult;
END;
$BODY$
  LANGUAGE 'plpgsql' VOLATILE
  COST 100;
```

Löschanweisung (PostgreSQL)

```
DROP FUNCTION IF EXISTS GET_NEXT_NUMBER(imoduleid numeric) CASCADE
```



Achtung

Bei PostgreSQL muss immer eine Löschanweisung angegeben werden.

Verwandte Seiten



Berechnete Attribute

CA_-Funktionen.

[Öffnen](#)



Datenquellen

Datenquellen.

[Öffnen](#)