Database objects

Language: Deutsch · **English**

Database objects

> **Custom views, functions and procedures as part of a nuclet – incl. order/delete statement and the CA_ convention.**

HOW-TO | APPLICATION DEVELOPER | UPDATED: JUL 2026 | APPLIES TO NUCLOS 4.2026.X

On this page

- What are database objects?
- Creating an object
- Entering the source code
- Related pages

What are database objects?

Database objects configure **custom views, functions, procedures etc. as part of a nuclet** – they are included in a nuclet transfer. Typical uses: DB functions for calculated attributes, views for virtual business objects, or packages for jobs.

Menu: Configuration Database Database objects

Dependencies under PostgreSQL

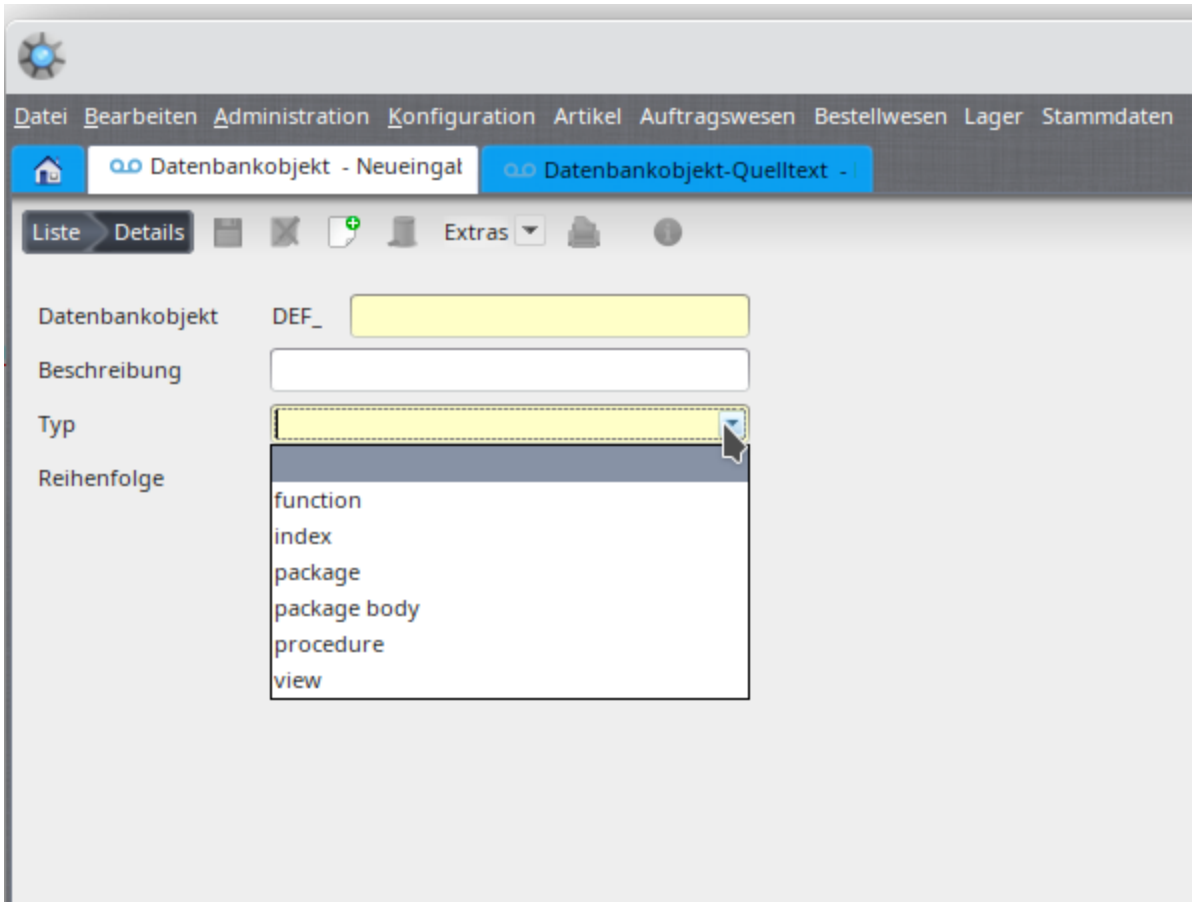
The **order** field (since Nuclos 3.10) defines dependencies: on change/transfer all objects are dropped in descending and recreated in ascending order. This requires a **delete statement** per object.

Creating an object

First create a **database object** with a suitable **type**: *function, index, package, package body, procedure, view*. A separate object can be stored per supported database system; the one for the active system is used.

Naming convention

If a function is to serve a calculated attribute, its name must start with `CA_` – only such objects appear for selection in the BO wizard.



Creating a database object and assigning a type.

Entering the source code

In the **database object source** (Configuration Database Database object sources) you enter the DB-specific code. If **active** is set, the object is created /updated on save. Nuclos prepends the schema/DB name automatically – do not add it yourself. The **delete statement** runs when the record is deleted.

? Unbekannter Anhang

The „database object source“ screen.

Function body (PostgreSQL)

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

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

Example

```
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;
```

Delete statement (PostgreSQL)

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

Caution

On PostgreSQL a delete statement must always be provided.

Related pages

Calculated attributes

CA_ functions.

Open

Data sources

Data sources.

Open