

Virtuelle Businessobjekte

🌐 Sprache: **Deutsch** · [English](#)

🔧 Virtuelle Businessobjekte

Businessobjekte auf Basis einer Datenbank-View – Pflichtspalten, ID-Factory, Anlage per Wizard und Schreibzugriff via UPDATE RULE.

HOW-TO

ANWENDUNGSENTWICKLER

STAND: JUL 2026

GILT FÜR NUCLOS 4.2026.X

Auf dieser Seite

- [Was ist ein virtuelles BO?](#)
- [Anforderungen an die View](#)
- [ID-Factory \(Beispiel PostgreSQL\)](#)
- [Virtuelles BO anlegen](#)
- [Virtuelles BO ändern](#)
- [Verwandte Seiten](#)

Was ist ein virtuelles BO?

Ein **virtuelles Businessobjekt** basiert nicht auf einer Tabelle, sondern auf einer **Datenbank-View**. Voraussetzung ist daher eine View (siehe [Datenbankobjekte](#)).

Anforderungen an die View

Die View muss mindestens diese Spalten enthalten (exakte Bezeichnung!):

intid,	-- Datensatzidentifizierer
strcreated,	-- Ersteller als String
datcreated,	-- Erstellungsdatum
strchanged,	-- Änderer als String
datchanged,	-- Änderungsdatum
intversion	-- Datensatzversion

- intid muss **eindeutig** sein, wenn referenziert wird.
- Referenzspalten beginnen mit intid_str...
- Datentypen ermittelt Nuclos automatisch; bei falschen Typen hilft ein expliziter Cast in der View (z.B. `db_funktion(intid)::numeric(9,2)`).



Benennungsregeln

Spaltennamen: **klein**, ohne Sonderzeichen, **max. 30 Zeichen**. Bezeichnungen in Datenquelle und BO sind case-sensitiv und müssen übereinstimmen. Gut: `datum`, `datum_ist` – schlecht: `Datum`, `"Datum (Ist)"`.

ID-Factory (Beispiel PostgreSQL)

Zum Schreiben über ein virtuelles BO wird eine ID-Factory (Sequence-Funktion) benötigt:

```
CREATE OR REPLACE FUNCTION Y25N_IDFACTORY()  
  RETURNS numeric  
  LANGUAGE 'plpgsql'  COST 100  
  VOLATILE  
AS $BODY$DECLARE  
  id numeric;  
BEGIN  
  SELECT NEXTVAL('idfactory') INTO id;  RETURN id;  
END;$BODY$;
```

Virtuelles BO anlegen

Im [Businessobjekt-Wizard](#) ein neues BO anlegen und in Schritt 2 über *Erweiterte Eigenschaften einblenden*:

- **Virtuelles Businessobjekt**: die vorbereitete View wählen (fehlt sie, ist sie evtl. ungültig).
- **Id Factory für VE** (optional): DB-Funktion, wenn über das BO geschrieben werden soll – das BO wird dann automatisch änderbar.
- **Tabellenname**: bei virtuellen BOs leer lassen.

Erweiterte Eigenschaften des virtuellen BO.

Schreibzugriff

Zum Anlegen/Löschen von Datensätzen über Nuclos ist zusätzlich eine **UPDATE RULE** nötig, die das Update auf die zugrunde liegende(n) Tabelle(n) ausführt.

Virtuelles BO ändern

Zuerst die View unter [Datenbankobjekte](#) anpassen und speichern, dann den [Businessobjekt-Wizard](#) erneut durchlaufen.

Anwendungsfall

Virtuelle BOs eignen sich auch für **feste Stammdaten** als Teil eines Nuclets (z.B. Herr/Frau, Ja/Nein, Himmelsrichtungen).

Verwandte Seiten

Businessobjekt

Datenmodell.

[Öffnen](#)

Datenbankobjekte

Views.

[Öffnen](#)

