

# Berechnetes Attribut

- Definition
- Anwendung
- Dynamische Hintergrundfarbe eines Attributs
  - Beispiel

## Definition

Menüaufruf: (Konfiguration) - (Datenquellen) - (Berechnetes Attribut)

Ein berechnetes Attribut ist ein einzelner Rückgabewert einer Datenbankabfrage, der im **Businessobjekt** mit einem Feld verknüpft werden kann. Somit kann dieser Wert in der Maske zur Laufzeit angezeigt werden, ohne dass für dessen Aktualisierung eine Aktion vom Benutzer nötig wird.

Berechnete Attribute haben gegenüber Funktionen in Datenbankobjekten den Vorteil, dass man bei der Berechnung noch ein paar Kontext-Variablen zur Verfügung hat, z.B.

- username
- mandator

Beim Mandanten / mandator muss der Platzhalter extra geklammert werden, z.B. so

### Abfrage für mandantenabhängige Nucletparameter

```
select
  pm.strvalue
from
  t_md_nucletparameter
  inner join t_md_mandator_param_value pm on t_md_nucletparameter.struid
where
  pm.struid_t_md_nuclet_param where strparameter = 'Nucletparameter x'
  and pm.struid_t_md_mandator = ('$mandator')
```



### Wichtiger Hinweis

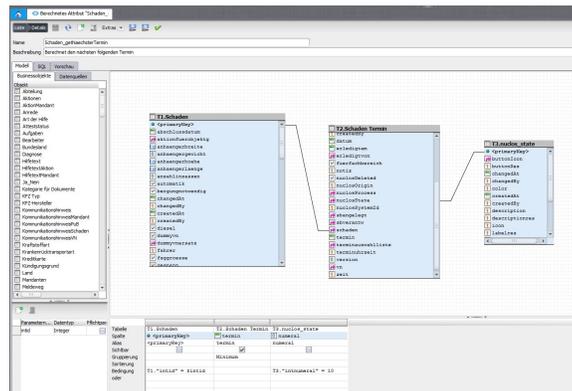
In der Datenquelle sollte nicht "t" als Tabellen-Alias verwendet werden, da dieser schon von Nuclio verwendet wird und es dadurch zu falschen Ergebnissen kommen kann.

## Anwendung

Ein Berechnetes Attribut kann über eine Datenquelle definiert werden.

Benötigt wird ein Parameter für die INTID des Datensatzes, in dem das Attribut eingeblendet werden soll. Dieses muss in der Abfrage entsprechend berücksichtigt werden, wenn es einen Bezug zum Hauptdatensatz geben soll.

Der Rückgabewert muss dem Wert entsprechen, der im Attribut angezeigt werden soll. Im **Businessobjekt** muss später bei der Verknüpfung der gleiche Datentyp gewählt werden.



Weitere Beispiele:

The screenshot shows a SQL query editor with the following content:

```

Name: Schaden_getSummeAlleSchadenProfil
Beschreibung: Liefert die Summe aller bisherigen Schäden des Versicherungsnehmers

SELECT coalesce(SUM(T2."@dateing"), 0.0) "dateing"
FROM hdbi_schaden T4
INNER JOIN hdbi_versicherungsmoeder T3 ON T4.intid_strom = T3.intid
INNER JOIN hdbi_schaden T1 ON T1.intid = T1.intid_strom
INNER JOIN hdbi_schadensabfragen T2 ON T4.intid = T2.intid_schaden
INNER JOIN c_md_state T5 ON T1.stuid_mdloostate = T5.stuid
WHERE T1."intid" = einid
AND
T2."instanznr" > 10

```

## Dynamische Hintergrundfarbe eines Attributs

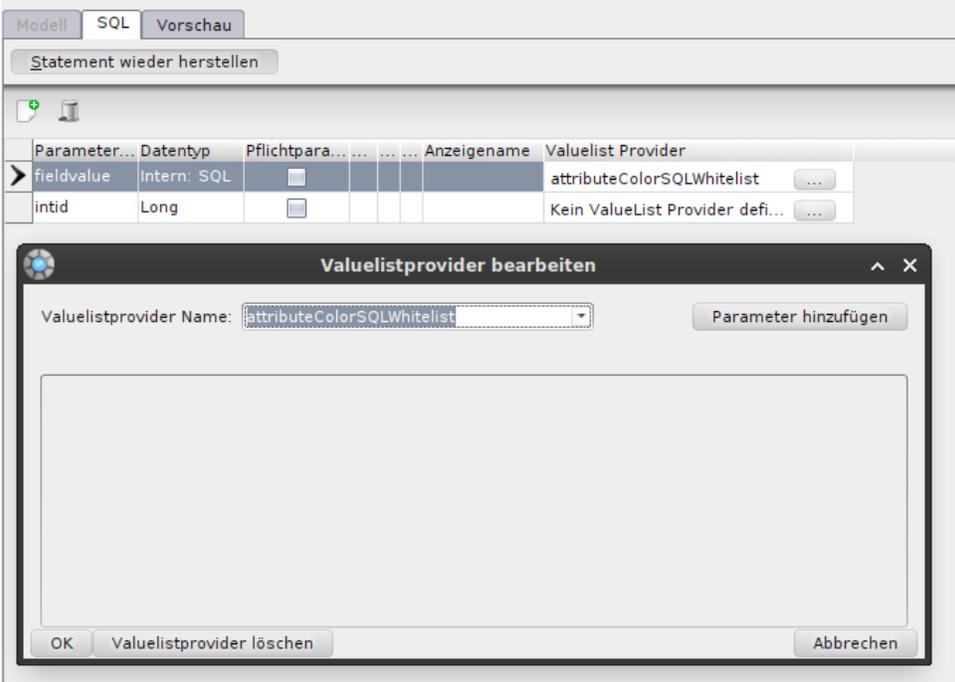
 Dieses Feature steht ab Nucløs 4.2024.5 zur Verfügung.

Der Richclient bietet die Möglichkeit die Hintergrundfarbe einer Komponente für ein Attribut mittels Groovy-Skript zu bestimmen (s.a. [Regeln \(clientseitig\)](#)).

Da Groovy-Skripte im Webclient nicht ausgewertet werden können, bietet Nucløs alternativ die Verwendung einer Datenquelle ("Berechnetes Attribut") an, die den Farbwert als hex-kodierten String zurückliefert, bspw #00ff00 für die Farbe "grün".

Diese Datenquellen werden wie gewohnt über den Menüpunkt "Datenquellen Berechnetes Attribut" angelegt. Es stehen die folgenden Parameter zur Verfügung:

Parameter	Datentyp	Beschreibung
<i>intid</i>	Long	Liefert die ID des Datensatzes
<i>strfield</i>	String	Name des Attributs, für das die Hintergrundfarbe bestimmt werden soll.
<i>struidfield</i>	String	UID des Attributs, für das die Hintergrundfarbe bestimmt werden soll.

<i>fieldvalue</i>	Intern: SQL	<p>Wert des Attributs, für das die Hintergrundfarbe bestimmt werden soll.</p> <p>Dieser Parameter wird durch ein SQL-Snippet ersetzt. Als Valuelist Provider für die SQL Whitelist muss <code>attributeColorSQLWhitelist</code> ausgewählt werden.</p> <p>Handelt es sich um einen Wert, der quotiert werden muss, wie bspw. einen Text, muss bei Anzeigename folgender Text eingetragen werden: ` `</p> <p>Bei Werten, die nicht quotiert werden müssen, wie bspw. booleschen oder numerischen Werten, bleibt der Anzeigeausdruck leer.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p><b>i</b> Falls die Hintergrundfarbe für ein berechnetes Attribut (Wert wird von Datenquelle oder DB-Funktion geliefert) bestimmt werden soll, wird der Parameter durch das SQL aus der Datenquelle, bzw. den Aufruf der DB-Funktion ersetzt.</p> </div>  <p>The screenshot shows a software interface with a table of parameters. The table has columns: Parameter..., Datentyp, Pflichtpara..., Anzeigename, and Valuelist Provider. The first row is selected, showing 'fieldvalue' with 'Intern: SQL' as the data type and 'attributeColorSQLWhitelist' as the provider. Below the table, a dialog box titled 'Valuelistprovider bearbeiten' is open, showing the same provider name in a dropdown menu and buttons for 'OK', 'Valuelistprovider löschen', and 'Abbrechen'.</p>
<i>username</i>	String	Benutzername des aktuellen Benutzers.
<i>mandator</i>	String	Liste der UUIDs der für den aktuellen Benutzer gültigen Mandanten

## Hierarchie zur Bestimmung der Hintergrundfarbe einer Komponente

**Richclient:** Groovy-Skript > Datenquelle > `nuclosrowcolor` (nur relevant für Ergebnisliste oder Unterformular) > statische Hintergrundfarbe aus Layout

**Webclient:** Datenquelle > `nuclosrowcolor` (nur relevant für Ergebnisliste oder Unterformular) > statische Hintergrundfarbe aus Layout

## Beispiel

```
SELECT
    CASE
        WHEN $fieldvalue IS NULL THEN null
        WHEN $fieldvalue > 100 THEN '#00ff00'
        ELSE '#ff0000'
    END as color
```