

**Datenbankmodell  
„Virtuelles Fahrzeug“  
Version 1.1**

Michael Bellair

Copyright © 2007 Michael Bellair.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled „GNU Free Documentation License“.

### **Zusammenfassung**

Das Datenbankmodell „Virtuelles Fahrzeug“ wird beschrieben. Dieses Dokument ist der GNU Free Documentation License (GNU-FDL) unterstellt und somit im Rahmen dieser Lizenz frei veränderbar. Jede veränderte Version ist jedoch selbst unter der GNU Free Documentation License (GNU-FDL) zu veröffentlichen.

Die Lizenz kann Anhang A entnommen werden. Eine deutsche Übersetzung ist unter <http://www.giese-online.de/gnufdl-de.html> zu finden.



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>7</b>
<b>1 Einleitung</b>	<b>9</b>
<b>2 Das Datenbankmodell</b>	<b>11</b>
<b>3 dok</b>	<b>12</b>
3.1 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	13
3.2 Schlüssel und Bedingungen . . . . .	14
3.3 dok.nimm__doku . . . . .	14
<b>4 intern</b>	<b>16</b>
4.1 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	17
4.2 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	18
4.3 Schlüssel und Bedingungen . . . . .	19
4.4 intern.aggreat__zk . . . . .	19
4.5 intern.gib__ere__abczk . . . . .	19
4.6 intern.gib__ere__buchstabe . . . . .	20
4.7 intern.gib__ere__restzk . . . . .	21
4.8 intern.gib__liste__abczkrestzk . . . . .	21
4.9 intern.gib__liste__tabellenpositionen . . . . .	22
4.10 intern.gib__naechsteid . . . . .	23
4.11 intern.gib__sqlselect . . . . .	24
4.12 intern.gib__stsname . . . . .	24
4.13 intern.gib__zu . . . . .	26
4.14 intern.ist__abczk . . . . .	26
4.15 intern.loesche__zeile . . . . .	27
4.16 intern.nimm__tlkommentar . . . . .	28
4.17 intern.nimm__tlkommentarfunktion . . . . .	29
4.18 intern.nimm__tlkommentarsicht . . . . .	30
4.19 intern.nimm__tlkommentarsts . . . . .	31
4.20 intern.nimm__zeile . . . . .	31
4.21 intern.pruefe__array . . . . .	33
4.22 intern.pruefe__dt . . . . .	35
4.23 intern.verbindezk . . . . .	38
4.24 intern.zeige__fremdschluesel . . . . .	38
4.25 intern.zeige__spaltenanzahl . . . . .	38
4.26 intern.zeige__tabspinformationen . . . . .	39
<b>5 prj</b>	<b>40</b>
5.1 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	41
5.2 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	42
5.3 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	43
5.4 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	44
5.5 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	45

5.6	Tabellenhierarchie der Ebenen 1 — 3 . . . . .	46
5.7	Tabellenhierarchie der Ebenen 1 — 3 . . . . .	47
5.8	Tabellenhierarchie der Ebenen 1 — 3 . . . . .	48
5.9	Schlüssel und Bedingungen . . . . .	49
5.10	prj.nimm__tabwerte . . . . .	53
5.11	prj.zeige__dokdoku . . . . .	61
5.12	prj.zeige__pl_dokausgabe_le . . . . .	62
5.13	prj.zeige__pl_prtdokdoku_le . . . . .	62
5.14	prj.zeige__pl_prtpsgefszenario . . . . .	63
5.15	prj.zeige__pl_prtpsgefszenario_le . . . . .	63
5.16	prj.zeige__pl_psgefabk . . . . .	63
5.17	prj.zeige__pl_psgefabk_le . . . . .	64
5.18	prj.zeige__pl_psgefam . . . . .	64
5.19	prj.zeige__pl_psgefam_le . . . . .	65
5.20	prj.zeige__pl_psgefstrkt . . . . .	65
5.21	prj.zeige__pl_psgefstrkt_le . . . . .	66
5.22	prj.zeige__pp_psgefszenario . . . . .	66
5.23	prj.zeige__ppl_dokdoku_le . . . . .	66
5.24	prj.zeige__prtprotokoll . . . . .	67
5.25	prj.zeige__prtprotokolltabellen . . . . .	67
5.26	prj.zeige__psabk . . . . .	68
5.27	prj.zeige__strstrkt . . . . .	68
<b>6</b>	<b>prt</b> . . . . .	<b>70</b>
6.1	Tabellenhierarchie der Ebenen 1 — 3 . . . . .	71
6.2	Schlüssel und Bedingungen . . . . .	72
6.3	prt.nimm__protokolleintrag . . . . .	72
6.4	prt.nimm__typ . . . . .	73
<b>7</b>	<b>ps</b> . . . . .	<b>75</b>
7.1	Tabellenhierarchie der Ebenen 1 — 3 . . . . .	76
7.2	Schlüssel und Bedingungen . . . . .	77
7.3	ps.nimm__abk . . . . .	77
7.4	ps.zeige__risikomatrix_en50126 . . . . .	78
<b>8</b>	<b>spr</b> . . . . .	<b>79</b>
8.1	Tabellenhierarchie der Ebenen 1 — 3 . . . . .	80
8.2	Schlüssel und Bedingungen . . . . .	81
8.3	spr.nimm__wgruppe . . . . .	81
8.4	spr.zeige__wgelemente . . . . .	84
8.5	spr.zeige__wgruppe . . . . .	84
<b>9</b>	<b>str</b> . . . . .	<b>85</b>
9.1	Tabellenhierarchie der Ebenen 1 — 3 . . . . .	86
9.2	Schlüssel und Bedingungen . . . . .	87
9.3	str.pruefe__fbwerte . . . . .	87

<b>10 tl</b>	<b>89</b>
10.1 Tabellenhierarchie der Ebenen 1 — 3 . . . . .	90
10.2 Schlüssel und Bedingungen . . . . .	91
10.3 tl.erzeuge__abczk . . . . .	92
10.4 tl.erzeuge__dbmodell . . . . .	92
10.5 tl.erzeuge__tldbgllossary . . . . .	93
10.6 tl.erzeuge__tlzk . . . . .	94
10.7 tl.gib__datei . . . . .	94
10.8 tl.gib__ere__tlbefehl . . . . .	95
10.9 tl.gib__ere__tlwort . . . . .	97
10.10tl.gib__ere__tlzzbefehl . . . . .	98
10.11tl.gib__liste__tlzzbefehl . . . . .	99
10.12tl.ist__tlwort . . . . .	100
10.13tl.nimm__datei . . . . .	101
10.14tl.nimm__tlwort . . . . .	102
10.15tl.nimm__erkannte__liste__tlobjekte . . . . .	104
<b>A GNU Free Documentation License</b>	<b>107</b>
<b>Verzeichnis der Tabellenspalten und Begriffe</b>	<b>115</b>





# 1 Einleitung

Es wird die Struktur des Datenbankmodells vorgestellt. Jedem Datenbankschema ist ein Abschnitt zugeordnet. Ziel und Zweck des Schemas werden beschrieben.



## **2 Das Datenbankmodell**

Datenbank des Virtuellen Fahrzeugs

### **3 dok**

Im Schema „Dokumente“ werden die in der Datenbank verwendeten Dokumentenreferenzen gehalten.

### 3.1 Tabellenhierarchie der Ebenen 1 — 3

dok.doku	id	int4	1
	idfrei	int4	
	notizen	varchar	
	intern_firma_id	int4	3
	spr_uebvar_id_kb	int4	13
	aspekt	varchar	

## 3.2 Schlüssel und Bedingungen

### Primärschlüssel

pk\_\_dok\_doku\_id für Spalte dok.doku.id

### Fremdschlüssel

fk\_\_dok\_doku\_\_intern\_firma\_id aus Spalte dok.doku.intern\_firma\_id

fk\_\_dok\_doku\_\_spr\_uebvar\_id\_kb aus Spalte dok.doku.spr\_uebvar\_id\_kb

### Unique Constraints

uc\_\_dok\_doku in Tabelle dok.doku für die Spalte(n) dok.doku.idfrei, dok.doku.intern\_firma\_id, dok.doku.spr\_uebvar\_id\_kb

### Check Constraints

cc\_\_dok\_doku in Tabelle dok.doku die Bedingung (((aspekt)::text = 'BEWEIS'::text)  
OR ((aspekt)::text = 'QUELLE'::text))

## 3.3 dok.nimm\_\_doku

### Funktionsname

dok.nimm\_\_doku( IN liste\_werte varchar[], OUT dokuid varchar )

### Funktionsparameter/Rückgabewert(e)

liste\_werte Liste der Werte, welche in dok.doku eingelesen werden sollen (Hinweise siehe Funktionsbeschreibung — Werteliste einlesen). Sie werden in der folgenden Reihenfolge in die Spalten

1. spr\_uebvar\_id\_kb
2. intern\_firma\_id
3. aspekt
4. notizen

eingetragen. Es müssen mindestens die ersten zwei Werte übergeben werden.

Für aspekt darf nur BEWEIS zur Kennzeichnung von Dokumenten, die beweisen, daß eine Sicherheitsanforderung umgesetzt wurde bzw. QUELLE zur Kennzeichnung von Dokumenten, die auf den Ursprung einer Sicherheitsanforderung verweisen.

dokuid Es wird die Datenbank-Identifikationsnummer der eingelesenen Dokumentenreferenz (dok.doku.id) zurückgegeben.

### Funktionsbeschreibung

Mittels dieser Funktion werden die im System zu referenzierenden Dokumente in dok.doku eingelesen.

Werteliste einlesen:

Dem Funktionsparameter `liste_werte` ist eine Werteliste (ARRAY) zu übergeben (die maximale Anzahl und die Reihenfolge der Werte ist fest vorgegeben — siehe oben). Soll nur eine Teilmenge der Werte übergeben werden, dann sind alle Werte von links nach rechts zu spezifizieren (nichtbenötigte Werte am Ende der Liste dürfen weggelassen werden).

## ACHTUNG

**Die Beispiele müssen aktualisiert werden !**

### Beispiele

-- Fehlermeldung. Der Parameter `liste_werte` darf nicht NULL sein.

```
SELECT * FROM dok.nimm__doku( NULL );
```

-- Fehlermeldung. Es sind mindestens zwei Werte zu übergeben.

```
SELECT * FROM dok.nimm__doku( ARRAY['08-15'] );
```

-- Fehlermeldung. Die Werte müssen größer 0 bzw. dürfen nicht leer sein.

```
SELECT * FROM dok.nimm__doku( ARRAY['', '1'] );
```

```
SELECT * FROM dok.nimm__doku( ARRAY['08-15', '0'] );
```

-- Dokumentenreferenz einlesen

```
SELECT dok_doku_id, doknr, intern_firma_id_eigentuemer, spr_uebvar_id_doktitel,  
spr_uebvar_id_kurzbeschr, dok_doku_notizen FROM dok.zeige__doku;
```

```
SELECT * FROM dok.nimm__doku( ARRAY['08-15', '1'] );
```

```
SELECT * FROM dok.nimm__doku( ARRAY['08-15', '1', spr.nimm__wgruppe( 'Dokumententitel',  
'1', '0' )] );
```

```
SELECT * FROM dok.nimm__doku( ARRAY['08-15', '1', spr.nimm__wgruppe( 'Dokumententitel',  
'1', '0' ), spr.nimm__wgruppe( 'Dokumentenkurzbeschreibung', '1', '0' )] );
```

```
SELECT * FROM dok.nimm__doku( ARRAY['08-15', '1', spr.nimm__wgruppe( 'Dokumententitel',  
'1', '0' ), spr.nimm__wgruppe( 'Dokumentenkurzbeschreibung', '1', '0' ), 'Zusatzinform  
)
```

```
SELECT dok_doku_id, doknr, intern_firma_id_eigentuemer, spr_uebvar_id_doktitel,  
spr_uebvar_id_kurzbeschr, dok_doku_notizen FROM dok.zeige__doku;
```

## 4 intern

Das Schema `Intern` dient der Bereitstellung von Tabellen, Datenstrukturen, usw. welche in allen Schemas verwendet werden können. Weiterhin wird die Basis der Logbuch-Funktionalität einschließlich der Logbuch-Funktion der Tabellen

- `abczk`,
- `restzk`

bereitgestellt.



## 4.1 Tabellenhierarchie der Ebenen 1 — 3

intern.abczk <sup>2</sup> ↙

id	int4
idfrei	int4
notizen	varchar
abczk	varchar

intern.alle

id	int4
idfrei	int4
notizen	varchar

intern.allewlog

id	int4
idfrei	int4
notizen	varchar
logdatuhrz	timestamp
logdqok	bool
logquelle	varchar
logentfernt	bool
logname	varchar

intern.firma <sup>3</sup> ↙

id	int4
idfrei	int4
notizen	varchar
name	varchar
name_k	varchar

## 4.2 Tabellenhierarchie der Ebenen 1 — 3

intern.restzk	
id	int4
idfrei	int4
notizen	varchar
restzk	varchar

4

## 4.3 Schlüssel und Bedingungen

### Primärschlüssel

pk\_\_intern\_abczk\_id für Spalte intern.abczk.id

pk\_\_intern\_firma\_id für Spalte intern.firma.id

pk\_\_intern\_restzk\_id für Spalte intern.restzk.id

### Unique Constraints

uc\_\_intern\_abczk in Tabelle intern.abczk für die Spalte(n) intern.abczk.idfrei, intern.abczk.a

uc\_\_intern\_firma in Tabelle intern.firma für die Spalte(n) intern.firma.idfrei, intern.firma.n  
intern.firma.name\_k

uc\_\_intern\_restzk in Tabelle intern.restzk für die Spalte(n) intern.restzk.idfrei,  
intern.restzk.restzk

### Check Constraints

cc\_\_intern\_alle in Tabelle intern.alle die Bedingung ((id = 0) AND (idfrei = 0))

cc\_\_intern\_allewlog in Tabelle intern.allewlog die Bedingung ((((((id = 0) AND (idfrei  
= 0)) AND (logdatuhrz IS NOT NULL)) AND (logname IS NOT NULL)) AND (logdquelle  
IS NULL)) AND (logdqok IS NULL)) AND (logentfernt = false))

## 4.4 intern.aggregat\_\_zk

### Funktionsname

intern.aggregat\_\_zk()

### Funktionsparameter/Rückgabewert(e)

-

### Funktionsbeschreibung

Diese Aggregatfunktion verbindet sämtliche Teilzeichenketten zu einer Zeichenkette.

## 4.5 intern.gib\_\_ere\_abczk

### Funktionsname

intern.gib\_\_ere\_abczk( OUT ere\_abczk varchar )

### Funktionsparameter/Rückgabewert(e)

ere\_abczk Ist der erweiterte reguläre Ausdruck zur Beschreibung einer AbcZK.

## Funktionsbeschreibung

Ziel der Funktion ist das Definieren und Zurückgeben eines erweiterten regulären Ausdrucks (ERE), welcher das Textmuster eine Zeichenkette beschreibt, die ausschließlich aus Buchstaben besteht - (AbcZK)- (siehe `intern.gib_ERE_Buchstabe(...)`).

## Beispiele

```
-- ERE zurückgeben
SELECT * FROM intern.gib_ERE_ABCZK();

-- hallo zurückgeben
SELECT substring( 'hallo' from intern.gib_ERE_ABCZK() );

-- hal zurückgeben
SELECT substring( 'hal2lo' from intern.gib_ERE_ABCZK() );
```

## 4.6 intern.gib\_\_ere\_buchstabe

### Funktionsname

```
intern.gib__ere_buchstabe( IN istanfang bool, OUT ere_buchstabe varchar )
```

### Funktionsparameter/Rückgabewert(e)

**istanfang** Der erweiterte reguläre Ausdruck eines Anfangsbuchstaben soll zurückgegeben werden

**ere\_buchstabe** Soll der erweiterte reguläre Ausdruck zur Beschreibung eines Anfangsbuchstaben zurückgegeben werden, ist `istAnfang = true` zu setzen. Soll der erweiterte reguläre Ausdruck zur Beschreibung eines Folgebuchstaben zurückgegeben werden, ist `istAnfang = false` zu setzen.

### Funktionsbeschreibung

Ziel der Funktion ist das Definieren und Zurückgeben eines erweiterten regulären Ausdrucks (ERE), welcher das Textmuster eines Anfangsbuchstaben oder Folgebuchstaben beschreibt. Die folgenden Buchstaben werden vom ERE erfaßt:

- A-Z
- a-z
- äöü
- ÄÖÜ
- ß (nur Folgebuchstabe)

## Beispiele

```
-- ERE für Anfangsbuchstaben zurückgeben
SELECT * FROM intern.gib__ERE_Buchstabe( true );

-- ERE für die Buchstaben im Wort zurückgeben
SELECT * FROM intern.gib__ERE_Buchstabe( false );

-- h zurückgeben
SELECT substring( 'h' from intern.gib__ERE_Buchstabe( true ) );

-- Leerzeichenkette zurückgeben
SELECT substring( 'ß' from intern.gib__ERE_Buchstabe( true ) );

-- ß zurückgeben
SELECT substring( 'ß' from intern.gib__ERE_Buchstabe( false ) );
```

## 4.7 intern.gib\_\_ere\_restzk

### Funktionsname

```
intern.gib__ere_restzk( OUT ere_restzk varchar )
```

### Funktionsparameter/Rückgabewert(e)

ere\_restzk Ist der erweiterte reguläre Ausdruck zur Beschreibung der Zeichenketten, die keine Buchstaben enthalten (in intern.restzk gültig sind).

### Funktionsbeschreibung

Ziel der Funktion ist das Definieren und Zurückgeben eines erweiterten regulären Ausdrucks (ERE), welcher das Textmuster der Zeichenkette beschreibt, die über die Tastatur eingetippt ist, aber nicht aus Buchstaben besteht.

## Beispiele

```
-- ERE zurückgeben
SELECT * FROM intern.gib__ERE_RestZK();
```

## 4.8 intern.gib\_\_liste\_abczkrestzk

### Funktionsname

```
intern.gib__liste_abczkrestzk( IN wortgruppe varchar, OUT zk varchar, OUT istabczk
bool )
```

### Funktionsparameter/Rückgabewert(e)

wortgruppe Wortgruppe, welche in abczk bzw. restzk zerlegt werden soll

zk Zeichenkette aus der wortgruppe, welche eine abczk oder restzk ist

istabczk Wenn istabczk = true dann ist die in zk vorliegende Zeichenkette eine abczk  
sonst ist sie eine restzk

### Funktionsbeschreibung

Ziel der Funktion ist das Zerlegen der Wortgruppe wortgruppe in abczk bzw. restzk mittels Erweiterter Regulaerer Ausdruecke (ERE).

Die Vorschrift zum Erkennen einer

- abczk kann intern.gib\_ERE.ABCZK() bzw.
- restzk kann intern.gib\_ERE.RestZK()

entnommen werden.

### Beispiele

-- Fehlermeldung. Die Wortgruppe ist anzugeben.

```
SELECT * FROM intern.gib_Liste_abczkrestzk( NULL );  
SELECT * FROM intern.gib_Liste_abczkrestzk( '' );
```

-- Es werden die Bestandteile der Wortgruppe zurückgegeben

```
SELECT '>>' || zk || '<' AS zk, istabczk FROM intern.gib_Liste_abczkrestzk( 'Das  
ist ein Array - This is an Array - 1984E' );
```

## 4.9 intern.gib\_\_liste\_tabellenpositionen

### Funktionsname

intern.gib\_\_liste\_tabellenpositionen( IN schname varchar, OUT tabname varchar, OUT  
ebene varchar )

### Funktionsparameter/Rückgabewert(e)

schname Name des Schemas dessen Tabellenpositionen (Ebenen innerhalb der Tabellenhierarchie) ermittelt werden.

tabname Tabellenname des Schemas

ebene Position (Ebene) der Tabelle im Schema

### Funktionsbeschreibung

Ziel der Funktion ist das Ermitteln der Positionen (Ebenen) sämtlicher Tabellen im Baum der Tabellenverbindungen (Fremdschlüssel) des Schemas schName.

### Beispiele

-- Fehlermeldung. Der Parameter schName darf nicht NULL oder Leer sein.

```
SELECT * FROM intern.gib__Liste_Tabellenpositionen( '' );
```

```
-- Tabellenpositionen ermitteln
```

```
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'dok' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'intern' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'kmp' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'prj' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'prt' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'ps' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'spr' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'str' );  
SELECT * FROM intern.gib__Liste_Tabellenpositionen( 'tl' );
```

## 4.10 intern.gib\_\_naechsteid

### Funktionsname

```
intern.gib__naechsteid( IN tabname varchar, OUT tabnameid varchar )
```

### Funktionsparameter/Rückgabewert(e)

tabname Name der Tabelle (ggf. einschliesslich Schemaname), aus der die nächste verfügbare Identifikationsnummer id ermittelt werden soll. (darf nicht leer sein).

tabnameid Nächste freie id

### Funktionsbeschreibung

Ziel der Funktion ist das Ermitteln der nächsten verfügbaren id der übergebenen Tabelle tabName. Es wird der kleinste mögliche Wert gesucht.

### Beispiele

```
-- Fehlermeldung. Der Parameter tabName darf nicht NULL sein.
```

```
SELECT intern.gib__naechsteid( NULL );
```

```
-- Eine wiederverwendbare id wurde nicht gefunden
```

```
SELECT max(id) FROM intern.abczk;
```

```
SELECT intern.gib__naechsteid( 'intern.abczk' );
```

```
-- Eine wiederverwendbare id wurde gefunden
```

```
SELECT intern.nimm__zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWort']  
) ;
```

```
SELECT intern.nimm__zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWortZwe']  
) ;
```

```
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
```

```
SELECT intern.loesche__zeile( 'intern.abczk', 'abczk = ''KeinRichtigesWort'' );
```

```
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
```

```
SELECT intern.gib__naechsteid( 'intern.abczk' );
```

## 4.11 intern.gib\_\_sqlselect

### Funktionsname

```
intern.gib__sqlselect( IN sel varchar, IN sel_from varchar[], IN sel_where varchar,  
OUT sqlselect varchar )
```

### Funktionsparameter/Rückgabewert(e)

**sel** Es ist die Liste der Parameter anzugeben, welche das Ergebnis der SELECT - Abfrage sind.  
Diese Variable darf nicht leer sein.

**sel\_from** Es ist die Liste (ARRAY) der Tabellen anzugeben, welche mittels SELECT - Abfrage ausgewertet werden sollen. Diese Variable darf nicht leer sein.

**sel\_where** Es ist eine WHERE - Bedingung anzugeben (ggf. einschließlich Semikolon). Diese Variable darf leer sein.

**sqlselect** Es wird die SELECT - Anweisung zurueckgegeben.

### Funktionsbeschreibung

Ziel der Funktion ist das Erzeugen einer SELECT - Anweisung, welche innerhalb einer PL/pgSQL - Funktion verwendet werden kann. Sie berücksichtigt das Konzept zur Wiederverwendung gelöschter id. Das abschliessende Semikolon ist explizit anzugeben.

### Beispiele

```
-- Fehlermeldung. Der Parameter sel darf nicht leer oder NULL sein.
```

```
SELECT intern.gib__sqlSELECT( '', '', '' );
```

```
-- Fehlermeldung. Der Parameter sel_from darf nicht leer oder NULL sein.
```

```
SELECT intern.gib__sqlSELECT( 'hsk', NULL, '' );
```

```
-- Fehlermeldung. Kein Element der Liste sel_from darf leer sein
```

```
SELECT intern.gib__sqlSELECT( 'hsk', ARRAY[''], '' );
```

```
-- SQL-Abfrage erzeugen
```

```
SELECT intern.gib__sqlSELECT( 'id, idfrei, notizen', 'intern.alle', '' );
```

```
SELECT intern.gib__sqlSELECT( 'id, idfrei, notizen', 'tt.muell, tt.test', ' AND  
tt.muell.id = tt.test.tt_muell_id' );
```

## 4.12 intern.gib\_\_stsname

### Funktionsname

```
intern.gib__stsname( IN befehl varchar, IN vollstname varchar, OUT name varchar
```



)

## Funktionsparameter/Rückgabewert(e)

**befehl** In Form eines fest vorgegebenen Befehls ist anzugeben, welcher Teil des vollständigen Spaltennamens zurückzugeben ist:

**sch** Schemaname

**tab** Tabellenname

**sp** Spaltenname

**scht** Schemaname und Tabellenname

**tabsp** Tabellenname und Spaltenname

Die Gültigkeit der übergebenen Befehle wird nicht überprüft. Es darf nicht NULL bzw. keine Leerzeichenkette angegeben werden.

**vollstname** Es ist der vollständige Spaltenname anzugeben. Es wird geprüft, ob die Syntax eingehalten wurde.

**name** Es wird der Name zurückgebene, welcher in **befehl** spezifiziert wurde.

## Funktionsbeschreibung

Ziel der Funktion ist das Extrahieren des Schema,- Tabellen- oder Spaltennamens aus dem vollständigen Spaltennamen. Der vollständige Spaltenname besteht aus dem Schemanamen, dem Tabellennamen und dem Spaltennamen welche mittels Punkt getrennt sind und aus Buchstaben a-z bzw. A-Z, Ziffern 0-9 bzw. Unterstrich bestehen.

Im Fehlerfall wird eine Leerzeichenkette zurückgegeben.

## Beispiele

-- Fehlermeldung. Es ist ein vollständiger Spaltenname anzugeben

```
SELECT * FROM intern.gib__stsname( 'sp', NULL );
```

```
SELECT * FROM intern.gib__stsname( 'sp', '' );
```

-- Fehlermeldung. Es ist Befehl anzugeben

```
SELECT * FROM intern.gib__stsname( NULL, 'intern.alle.notizen' );
```

```
SELECT * FROM intern.gib__stsname( '', 'intern.alle.notizen' );
```

-- Fehlermeldung. Der vollständige Spaltenname ist in der Syntax schema.tabelle.spalte anzugeben

```
SELECT * FROM intern.gib__stsname( 'sch', 'i.a' );
```

```
SELECT * FROM intern.gib__stsname( 'sch', '.a.n' );
```

```
SELECT * FROM intern.gib__stsname( 'sch', 'i..n' );
```

```
SELECT * FROM intern.gib__stsname( 'sch', 'i.a.' );
```

```
SELECT * FROM intern.gib__stsname( 'sch', '..' );
```

```
SELECT * FROM intern.gib__stsname( 'sch', 'iβ.a.n' );
```

```
SELECT * FROM intern.gib__stsname( 'sch', 'i.a2.n' );
```

```
SELECT * FROM intern.gib__stsname( 'sch', 'i.a.n,' );
```

```

SELECT * FROM intern.gib__stsname( 'sch', 'i.a.n.falsch' );

-- Fehlermeldung. Der übergebene befehl ist falsch
SELECT * FROM intern.gib__stsname( 'unbekannterBefehl', 'intern.alle.notizen' );

-- Schema, Tabelle, Spalte, ... zurückgeben
SELECT * FROM intern.gib__stsname( 'sch', 'intern.alle.notizen' );
SELECT * FROM intern.gib__stsname( 'tab', 'intern.alle.notizen' );
SELECT * FROM intern.gib__stsname( 'sp', 'intern.alle.notizen' );
SELECT * FROM intern.gib__stsname( 'schartab', 'intern.alle.notizen' );
SELECT * FROM intern.gib__stsname( 'tabsp', 'intern.alle.notizen' );
SELECT * FROM intern.gib__stsname( 'tabsp', 'int2ern.al.le.notizen85' );

```

## 4.13 intern.gib\_\_zu

### Funktionsname

intern.gib\_\_zu( OUT zu bpchar )

### Funktionsparameter/Rückgabewert(e)

zu Das Ersatzzeichen für den Zeilenumbruch

### Funktionsbeschreibung

Ziel der Funktion ist das Zurückgeben des im System zugelassenen Ersatzzeichens für den Zeilenumbruch.

Alle Texte, welche ein Zeilenumbruchzeichen enthalten, sollen mit dem hier zurückgegebenen Zeichen erstellt (gespeichert werden). Das Bearbeiten der Datenbankdumps oder der Texte, welche aus der Datenbank ausgelesen werden, wird vereinfacht.

### Beispiele

```

-- Es wird das Ersatzzeichen für den Zeilenumbruch zurückgegeben
SELECT * FROM intern.gib__ZU();

```

## 4.14 intern.ist\_\_abczk

### Funktionsname

intern.ist\_\_abczk( IN abczk varchar, OUT istabczk bool )

### Funktionsparameter/Rückgabewert(e)

abczk Es ist die zu testende Zeichenkette anzugeben.

istabczk Wenn eine AbcZK vorliegt wird true zurückgegeben, sonst false.

## Funktionsbeschreibung

Ziel der Funktion ist das Prüfen, ob die übergebene Zeichenkette `abczk` als Zeichenkette interpretiert werden kann, welche ausschließlich aus Buchstaben besteht (AbcZK). Die gültigen Buchstaben können `intern.gib__ERE_ABCZK()` entnommen werden.

## Beispiele

```
-- Es wird false zurückgegeben
SELECT * FROM intern.ist__abczk( '' );
SELECT * FROM intern.ist__abczk( 'ein-/auslesen' );
SELECT * FROM intern.ist__abczk( 'ß' );
SELECT * FROM intern.ist__abczk( 'ßbeginnt' );

-- Es wird true zurückgegeben
SELECT * FROM intern.ist__abczk( 'I' );
SELECT * FROM intern.ist__abczk( 'daß' );
```

## 4.15 intern.loesche\_\_zeile

### Funktionsname

```
intern.loesche__zeile( IN tabname varchar, IN sql_where varchar, IN beachteri varchar
)
```

### Funktionsparameter/Rückgabewert(e)

`tabname` Name einschliesslich Schemaname der Tabelle, aus der die Zeile gelöscht werden soll (darf nicht leer sein)

`sql_where` SQL-Bedingung welche die zu löschende(n) Zeile(n) beschreibt (darf leer sein)

`beachteri` Es ist `true` anzugeben, wenn die referentielle Integrität beachtet werden soll. Sonst ist `false` anzugeben.

### Funktionsbeschreibung

Ziel der Funktion ist das Löschen von Zeilen (außer der Zeile mit `id = 0`) unter Berücksichtigung der Bedingungen, welche mittels `sql_where` spezifiziert sind. Wenn `sql_where` leer ist, werden alle Zeilen gelöscht.

Es ist besonders zu beachten, daß `tabName` den Schemanamen enthalten muss !

## Beispiele

```
-- Fehlermeldung. Der Tabellename ist anzugeben
SELECT intern.loesche__zeile( NULL, '' );

-- Fehlermeldung. Die übergebene Tabelle ist nicht vorhanden
SELECT intern.loesche__zeile( 'intern.abc', '' );
```

```

-- Fehlermeldung. Die Zeilen können nicht gelöscht werden, da weitere Zeilen abhängen.
SELECT intern.loesche_zeile( 'intern.abczk', '' );

-- Fehlermeldung. Die übergebe WHERE - Bedingung ist ungültig
SELECT intern.loesche_zeile( 'intern.abczk', 'intern.abczk.id = 99999999' );

-- Tabelleninhalt anzeigen
SELECT intern.nimm_zeile( 'tl.datei', ARRAY['dname'], ARRAY['Dateiname-1'] );
SELECT intern.nimm_zeile( 'tl.datei', ARRAY['dname'], ARRAY['Dateiname-2'] );
SELECT * FROM tl.datei WHERE id < 5;

-- Gesamten Inhalt der Tabelle löschen
SELECT intern.loesche_zeile( 'tl.datei', NULL );

-- Zeile mit der id = 1 löschen
SELECT intern.loesche_zeile( 'tl.datei', 'tl.datei.id = 1' );

-- Tabelleninhalt anzeigen
SELECT * FROM tl.datei WHERE id < 5;

```

## 4.16 intern.nimm\_tlkomentar

### Funktionsname

```
intern.nimm_tlkomentar( IN sqldbobj varchar, IN kommentar varchar )
```

### Funktionsparameter/Rückgabewert(e)

`sqldbobj` Es ist das Datenbankobjekt zu benennen (SQL) welchem der Kommentar hinzugefügt werden soll.

`kommentar` Der Kommentar (einschl. TL-Befehlen). `'\'` sind zu maskieren !

### Funktionsbeschreibung

Ziel der Funktion ist das Hinzufügen eines Kommentares zu einem Datenbankobjekt. Im Kommentar können TL-Befehle verwendet werden, da die Dokumentation zur Datenbank mittels  $\text{\TeX}/\text{\LaTeX}$  formatiert wird.

Zeilenumbrüche werden mittels `intern.gib_ZU()` ersetzt.

### Beispiele

```

-- Fehlermeldung. Der Parameter sqlDBObj darf nicht leer oder NULL sein.
SELECT intern.nimm_TLKommentar( '', '' );
SELECT intern.nimm_TLKommentar( NULL, '' );

-- Kommentar gelöscht
SELECT intern.nimm_TLKommentar( 'DATABASE ' || current_database(), '' );

```

```

-- Kommentar hinzugefügt
SELECT intern.nimm__TLKommentar( 'DATABASE ' || current_database(), 'Neuer Kommentar'
);

-- prüfen, ob Kommentar tatsächlich hinzugefügt wurde
SELECT shobj_description( ( SELECT oid FROM pg_database WHERE pg_database.datname
= current_database() LIMIT 1 ), 'pg_database');

```

## 4.17 intern.nimm\_\_tlkommentarfunktion

### Funktionsname

```
intern.nimm__tlkommentarfunktion( IN fkt varchar, IN fktparam varchar[], IN beschr
varchar, IN beispiel varchar[] )
```

### Funktionsparameter/Rückgabewert(e)

**fkt** Es ist der Name der Funktion anzugeben, der der Kommentar zugeordnet werden soll. Er darf nicht leer oder NULL sein.

**fktparam** In Form eines ARRAY sind in der Reihenfolge ihres Auftretens sämtliche Beschreibungen der Funktionsparameter anzugeben. Es wird nicht zwischen Eingabeparametern IN bzw. Rückgabeparametern OUT / INOUT unterschieden. Sollte die Funktion über keine Parameter verfügen, ist NULL anzugeben.

**beschr** Es ist die Beschreibung der Funktion einzugeben. Sie darf nicht leer oder NULL sein.

**beispiel** Es können mehrere Beispiele zur Demonstration der Nutzung der Funktion angegeben werden. Sie dürfen NULL sein.

### Funktionsbeschreibung

Ziel der Funktion ist das Hinzufügen einer Beschreibung (eines Kommentars) zu einer Funktion. Hinweise zur Gültigkeit können `intern.nimm__TLKommentar(...)` entnommen werden.

### Beispiele

```

-- Fehlermeldung. Die Parameter fkt und beschr dürfen nicht NULL sein.
SELECT intern.nimm__TLKommentarFunktion( NULL, NULL, '', NULL );
SELECT intern.nimm__TLKommentarFunktion( 'intern.nimm__TLKommentarFunktion', NULL,
'', NULL );

```

```
-- Kommentar einlesen
```

```

SELECT intern.nimm__TLKommentarFunktion( 'intern.nimm__TLKommentarFunktion( fkt
varchar, fktparam varchar[], beschr varchar, beispiel varchar[] )', NULL, 'Testbeschre
NULL );
\df+ intern.nimm__tlkommentarfunktion

```

## 4.18 intern.nimm\_\_tlkommentarsicht

### Funktionsname

intern.nimm\_\_tlkommentarsicht( IN sicht varchar, IN spnamen varchar[], IN spbeschr varchar[], IN beschr varchar )

### Funktionsparameter/Rückgabewert(e)

sicht Es ist der vollständige Name der Sicht anzugeben. Er darf nicht leer oder NULL sein.

spnamen In Form eines ARRAY sind in der Reihenfolge ihres Auftretens sämtliche Namen der Spalten anzugeben, welche von der Sicht bereitgestellt werden. Es muß mindestens eine Spalte beschrieben werden.

spbeschr In Form eines ARRAY sind in der Reihenfolge ihres Auftretens sämtliche Beschreibungen der Spalten anzugeben, welche von der Sicht bereitgestellt werden. Es muß mindestens eine Spalte beschrieben werden.

beschr Es ist die Sicht zu beschreiben. Es muß eine Beschreibung angegeben werden.

### Funktionsbeschreibung

Ziel der Funktion ist das Hinzufügen einer Beschreibung (eines Kommentars) zu einer Sicht. Hinweise zur Gültigkeit können intern.nimm\_\_TLKommentar(...) entnommen werden.

### Beispiele

-- Fehlermeldung. Der Name der Sicht fehlt

```
SELECT intern.nimm__TLKommentarSicht( NULL, NULL, NULL, '' );
```

-- Fehlermeldung. Der Kommentar fehlt

```
SELECT intern.nimm__TLKommentarSicht( 'intern.zeige__fremdschluessel', ARRAY['spName'], ARRAY['spBeschreibung'], '' );
```

-- Fehlermeldung. Es ist mindestens eine Spalte zu beschreiben

```
SELECT intern.nimm__TLKommentarSicht( 'intern.zeige__fremdschluessel', NULL, ARRAY['spBeschreibung', 'Kommentar' ] );
```

```
SELECT intern.nimm__TLKommentarSicht( 'intern.zeige__fremdschluessel', ARRAY['spName'], NULL, 'Kommentar' );
```

-- Fehlermeldung. Die Anzahl der Spaltennamen muß mit der Anzahl der Spaltenbeschreibungen übereinstimmen

```
SELECT intern.nimm__TLKommentarSicht( 'intern.zeige__fremdschluessel', ARRAY['spName1', 'spName2'], ARRAY['spBeschreibung'], 'Kommentar' );
```

-- Sicht kommentieren, Ergebnis überprüfen

```
SELECT intern.nimm__TLKommentarSicht( 'intern.zeige__fremdschluessel', ARRAY['SpName'], ARRAY['SpBeschreibung'], 'Beschreibung' );
```

```
\dv+ intern.zeige__fremdschluessel
```

## 4.19 intern.nimm\_tlkomentarsts

### Funktionsname

intern.nimm\_tlkomentarsts( IN sts varchar, IN kommentar varchar )

### Funktionsparameter/Rückgabewert(e)

sts Es ist der vollständige Schemaname, Tabellenname oder Spaltenname anzugeben.

kommentar Es ist der Kommentar anzugeben

### Funktionsbeschreibung

Ziel der Funktion ist das Hinzufügen eines Kommentares zu einem Schema, einer Tabelle oder einer Spalte. Es ist der vollständige Name anzugeben (Bsp.: `schema.tabelle.spalte`). Anhand der Anzahl der Datenbankobjekte wird entschieden, ob ein Schema, eine Tabelle oder eine Spalte vorliegt.

Hinweise zur Gültigkeit können `intern.nimm_TLKommentar(...)` entnommen werden.

### Beispiele

```
-- Fehlermeldung. Der Parameter sts darf nicht NULL sein.
```

```
SELECT intern.nimm_TLKommentarSTS( '', '' );
```

```
SELECT intern.nimm_TLKommentarSTS( NULL, '' );
```

```
-- Kommentar gelöscht
```

```
SELECT intern.nimm_TLKommentarSTS( 'intern.alle.id', '' );
```

```
-- Kommentar hinzugefügt
```

```
SELECT intern.nimm_TLKommentarSTS( 'intern.alle.id', 'Neuer Kommentar' );
```

```
\d+ intern.alle
```

## 4.20 intern.nimm\_zeile

### Funktionsname

intern.nimm\_zeile( IN tabname varchar, IN liste\_spname varchar[], IN liste\_werte varchar[], OUT tabnameid varchar )

### Funktionsparameter/Rückgabewert(e)

tabname Name der Tabelle (ggf. einschliesslich Schemaname), in die die Werte `liste_werte[]` eingelesen werden sollen (darf nicht leer sein).

liste\_spname Liste der Spalten, welche die Werte aufnehmen sollen. Es muß mindestens ein Spaltenname übergeben werden. Es dürfen keine Leerzeichenketten übergeben werden.

liste\_werte Liste der Werte, welche eingefügt werden sollen. Es muß mindestens ein Wert übergeben werden. Es dürfen keine Leerzeichenketten übergeben werden. Weiterhin muß dieselbe Anzahl Werte übergeben werden, wie Spalten benannt sind.

tablenameid id der Zeile, welche die übergebenen Werte aufgenommen hat

### Funktionsbeschreibung

Ziel der Funktion ist das Einlesen von Werten. Der Spaltenliste `liste_spName[]` darf jede in der Tabelle `tabName` vorhandene Spalte übergeben werden. Die Spalte `idfrei` darf nicht übergeben werden, da sie ausschließlich vom System gesetzt wird.

Die übergebenen Werte `liste_werte[]` werden in der DB gesucht und die `id` der Zeile zurückgegeben, welche die Werte bereits enthält. Sollte keine Zeile gefunden werden, wird die nächste verfügbare `id` ermittelt und die Werte eingelesen.

Sollte `liste_spName[]` die Spalte `id` explizit benannt sein, so wird unabhängig vom Wert der Spalte `idfrei` die übergebene `id` gesucht, der gesamte Inhalt gelöscht und die neuen Werte eingetragen. Wurde `id` nicht gefunden, wird eine Zeile mit der übergebenen `id` hinzugefügt.

### Beispiele

-- Fehlermeldung. Der Parameter `tabName` darf nicht NULL oder leer sein.

```
SELECT intern.nimm_zeile( NULL, ARRAY[''], ARRAY[''] );
```

-- Fehlermeldung. Die Parameter `liste_spName` und `liste_werte` müssen mindestens ein Element enthalten.

```
SELECT intern.nimm_zeile( 'intern.abczk', NULL, NULL );
```

```
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], NULL );
```

-- Fehlermeldung. Die Parameter `liste_spName` und `liste_werte` dürfen keine Leerzeichenketten enthalten.

```
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY[''], ARRAY[''] );
```

```
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY[''] );
```

-- Fehlermeldung. Der Parameter `liste_wert` muß dieselbe Anzahl Elemente enthalten, wie der Parameter `liste_spName`.

```
SELECT intern.nimm_zeile( 'intern.abczk', 'notizen, abczk', 'zeichenkette für notizen' );
```

-- Fehlermeldung. Die Spalte `idfrei` darf nicht explizit gesetzt werden

```
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['idfrei'], ARRAY['0'] );
```

-- Der Wert war bereits in der Datenbank enthalten

```
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWort'] );
```

```
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
```

```
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWort'] );
```

```
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
```

-- Wiederverwendung einer gelöschten `id`

```
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWort'] );
```



```

SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWortZwe
');
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
SELECT intern.loesche_zeile( 'intern.abczk', 'abczk = ''KeinRichtigesWort'' ');
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWortEin
');
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;

-- Überschreiben einer Zeile mit vorgegebener id
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWort']
);
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['abczk'], ARRAY['KeinRichtigesWortZwe
');
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['id', 'abczk'], ARRAY['2399',
'KeinRichtigesWortEins'] );
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;

-- Einfügen einer Zeile mit vorgegebener id
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;
SELECT intern.nimm_zeile( 'intern.abczk', ARRAY['id', 'abczk'], ARRAY['999999',
'NichtVorhandeneId'] );
SELECT * FROM intern.abczk ORDER BY id DESC LIMIT 5;

```

## 4.21 intern.pruefe\_\_array

### Funktionsname

intern.pruefe\_\_array( IN liste varchar[], IN minel int4, IN maxel int4, IN fktname varchar, IN paramname varchar )

### Funktionsparameter/Rückgabewert(e)

**liste** Es ist die Liste anzugeben, deren Plausibilität geprüft werden soll.

**minel** Es ist die Anzahl der Listenelemente anzugeben, welche mindestens übergeben werden sollen. Wenn kleiner 1 oder NULL übergeben wird, muß die Liste NULL sein.

**maxel** Es ist die Anzahl der Listenelemente anzugeben, welche maximal übergeben werden sollen. Wenn kleiner 1 oder NULL übergeben wird, darf die Liste eine beliebige Anzahl Elemente besitzen. Andernfalls darf die Anzahl der Elemente den übergebenen Wert nicht überschreiten.

**fktname** Es ist der Name der Funktion anzugeben, welche die Prüfroutine aufruft. Der Funktionsname wird in der Fehlermeldung verwendet. Sollte er leer oder NULL sein, wird intern.pruefe\_\_array(...) verwendet.

**paramname** Es ist der Name des Parameters anzugeben, welcher die zu prüfende Liste **liste** beinhaltet. Der Parametername wird in der Fehlermeldung verwendet. Sollte er leer oder NULL sein, wird **liste** verwendet.

### Funktionsbeschreibung

Ziel der Funktion ist die Plausibilitätsprüfung der mittels des Parameters **liste** übergebenen Liste (ARRAY) anhand der Randbedingungen **minel** und **maxel**. Wenn die Liste nicht NULL sein darf, darf keines der Listenelemente leer sein. Sollte die Plausibilitätsprüfung negativ ausfallen, wird eine Fehlermeldung ausgegeben.

### Beispiele

Fehlermeldung mit Standardfunktions- und Parameternamen ausgeben, wenn **liste** nicht NULL sein darf.

```
SELECT * FROM intern.pruefe__array( NULL, 1, 1, NULL, NULL );  
SELECT * FROM intern.pruefe__array( NULL, 1, 1, '', '' );
```

Fehlermeldung mit übergebenen Funktions- und Parameternamen ausgeben, wenn **liste** nicht NULL sein darf.

```
SELECT * FROM intern.pruefe__array( NULL, 1, 1, 'fktname(...)', 'listenName' );
```

Fehlermeldung mit Standardfunktions- und Parameternamen ausgeben, wenn **liste** NULL sein muß.

```
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', 'zwei' ], NULL, 1, NULL, NULL );  
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', 'zwei' ], 0, 1, '', '' );  
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', 'zwei' ], -5, 1, NULL, NULL );
```

Fehlermeldung mit übergebenen Funktions- und Parameternamen ausgeben, wenn **liste** NULL sein muß.

```
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', 'zwei' ], NULL, 1, 'fktname(...)', 'listenName' );
```

Fehlermeldung ausgeben, wenn **liste** mindesten 2 Elemente enthalten muß.

```
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins' ], 2, 0, NULL, NULL );  
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins' ], 2, NULL, 'fktname(...)', 'listenName' );
```

Fehlermeldung ausgeben, wenn **liste** maximal 1 Element enthalten darf.

```
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', 'zwei' ], 1, 1, '', '' );  
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', 'zwei' ], 1, 1, 'fktname(...)', 'listenName' );
```

Fehlermeldung ausgeben, wenn **liste** maximal 1 Element enthalten darf.

```
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', '', 'drei' ], 1, 3, NULL, NULL );  
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', '', 'drei' ], 1, 3, '', '' );
```

```
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', '', 'drei' ], 1, 3, 'fktname(...)', 'listenName' );
```

keine Fehlermeldung ausgeben, weil liste NULL sein darf.

```
SELECT * FROM intern.pruefe__array( NULL, NULL, 1, 'fktname(...)', 'listenName' );
SELECT * FROM intern.pruefe__array( NULL, 0, 1, 'fktname(...)', 'listenName' );
SELECT * FROM intern.pruefe__array( NULL, -5, 1, 'fktname(...)', 'listenName' );
```

keine Fehlermeldung ausgeben, weil liste minimal 1 und maximal 2 Element enthalten darf.

```
SELECT * FROM intern.pruefe__array( ARRAY[ 'eins', 'zwei' ], 1, 2, 'fktname(...)', 'listenName' );
```

## 4.22 intern.pruefe\_\_dt

### Funktionsname

```
intern.pruefe__dt( IN dt varchar, IN wert varchar, IN ug varchar, IN og varchar,
IN fktname varchar, IN paramname varchar )
```

### Funktionsparameter/Rückgabewert(e)

**dt** Es ist einer der möglichen Datentypen (siehe Funktionsbeschreibung — Gültige Datentypen / Wertebereich) anzugeben

**wert** Es ist der Wert anzugeben, dessen Plausibilität geprüft werden soll. Er darf nicht NULL oder leer sein.

**ug** Es ist die untere Grenze des Wertes anzugeben (weitere Hinweise siehe Funktionsbeschreibung — Gültige Datentypen / Wertebereich). Wird NULL oder eine Leerzeichenkette angegeben, gilt die in PostgreSQL implementierte untere Grenze.

**og** Es ist die obere Grenze des Wertes anzugeben. Wird NULL oder eine Leerzeichenkette angegeben, gilt die in PostgreSQL implementierte obere Grenze.

**fktname** Es ist der Name der Funktion anzugeben, welche die Prüfroutine aufruft. Der Funktionsname wird in der Fehlermeldung verwendet. Sollte er leer oder NULL sein, wird `intern.pruefe__dt(...)` verwendet.

**paramname** Es ist der Name des Parameters anzugeben, welcher den zu prüfenden Wert `wert` beinhaltet. Der Parametername wird in der Fehlermeldung verwendet. Sollte er leer oder NULL sein, wird `wert` verwendet.

### Funktionsbeschreibung

Ziel der Funktion ist die Plausibilitätsprüfung elementarer Datentypen. Die Datentypen und der Wertebereich werden als Zeichenketten übergeben. Sollte die Plausibilitätsprüfung negativ

ausfallen, wird eine Fehlermeldung ausgegeben.

Der im PostgreSQL implementierte Wertebereich der Datentypen kann der Postgres-Dokumentation entnommen werden. Er wird von PostgreSQL automatisch und somit innerhalb dieser Rputine nicht explizit überprüft.

Gültige Datentypen / Wertebereich

Es werden die folgenden Datentypen geprüft:

**bw** Boolescher Wert. Die untere / obere Grenze wird nicht überprüft.

**du** Datum / Uhrzeit. Die untere / obere Grenze wird überprüft.

**gz** Ganzzahl (integer). Die untere / obere Grenze wird überprüft.

**zk** Zeichenkette. Die untere Grenze wird nicht überprüft. Die obere Grenze wird überprüft.

## Beispiele

Fehlermeldung. Der übergebene Datentyp darf nicht NULL oder leer sein

```
SELECT * FROM intern.pruefe_dt( NULL, '1', NULL, NULL, NULL, NULL );
```

Fehlermeldung. Der übergebene Datentyp ist unbekannt

```
SELECT * FROM intern.pruefe_dt( 'unbekannt', '1', NULL, NULL, 'fktname(...)',  
'wertName' );
```

Fehlermeldung: Es ist kein boolean

```
SELECT * FROM intern.pruefe_dt( 'bw', 'keinBool', 'ignoriert', 'ignoriert', NULL,  
NULL );
```

```
SELECT * FROM intern.pruefe_dt( 'bw', 'keinBool', 'ignoriert', 'ignoriert', 'fktname(...)',  
'wertName' );
```

Fehlermeldung. Der übergebene Wert darf nicht NULL oder leer sein

```
SELECT * FROM intern.pruefe_dt( NULL, NULL, NULL, NULL, NULL, NULL );
```

```
SELECT * FROM intern.pruefe_dt( NULL, '', NULL, NULL, 'fktname(...)', 'wertName'  
);
```

Fehlermeldung. Das Datum befindet sich außerhalb der zulässigen Grenzen

```
SELECT * FROM intern.pruefe_dt( 'du', '2007-04-16', '2007-04-17', NULL, 'fktname(...)',  
'datUhrz' );
```

```
SELECT * FROM intern.pruefe_dt( 'du', '2007-04-18', NULL, '2007-04-17', 'fktname(...)',  
'datUhrz' );
```

```
SELECT * FROM intern.pruefe_dt( 'du', '2007-04-19', '2007-04-17', '2007-04-18',  
'fktname(...)', 'datUhrz' );
```

Fehlermeldung. Die Ganzzahl (integer) befindet sich außerhalb der zulässigen Grenzen

```
SELECT * FROM intern.pruefe_dt( 'gz', '-2', '-1', NULL, 'fktname(...)', 'gz_int'  
);
```

```

SELECT * FROM intern.pruefe_dt( 'gz', '2', NULL, '1', 'fktname(...)', 'gz_int'
);
SELECT * FROM intern.pruefe_dt( 'gz', '-5', '-1', '1', 'fktname(...)', 'gz_int'
);

```

Fehlermeldung. Die übergebene Zeichenkette ist zu lang

```

SELECT * FROM intern.pruefe_dt( 'zk', '1234567890', NULL, '9', NULL, NULL );
SELECT * FROM intern.pruefe_dt( 'zk', '1234567890', NULL, '9', 'fktname(...)',
'zkWert' );

```

keine Fehlermeldung

```

SELECT * FROM intern.pruefe_dt( 'bw', 'TRUE', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', 'T', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', 'Y', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', 'YES', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', '1', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', 'FALSE', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', 'F', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', 'N', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', 'NO', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'bw', '0', NULL, NULL, 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'du', '2007-04-18', NULL, NULL, 'fktname(...)',
'wertName' );
SELECT * FROM intern.pruefe_dt( 'du', '2007-04-18', '2007-04-17', '2007-04-19',
'fktname(...)', 'wertName' );
SELECT * FROM intern.pruefe_dt( 'gz', '2123456', NULL, NULL, 'fktname(...)', 'wertNam
);
SELECT * FROM intern.pruefe_dt( 'gz', '0', '-1', '1', 'fktname(...)', 'wertName'
);
SELECT * FROM intern.pruefe_dt( 'zk', '123456789', 'ignoriert', '9', 'fktname(...)',
'wertName' );

```

## 4.23 intern.verbindezk

### Funktionsname

intern.verbindezk( IN zk varchar, IN zkadd varchar, OUT zkges varchar )

### Funktionsparameter/Rückgabewert(e)

zk Erste Zeichenkette.

zkadd Zweite Zeichenkette.

zkges Erste + Zweite Zeichenkette.

### Funktionsbeschreibung

Ziel der Funktion ist das Verbinden beider Zeichenketten. Diese Funktion wird in der Aggregatfunktion `intern.erzeuge_zk(...)` verwendet.

### Beispiele

```
intern.verbindezk( 'Anfang ', '- Ende' );
```

## 4.24 intern.zeige\_fremdschlüssel

### Beschreibung der Spalten

fk Namen der Fremdschlüssel in der Datenbank

spstart Schema.Tabelle.Name der Startspalte des Fremdschlüssels

spziel Schema.Tabelle.Name der Zielspalte des Fremdschlüssels

### Beschreibung der Sicht

Es werden alle im System vorhandenen Fremdschlüssel einschließlich der Start- und Zielspalten aufgelistet.

## 4.25 intern.zeige\_spaltenanzahl

### Beschreibung der Spalten

tabname Tabellenname

spanz Anzahl der Spalten der Tabelle

### Beschreibung der Sicht

Es wird die Anzahl der in der Tabelle enthaltenen Spalten (ohne Systemspalten) aufgelistet.

## 4.26 intern.zeige\_tabsinformationen

### Beschreibung der Spalten

`sname` Schema.Tabelle.Name der Spalte

`dtyp` pg\_type.typname

`spnr` pg\_attribute.attnum

`beschr_spalte` pg\_description.description für die Spalte

`beschr_tabelle` pg\_description.description für die Tabelle

### Beschreibung der Sicht

Es werden alle im System vorhandenen Spalten (einschließlich Schema- und Tabellennamen) und deren Beschreibung aufgelistet.

## 5 prj

Ausschließlich im Schema „Projekte“ wird der Projektbezug für alle weiteren Schemas hergestellt. Die Definition des Begriffes „Projekt“ hängt selbstverständlich von der Anwendung ab.

In der vorliegenden Datenbank kann mittels des Projektes „DBDokumentation“ die Dokumentation der Datenbank aus den Kommentaren der verschiedenen Datenbankobjekte erzeugt werden.

Dazu sind in der Kommandozeile (Eingabeaufforderung engl. Shell) die folgenden Befehle einzugeben:

1. `psql -U micha vf -P tuples_only -P format=unaligned -P footer -c ‘‘SELECT replace( makefile, '-U micha vf', '-U ' || current_user || ' ' || current_database() ) FROM prj.projekte WHERE prjname = 'DBDokumentation';‘‘ -o makefile`  
Anstelle des Benutzers `micha` im shell-Befehl `psql -U micha vf ...` und des Datenbanknamens `vf` ist der eigene Benutzername bzw. Datenbankname anzugeben
2. `make`

Das genannte Verfahren kann für sämtliche Projekte, denen ein `makefile` zugeordnet ist, angewendet werden. Mögliche Varianten zum Erzeugen unterschiedlicher Dokumente können dem `makefile` des entsprechenden Projektes entnommen werden.

### Benennen der Tabellen

`prj.p_*` Projektbezug herstellen

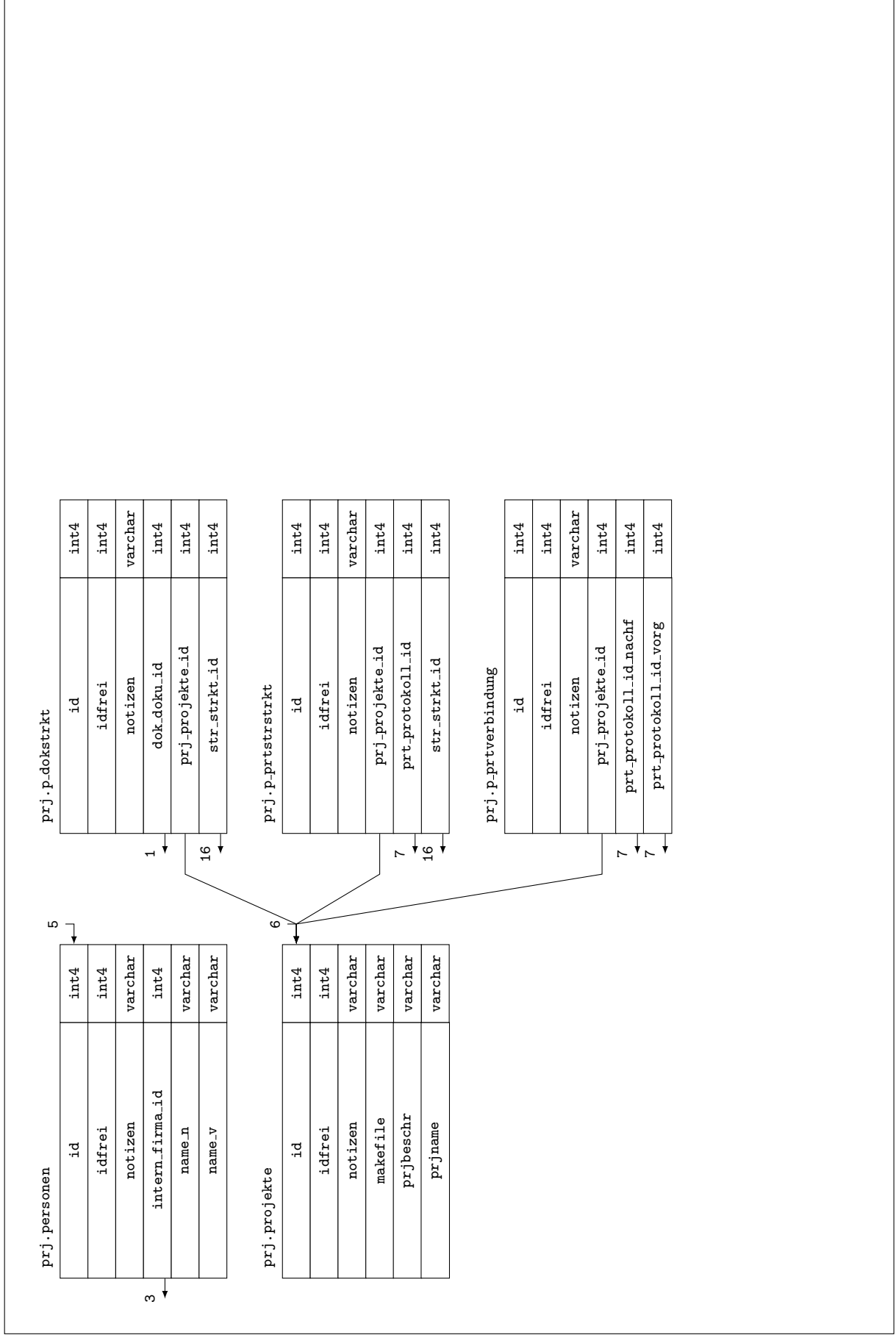
`prj.pp_*` projektbezogene Personenzuordnung

`prj.pl_*` projektbezogene Logbuchfunktionalität

`prj.pplu_*` projektbezogene Logbuchfunktionalität und Personenzuordnung



## 5.1 Tabellenhierarchie der Ebenen 1 — 3



## 5.2 Tabellenhierarchie der Ebenen 1 — 3

prj.p-psabk

id	int4
idfrei	int4
notizen	varchar
prj_projekte_id	int4
ps_abk_id	int4
spr_nebvar_id	int4
spr_nebvar_id_1	int4
spr_nebvar_id_zus	int4

6  
9  
13  
13  
13

prj.p-strverbindung

id	int4
idfrei	int4
notizen	varchar
fbtor	varchar
prj_projekte_id	int4
str_strkt_id_nachf	int4
str_strkt_id_vorg	int4

6  
16  
16

### 5.3 Tabellenhierarchie der Ebenen 1 — 3

id	int4
idfrei	int4
notizen	varchar
logdatuhrz	timestamp
logdqok	bool
logdquelle	varchar
logentfernt	bool
logname	varchar
datum	date
dok.doku_id	int4
istfrei	bool
prj_projekte_id	int4
stand	varchar

1 ↓

6 ↓

## 5.4 Tabellenhierarchie der Ebenen 1 — 3

prj.pl\_prttdokdoku

	id	int4
	idfrei	int4
	notizen	varchar
	logdatuhrz	timestamp
	logdqok	bool
	logdquelle	varchar
	logentfernt	bool
	logname	varchar
1	dok.doku_id	int4
	nr	varchar
6	prj.projekte_id	int4
7	prt.protokoll_id	int4
13	spr.uebvar_id	int4
13	spr.uebvar_id_zus	int4

prj.pl\_prtpsgefszenario

	id	int4
	idfrei	int4
	notizen	varchar
	logdatuhrz	timestamp
	logdqok	bool
	logdquelle	varchar
	logentfernt	bool
	logname	varchar
6	prj.projekte_id	int4
7	prt.protokoll_id	int4
11	ps.gefszenario_id	int4

## 5.5 Tabellenhierarchie der Ebenen 1 — 3

prj.pl.psgfabk

id	int4
idfrei	int4
notizen	varchar
logdatuhrz	timestamp
logdqok	bool
logdquelle	varchar
logentfernt	bool
logname	varchar
prj_projekte_id	int4
ps_abk_id	int4
ps_gefszenario_id	int4
spr_uebvar_id_zus	int4

6  
9  
11  
13

prj.pl.psgfaunm

id	int4
idfrei	int4
notizen	varchar
logdatuhrz	timestamp
logdqok	bool
logdquelle	varchar
logentfernt	bool
logname	varchar
prj_projekte_id	int4
ps_anmerkung_id	int4
ps_gefszenario_id	int4
spr_uebvar_id	int4

6  
10  
11  
13

## 5.6 Tabellenhierarchie der Ebenen 1 — 3

prj.pl\_psegefstrkt

id	int4
idfrei	int4
notizen	varchar
logdatuhrz	timestamp
logdqok	bool
logdquelle	varchar
logentfernt	bool
logname	varchar
prj_projekte_id	int4
ps_gefszenario_id	int4
str_strkt_id	int4

6  
11  
16

prj.pp\_prtprotokoll

id	int4
idfrei	int4
notizen	varchar
datuhrz	timestamp
duanfrage	timestamp
erledigt	bool
intern	bool
prj_personen_id	int4
prj_projekte_id	int4
prt_protokoll_id	int4
spr_uebvar_id	int4
teilnehmer	bool
termin	timestamp

5  
6  
7  
13

## 5.7 Tabellenhierarchie der Ebenen 1 — 3

prj.pp.psgfszenario

id	int4
idfrei	int4
notizen	varchar
prj_personen_id	int4
prj_projekte_id	int4
ps_gfszenario_id	int4
spr_uebvar_id	int4
spr_uebvar_id_reprgfszen	int4
spr_uebvar_id_ursp	int4

5 ↓  
6 ↓  
11 ↓  
13 ↓  
13 ↓  
13 ↓

prj.pp.strstrkt

id	int4
idfrei	int4
notizen	varchar
prj_personen_id	int4
prj_projekte_id	int4
str_strkt_id	int4
fbereignis	varchar
spr_uebvar_id	int4

5 ↓  
6 ↓  
16 ↓  
13 ↓

## 5.8 Tabellenhierarchie der Ebenen 1 — 3

prj.ppl.dokdoku

id	int4
idfrei	int4
notizen	varchar
logdatuhrz	timestamp
logdqok	bool
logdquelle	varchar
logentfernt	bool
logname	varchar
dok.doku_id	int4
prj_personen_id	int4
prj_projekte_id	int4

1  
5  
6



## 5.9 Schlüssel und Bedingungen

### Primärschlüssel

pk\_\_prj\_p\_dokstrkt\_id für Spalte prj.p\_dokstrkt.id  
pk\_\_prj\_p\_prtstrstrkt\_id für Spalte prj.p\_prtstrstrkt.id  
pk\_\_prj\_p\_prtverbindung\_id für Spalte prj.p\_prtverbindung.id  
pk\_\_prj\_p\_psabk\_id für Spalte prj.p\_psabk.id  
pk\_\_prj\_p\_strverbindung\_id für Spalte prj.p\_strverbindung.id  
pk\_\_prj\_personen\_id für Spalte prj.personen.id  
pk\_\_prj\_pl\_dokausgabe\_id für Spalte prj.pl\_dokausgabe.id  
pk\_\_prj\_pl\_prtdokdoku\_id für Spalte prj.pl\_prtdokdoku.id  
pk\_\_prj\_pl\_prtpsgefszenario\_id für Spalte prj.pl\_prtpsgefszenario.id  
pk\_\_prj\_pl\_psgefabk\_id für Spalte prj.pl\_psgefabk.id  
pk\_\_prj\_pl\_psgefstrkt\_id für Spalte prj.pl\_psgefstrkt.id  
pk\_\_prj\_pp\_protokoll\_id für Spalte prj.pp\_prtprotokoll.id  
pk\_\_prj\_pp\_psgefszenario\_id für Spalte prj.pp\_psgefszenario.id  
pk\_\_prj\_pp\_strstrkt\_id für Spalte prj.pp\_strstrkt.id  
pk\_\_prj\_ppl\_dokdoku\_id für Spalte prj.ppl\_dokdoku.id  
pk\_\_prj\_projekte\_id für Spalte prj.projekte.id

### Fremdschlüssel

fk\_\_prj\_p\_dokstrkt\_\_dok\_doku\_id aus Spalte prj.p\_dokstrkt.dok\_doku\_id  
fk\_\_prj\_p\_dokstrkt\_\_prj\_projekte\_id aus Spalte prj.p\_dokstrkt.prj\_projekte\_id  
fk\_\_prj\_p\_dokstrkt\_\_str\_strkt\_id aus Spalte prj.p\_dokstrkt.str\_strkt\_id  
fk\_\_prj\_p\_prtstrstrkt\_\_prj\_projekte\_id aus Spalte prj.p\_prtstrstrkt.prj\_projekte\_id  
fk\_\_prj\_p\_prtstrstrkt\_\_prt\_protokoll\_id aus Spalte prj.p\_prtstrstrkt.prt\_protokoll\_id  
fk\_\_prj\_p\_prtstrstrkt\_\_str\_strkt\_id aus Spalte prj.p\_prtstrstrkt.str\_strkt\_id  
fk\_\_prj\_p\_prtverbindung\_\_prj\_projekte\_id aus Spalte prj.p\_prtverbindung.prj\_projekte\_id  
fk\_\_prj\_p\_prtverbindung\_\_prt\_protokoll\_id\_nachf aus Spalte prj.p\_prtverbindung.prt\_protokol  
fk\_\_prj\_p\_prtverbindung\_\_prt\_protokoll\_id\_vorg aus Spalte prj.p\_prtverbindung.prt\_protokoll

fk\_prj\_p\_psabk\_prj\_projekte\_id aus Spalte prj.p\_psabk.prj\_projekte\_id  
fk\_prj\_p\_psabk\_ps\_abk\_id aus Spalte prj.p\_psabk.ps\_abk\_id  
fk\_prj\_p\_psabk\_spr\_uebvar\_id aus Spalte prj.p\_psabk.spr\_uebvar\_id  
fk\_prj\_p\_psabk\_spr\_uebvar\_id\_l aus Spalte prj.p\_psabk.spr\_uebvar\_id\_l  
fk\_prj\_p\_psabk\_spr\_uebvar\_id\_zus aus Spalte prj.p\_psabk.spr\_uebvar\_id\_zus  
fk\_prj\_p\_strverbindung\_prj\_projekte\_id aus Spalte prj.p\_strverbindung.prj\_projekte\_id  
fk\_prj\_p\_strverbindung\_str\_strkt\_id\_nachf aus Spalte prj.p\_strverbindung.str\_strkt\_id\_nachf  
fk\_prj\_p\_strverbindung\_str\_strkt\_id\_vorg aus Spalte prj.p\_strverbindung.str\_strkt\_id\_vorg  
fk\_prj\_personen\_intern\_firma\_id aus Spalte prj.personen.intern\_firma\_id  
fk\_prj\_pl\_dokausgabe\_dok\_doku\_id aus Spalte prj.pl\_dokausgabe.dok\_doku\_id  
fk\_prj\_pl\_dokausgabe\_prj\_projekte\_id aus Spalte prj.pl\_dokausgabe.prj\_projekte\_id  
fk\_prj\_pl\_prtdokdoku\_dok\_doku\_id aus Spalte prj.pl\_prtdokdoku.dok\_doku\_id  
fk\_prj\_pl\_prtdokdoku\_prj\_projekte\_id aus Spalte prj.pl\_prtdokdoku.prj\_projekte\_id  
fk\_prj\_pl\_prtdokdoku\_prt\_protokoll\_id aus Spalte prj.pl\_prtdokdoku.prt\_protokoll\_id  
fk\_prj\_pl\_prtdokdoku\_spr\_uebvar\_id aus Spalte prj.pl\_prtdokdoku.spr\_uebvar\_id  
fk\_prj\_pl\_prtdokdoku\_spr\_uebvar\_id\_zus aus Spalte prj.pl\_prtdokdoku.spr\_uebvar\_id\_zus  
fk\_prj\_pl\_prtpsgefszenario\_prj\_projekte\_id aus Spalte prj.pl\_prtpsgefszenario.prj\_projekte\_id  
fk\_prj\_pl\_prtpsgefszenario\_prt\_protokoll\_id aus Spalte prj.pl\_prtpsgefszenario.prt\_protokoll\_id  
fk\_prj\_pl\_prtpsgefszenario\_ps\_gefszenario\_id aus Spalte prj.pl\_prtpsgefszenario.ps\_gefszenario\_id  
fk\_prj\_pl\_psgefabk\_prj\_projekte\_id aus Spalte prj.pl\_psgefabk.prj\_projekte\_id  
fk\_prj\_pl\_psgefabk\_ps\_abk\_id aus Spalte prj.pl\_psgefabk.ps\_abk\_id  
fk\_prj\_pl\_psgefabk\_ps\_gefszenario\_id aus Spalte prj.pl\_psgefabk.ps\_gefszenario\_id  
fk\_prj\_pl\_psgefabk\_spr\_uebvar\_id\_zus aus Spalte prj.pl\_psgefabk.spr\_uebvar\_id\_zus  
fk\_prj\_pl\_psgefanm\_prj\_projekte\_id aus Spalte prj.pl\_psgefanm.prj\_projekte\_id  
fk\_prj\_pl\_psgefanm\_ps\_anmerkung\_id aus Spalte prj.pl\_psgefanm.ps\_anmerkung\_id  
fk\_prj\_pl\_psgefanm\_ps\_gefszenario\_id aus Spalte prj.pl\_psgefanm.ps\_gefszenario\_id  
fk\_prj\_pl\_psgefanm\_spr\_uebvar\_id aus Spalte prj.pl\_psgefanm.spr\_uebvar\_id  
fk\_prj\_pl\_psgefstrkt\_prj\_projekte\_id aus Spalte prj.pl\_psgefstrkt.prj\_projekte\_id

fk\_prj\_pl\_psgefstrkt\_ps\_gefszenario\_id aus Spalte prj.pl\_psgefstrkt.ps\_gefszenario\_id  
 fk\_prj\_pl\_psgefstrkt\_str\_strkt\_id aus Spalte prj.pl\_psgefstrkt.str\_strkt\_id  
 fk\_prj\_pp\_protokoll\_prj\_personen\_id aus Spalte prj.ppprtprotokoll.prj\_personen\_id  
 fk\_prj\_pp\_protokoll\_prj\_projekte\_id aus Spalte prj.ppprtprotokoll.prj\_projekte\_id  
 fk\_prj\_pp\_protokoll\_prt\_protokoll\_id aus Spalte prj.ppprtprotokoll.prt\_protokoll\_id  
 fk\_prj\_pp\_protokoll\_spr\_uebvar\_id aus Spalte prj.ppprtprotokoll.spr\_uebvar\_id  
 fk\_prj\_pp\_psgefszenario\_prj\_personen\_id aus Spalte prj.pp\_psgefszenario.prj\_personen\_id  
 fk\_prj\_pp\_psgefszenario\_prj\_projekte\_id aus Spalte prj.pp\_psgefszenario.prj\_projekte\_id  
 fk\_prj\_pp\_psgefszenario\_ps\_gefszenario\_id aus Spalte prj.pp\_psgefszenario.ps\_gefszenario\_id  
 fk\_prj\_pp\_psgefszenario\_spr\_uebvar\_id aus Spalte prj.pp\_psgefszenario.spr\_uebvar\_id  
 fk\_prj\_pp\_psgefszenario\_spr\_uebvar\_id\_reprgefszen aus Spalte prj.pp\_psgefszenario.spr\_uebvar\_id\_reprgefszen  
 fk\_prj\_pp\_psgefszenario\_spr\_uebvar\_id\_ursp aus Spalte prj.pp\_psgefszenario.spr\_uebvar\_id\_ursp  
 fk\_prj\_pp\_strstrkt\_prj\_personen\_id aus Spalte prj.pp\_strstrkt.prj\_personen\_id  
 fk\_prj\_pp\_strstrkt\_prj\_projekte\_id aus Spalte prj.pp\_strstrkt.prj\_projekte\_id  
 fk\_prj\_pp\_strstrkt\_spr\_uebvar\_id aus Spalte prj.pp\_strstrkt.spr\_uebvar\_id  
 fk\_prj\_pp\_strstrkt\_str\_strkt\_id aus Spalte prj.pp\_strstrkt.str\_strkt\_id  
 fk\_prj\_ppl\_dokdoku\_dok\_doku\_id aus Spalte prj.ppl\_dokdoku.dok\_doku\_id  
 fk\_prj\_ppl\_dokdoku\_prj\_personen\_id aus Spalte prj.ppl\_dokdoku.prj\_personen\_id  
 fk\_prj\_ppl\_dokdoku\_prj\_projekte\_id aus Spalte prj.ppl\_dokdoku.prj\_projekte\_id

### Unique Constraints

uc\_prj\_p\_dokstrkt in Tabelle prj.p\_dokstrkt für die Spalte(n) prj.p\_dokstrkt.idfrei,  
 prj.p\_dokstrkt.dok\_doku\_id, prj.p\_dokstrkt.prj\_projekte\_id, prj.p\_dokstrkt.str\_strkt\_id  
 uc\_prj\_p\_prtstrstrkt in Tabelle prj.p\_prtstrstrkt für die Spalte(n) prj.p\_prtstrstrkt.idfrei,  
 prj.p\_prtstrstrkt.prj\_projekte\_id, prj.p\_prtstrstrkt.prt\_protokoll\_id, prj.p\_prtstrstrkt.spr\_uebvar\_id  
 uc\_prj\_p\_prtverbindung in Tabelle prj.p\_prtverbindung für die Spalte(n) prj.p\_prtverbindung.idfrei,  
 prj.p\_prtverbindung.prj\_projekte\_id, prj.p\_prtverbindung.prt\_protokoll\_id\_nachf,  
 prj.p\_prtverbindung.prt\_protokoll\_id\_vorg  
 uc\_prj\_p\_psabk in Tabelle prj.p\_psabk für die Spalte(n) prj.p\_psabk.idfrei, prj.p\_psabk.prj\_personen\_id,  
 prj.p\_psabk.ps\_abk\_id

uc\_\_prj\_p\_strverbindung in Tabelle prj.p\_strverbindung für die Spalte(n) prj.p\_strverbindung.idfrei,  
prj.p\_strverbindung.prj\_projekte\_id, prj.p\_strverbindung.str\_strkt\_id\_nachf,  
prj.p\_strverbindung.str\_strkt\_id\_vorg

uc\_\_prj\_personen in Tabelle prj.personen für die Spalte(n) prj.personen.idfrei, prj.personen.name\_  
prj.personen.name\_v, prj.personen.intern\_firma\_id

uc\_\_prj\_pl\_dokausgabe in Tabelle prj.pl\_dokausgabe für die Spalte(n) prj.pl\_dokausgabe.idfrei,  
prj.pl\_dokausgabe.logentfernt, prj.pl\_dokausgabe.logdatuhrz, prj.pl\_dokausgabe.dok\_dok  
prj.pl\_dokausgabe.prj\_projekte\_id

uc\_\_prj\_pl\_prtdokdoku in Tabelle prj.pl\_prtdokdoku für die Spalte(n) prj.pl\_prtdokdoku.idfrei,  
prj.pl\_prtdokdoku.logentfernt, prj.pl\_prtdokdoku.logdatuhrz, prj.pl\_prtdokdoku.dok\_dok  
prj.pl\_prtdokdoku.prj\_projekte\_id, prj.pl\_prtdokdoku.prt\_protokoll\_id, prj.pl\_prtdokdoku

uc\_\_prj\_pl\_prtpsgefszenario in Tabelle prj.pl\_prtpsgefszenario für die Spalte(n) prj.pl\_prtpsgefsze  
prj.pl\_prtpsgefszenario.prj\_projekte\_id, prj.pl\_prtpsgefszenario.prt\_protokoll\_id,  
prj.pl\_prtpsgefszenario.ps\_gefszenario\_id, prj.pl\_prtpsgefszenario.logentfernt,  
prj.pl\_prtpsgefszenario.logdatuhrz

uc\_\_prj\_pl\_psgefabk in Tabelle prj.pl\_psgefabk für die Spalte(n) prj.pl\_psgefabk.idfrei,  
prj.pl\_psgefabk.prj\_projekte\_id, prj.pl\_psgefabk.ps\_abk\_id, prj.pl\_psgefabk.ps\_gefszena  
prj.pl\_psgefabk.logentfernt, prj.pl\_psgefabk.logdatuhrz

uc\_\_prj\_pl\_psgefam in Tabelle prj.pl\_psgefam für die Spalte(n) prj.pl\_psgefam.idfrei,  
prj.pl\_psgefam.prj\_projekte\_id, prj.pl\_psgefam.ps\_anmerkung\_id, prj.pl\_psgefam.ps\_ge  
prj.pl\_psgefam.logentfernt, prj.pl\_psgefam.logdatuhrz

uc\_\_prj\_pl\_psgefstrkt in Tabelle prj.pl\_psgefstrkt für die Spalte(n) prj.pl\_psgefstrkt.idfrei,  
prj.pl\_psgefstrkt.prj\_projekte\_id, prj.pl\_psgefstrkt.ps\_gefszenario\_id, prj.pl\_psgefstr  
prj.pl\_psgefstrkt.logentfernt, prj.pl\_psgefstrkt.logdatuhrz

uc\_\_prj\_pp\_protokoll in Tabelle prj.pp\_prtprotokoll für die Spalte(n) prj.pp\_prtprotokoll.idfrei,  
prj.pp\_prtprotokoll.prj\_projekte\_id, prj.pp\_prtprotokoll.prt\_protokoll\_id, prj.pp\_prtpr  
prj.pp\_prtprotokoll.prj\_personen\_id

uc\_\_prj\_pp\_psgefszenario in Tabelle prj.pp\_psgefszenario für die Spalte(n) prj.pp\_psgefszenario.idf  
prj.pp\_psgefszenario.prj\_projekte\_id, prj.pp\_psgefszenario.ps\_gefszenario\_id

uc\_\_prj\_pp\_strstrkt in Tabelle prj.pp\_strstrkt für die Spalte(n) prj.pp\_strstrkt.idfrei,  
prj.pp\_strstrkt.prj\_personen\_id, prj.pp\_strstrkt.prj\_projekte\_id, prj.pp\_strstrkt.str\_s

uc\_\_prj\_ppl\_dokdoku in Tabelle prj.ppl\_dokdoku für die Spalte(n) prj.ppl\_dokdoku.idfrei,  
prj.ppl\_dokdoku.logentfernt, prj.ppl\_dokdoku.logdatuhrz, prj.ppl\_dokdoku.dok\_doku\_id,  
prj.ppl\_dokdoku.prj\_personen\_id, prj.ppl\_dokdoku.prj\_projekte\_id

uc\_\_prj\_projekte in Tabelle prj.projekte für die Spalte(n) prj.projekte.idfrei, prj.projekte.prjna

## Check Constraints

cc\_prj\_p\_strverbindung in Tabelle prj.p\_strverbindung die Bedingung (((str\_strkt\_id\_nachf = 0) AND (str\_strkt\_id\_vorg = 0)) OR ((id > 0) AND (str\_strkt\_id\_vorg > 0)))

cc\_prj\_pp\_protokoll in Tabelle prj.pp\_prtprotokoll die Bedingung (((((((((((id = 0) AND (idfrei = 0)) AND (prj\_personen\_id = 0)) AND (prj\_projekte\_id = 0)) AND (prt\_protokoll\_id = 0)) AND (spr\_uebvar\_id = 0)) OR ((id > 0) AND (idfrei > 0))) OR (((((id > 0) AND (idfrei = 0)) AND (prj\_personen\_id = 0)) AND (prj\_projekte\_id > 0)) AND (prt\_protokoll\_id > 0))) OR (((((((id > 0) AND (idfrei = 0)) AND (prj\_personen\_id > 0)) AND (prj\_projekte\_id > 0)) AND (prt\_protokoll\_id > 0)) AND (spr\_uebvar\_id = 0)) AND (teilnehmer = true))) OR (((((((id > 0) AND (idfrei = 0)) AND (prj\_personen\_id > 0)) AND (prj\_projekte\_id > 0)) AND (prt\_protokoll\_id > 0)) AND (spr\_uebvar\_id = 0)) AND (teilnehmer = false)) AND (termin IS NOT NULL)))

## 5.10 prj.nimm\_\_tabwerte

### Funktionsname

prj.nimm\_\_tabwerte( IN tabname varchar, IN tabwerte varchar[], OUT tabid varchar )

### Funktionsparameter/Rückgabewert(e)

**tabname** Name der Tabelle (ohne Schemanamen) des Schemas **prj**, welche die Werte übernehmen soll (Details siehe Funktionsbeschreibung — Werteliste einlesen).

**tabwerte** Liste der Werte, welche in die aktuelle Tabelle eingelesen werden sollen (Details siehe Funktionsbeschreibung — Werteliste einlesen).

**tabid** Es wird die Identifikationsnummer des aktuell eingelesenen Wertes **prj.\*.id** zurückgegeben.

### Funktionsbeschreibung

Mittels dieser Funktion werden in die Tabellen des Schemas **prj** die Werte eingelesen.

Werteliste einlesen:

Dem Funktionsparameter **tabwerte** ist eine Werteliste (**ARRAY**) zu übergeben (die maximale Anzahl und die Reihenfolge der Werte ist fest vorgegeben — siehe oben). Soll nur eine Teilmenge der Werte übergeben werden, dann sind alle Werte von links nach rechts zu spezifizieren (nichtbenötigte Werte am Ende der Liste dürfen weggelassen werden).

### personen

1. intern\_firma\_id
2. name\_n
3. name\_v
4. notizen

Es müssen mindestens die ersten drei Werte übergeben werden. Der erste Wert muß größer 0 sein.

### **projekte**

1. prjname
2. prjbeschr
3. makefile
4. notizen

Es müssen mindestens die ersten beiden Werte übergeben werden.

### **p\_dokstrkt, p\_prtstrstrkt, p\_prtverbindung, p\_psabk, p\_strverbindung**

1. prj\_projekte\_id
2. dok\_doku\_id, prt\_protokoll\_id, prt\_protokoll\_id\_nachf,  
ps\_abk\_id, str\_strkt\_id\_vorg
3. dok\_strkt\_id, str\_strkt\_id, prt\_protokoll\_id\_vorg,  
spr\_uebvar\_id, str\_strkt\_id\_nachf
4. notizen, notizen, notizen,  
spr\_uebvar\_id\_1, fctor
5. ---, ---, ---,  
spr\_uebvar\_id\_zus, notizen
6. ---, ---, ---,  
notizen, ---

Es müssen mindestens die ersten drei Werte übergeben werden. prj\_projekte\_id darf 0 sein, die anderen beiden Werte müssen größer 0 sein.

### **pl\_dokausgabe, pl\_prtdokdoku, pl\_prtpsgefszenario, pl\_psgefabk, pl\_psgefann, pl\_psgefstrkt**

1. prj\_projekte\_id
2. dok\_doku\_id, dok\_doku\_id, ps\_gefszenario\_id,  
ps\_gefszenario\_id, ps\_gefszenario\_id, ps\_gefszenario\_id
3. logentfernt, prt\_protokoll\_id, prt\_protokoll\_id,  
logentfernt, logentfernt, str\_strkt\_id
4. stand, nr, notizen,  
ps\_abk\_id, ps\_anmerkung\_id, notizen

5. datum, spr\_uebvar\_id, logentfernt,  
spr\_uebvar\_id\_zus, spr\_uebvar\_id, logentfernt
6. istfrei, logentfernt, logdatuhrz,  
notizen, notizen, logdatuhrz
7. notizen, spr\_uebvar\_id\_zus, logdquelle,  
logdatuhrz, logdatuhrz, logdquelle
8. logdatuhrz, notizen, logname,  
logdquelle, logdquelle, logname
9. logdquelle, logdatuhrz, logdqok,  
logname, logname, logdqok
10. logname, logdquelle, ---,  
logdqok, logdqok, ---
11. logdqok, logname, ---,  
---, ---, ---
12. ---, logdqok, ---,  
---, ---, ---

Es müssen mindestens die ersten beiden Werte übergeben werden. `prj_projekte_id` darf 0 sein, der andere Wert muß größer 0 sein. Bei diesen Tabellen (Tabellen mit Logbuchfunktion) wird nicht nach bereits eingelesenen Einträgen gesucht. Somit ist es möglich mehrfach dieselben Einträge einzulesen.

**pp\_prtprotokoll, pp\_prtprotokoll\_pers, pp\_psgefszenario,  
pp\_strstrkt, ppl\_dokdoku**

1. prj\_projekte\_id
2. prt\_protokoll\_id, prt\_protokoll\_id, ps\_gefszenario\_id,  
str\_strkt\_id, dok\_doku\_id
3. spr\_uebvar\_id, teilnehmer, prj\_personen\_id,  
prj\_personen\_id, prj\_personen\_id
4. datuhrz, prj\_personen\_id, spr\_uebvar\_id,  
spr\_uebvar\_id, logentfernt
5. intern, termin, spr\_uebvar\_id\_reprgefszen,  
fbereignis, notizen Für die Spalten `spr_uebvar_id_fzg` und `spr_uebvar_id` muß der Wert größer 0 sein.
6. erledigt, duanfrage, spr\_uebvar\_id\_ursp,  
notizen, logdatuhrz

- 7. notizen, notizen, notizen,  
---, logdquelle
- 8. ---, ---, ---,  
---, logname
- 9. ---, ---, ---,  
---, logdqok

Bei den Tabellen `ppl_dokdoku` müssen die ersten drei und bei den Tabellen `pp_psgfszenario`, `pp_strstrkt` die ersten zwei Werte übergeben werden. `prj_projekte_id` darf 0 sein, die anderen Drei bzw. Zwei Werte müssen größer 0 sein.

Bei der Tabelle `p1_prtdokdoku` müssen die ersten fünf Werte übergeben werden.

Die Tabelle `pp_prtprotokoll` deckt drei Aspekte ab:

1. Protokolleintrag einem Projekt zuordnen, projektspezifisch modifizieren und Randbedingungen setzen (`pp_prtprotokoll` wählen);
2. dem Protokolleintrag die beteiligten Personen zuordnen (`pp_prtprotokoll_pers` wählen);
3. dem Protokolleintrag der eine Frage / Aufgabe ist, die relevanten Informationen zuordnen (`pp_prtprotokoll_pers` wählen);

wobei ein Protokolleintrag ein Gefahrenprotokolleintrag, eine Sicherheitsüberprüfung oder ein Terminplaneintrag (je nach Sinnfälligkeit) sein kann.

Zur Unterscheidung der drei Aspekte werden die Spalten `prj_personen_id` und `teilnehmer` verwendet. Es gilt für

1. `prj_personen_id = 0` und `teilnehmer = NULL`
2. `prj_personen_id > 0` und `teilnehmer = true`
3. `prj_personen_id > 0` und `teilnehmer = false`

Die Spalten der Tabelle `pp_prtprotokoll` werden zur Abbildung der drei Aspekte wie folgt verwendet:

```

notizen ..... 1
datuhrz ..... 1
duanfrage ..... 3
erledigt ..... 1
intern ..... 1
prj_personen_id ..... 2 3
prj_projekte_id ... 1 2 3
prt_protokoll_id .. 1 2 3
spr_uebvar_id ..... 1
teilnehmer ..... 2 3
termin ..... 3

```



## Beispiele

ACTUNG ! Die Beispiele sind nicht aktualisiert --- muß später bereinigt werden.

```
-- Fehlermeldung. Ungültige Tabelle !
SELECT * FROM prj.nimm__tabwerte( NULL, NULL );

-- Fehlermeldung. Bei allen Tabellen müssen mindestens zwei Elemente übergeben
werden
SELECT * FROM prj.nimm__tabwerte( 'projekte', ARRAY['1'] );

-- Fehlermeldung. Die übergebenen Parameter müssen größer 0 sein.
SELECT * FROM prj.nimm__tabwerte( 'p_dokkomp', ARRAY['0', '1'] );
SELECT * FROM prj.nimm__tabwerte( 'p_dokkomp', ARRAY['1', '0'] );
SELECT * FROM prj.nimm__tabwerte( 'pl_psgefanm', ARRAY['1', '1', '0'] );
SELECT * FROM prj.nimm__tabwerte( 'ppu_kmpkomp', ARRAY['1', '1', '1', '', '0'] );
SELECT * FROM prj.nimm__tabwerte( 'pp_prtprotokoll', ARRAY['1', '1', '1', '', '',
'', '', '', '', '0'] );

-- Fehlermeldung. Es dürfen nur die zulässigen Begriffe verwendet werden
SELECT * FROM prj.nimm__tabwerte( 'pp_strstrkt', ARRAY['1', '22', '1', '99', 'ttt']
);
SELECT * FROM prj.nimm__tabwerte( 'p_strverbindung', ARRAY['1', '1', 'ttt'] );

-- prj.projekte einlesen
SELECT * FROM prj.projekte;
SELECT * FROM prj.nimm__tabwerte( 'projekte', ARRAY['testProj', 'Testprojekt',
'Inhalt des makefile', 'Zusatzinformationen'] );
SELECT * FROM prj.projekte;

-- dok.* einlesen
SELECT * FROM prj.pp_dokausgabe;
SELECT * FROM prj.nimm__tabwerte( 'pp_dokausgabe', ARRAY[prj.nimm__tabwerte( 'projekte'
ARRAY['testProj', 'Testprojekt'] ), dok.nimm__tabwerte( 'ausgabe', ARRAY[ dok.nimm__doku(
ARRAY['08-15', '1'] ), '01'] ), prj.nimm__tabwerte( 'personen', ARRAY['1', 'Bellair',
'Michael'] ), 'Zusatzinformationen'] );
SELECT * FROM prj.pp_dokausgabe;
SELECT * FROM prj.p_dokkomp;
SELECT * FROM prj.nimm__tabwerte( 'p_dokkomp', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), dok.nimm__tabwerte( 'komp', ARRAY[dok.nimm__doku(
ARRAY['08-15', '1'] ), kmp.nimm__komp( ARRAY['1', '47/11'] )] ), 'Zusatzinformationen'
);
SELECT * FROM prj.p_dokkomp;
SELECT * FROM prj.p_dokstrkt;
SELECT * FROM prj.nimm__tabwerte( 'p_dokstrkt', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), dok.nimm__tabwerte( 'strkt', ARRAY[dok.nimm__doku(
ARRAY['08-15', '1'] ), str.nimm__strkt( ARRAY['1', '-BAAA', spr.nimm__wgruppe( 'Struktu
```

```

'1', '0' )) ] ) ), 'Zusatzinformationen' ] );
SELECT * FROM prj.p_dokstrkt;

-- kmp.* einlesen
SELECT * FROM prj.ppu_kmpkomp;
SELECT * FROM prj.nimm__tabwerte( 'ppu_kmpkomp', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] )], kmp.nimm__komp( ARRAY['1', '08-15'] ), prj.nimm__tabwer
'personen', ARRAY['1', 'Bellair', 'Michael'] ), 'kmp-id-fzg', spr.nimm__wgruppe(
'Komponentenbezeichnung-Fahrezeughersteller', '1', '0' ), 'Zusatzinformationen'
);
SELECT * FROM prj.ppu_kmpkomp;
SELECT * FROM prj.p_kmpstrkt;
SELECT * FROM prj.nimm__tabwerte( 'p_kmpstrkt', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] )], kmp.nimm__strkt( ARRAY[kmp.nimm__komp( ARRAY['1',
'08-15'] ), str.nimm__strkt( ARRAY['1', '-BAAA', spr.nimm__wgruppe( 'Strukturpunkt',
'1', '0' )) ] ), 'Zusatzinformationen' ] ), 'Zusatzinformationen' ] );
SELECT * FROM prj.p_kmpstrkt;

-- prt.* einlesen
SELECT * FROM prj.p_prtdokausgabe;
SELECT * FROM prj.nimm__tabwerte( 'p_prtdokausgabe', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] )], prt.nimm__tabwerte( 'dokausgabe', ARRAY[prt.nimm__proto
ARRAY[spr.nimm__wgruppe( 'Aktueller Protokolleintrag.', '1', '0' ), 'PE_OK'] ),
dok.nimm__tabwerte( 'ausgabe', ARRAY[ dok.nimm__doku( ARRAY['08-15', '1'] ), '01'
), 'Zusatzinformationen' ] ), 'Zusatzinformationen' ] );
SELECT * FROM prj.p_prtdokausgabe;
SELECT * FROM prj.p_prtkomp;
SELECT * FROM prj.nimm__tabwerte( 'p_prtkomp', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] )], prt.nimm__tabwerte( 'komp', ARRAY[prt.nimm__protokollei
ARRAY[spr.nimm__wgruppe( 'Aktueller Protokolleintrag.', '1', '0' ), 'PE_OK'] ),
kmp.nimm__komp( ARRAY['1', '47/11'] ) ] ), 'Zusatzinformationen' ] );
SELECT * FROM prj.p_prtkomp;
SELECT * FROM prj.pp_prtprotokoll;
SELECT * FROM prj.nimm__tabwerte( 'pp_prtprotokoll', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] )], prt.nimm__protokolleintrag( ARRAY[spr.nimm__wgruppe(
'Aktueller Protokolleintrag.', '1', '0' ), 'PE_OK'] ), prj.nimm__tabwerte( 'personen',
ARRAY['1', 'Bellair', 'Michael'] ), 'true', '2007-06-08', '2007-06-20', '2007-07-07',
'true', 'true', spr.nimm__wgruppe( 'Projektabhängiger, aktueller Protokolleintrag.',
'1', '0' ), 'Zusatzinformationen' ] );
SELECT * FROM prj.pp_prtprotokoll;
SELECT * FROM prj.p_prtstrstrkt;
SELECT * FROM prj.nimm__tabwerte( 'p_prtstrstrkt', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] )], prt.nimm__tabwerte( 'strkt', ARRAY[prt.nimm__protokolle
ARRAY[spr.nimm__wgruppe( 'Aktueller Protokolleintrag.', '1', '0' ), 'PE_OK'] ),
str.nimm__strkt( ARRAY['1', '-BAAA', spr.nimm__wgruppe( 'Strukturpunkt', '1', '0'
)]))], 'Zusatzinformationen' ] );

```

```

SELECT * FROM prj.p_prtstrstrkt;
SELECT * FROM prj.p_prtverbindung;
SELECT * FROM prj.nimm__tabwerte( 'p_prtverbindung', ARRAY[prj.nimm__tabwerte( 'projekt',
ARRAY['testProj', 'Testprojekt'] ), prt.nimm__tabwerte( 'verbindung', ARRAY[prt.nimm__p
ARRAY[spr.nimm__wgruppe( 'Aktueller Protokolleintrag.', '1', '0' ), 'PE_OK'] ),
prt.nimm__protokolleintrag( ARRAY[spr.nimm__wgruppe( 'Zugeordneter Aktueller Protokolle
'1', '0' ), 'PE_OK'] )]), 'Zusatzinformationen' );
SELECT * FROM prj.p_prtverbindung;

-- ps.* einlesen
SELECT * FROM prj.p_psabk;
SELECT * FROM prj.nimm__tabwerte( 'p_psabk', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), ps.nimm__abk( ARRAY[spr.nimm__wgruppe( 'abk',
'1', '0' ), spr.nimm__wgruppe( 'Abkürzung', '1', '0' ), 'BP'] ), spr.nimm__wgruppe(
'projabk', '1', '0' ), spr.nimm__wgruppe( 'projektabhängige Abkürzung', '1', '0'
), 'Zusatzinformationen' ] );
SELECT * FROM prj.p_psabk;
SELECT * FROM prj.pl_psgefabk;
SELECT * FROM prj.nimm__tabwerte( 'pl_psgefabk', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), ps.nimm__aspekte( 'gefabk', ARRAY[ ps.nimm__gefszen
ARRAY[spr.nimm__wgruppe( 'Gefahrenszenario', '1', '0' ), spr.nimm__wgruppe( 'Ursprung
des Gefahrenszenarios', '1', '0' )] ), ps.nimm__abk( ARRAY[spr.nimm__wgruppe( 'abk',
'1', '0' ), spr.nimm__wgruppe( 'Abkürzung', '1', '0' ), 'BP'] )] ), '2007-06-06',
'datenquelle', 'micha', 'true', 'Zusatzinformationen' ] );
SELECT * FROM prj.pl_psgefabk;
SELECT * FROM prj.pl_psgefam;
SELECT * FROM prj.nimm__tabwerte( 'pl_psgefam', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), ps.nimm__aspekte( 'gefam', ARRAY[ps.nimm__gefszena
ARRAY[spr.nimm__wgruppe( 'Gefahrenszenario', '1', '0' ), spr.nimm__wgruppe( 'Ursprung
des Gefahrenszenarios', '1', '0' )] ), spr.nimm__wgruppe( 'Anmerkungen', '1', '0'
)] ), spr.nimm__wgruppe( 'projektspezifische Anmerkungen', '1', '0' ), 'true',
'2007-06-06', 'datenquelle', 'micha', 'true', 'Zusatzinformationen' ] );
SELECT * FROM prj.pl_psgefam;
SELECT * FROM prj.pl_psgefkom;
SELECT * FROM prj.nimm__tabwerte( 'pl_psgefkom', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), ps.nimm__aspekte( 'gefkom', ARRAY[ps.nimm__gefszen
ARRAY[spr.nimm__wgruppe( 'Gefahrenszenario', '1', '0' ), spr.nimm__wgruppe( 'Ursprung
des Gefahrenszenarios', '1', '0' )] ), kmp.nimm__komp( ARRAY['1', '08-15'] )] ),
'2007-06-06', 'datenquelle', 'micha', 'true', 'Zusatzinformationen' ] );
SELECT * FROM prj.pl_psgefkom;
SELECT * FROM prj.pl_psgefnetz;
SELECT * FROM prj.nimm__tabwerte( 'pl_psgefnetz', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), ps.nimm__aspekte( 'gefnetz', ARRAY[ps.nimm__gefszen
ARRAY[spr.nimm__wgruppe( 'Gefahrenszenario', '1', '0' ), spr.nimm__wgruppe( 'Ursprung
des Gefahrenszenarios', '1', '0' )] ), str.nimm__strkt( ARRAY['1', '-BAAA', spr.nimm__wg
'Strukturpunkt', '1', '0' ])] ), '2007-06-06', 'datenquelle', 'micha', 'true',

```

```

'Zusatzinformationen'] );
SELECT * FROM prj.pl_psgefnetzt;
SELECT * FROM prj.pl_prtpsgefszenario;
SELECT * FROM prj.nimm__tabwerte( 'pl_prtpsgefszenario', ARRAY[prj.nimm__tabwerte(
'projekte', ARRAY['testProj', 'Testprojekt'] ), ps.nimm__aspekte( 'gefprot', ARRAY[ps.nimm__
ARRAY[spr.nimm__wgruppe( 'Gefahrenszenario', '1', '0' ), spr.nimm__wgruppe( 'Ursprung
des Gefahrenszenarios', '1', '0' )]], prt.nimm__protokolleintrag( ARRAY[spr.nimm__wgruppe(
'Zugeordneter Aktueller Protokolleintrag.', '1', '0' ), 'PE_OK'] )) ], 'true',
'2007-06-06', 'datenquelle', 'micha', 'true', 'Zusatzinformationen'] );
SELECT * FROM prj.pl_prtpsgefszenario;
SELECT * FROM prj.pl_psgefstrkt;
SELECT * FROM prj.nimm__tabwerte( 'pl_psgefstrkt', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), ps.nimm__aspekte( 'gefstrkt', ARRAY[ps.nimm__gefszenari
ARRAY[spr.nimm__wgruppe( 'Gefahrenszenario', '1', '0' ), spr.nimm__wgruppe( 'Ursprung
des Gefahrenszenarios', '1', '0' )]], str.nimm__strkt( ARRAY['1', '-BAAA', spr.nimm__wgrupp
'Strukturpunkt', '1', '0' ])] )], 'true', '2007-06-06', 'datenquelle', 'micha',
'true', 'Zusatzinformationen'] );
SELECT * FROM prj.pl_psgefstrkt;
SELECT * FROM prj.pp_psgefszenario;
SELECT * FROM prj.nimm__tabwerte( 'pp_psgefszenario', ARRAY[prj.nimm__tabwerte(
'projekte', ARRAY['testProj', 'Testprojekt'] ), ps.nimm__gefszenario( ARRAY[spr.nimm__wgrupp
'Gefahrenszenario', '1', '0' ), spr.nimm__wgruppe( 'Ursprung des Gefahrenszenarios',
'1', '0' )]], prj.nimm__tabwerte( 'personen', ARRAY['1', 'Bellair', 'Michael']
), spr.nimm__wgruppe( 'projektspezifisch formuliertes Gefahrenszenario', '1', '0'
), spr.nimm__wgruppe( 'projektspezifisch formulierter Ursprung des Gefahrenszenarios',
'1', '0' ), 'Zusatzinformationen'] );
SELECT * FROM prj.pp_psgefszenario;

-- str.* einlesen
SELECT * FROM prj.pp_strstrkt;
SELECT * FROM prj.nimm__tabwerte( 'pp_strstrkt', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), str.nimm__strkt( ARRAY['1', '-BAAA', spr.nimm__wgruppe(
'Strukturpunkt', '1', '0' )]], prj.nimm__tabwerte( 'personen', ARRAY['1', 'Bellair',
'Michael'] ), spr.nimm__wgruppe( 'Strukturpunkt übersetzt ?', '1', '0' ), 'HAUS',
'Zusatzinformationen'] );
SELECT * FROM prj.pp_strstrkt;
SELECT * FROM prj.p_strverbindung;
SELECT * FROM prj.nimm__tabwerte( 'p_strverbindung', ARRAY[prj.nimm__tabwerte( 'projekte',
ARRAY['testProj', 'Testprojekt'] ), str.nimm__verbindung( ARRAY[str.nimm__strkt(
ARRAY['1', '-BAAA1', spr.nimm__wgruppe( 'Strukturpunkt-1', '1', '0' )]], str.nimm__strkt(
ARRAY['1', '-BAAA2', spr.nimm__wgruppe( 'Strukturpunkt-2', '1', '0' )]] )], 'UND',
'Zusatzinformationen'] );
SELECT * FROM prj.p_strverbindung;

```

## 5.11 prj.zeige\_\_dokdoku

### Beschreibung der Spalten

dok\_\_doku\_\_id siehe dok.doku.id

dok\_\_doku\_\_aspekt siehe dok.doku.aspekt

dok\_\_doku\_\_notizen siehe dok.doku.notizen

dok\_\_doku\_\_intern\_firma\_id siehe dok.doku.intern\_firma\_id

dok\_\_doku\_\_spr\_uebvar\_id\_kb siehe dok.doku.spr\_uebvar\_id\_kb

prj\_\_pl\_\_prtdokdoku\_\_id siehe prj.pl\_prtdokdoku.id

prj\_\_pl\_\_prtdokdoku\_\_prj\_projekte\_id siehe prj.pl\_prtdokdoku.prj\_projekte\_id

prj\_\_pl\_\_prtdokdoku\_\_prt\_protokoll\_id siehe prj.pl\_prtdokdoku.prt\_protokoll\_id

prj\_\_pl\_\_prtdokdoku\_\_nr siehe prj.pl\_prtdokdoku.nr

prj\_\_pl\_\_prtdokdoku\_\_spr\_uebvar\_id siehe prj.pl\_prtdokdoku.spr\_uebvar\_id

prj\_\_pl\_\_prtdokdoku\_\_spr\_uebvar\_id\_zus siehe prj.pl\_prtdokdoku.spr\_uebvar\_id\_zus

prj\_\_pl\_\_prtdokdoku\_\_notizen siehe prj.pl\_prtdokdoku.notizen

prj\_\_pl\_\_prtdokdoku\_\_logdatuhrz siehe prj.pl\_prtdokdoku.logdatuhrz

prj\_\_pl\_\_prtdokdoku\_\_logdqok siehe prj.pl\_prtdokdoku.logdqok

prj\_\_pl\_\_prtdokdoku\_\_logdquelle siehe prj.pl\_prtdokdoku.logdquelle

prj\_\_pl\_\_prtdokdoku\_\_logentfernt siehe prj.pl\_prtdokdoku.logentfernt

prj\_\_pl\_\_prtdokdoku\_\_logname siehe prj.pl\_prtdokdoku.logname

prj\_\_pl\_\_dokausgabe\_\_id siehe prj.pl\_dokausgabe.id

prj\_\_pl\_\_dokausgabe\_\_prj\_projekte\_id siehe prj.pl\_dokausgabe.prj\_projekte\_id

prj\_\_pl\_\_dokausgabe\_\_stand siehe prj.pl\_dokausgabe.stand

prj\_\_pl\_\_dokausgabe\_\_datum siehe prj.pl\_dokausgabe.datum

prj\_\_pl\_\_dokausgabe\_\_istfrei siehe prj.pl\_dokausgabe.istfrei

prj\_\_pl\_\_dokausgabe\_\_notizen siehe prj.pl\_dokausgabe.notizen

prj\_\_pl\_\_dokausgabe\_\_logdatuhrz siehe prj.pl\_dokausgabe.logdatuhrz

prj\_\_pl\_\_dokausgabe\_\_logdqok siehe prj.pl\_dokausgabe.logdqok

prj\_\_pl\_\_dokausgabe\_\_logdquelle siehe prj.pl\_dokausgabe.logdquelle

prj\_\_pl\_dokausgabe\_\_logentfernt siehe prj.pl\_dokausgabe.logentfernt  
prj\_\_pl\_dokausgabe\_\_logname siehe prj.pl\_dokausgabe.logname  
prj\_\_ppl\_dokdoku\_\_id siehe prj.ppl\_dokdoku.id  
prj\_\_ppl\_dokdoku\_\_prj\_personen\_id siehe prj.ppl\_dokdoku.prj\_personen\_id  
prj\_\_ppl\_dokdoku\_\_prj\_projekte\_id siehe prj.ppl\_dokdoku.prj\_projekte\_id  
prj\_\_ppl\_dokdoku\_\_notizen siehe prj.ppl\_dokdoku.notizen  
prj\_\_ppl\_dokdoku\_\_logdatuhrz siehe prj.ppl\_dokdoku.logdatuhrz  
prj\_\_ppl\_dokdoku\_\_logdqok siehe prj.ppl\_dokdoku.logdqok  
prj\_\_ppl\_dokdoku\_\_logdquelle siehe prj.ppl\_dokdoku.logdquelle  
prj\_\_ppl\_dokdoku\_\_logentfernt siehe prj.ppl\_dokdoku.logentfernt  
prj\_\_ppl\_dokdoku\_\_logname siehe prj.ppl\_dokdoku.logname  
prj\_\_p\_dokstrkt\_\_id siehe prj.p\_dokstrkt.id  
prj\_\_p\_dokstrkt\_\_notizen siehe prj.p\_dokstrkt.notizen  
prj\_\_p\_dokstrkt\_\_prj\_projekte\_id siehe prj\_projekte\_id  
prj\_\_p\_dokstrkt\_\_str\_strkt\_id siehe prj.p\_dokstrkt.str\_strkt\_id

### **Beschreibung der Sicht**

Es werden die Dokumente (siehe dok.doku, prj.pl\_prtdokdoku, prj.pl\_dokausgabe, prj.ppl\_dokdoku) angezeigt.

## **5.12 prj.zeige\_\_pl\_dokausgabe\_le**

### **Beschreibung der Spalten**

prj\_\_pl\_dokausgabe\_\_id siehe prj.pl\_dokausgabe.id

### **Beschreibung der Sicht**

Zu jedem in der Tabelle prj.pl\_dokausgabe enthaltenen Dokument wird unter Berücksichtigung des Projektes die Identifikationsnummer prj.pl\_dokausgabe.id der jeweils letzten zugeordneten Ausgabe angezeigt (Logbuchfunktion).

## **5.13 prj.zeige\_\_pl\_prtdokdoku\_le**

### **Beschreibung der Spalten**

prj\_\_pl\_prtdokdoku\_\_id siehe prj.pl\_prtdokdoku.id

### **Beschreibung der Sicht**

Zu jeder Sicherheitsanforderung, Protokolleintrag, Terminplaneintrag (siehe Tabelle prt.protokoll) werden unter Berücksichtigung des Projektes die Identifikationsnummern prj.pl\_prtdokdoku.id die zugeordneten Dokumente (siehe dok.doku) angezeigt (Logbuchfunktion).

## 5.14 prj.zeige\_\_pl\_prtpsgefszenario

### Beschreibung der Spalten

id siehe prj.pl\_prtpsgefszenario.id

notizen siehe prj.pl\_prtpsgefszenario.notizen

logdatuhrz siehe prj.pl\_prtpsgefszenario.logdatuhrz

logdqok siehe prj.pl\_prtpsgefszenario.logdqok

logdquelle siehe prj.pl\_prtpsgefszenario.logdquelle

logentfernt siehe prj.pl\_prtpsgefszenario.logentfernt

logname siehe prj.pl\_prtpsgefszenario.logname

prj\_projekte\_id siehe prj.pl\_prtpsgefszenario.prj\_projekte\_id

prt\_protokoll\_id siehe prj.pl\_prtpsgefszenario.prt\_protokoll\_id

ps\_gefszenario\_id siehe prj.pl\_prtpsgefszenario.ps\_gefszenario\_id

### Beschreibung der Sicht

Es werden die projektspezifischen Sicherheitsanforderungen, Gefahrenprotokolleinträge bzw. Terminplaneinträge angezeigt (siehe prj.pl\_prtpsgefszenario).

## 5.15 prj.zeige\_\_pl\_prtpsgefszenario\_le

### Beschreibung der Spalten

prj\_\_pl\_prtpsgefszenario\_\_id siehe prj.pl\_prtpsgefszenario.id

### Beschreibung der Sicht

Zu jedem in der Tabelle prj.pl\_prtpsgefszenario enthaltenen Aspekt wird der jeweils letzte zugeordnete in ps verwendete Protokolleintrag angezeigt (Logbuchfunktion). Die unterschiedlichen Aspekte, zu denen das Logbuch geführt wird, werden anhand der Spalten prj.pl\_prtpsgefszenario.prj\_projekte\_id, prj.pl\_prtpsgefszenario.prt\_protokoll\_id, prj.pl\_prtpsgefszenario.ps\_gefszenario\_id unterschieden.

## 5.16 prj.zeige\_\_pl\_psgefabk

### Beschreibung der Spalten

id siehe prj.pl\_psgefabk.id

notizen siehe prj.pl\_psgefabk.notizen

logdatuhrz siehe prj.pl\_psgefabk.logdatuhrz

logdqok siehe prj.pl\_psgefabk.logdqok

logdquelle siehe prj.pl\_psgefabk.logdquelle

logentfernt siehe prj.pl\_psgefabk.logentfernt

logname siehe prj.pl\_psgefabk.logname

prj\_projekte\_id siehe prj.pl\_psgefabk.prj\_projekte\_id

ps\_abk\_id siehe prj.pl\_psgefabk.ps\_abk\_id

ps\_gefszenario\_id siehe prj.pl\_psgefabk.ps\_gefszenario\_id

spr\_uebvar\_id\_zus siehe prj.pl\_psgefabk.spr\_uebvar\_id\_zus

### **Beschreibung der Sicht**

Es werden die projektspezifischen (siehe prj.pl\_psgefabk) Abkürzungen angezeigt, die den Gefahrenszenarien zugeordnet sind.

## **5.17 prj.zeige\_\_pl\_psgefabk\_le**

### **Beschreibung der Spalten**

prj\_\_prj.pl\_psgefabk\_\_id siehe prj.pl\_psgefabk.id

### **Beschreibung der Sicht**

Zu jedem in der Tabelle prj.pl\_psgefabk enthaltenen Aspekt wird die jeweils letzte zugeordnete in ps verwendete Abkürzung angezeigt (Logbuchfunktion). Die unterschiedlichen Aspekte, zu denen das Logbuch geführt wird, werden anhand der Spalten prj.pl\_psgefabk.prj\_projekte\_id, prj.pl\_psgefabk.ps\_gefszenario\_id, prj.pl\_psgefabk.ps\_abk\_id unterschieden.

## **5.18 prj.zeige\_\_pl\_psgefanm**

### **Beschreibung der Spalten**

id siehe prj.pl\_psgefanm.id

notizen siehe prj.pl\_psgefanm.notizen

logdatuhrz siehe prj.pl\_psgefanm.logdatuhrz

logdqok siehe prj.pl\_psgefanm.logdqok

logdquelle siehe prj.pl\_psgefanm.logdquelle

logentfernt siehe prj.pl\_psgefanm.logentfernt

logname siehe prj.pl\_psgefanm.logname

prj\_projekte\_id siehe prj.pl\_psgefanm.prj\_projekte\_id

ps\_anmerkung\_id siehe prj.pl\_psgefanm.ps\_anmerkung\_id



ps\_gefszenario\_id siehe prj.pl\_psgefanm.ps\_gefszenario\_id

spr\_uebvar\_id siehe prj.pl\_psgefanm.spr\_uebvar\_id

### **Beschreibung der Sicht**

Es werden die projektspezifischen (siehe prj.pl\_psgefanm) Anmerkungen angezeigt.

## **5.19 prj.zeige\_\_pl\_psgefanm\_le**

### **Beschreibung der Spalten**

prj\_\_pl\_psgefanm\_\_id siehe prj.pl\_psgefanm.id

### **Beschreibung der Sicht**

Zu jedem in der Tabelle prj.pl\_psgefanm enthaltenen Aspekt wird die jeweils letzte zugeordnete in ps verwendete Abkürzung angezeigt (Logbuchfunktion). Die unterschiedlichen Aspekte, zu denen das Logbuch geführt wird, werden anhand der Spalten prj.pl\_psgefanm.prj\_projekte\_id, prj.pl\_psgefanm.ps\_gefszenario\_id unterschieden.

## **5.20 prj.zeige\_\_pl\_psgefstrkt**

### **Beschreibung der Spalten**

id siehe prj.pl\_psgefstrkt.id

notizen siehe prj.pl\_psgefstrkt.notizen

logdatuhrz siehe prj.pl\_psgefstrkt.logdatuhrz

logdqok siehe prj.pl\_psgefstrkt.logdqok

logdquelle siehe prj.pl\_psgefstrkt.logdquelle

logentfernt siehe prj.pl\_psgefstrkt.logentfernt

logname siehe prj.pl\_psgefstrkt.logname

prj\_projekte\_id siehe prj.pl\_psgefstrkt.prj\_projekte\_id

ps\_gefszenario\_id siehe prj.pl\_psgefstrkt.ps\_gefszenario\_id

str\_strkt\_id siehe prj.pl\_psgefstrkt.str\_strkt\_id

### **Beschreibung der Sicht**

Es werden die projektspezifischen Strukturpunkte angezeigt (siehe prj.pl\_psgefstrkt).

## 5.21 prj.zeige\_\_pl\_psgefstrkt\_le

### Beschreibung der Spalten

prj\_\_pl\_psgefstrkt\_\_id siehe prj.pl\_psgefstrkt.id

### Beschreibung der Sicht

Zu jedem in der Tabelle prj.pl\_psgefstrkt enthaltenen Aspekt wird der jeweils letzte zugeordnete in ps verwendete Strukturpunkt angezeigt (Logbuchfunktion). Die unterschiedlichen Aspekte, zu denen das Logbuch geführt wird, werden anhand der Spalten prj.pl\_psgefstrkt.prj\_projekte\_\_prj.pl\_psgefstrkt.ps\_gefszenario\_id unterschieden.

## 5.22 prj.zeige\_\_pp\_psgefszENARIO

### Beschreibung der Spalten

id siehe prj.pp\_psgefszENARIO.id

notizen siehe prj.pp\_psgefszENARIO.notizen

prj\_personen\_id siehe prj.pp\_psgefszENARIO.prj\_personen\_id

prj\_projekte\_id siehe prj.pp\_psgefszENARIO.prj\_projekte\_id

ps\_gefszenario\_id siehe prj.pp\_psgefszENARIO.ps\_gefszenario\_id

spr\_uebvar\_id siehe prj.pp\_psgefszENARIO.spr\_uebvar\_id

spr\_uebvar\_id\_reprgefszen siehe prj.pp\_psgefszENARIO.spr\_uebvar\_id\_reprgefszen

spr\_uebvar\_id\_ursp siehe prj.pp\_psgefszENARIO.spr\_uebvar\_id\_ursp

### Beschreibung der Sicht

Es werden die projektspezifischen Gefahrenszenarien angezeigt (siehe prj.pp\_psgefszENARIO).

## 5.23 prj.zeige\_\_ppl\_dokdoku\_le

### Beschreibung der Spalten

prj\_\_ppl\_dokdoku\_\_id siehe prj.ppl\_dokdoku.id

### Beschreibung der Sicht

Zu jedem in der Tabelle prj.ppl\_dokdoku enthaltenen Dokument wird unter Berücksichtigung des Projektes die Identifikationsnummer prj.ppl\_dokdoku.id des Ansprechpartners (ggf. der verantw. Person) angezeigt (Logbuchfunktion).

## 5.24 prj.zeige\_\_prtprotokoll

### Beschreibung der Spalten

prt\_\_typ\_\_id siehe prt.typ.id

prt\_\_typ\_\_notizen siehe prt.typ.notizen

prt\_\_typ\_\_aspekt siehe prt.typ.aspekt

prt\_\_typ\_\_spr\_uebvar\_id siehe prt.typ.spr\_uebvar\_id

prt\_\_protokoll\_\_id siehe prt.protokoll.id

prt\_\_protokoll\_\_notizen siehe prt.protokoll.notizen

prj\_\_pp\_prtprotokoll\_\_id siehe prj.pp\_prtprotokoll.id

prj\_\_pp\_prtprotokoll\_\_notizen siehe prj.pp\_prtprotokoll.notizen

prj\_\_pp\_proto\_koll\_\_datuhrz siehe prj.pp\_prtprotokoll.datuhrz

prj\_\_pp\_prtprotokoll\_\_duanfrage siehe prj.pp\_prtprotokoll.duanfrage

prj\_\_pp\_prtprotokoll\_\_erledigt siehe prj.pp\_prtprotokoll.erledigt

prj\_\_pp\_prtprotokoll\_\_intern siehe prj.pp\_prtprotokoll.intern

prj\_\_pp\_prtprotokoll\_\_prj\_personen\_id siehe prj.pp\_prtprotokoll.prj\_personen\_id

prj\_\_pp\_prtprotokoll\_\_prj\_projekte\_id siehe prj.pp\_prtprotokoll.prj\_projekte\_id

prj\_\_pp\_prtprotokoll\_\_spr\_uebvar\_id siehe prj.pp\_prtprotokoll.spr\_uebvar\_id

prj\_\_pp\_prtprotokoll\_\_teilnehmer siehe prj.pp\_prtprotokoll.teilnehmer

prj\_\_pp\_prtprotokoll\_\_termin siehe prj.pp\_prtprotokoll.termin

### Beschreibung der Sicht

Es werden die projektspezifischen Sicherheitsanforderungen, Gefahrenprotokolleinträge und Terminplaneinträge angezeigt (siehe prj.pp\_prtprotokoll).

## 5.25 prj.zeige\_\_prtprotokolltabellen

### Beschreibung der Spalten

prt\_\_protokoll\_\_id siehe prt.protokoll.id bzw. prj.p\_prtverbindung.prt\_protokoll\_id\_nachf

prj\_\_p\_prtstrstrkt\_\_id siehe prj.p\_prtstrstrkt.id

prj\_\_p\_prtstrstrkt\_\_notizen siehe prj.p\_prtstrstrkt.notizen

prj\_\_p\_prtstrstrkt\_\_prj\_projekte\_id siehe prj.p\_prtstrstrkt.prj\_projekte\_id

prj\_\_p\_prtstrstrkt\_\_str\_strkt\_id siehe prj.p\_prtstrstrkt.str\_strkt\_id

prj\_\_p\_prtverbindung\_\_id siehe prj.p\_prtverbindung.id

prj\_\_p\_prtverbindung\_\_notizen siehe prj.p\_prtverbindung.notizen

prj\_\_p\_prtverbindung\_\_prj\_projekte\_id siehe prj.p\_prtverbindung.prj\_projekte\_id

prj\_\_p\_prtverbindung\_\_prt\_protokoll\_id\_vorg siehe prj.p\_prtverbindung.prt\_protokoll\_id\_vorg

### **Beschreibung der Sicht**

Es werden die projektspezifischen Zusatzinformationen (siehe prj.p\_prtstrstrkt, prj.p\_prtverbindung) zu den Sicherheitsanforderungen, Gefahrenprotokolleinträgen bzw. Terminplaneinträge angezeigt.

## **5.26 prj.zeige\_\_psabk**

### **Beschreibung der Spalten**

ps\_\_abk\_\_id siehe ps.abk.id

ps\_\_abk\_\_notizen siehe ps.abk.notizen

ps\_\_abk\_\_typ siehe ps.abk.typ

prj\_\_p\_psabk\_\_id siehe prj.p\_psabk.id

prj\_\_p\_psabk\_\_notizen siehe prj.p\_psabk.notizen

prj\_\_p\_psabk\_\_prj\_projekte\_id siehe prj.p\_psabk.prj\_projekte\_id

prj\_\_p\_psabk\_\_spr\_uebvar\_id siehe prj.p\_psabk.spr\_uebvar\_id

prj\_\_p\_psabk\_\_spr\_uebvar\_id\_l siehe prj.p\_psabk.spr\_uebvar\_id\_l

prj\_\_p\_psabk\_\_spr\_uebvar\_id\_zus siehe prj.p\_psabk.spr\_uebvar\_id\_zus

### **Beschreibung der Sicht**

Es werden die projektspezifischen (siehe prj.p\_psabk) Abkürzungen angezeigt.

## **5.27 prj.zeige\_\_strstrkt**

### **Beschreibung der Spalten**

str\_\_name\_\_id siehe str.name.id

str\_\_name\_\_name siehe str.name.name

str\_\_name\_\_name\_k siehe str.name.name\_k

str\_\_name\_\_notizen siehe str.name.notizen

str\_\_strkt\_\_id siehe str.strkt.id

`str__strkt__notizen` siehe `str.strkt.notizen`

`str__strkt__nr` siehe `str.strkt.nr`

`prj__pp_strstrkt__id` siehe `prj.pp_strstrkt.id` UND `prj.p_strverbindung.str_strkt_id_vorg`

`prj__pp_strstrkt__notizen` siehe `prj.pp_strstrkt.notizen`

`prj__pp_strstrkt__fbereignis` siehe `prj.pp_strstrkt.fbereignis`

`prj__pp_strstrkt__prj_personen_id` siehe `prj.pp_strstrkt.prj_personen_id`

`prj__projekte_id` siehe `prj.pp_strstrkt.prj_projekte_id` UND `prj.p_strverbindung.prj_projekt`

`prj__pp_strstrkt__spr_uebvar_id` siehe `prj.pp_strstrkt.spr_uebvar_id`

`prj__p_strverbindung__id` siehe `prj.p_strverbindung.id`

`prj__p_strverbindung__notizen` siehe `prj.p_strverbindung.notizen`

`prj__p_strverbindung__fbtor` siehe `prj.p_strverbindung.fbtor`

`prj__p_strverbindung__str_strkt_id_nachf` siehe `prj.p_strverbindung.str_strkt_id_nachf`

### **Beschreibung der Sicht**

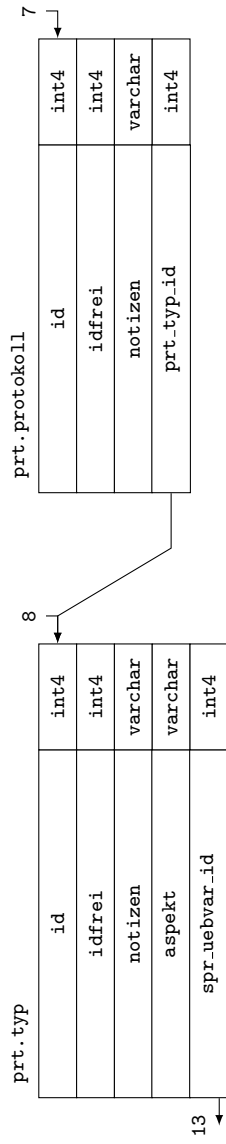
Es werden die projektübergreifend gültigen (siehe `str.name`, `str.strkt`) und die projektspezifischen Strukturpunkte (siehe `prj.pp_strstrkt`, `prj.p_strverbindung`) angezeigt.

## 6 prt

Im Schema „Protokoll“ werden neben Protokolleinträgen (PE) auch Sicherheitsanforderungen (SF) und Terminplaneinträge (TP) gespeichert.

Mittels der Tabelle `prt.verbindung` können Verbindungen zwischen den PE, SF, TP hergestellt werden (sind auch aspektübergreifend [PE - SF - TE] sinnvoll).

## 6.1 Tabellenhierarchie der Ebenen 1 — 3



## 6.2 Schlüssel und Bedingungen

### Primärschlüssel

`pk_prt_protokoll_id` für Spalte `prt.protokoll_id`

`pk_prt_typ_id` für Spalte `prt.typ_id`

### Fremdschlüssel

`fk_prt_protokoll_prt_typ_id` aus Spalte `prt.protokoll.prt_typ_id`

`fk_prt_typ_spr_uebvar_id` aus Spalte `prt.typ.spr_uebvar_id`

### Unique Constraints

`uc_prt_typ` in Tabelle `prt.typ` für die Spalte(n) `prt.typ.idfrei`, `prt.typ.aspekt`, `prt.typ.spr_uebva`

### Check Constraints

`cc_prt_typ` in Tabelle `prt.typ` die Bedingung (((`aspekt IS NULL`) OR ((`aspekt>::text = 'PE'::text`)) OR ((`aspekt>::text = 'SF'::text`)) OR ((`aspekt>::text = 'TP'::text`)))

## 6.3 prt.nimm\_protokolleintrag

### Funktionsname

`prt.nimm_protokolleintrag( IN tabwerte varchar[], OUT protokollid varchar )`

### Funktionsparameter/Rückgabewert(e)

`tabwerte` Liste der Werte, welche in `prt.protokoll` eingelesen werden sollen (Details siehe Funktionsbeschreibung — Werteliste einlesen). Sie werden in der folgenden Reihenfolge in die Spalten

1. `prt_typ_id` - Identifikationsnummer des Typs des Protokolleintrages.
2. `notizen`

Es müssen mindestens die ersten beiden Werte übergeben werden.

`protokollid` Es wird die Identifikationsnummer des eingelesenen Protokolleintrages zurückgegeben (`prt.protokoll_id`)

### Funktionsbeschreibung

Ziel der Funktion ist das Einlesen eines Protokolleintrages, einer Sicherheitsanforderung, eines Terminplaneintrages.

Werteliste einlesen:

Dem Funktionsparameter `tabwerte` ist eine Werteliste (`ARRAY`) zu übergeben (die maximale Anzahl und die Reihenfolge der Werte ist fest vorgegeben — siehe oben). Soll nur eine Teilmenge der Werte übergeben werden, dann sind alle Werte von links nach rechts zu spezifizieren (nichtbenötigte Werte am Ende der Liste dürfen weggelassen werden).



## Beispiele

-- Fehlermeldung. Die Mindestanzahl der Elemente muß stimmen.

```
SELECT * FROM prt.nimm_protokolleintrag( ARRAY['1'] );
```

-- Fehlermeldung. Der Parameter tabwerte[1] muß größer 0 sein.

```
SELECT * FROM prt.nimm_protokolleintrag( ARRAY['0', '1'] );
```

-- Fehlermeldung. Der Parameter tabwerte[2] muß größer 0 sein.

```
SELECT * FROM prt.nimm_protokolleintrag( ARRAY['1', '0'] );
```

-- Protokolleintrag einlesen

```
SELECT prt_protokoll_id, prt_typ_id, prt_protokoll_spr_uebvar_id, prt_protokoll_notizen  
FROM prt.zeige_protokolleintrag;
```

```
SELECT * FROM prt.nimm_typ( ARRAY[spr.nimm_wgruppe( 'Frage', '1', '0' ), 'PE']  
);
```

```
SELECT * FROM prt.nimm_protokolleintrag( ARRAY[spr.nimm_wgruppe( 'Aktueller Protokoll  
'1', '0' ), '1'] );
```

```
SELECT * FROM prt.nimm_protokolleintrag( ARRAY[spr.nimm_wgruppe( 'Aktueller Protokoll  
'1', '0' ), '1'] );
```

```
SELECT * FROM prt.nimm_protokolleintrag( ARRAY[spr.nimm_wgruppe( 'Weiterer Protokolle  
'1', '0' ), '1', 'interne Notizen zum PE'] );
```

```
SELECT prt_protokoll_id, prt_typ_id, prt_protokoll_spr_uebvar_id, prt_protokoll_notizen  
FROM prt.zeige_protokolleintrag;
```

## 6.4 prt.nimm\_typ

### Funktionsname

```
prt.nimm_typ( IN tabwerte varchar[], OUT typid varchar )
```

### Funktionsparameter/Rückgabewert(e)

**tabwerte** Liste der Werte, welche in `prt.typ` eingelesen werden sollen (Details siehe Funktionsbeschreibung — Werteliste einlesen). Sie werden in der folgenden Reihenfolge in die Spalten

1. `spr_uebvar_id` - Identifikationsnummer der Wortgruppe, welche den Typ des Protokolleintrages / der Sicherheitsanforderung / den Terminplaneintrag darstellt. Sie muß größer 0 sein.
2. `aspekt` (gültige Kurzbezeichnungen siehe Funktionsbeschreibung — Aspekt - gültige Kurzbezeichnungen. Sie werden auf Plusibilität geprüft.),
3. `notizen`

Es müssen mindestens die ersten beiden Werte übergeben werden.

**typid** Es wird die Identifikationsnummer des eingelesenen Typs des Protokolleintrages / der Sicherheitsanforderung / den Terminplaneintrag zurückgegeben (`prt.typ.id`)

### **Funktionsbeschreibung**

Ziel der Funktion ist das Einlesen eines Typs des Protokolleintrages, der Sicherheitsanforderung, des Terminplaneintrages.

### **Werteliste einlesen:**

Dem Funktionsparameter `tabwerte` ist eine Werteliste (`ARRAY`) zu übergeben (die maximale Anzahl und die Reihenfolge der Werte ist fest vorgegeben — siehe oben). Soll nur eine Teilmenge der Werte übergeben werden, dann sind alle Werte von links nach rechts zu spezifizieren (nichtbenötigte Werte am Ende der Liste dürfen weggelassen werden).

Es wird empfohlen, zum Einlesen des ersten Parameters die Funktion `spr.nimm_wgruppe(...)` zu verwenden (Hinweise zum Umgang siehe dort).

### **Aspekt - gültige Kurzbezeichnungen:**

Zur Klassifizierung der Aspekte sind die folgenden Kurzbezeichnungen zu verwenden.

**PE** Protokolleintrag

**SF** Sicherheitsanforderung

**TP** Terminplaneintrag

### **Beispiele**

-- Fehlermeldung. Die Mindestanzahl der Elemente muß stimmen.

```
SELECT * FROM prt.nimm__typ( ARRAY['1'] );
```

-- Fehlermeldung. Der Parameter `tabwerte[1]` muß größer 0 sein.

```
SELECT * FROM prt.nimm__typ( ARRAY['0', 'PE'] );
```

-- Fehlermeldung. Der Parameter `tabwerte[2]` muß gültige Werte enthalten.

```
SELECT * FROM prt.nimm__typ( ARRAY['1', 'C'] );
```

-- Typ einlesen

```
SELECT prt.typ_id, prt.typ_notizen, prt.typ_aspekt, prt.typ__spr_uebvar_id
```

```
FROM prt.zeige__protokolleintrag;
```

```
SELECT * FROM prt.nimm__typ( ARRAY[spr.nimm_wgruppe( 'Frage', '1', '0' ), 'PE'] );
```

```
SELECT * FROM prt.nimm__typ( ARRAY[spr.nimm_wgruppe( 'Technische Maßnahme', '1', '0' ), 'SF'] );
```

```
SELECT * FROM prt.nimm__typ( ARRAY[spr.nimm_wgruppe( 'Meilenstein', '1', '0' ), 'TP', 'interne Notizen zum TP'] );
```

```
SELECT prt.typ_id, prt.typ_notizen, prt.typ_aspekt, prt.typ__spr_uebvar_id
```

```
FROM prt.zeige__protokolleintrag;
```

## 7 ps

Das Schema „Produktsicherheit“ dient der Verwaltung sämtlicher Aspekte zur Produktsicherheit.

Die Zuordnung von technischen Systemen / Komponenten bzw. Strukturpunkten erfolgt **explizit**

- str.strkt --- kmp.strkt --- kmp.komp --- ps.gefkomp --- ps.gefszenario
- str.strkt --- ps.gefstrkt --- ps.gefszenario

**implizit über prt**

- str.strkt --- kmp.strkt --- kmp.komp --- prt.komp --- prt.protokoll --- ps.gefpro  
--- ps.gefszenario
- str.strkt --- prt.strkt --- prt.protokoll --- ps.gefprot --- ps.gefszenario

**implizit über prt.dokausgabe**

- str.strkt --- kmp.strkt --- kmp.komp --- dok.komp --- dok.doku --- dok.ausgabe  
--- prt.dokausgabe --- prt.protokoll --- ps.gefprot --- ps.gefszenario
- str.strkt --- dok.strkt --- dok.doku --- dok.ausgabe --- prt.dokausgabe ---  
prt.protokoll --- ps.gefprot --- ps.gefszenario

## 7.1 Tabellenhierarchie der Ebenen 1 — 3

9

ps.abk	id	int4
	idfrei	int4
	notizen	varchar
	typ	varchar

10

ps.anmerkung	id	int4
	idfrei	int4
	notizen	varchar

11

ps.gefszenario	id	int4
	idfrei	int4
	notizen	varchar

## 7.2 Schlüssel und Bedingungen

### Primärschlüssel

`pk_ps_abk_id` für Spalte `ps.abk.id`

`pk_ps_anmerkung_id` für Spalte `ps.anmerkung.id`

`pk_ps_gefszenario_id` für Spalte `ps.gefszenario.id`

### Unique Constraints

`uc_ps_abk` in Tabelle `ps.abk` für die Spalte(n) `ps.abk.idfrei`, `ps.abk.notizen`, `ps.abk.typ`

## 7.3 `ps.nimm_abk`

### Funktionsname

`ps.nimm_abk( IN tabwerte varchar[], OUT abkid varchar )`

### Funktionsparameter/Rückgabewert(e)

`tabwerte` Liste der Werte, welche in `ps.abk` eingelesen werden sollen (Details siehe Funktionsbeschreibung — Werteliste einlesen). Sie werden in der folgenden Reihenfolge in die Spalten

1. `typ` gültige Kurzbezeichnungen siehe Funktionsbeschreibung — Typ - gültige Kurzbezeichnungen. Sie werden auf Plusibilität geprüft.,
2. `notizen`

eingetragen. Es müssen beide Werte übergeben werden.

`abkid` Es wird die Identifikationsnummer der Abkürzung `ps.abk.id` zurückgegeben.

### Funktionsbeschreibung

Ziel der Funktion ist das Einlesen der Abkürzungen, welche einem Gefahrenszenario zugeordnet werden können..

#### *Werteliste einlesen:*

Dem Funktionsparameter `tabwerte` ist eine Werteliste (`ARRAY`) zu übergeben (die maximale Anzahl und die Reihenfolge der Werte ist fest vorgegeben — siehe oben). Soll nur eine Teilmenge der Werte übergeben werden, dann sind alle Werte von links nach rechts zu spezifizieren (nichtbenötigte Werte am Ende der Liste dürfen weggelassen werden).

#### *Typ - gültige Kurzbezeichnungen:*

**GP** gefährdete Person (die dem Gefahrenszenario ausgesetzt Personen)

**RG** Gefahrenstufe — Auswirkungen bei Eintreten des Gefahrenszenarios (zur Risikobewertung lt. EN 50126)

**RH** Häufigkeit des Auftretens des Gefahrenszenarios (zur Risikobewertung lt. EN 50126)

**RR** Risikostufe — Bewertung des Risikos des Gefahrenszenarios (zur Risikobewertung lt. EN 50126)

### Beispiele

-- Fehlermeldung. Die Mindestanzahl der Elemente muß stimmen.

```
SELECT * FROM ps.nimm__abk( ARRAY['1', '1'] );
```

-- Fehlermeldung. Der Parameter tabwerte[1] bzw. tabwerte[2] muß größer 0 sein.

```
SELECT * FROM ps.nimm__abk( ARRAY['0', '1', 'C' ] );
```

```
SELECT * FROM ps.nimm__abk( ARRAY['1', '0', 'C' ] );
```

-- Fehlermeldung. Der Parameter tabwerte[3] muß gültige Werte enthalten.

```
SELECT * FROM ps.nimm__abk( ARRAY['1', '1', 'C' ] );
```

-- Standardverhalten

```
SELECT * FROM ps.abk;
```

```
SELECT * FROM ps.nimm__abk( ARRAY[spr.nimm__wgruppe( 'abk', '1', '0' ), spr.nimm__wgruppe( 'Abkürzung', '1', '0' ), spr.nimm__wgruppe( 'Zusatzinfo', '1', '0' ), 'GP'] );
```

```
SELECT * FROM ps.nimm__abk( ARRAY[spr.nimm__wgruppe( 'abk-1', '1', '0' ), spr.nimm__wgruppe( 'Abkürzung-1', '1', '0' ), spr.nimm__wgruppe( 'Zusatzinfo', '1', '0' ), 'GP', 'Notizen'] );
```

```
SELECT * FROM ps.abk;
```

## 7.4 ps.zeige\_\_risikomatrix\_en50126

### Beschreibung der Spalten

ps\_\_abk\_\_id\_hfg siehe ps.abk.id der Häufigkeit des Gefahrenszenarios

ps\_\_abk\_\_id\_gstufe siehe ps.abk.id der Auswirkung des Gefahrenszenarios (Gefahrenstufe)

ps\_\_abk\_\_id\_rstufe siehe ps.abk.id der Risikobewertung des Gefahrenszenarios (Risikostufe)

### Beschreibung der Sicht

Es werden die Identifikationsnummern der Abkürzungen ps.abk.id zur Risikobewertung nach EN 50 126-1 ausgelesen.

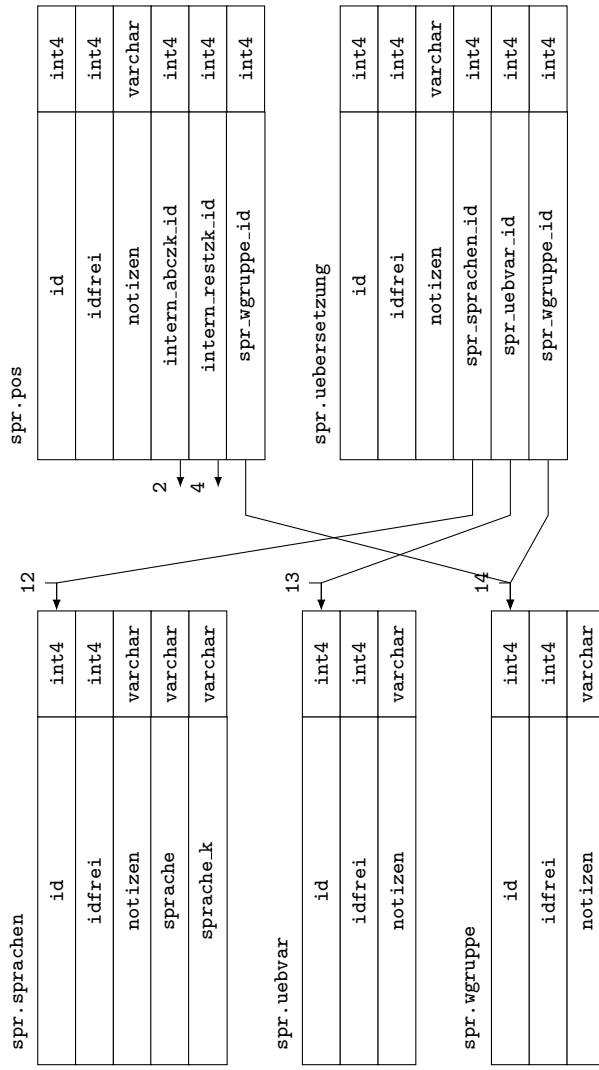
## 8 spr

Das Schema `Sprache` dient der sprachenabhängigen Verwaltung von Wortgruppen. Jeder Wortgruppe `spr.wgruppe.id` können mehrere unterschiedliche Sprachen `spr.sprachen.id` mittels unterschiedlicher Übersetzungsvarianten `spr.uebvar.id` zugeordnet werden. Die Tabelle `spr.uebersetzung` dient der Zuordnung der Sprachen / Übersetzungsvarianten zur Wortgruppe.

Innerhalb einer Übersetzungsvariante `spr.uebersetzung.spr_uebvar_id` tritt jede Sprache maximal einmal auf.

Die Tabelle `spr.pos` dient der Bildung der Wortgruppen.

## 8.1 Tabellenhierarchie der Ebenen 1 — 3





## 8.2 Schlüssel und Bedingungen

### Primärschlüssel

pk\_\_spr\_pos\_id für Spalte spr.pos.id

pk\_\_spr\_sprachen\_id für Spalte spr.sprachen.id

pk\_\_spr\_uebersetzung\_id für Spalte spr.uebersetzung.id

pk\_\_spr\_uebvar\_id für Spalte spr.uebvar.id

pk\_\_spr\_wgruppe\_id für Spalte spr.wgruppe.id

### Fremdschlüssel

fk\_\_spr\_pos\_intern\_abczk\_id aus Spalte spr.pos.intern\_abczk\_id

fk\_\_spr\_pos\_intern\_restzk\_id aus Spalte spr.pos.intern\_restzk\_id

fk\_\_spr\_pos\_spr\_wgruppe\_id aus Spalte spr.pos.spr\_wgruppe\_id

fk\_\_spr\_uebersetzung\_spr\_sprachen\_id aus Spalte spr.uebersetzung.spr\_sprachen\_id

fk\_\_spr\_uebersetzung\_spr\_uebvar\_id aus Spalte spr.uebersetzung.spr\_uebvar\_id

fk\_\_spr\_uebersetzung\_spr\_wgruppe\_id aus Spalte spr.uebersetzung.spr\_wgruppe\_id

### Unique Constraints

uc\_\_spr\_sprachen in Tabelle spr.sprachen für die Spalte(n) spr.sprachen.idfrei, spr.sprachen.s

uc\_\_spr\_uebersetzung in Tabelle spr.uebersetzung für die Spalte(n) spr.uebersetzung.idfrei,  
spr.uebersetzung.spr\_sprachen\_id, spr.uebersetzung.spr\_uebvar\_id

### Check Constraints

cc\_\_spr\_pos in Tabelle spr.pos die Bedingung (((((intern\_abczk\_id = 0) AND (intern\_restzk\_id  
= 0)) AND (spr\_wgruppe\_id = 0)) OR (((intern\_abczk\_id > 0) AND (intern\_restzk\_id  
= 0)) AND (spr\_wgruppe\_id > 0))) OR (((intern\_abczk\_id = 0) AND (intern\_restzk\_id  
> 0)) AND (spr\_wgruppe\_id > 0)))

## 8.3 spr.nimm\_\_wgruppe

### Funktionsname

spr.nimm\_\_wgruppe( IN wortgruppe varchar, IN sprachenid varchar, INOUT spruebvarid  
varchar )

### Funktionsparameter/Rückgabewert(e)

wortgruppe Es ist die einzulesende Wortgruppe anzugeben. Sie darf nicht leer oder NULL sein.

**sprachenid** Es ist die Identifikationsnummer `spr.sprachen.id` der Sprache anzugeben, in der die Wortgruppe eingelesen werden soll. Der Wert muß angegeben werden und größer/gleich 0 sein.

**spruebvarid** Es wird die Identifikationsnummer (`spr.uebvar.id`) der Übersetzungsvariante zurückgegeben, welcher die aktuelle Wortgruppe einschliesslich Sprache zugeordnet wurde. Wird ein Wert größer 0 übergeben, so soll die aktuelle Wortgruppe einschliesslich Sprache dieser Übersetzungsvariante hinzugefügt werden (Modus „Übersetzen“). Andernfalls liegt der Modus „Einlesen“ vor.

## **Funktionsbeschreibung**

Ziel der Funktion ist

- das Einlesen von Wortgruppen im Modus „Einlesen“ bzw.
- das Übersetzen einer im System vorhandenen Referenzwortgruppe `spruebvarid` mittels aktueller Wortgruppe `wortgruppe` in der aktuellen Sprache `spr.sprachen.id` im Modus „Übersetzen“.

### **Modus „Einlesen“**

Konnte die aktuelle Wortgruppe `wortgruppe` zur gewünschten Sprache `spr.sprachen.id` im System gefunden werden, wird die erste passende Übersetzungsvariante `spr.uebvar.id` zurückgegeben. Andernfalls wird sie mit der nächsten verfügbaren `spr.uebvar.id` eingelesen.

### **Modus „Übersetzen“**

Im System werden Übersetzungsvarianten mittels `spr.uebvar.id` gebildet. Sie stellen den konstanten Bezug auf Wortgruppen verschiedener Sprachen, welche in einem gemeinsamen Kontext stehen, dar. Innerhalb einer Übersetzungsvariante darf nur eine Übersetzung je Sprache zugeordnet werden. Somit ist es möglich unter Angabe der Sprache und Identifikationsnummer der Übersetzungsvariante die korrekte Wortgruppe zu ermitteln.

Zum korrekten Verständnis der Funktionalität im Modus „Übersetzen“ sind die folgenden Randbedingungen zu beachten:

1. Wird die aktuelle Wortgruppe zur gewünschten Sprache innerhalb der vorgegebenen Übersetzungsvariante gefunden, so wird die Identifikationsnummer der Übersetzungsvariante zurückgegeben.
2. Wird innerhalb der vorgegebenen Übersetzungsvariante keine Übersetzung in der gewünschten Sprache gefunden, wird die aktuelle Wortgruppe zur gewünschten Sprache eingelesen.
3. Wird die vorgegebenen Übersetzungsvariante nicht gefunden, so wird eine Fehlermeldung ausgegeben und das Einlesen abgebrochen.
4. Wird innerhalb der vorgegebenen Übersetzungsvariante eine Übersetzung in der gewünschten Sprache gefunden (Sprache ist bereits belegt), so wird eine Fehlermeldung ausgegeben und das Einlesen abgebrochen.

## Beispiele

```
-- Fehlermeldung, die Wortgruppe ist anzugeben
SELECT * FROM spr.nimm_wgruppe( NULL, '1', NULL );
SELECT * FROM spr.nimm_wgruppe( '', '1', NULL );

-- Fehlermeldung, die Identifikationsnummer der Sprache muß größer 0 sein
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Test .', '0', NULL );

-- Fehlermeldung, die Identifikationsnummer der Übersetzungsvariante muß größer/gleich
0 sein
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Test .', '1', NULL );
SELECT * FROM spr.nimm_wgruppe( 'das ist ein Array', '1', '-1' );

-- Fehlermeldung, die Identifikationsnummer der Übersetzungsvariante muß im System
vorhanden sein
SELECT * FROM spr.nimm_wgruppe( 'das ist ein Array', '1', '999999' );

-- Fehlermeldung, die Übersetzungsvariante ist bereits mit einer Übersetzung in
der gewünschten Sprache belegt
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Array.', '1', '0' );
SELECT * FROM spr.nimm_wgruppe( 'This is an Array.', '2', '1' );
SELECT * FROM spr.zeige_wgruppe;
SELECT * FROM spr.nimm_wgruppe( '-----This is an Array.', '2', '1' );

-- Modus „Einlesen\“:
-- Die aktuelle Wortgruppe konnte nicht im System gefunden werden. Sie wird neu
angelegt.
SELECT * FROM spr.zeige_wgruppe;
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Array.', '1', '0' );
SELECT * FROM spr.zeige_wgruppe;

-- Modus „Einlesen\“:
-- Die aktuelle Wortgruppe konnte im System gefunden werden.
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Array.', '1', '0' );
SELECT * FROM spr.zeige_wgruppe ORDER BY spr.uebvar_id DESC LIMIT 5;
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Array.', '1', '0' );
SELECT * FROM spr.zeige_wgruppe ORDER BY spr.uebvar_id DESC LIMIT 5;

-- Modus „Übersetzen\“:
-- Die aktuelle Wortgruppe konnte nicht im System gefunden werden. Die Übersetzungsvar
wird neu angelegt.
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Array.', '1', '0' );
SELECT * FROM spr.zeige_wgruppe;
SELECT * FROM spr.nimm_wgruppe( 'This is an Array.', '2', '1' );
SELECT * FROM spr.zeige_wgruppe;
```

```

-- Modus „Übersetzen\“:
-- Die aktuelle Wortgruppe konnte im System gefunden werden.
SELECT * FROM spr.nimm_wgruppe( 'Das ist ein Array.', '1', '0' );
SELECT * FROM spr.nimm_wgruppe( 'This is an Array.', '2', '1' );
SELECT * FROM spr.zeige_wgruppe;
SELECT * FROM spr.nimm_wgruppe( 'This is an Array.', '2', '1' );
SELECT * FROM spr.zeige_wgruppe;

```

## 8.4 spr.zeige\_wgelemente

### Beschreibung der Spalten

spr\_uebersetzung\_id siehe spr.uebersetzung.id

spr\_uebvar\_id siehe spr.uebvar.id

spr\_sprachen\_id siehe spr.sprachen.id

spr\_wgruppe\_id siehe spr.wgruppe.id

wgelement siehe intern.abczk.id ODER intern.restzk.id

### Beschreibung der Sicht

Es werden die Elemente der Wortgruppen zurückgegeben.

## 8.5 spr.zeige\_wgruppe

### Beschreibung der Spalten

spr\_uebersetzung\_id siehe spr.zeige\_wgelemente.spr\_uebersetzung\_id

spr\_uebvar\_id siehe spr.zeige\_wgelemente.spr\_uebvar\_id

spr\_sprachen\_id siehe spr.zeige\_wgelemente.spr\_sprachen\_id

spr\_wgruppe\_id siehe spr.zeige\_wgelemente.spr\_wgruppe\_id

wgelement siehe spr.zeige\_wgelemente.wgelement

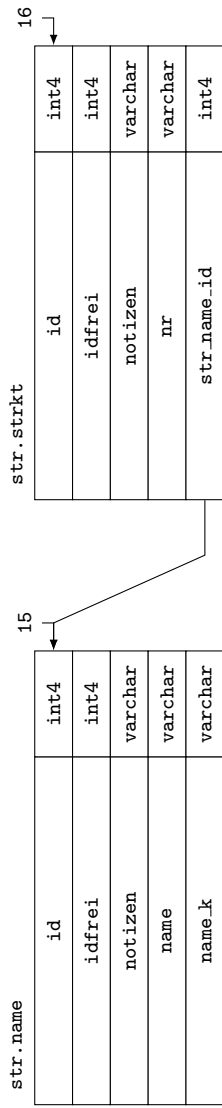
### Beschreibung der Sicht

Es werden die vollständigen Wortgruppen zurückgegeben.

## **9 str**

Das Schema „Strukturen“ dient der Verwaltung sämtlicher Strukturen (z.B. Fahrzeugstruktur, Gefahrennetz, Fehlerbaumstrukturen)

## 9.1 Tabellenhierarchie der Ebenen 1 — 3



## 9.2 Schlüssel und Bedingungen

### Primärschlüssel

pk\_\_str\_name\_id für Spalte str.name.id

pk\_\_str\_strkt\_id für Spalte str.strkt.id

### Fremdschlüssel

fk\_\_str\_strkt\_\_str\_name\_id aus Spalte str.strkt.str\_name\_id

### Unique Constraints

uc\_\_str\_name in Tabelle str.name für die Spalte(n) str.name.idfrei, str.name.name, str.name.name\_k

uc\_\_str\_str\_strkt in Tabelle str.strkt für die Spalte(n) str.strkt.idfrei, str.strkt.nr, str.strkt.str\_name\_id

## 9.3 str.pruefe\_\_fbwerte

### Funktionsname

str.pruefe\_\_fbwerte( IN aspekt varchar, IN wert varchar, IN fktname varchar, IN paramname varchar )

### Funktionsparameter/Rückgabewert(e)

aspekt Sollen Fehlerbaum-Ereignisse geprüft werden, ist FB-Ereignis anzugeben. Sollen Fehlerbaum-Tore geprüft werden, ist FB-Tor anzugeben.

wert Es ist der Wert anzugeben, dessen Plausibilität geprüft werden soll.

fktname Es ist der Name der Funktion anzugeben, welche die Prüfroutine aufruft. Der Funktionsname wird in der Fehlermeldung verwendet. Sollte er leer oder NULL sein, wird str.pruefe\_\_fbwerte(...) verwendet.

paramname Es ist der Name des Parameters anzugeben, welcher den zu prüfenden Wert wert beinhaltet. Der Parametername wird in der Fehlermeldung verwendet. Sollte er leer oder NULL sein, wird wert verwendet.

### Funktionsbeschreibung

Ziel der Funktion ist die Plausibilitätsprüfung gültiger Fehlerbaum-Ereignisse bzw. -Tore.

FB-Ereignis - gültige Werte nach DIN IEC 61025:1993-12):

**EBB** Ereignis-Beschreibungs-Block

**GE** Grundereignis

**NUE** Nicht weiter untersuchtes Ereignis

**AUE** Anderweitig untersuchtes Ereignis

**HAUS** Haus

**NULL** Null-Ereignis

**VWVA** Verweisungs-Zeichen von außen

**VWNA** Verweisungs-Zeichen nach außen

FB-Tor - gültige Werte nach DIN IEC 61025:1993-12):

**UND** UND-Verknüpfung

**ODER** ODER-Verknüpfung

**XODER** EXCLUSIV-ODER-Verknüpfung

**NICHT** NICHT-Verknüpfung

**BED** Bedingungs-Verknüpfung

**RED** Redundante Struktur

**ALLG** Verknüpfung (allgemein)

### Beispiele

-- Fehlermeldung. Der übergebene Aspekt ist ungültig.

```
SELECT * FROM str.pruefe_fbwerte( NULL, 'HAUS', NULL, NULL );
```

-- Fehlermeldung. Der übergebene Wert darf Nicht NULL oder leer sein, bzw. muß dem gültigen Wert entsprechen

```
SELECT * FROM str.pruefe_fbwerte( 'FB-Ereignis', NULL, NULL, NULL );
```

```
SELECT * FROM str.pruefe_fbwerte( 'FB-Ereignis', 'ungueltig', NULL, NULL );
```

```
SELECT * FROM str.pruefe_fbwerte( 'FB-Tor', 'ungueltig', 'fkname', 'fktparam' );
```

-- Die Werte sind gültig

```
SELECT * FROM str.pruefe_fbwerte( 'FB-Ereignis', 'HAUS', NULL, NULL );
```

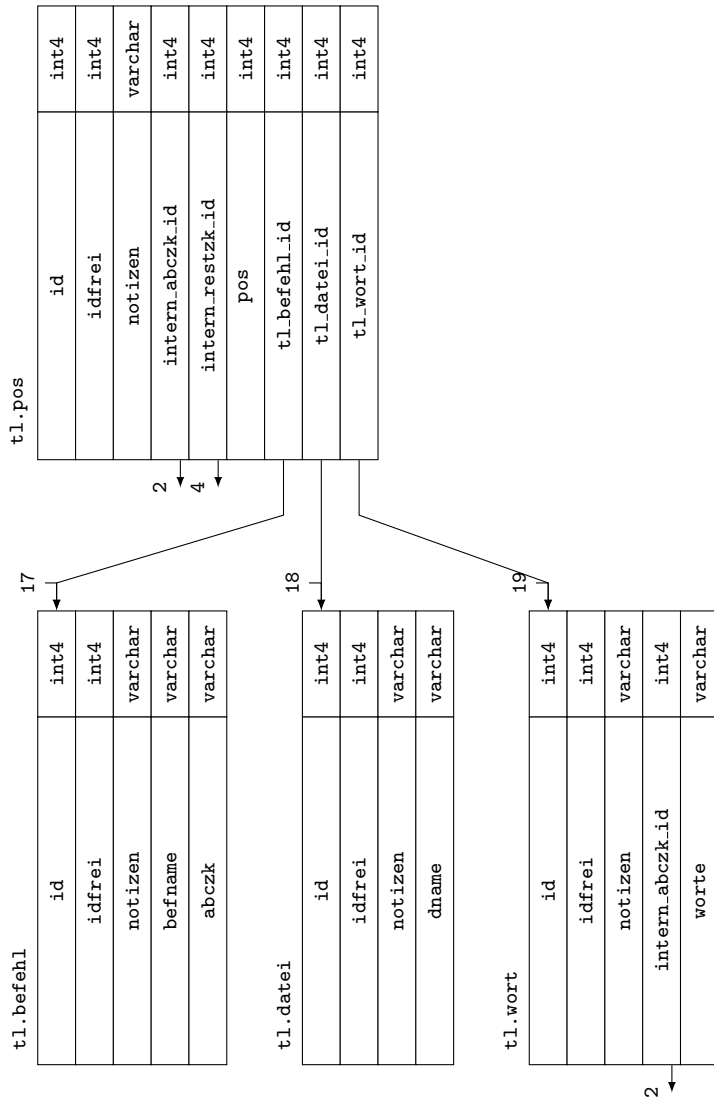
```
SELECT * FROM str.pruefe_fbwerte( 'FB-Tor', 'UND', NULL, NULL );
```



## 10 tl

Das Schema „Tex/Latex“ dient der Verwaltung der verwendeten  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  - Befehle,  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  - Wörter und sonstigen in  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  - Dateien auftretenden Zeichenketten.

## 10.1 Tabellenhierarchie der Ebenen 1 — 3



## 10.2 Schlüssel und Bedingungen

### Primärschlüssel

pk\_\_tl\_befehl\_id für Spalte tl.befehl.id

pk\_\_tl\_datei\_id für Spalte tl.datei.id

pk\_\_tl\_pos\_id für Spalte tl.pos.id

pk\_\_tl\_wort\_id für Spalte tl.wort.id

### Fremdschlüssel

fk\_\_tl\_pos\_\_intern\_abczk\_id aus Spalte tl.pos.intern\_abczk\_id

fk\_\_tl\_pos\_\_intern\_restzk\_id aus Spalte tl.pos.intern\_restzk\_id

fk\_\_tl\_pos\_\_tl\_befehl\_id aus Spalte tl.pos.tl\_befehl\_id

fk\_\_tl\_pos\_\_tl\_datei\_id aus Spalte tl.pos.tl\_datei\_id

fk\_\_tl\_pos\_\_tl\_wort\_id aus Spalte tl.pos.tl\_wort\_id

fk\_\_tl\_wort\_\_intern\_abczk\_id aus Spalte tl.wort.intern\_abczk\_id

### Unique Constraints

uc\_\_tl\_befehl in Tabelle tl.befehl für die Spalte(n) tl.befehl.idfrei, tl.befehl.befname, tl.befehl.abczk

uc\_\_tl\_datei in Tabelle tl.datei für die Spalte(n) tl.datei.idfrei, tl.datei.dname

uc\_\_tl\_pos in Tabelle tl.pos für die Spalte(n) tl.pos.idfrei, tl.pos.pos, tl.pos.tl\_datei\_id

uc\_\_tl\_wort in Tabelle tl.wort für die Spalte(n) tl.wort.idfrei, tl.wort.intern\_abczk\_id, tl.wort.worte

### Check Constraints

cc\_\_tl\_pos in Tabelle tl.pos die Bedingung (((((((intern\_abczk\_id = 0) AND (intern\_restzk\_id = 0)) AND (tl\_befehl\_id = 0)) AND (tl\_wort\_id = 0)) OR (((intern\_abczk\_id > 0) AND (intern\_restzk\_id = 0)) AND (tl\_befehl\_id = 0)) AND (tl\_wort\_id = 0))) OR (((intern\_abczk\_id = 0) AND (intern\_restzk\_id > 0)) AND (tl\_befehl\_id = 0)) AND (tl\_wort\_id = 0))) OR (((intern\_abczk\_id = 0) AND (intern\_restzk\_id = 0)) AND (tl\_befehl\_id > 0)) AND (tl\_wort\_id = 0))) OR (((intern\_abczk\_id = 0) AND (intern\_restzk\_id = 0)) AND (tl\_befehl\_id = 0)) AND (tl\_wort\_id > 0)))

## 10.3 tl.erzeuge\_\_abczk

### Funktionsname

tl.erzeuge\_\_abczk( IN tlwort varchar, OUT abczk varchar )

### Funktionsparameter/Rückgabewert(e)

tlwort Es ist das Wort anzugeben, aus welchem die AbCZk extrahiert werden soll. ACHTUNG ! Das '\ ' ist doppelt anzugeben (zu maskieren). Bsp.: tl.erzeuge\_\_ABCZk( 'Änderung' )

abczk Es wird die aus dem TL-Wort erzeugte AbCZk zurückgegeben.

### Funktionsbeschreibung

Ziel der Funktion ist das Erzeugen einer AbCZk abczk aus dem übergebenen TL-Wort tlwort. Es werden alle TL-Zweizeichenbefehle in tlwort mittels der in tl.befehl.abczk zugeordneten AbCZk ersetzt.

Sollte tlwort leer sein, oder beim Erzeugen ein Fehler auftreten bzw. abczk kein AbCZk darstellen, wird eine Fehlermeldung generiert.

### Beispiele

-- Fehlermeldung. Der Parameter tlwort darf nicht leer oder NULL sein.

```
SELECT * FROM tl.erzeuge__ABCZk( '' );
```

-- Fehlermeldung. Es konnten keine TL-Zweizeichenbefehle gefunden werden.

```
SELECT * FROM tl.erzeuge__ABCZk( 'Änderung' );
```

-- Änderung zurückgeben

```
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['abczk', 'befname'], ARRAY['Ä', 'Ä'] );
```

```
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['befname'], ARRAY[E''] );
```

```
SELECT * FROM tl.erzeuge__ABCZk( E'Änderung' );
```

## 10.4 tl.erzeuge\_\_dbmodell

### Funktionsname

tl.erzeuge\_\_dbmodell( OUT tldateinhalt varchar )

### Funktionsparameter/Rückgabewert(e)

tldateinhalt Es wird der Inhalt der TeX/LaTeX-Datei zurückgegeben, welche das Datenbankmodell enthält.

### Funktionsbeschreibung

Ziel der Funktion ist das Erzeugen einer TeX/LaTeX-Datei, welche das Datenbankmodell enthält. Sie wird aus den Kommentaren erzeugt, welche den verschiedenen Datenbankobjekten zugeordnet wurden. Es werden die Kommentare der Datenbank, Schemas, Tabellen,

Spalten, Funktionen, Primär- und Fremdschlüssel, sowie die `unique` und `check constraints` verwendet.

In den Kommentaren sind `\` entsprechend zu doppeln. Unterstrichen werden automatisch `\` vorangestellt, weshalb zum momentanen Zeitpunkt keine mathematischen Formeln eingegeben werden können.

Hinweise zur Benutzung:

Um nicht auf eine separate Ausleseroutine angewiesen zu sein, wird folgende (in `psql`) auszuführende Handlungsweise empfohlen:

```
\pset tuples_only \pset format unaligned \pset footer
SELECT * FROM tl.erzeuge_DBModell() \o dateiname.tex
;
```

Alternativ kann in der Kommandozeile

```
psql -U micha vf -P tuples_only -P format=unaligned -P footer -c 'SELECT * FROM
tl.erzeuge_DBModell()' -o dateiname.tex
eingegeben werden.
```

## Beispiele

```
-- Datenbankmodell ausgeben
SELECT * FROM tl.erzeuge_DBModell();
```

## 10.5 tl.erzeuge\_tldbgl

### Funktionsname

```
tl.erzeuge_tldbgl( IN begriff varchar, IN gloprfx varchar, IN bechreibung
varchar, OUT tldbgl varchar )
```

### Funktionsparameter/Rückgabewert(e)

`begriff` Der im Textdokument und Glossar darzustellende Begriff (darf nicht leer sein).

`gloprfx` Im Glossar wird `gloPrfx` dem `begriff` vorangestellt (zur Positionierung im Glossar)  
- darf leer sein.

`bechreibung` Die im Glossar dem `begriff` zuzuordnende Beschreibung (darf leer sein).

`tldbgl` Der im TL-Dokument einzusetzende `\dbGlossary...` - Befehl.

### Funktionsbeschreibung

Ziel der Funktion ist das Erzeugen eines `\dbGlossary...` - Befehls welcher im TL-Dokument verwendet werden kann.

Sollte `bechreibung` leer sein, wird ausschließlich `begriff` zurückgegeben.

## Beispiele

```

-- Fehlermeldung. Der Parameter begriff darf nicht leer oder NULL sein.
SELECT * FROM tl.erzeuge_TLdbGlossary( NULL, '', '' );
SELECT * FROM tl.erzeuge_TLdbGlossary( '', '', '' );

-- Glossareintrag zurückgeben
SELECT * FROM tl.erzeuge_TLdbGlossary( 'cc_intern_alle', 'intern.alle', NULL );
SELECT * FROM tl.erzeuge_TLdbGlossary( 'cc_intern_alle', 'intern.alle', '' );
SELECT * FROM tl.erzeuge_TLdbGlossary( 'cc_intern_alle', 'intern.alle', 'Es sind
die Zeichenketten einzugeben.' );

```

## 10.6 tl.erzeuge\_tlzk

### Funktionsname

```
tl.erzeuge_tlzk( IN zk varchar, OUT tlzk varchar )
```

### Funktionsparameter/Rückgabewert(e)

zk Es ist die in  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  zu wandelnde Zeichenkette anzugeben.

tlzk Es wird die in  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  gewandelte Zeichenkette zurückgegeben.

### Funktionsbeschreibung

Ziel der Funktion ist das Umwandeln von Zeichenketten in  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Zeichenketten, welche im Absatzmodus akzeptiert werden können.

### Beispiele

```
-- Es wird eine Leerzeichenkette bzw. NULL zurückgegeben
```

```
SELECT * FROM tl.erzeuge_TLzk( '' );
SELECT * FROM tl.erzeuge_TLzk( NULL );
```

```
-- Es wird intern_abczk_id zurückgegeben
```

```
SELECT * FROM tl.erzeuge_TLzk( 'intern_abczk_id' );
```

```
-- Es wird normale Zk zurückgegeben
```

```
SELECT * FROM tl.erzeuge_TLzk( 'normale Zk' );
```

## 10.7 tl.gib\_datei

### Funktionsname

```
tl.gib_datei( IN dateiname varchar, OUT dateiinhalte varchar )
```

### Funktionsparameter/Rückgabewert(e)

dateiname Vollständiger Dateiname (einschl. Dateierweiterung) der auszugebenden TL-Datei

dateiinhalt Vollständiger Inhalt der auszugebenden TL-Datei

### Funktionsbeschreibung

Ziel der Funktion ist das Herausgeben des Inhalts der mittels `dateiname` spezifizierten  $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -Quelltextdatei (TL-Datei).

Hinweise zur Benutzung:

Um nicht auf eine separate Ausleseroutine angewiesen zu sein, wird folgende (in `psql`) auszuführende Handlungsweise empfohlen:

```
\pset tuples_only
\pset format unaligned
\pset footer
SELECT * FROM tl.gib__datei( 'dateiname.tex' ) \o dateiname.tex
;
\q
```

Alternativ kann in der Kommandozeile

```
psql -U micha vf -P tuples_only -P format=unaligned -P footer -c ''SELECT * FROM
tl.gib__datei( 'dateiname.tex' );'' -o dateiname.tex
eingegeben werden.
```

### Beispiele

-- Fehlermeldung. Der Parameter `dateiname` darf nicht leer oder NULL sein.

```
SELECT * FROM tl.gib__datei( NULL );
```

-- Dateiinhalt einlesen und wieder ausgeben

```
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['abczk', 'befname'], ARRAY['Ä', '
Ä'] );
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['befname'], ARRAY[E'\-'] );
SELECT intern.nimm__zeile( 'tl.datei', ARRAY['dname'], ARRAY['dateiname.tex'] );
SELECT * FROM tl.nimm__datei(
E'Än\ -de\ -rung
Än\ -de\ -rungHo\ -sen\ -bein
'' 2006-01-01 #1 Änderung
norm-dt_01050_din_en
\befehl'
, 'dateiname.tex' );
SELECT * FROM tl.gib__datei( 'dateiname.tex' );
```

## 10.8 tl.gib\_\_ere\_tlbefehl

### Funktionsname

```
tl.gib__ere_tlbefehl( OUT ere_tlbefehl varchar )
```

## Funktionsparameter/Rückgabewert(e)

`ere_tlbehehl` Ist der erweiterte reguläre Ausdruck zur Beschreibung eines TL-Befehls

### Funktionsbeschreibung

Ziel der Funktion ist das Definieren und Zurückgeben eines erweiterten regulären Ausdrucks (ERE), welcher das Textmuster eines  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  - Befehls (TL-Befehl) beschreibt.

TL-Befehle beginnen mit einem `o` oder `@` oder Buchstaben dem ein Buchstabe folgen muß. TL-Befehle, welche mit einem Buchstaben beginnen, müssen mindestens ein `-` `_` oder eine Ziffer enthalten.

TL-Zweizeichenbefehle werden über `tl.gib_ERE_TLZZBefehl()` abgefangen.

TL-Befehle, welche mit `-` `_` oder einer Ziffer beginnen, werden bis zum ersten Buchstaben `intern.gib_ERE_RestZK()` abgebildet (TL-Befehl wird zerteilt).

Klammern und Sterne welche Teil eines TL-Befehls sein können, werden mittels `intern.gib_ERE_RestZK()` abgebildet (TL-Befehl wird zerteilt).

Es ist besonders zu beachten, daß auch `ABCZk` (siehe `intern.gib_ERE_ABCZK()`) beschrieben werden, solange sie keine deutschen Sonderzeichen (ÄÖÜ, ...) enthalten ! Das ist von folgenden Routinen abzufangen !

### Beispiele

-- ERE zurückgeben

```
SELECT * FROM tl.gib_ERE_TLBefehl();
```

-- Es wird `test` zurückgegeben

```
SELECT substring( 'test' from tl.gib_ERE_TLBefehl() );
```

-- Es wird `\befehl` zurückgeben

```
SELECT substring( '\befehl' from tl.gib_ERE_TLBefehl() );
```

-- Es wird `@manual` zurückgeben

```
SELECT substring( '@manual' from tl.gib_ERE_TLBefehl() );
```

-- Es wird `abs2def` zurückgegeben

```
SELECT substring( 'abs2def' from tl.gib_ERE_TLBefehl() );
```

-- Es wird `absdef2` zurückgegeben

```
SELECT substring( 'absdef2' from tl.gib_ERE_TLBefehl() );
```

-- Es wird `norm-dt_01050_din_en` zurückgegeben

```
SELECT substring( 'norm-dt_01050_din_en' from tl.gib_ERE_TLBefehl() );
```

-- Es wird eine Leerzeichenkette zurückgegeben

```
SELECT substring( '\ ' from tl.gib_ERE_TLBefehl() );
```



```

-- Es wird eine Leerzeichenkette zurückgeben
SELECT substring( '\\\ ' from tl.gib_ERE_TLBefehl() );

-- Es wird eine Leerzeichenkette zurückgeben
SELECT substring( '\@befehl' from tl.gib_ERE_TLBefehl() );

-- Es wird \bef zurückgeben (leider)
SELECT substring( '\bef@befehl' from tl.gib_ERE_TLBefehl() );

-- Es wird eine Leerzeichenkette zurückgeben
SELECT substring( '2absdef' from tl.gib_ERE_TLBefehl() );

-- Es wird eine Leerzeichenkette zurückgeben
SELECT substring( '-no-dt_01050_din_en' from tl.gib_ERE_TLBefehl() );

-- Es wird eine Leerzeichenkette zurückgeben
SELECT substring( '_no-dt_01050_din_en' from tl.gib_ERE_TLBefehl() );

-- Es wird eine Leerzeichenkette zurückgeben
SELECT substring( '2no-dt_01050_din_en' from tl.gib_ERE_TLBefehl() );

```

## 10.9 tl.gib\_ere\_tlwort

### Funktionsname

```
tl.gib_ere_tlwort( OUT ere_tlwort varchar )
```

### Funktionsparameter/Rückgabewert(e)

`ere_tlwort` Ist der erweiterte reguläre Ausdruck zur Beschreibung eines TL-Wortes.

### Funktionsbeschreibung

Ziel der Funktion ist das Definieren und Zurückgeben eines erweiterten regulären Ausdrucks (ERE), welcher das Textmuster eines TL-Wortes beschreibt.

Ein TL-Wort beginnt mit einem Buchstaben oder mit einem in Kombination mit dem zulässigen Buchstaben.

In der Folge setzt es sich aus TL-Zweizeichenbefehlen (siehe `tl.gib_ERE_TLZZBefehl()`) und `ABCZk` (siehe `intern.gib_ERE_ABCZK()`) zusammen.

Es ist besonders zu beachten, daß auch ASCII-Wörter (siehe `intern.gib_ERE_ABCZK()`) beschrieben werden, solange sie keine deutschen Sonderzeichen (ÄÖÜ, ...) enthalten ! Das ist von folgenden Routinen abzufangen !

### Beispiele

```

-- ERE zurückgeben
SELECT * FROM tl.gib_ERE_TLWort();

```

```

-- Es wird test zurückgegeben (leider)
SELECT substring( 'test' from tl.gib__ERE_TLWort() );

-- Es wird eine Leerzeichenkette zurückgegeben
SELECT substring( E'\Aistfalsch' from tl.gib__ERE_TLWort() );

-- Es wird Äßn-de-rung zurückgegeben
SELECT substring( E'Äßnderung' from tl.gib__ERE_TLWort() );

-- Es wird Äßnde-rungä zurückgegeben
SELECT substring( E'Äßnderungä' from tl.gib__ERE_TLWort() );

-- Es wird ßnderungä zurückgegeben (leider)
SELECT substring( E'ßnderungä' from tl.gib__ERE_TLWort() );

-- Es wird Äßmann zurückgegeben
SELECT substring( 'Äßmann' from tl.gib__ERE_TLWort() );

```

## 10.10 tl.gib\_\_ere\_tlzzbefehl

### Funktionsname

tl.gib\_\_ere\_tlzzbefehl( IN istanfang bool, OUT ere\_tlzzbefehl varchar )

### Funktionsparameter/Rückgabewert(e)

**istanfang** Der erweiterte reguläre Ausdruck des Anfangs eines TL-Zweizeichenbefehls soll zurückgegeben werden

**ere\_tlzzbefehl** Ist der erweiterte reguläre Ausdruck zur Beschreibung eines TL-Zweizeichenbefehls.

### Funktionsbeschreibung

Ziel der Funktion ist das Definieren und Zurückgeben eines erweiterten regulären Ausdrucks (ERE), welcher das Textmuster eines TL-Zweizeichenbefehls beschreibt. Die folgenden TL-Zweizeichenbefehle werden vom ERE erfaßt:

- Ä
- ä
- Ö
- ö
- Ü
- ü
-

•

## Beispiele

```
-- ERE zurückgeben
SELECT * FROM tl.gib_ERE_TLZZBefehl( true );
SELECT * FROM tl.gib_ERE_TLZZBefehl( false );

-- Leerzeichenkette zurückgeben
SELECT substring( ' ' from tl.gib_ERE_TLZZBefehl( true ) );

-- ü zurückgeben
SELECT substring( 'ü' from tl.gib_ERE_TLZZBefehl( true ) );

-- zurückgeben
SELECT substring( ' ' from tl.gib_ERE_TLZZBefehl( false ) );
```

## 10.11 tl.gib\_\_liste\_tlzzbefehl

### Funktionsname

tl.gib\_\_liste\_tlzzbefehl( IN tlwort varchar, OUT listetlzzbefehl varchar[] )

### Funktionsparameter/Rückgabewert(e)

**tlwort** Es ist das Wort anzugeben, aus welchem die TL-Zweizeichenbefehle extrahiert werden sollen. ACHTUNG! Das '\ ' ist doppelt anzugeben (zu maskieren). Bsp.: tl.gib\_\_Liste\_TLZZBefehl 'Änderung' )

**listetlzzbefehl** Es wird die Liste der gefundenen TL-Zweizeichenbefehle als ARRAY vom Typ VARCHAR zurückgegeben. Sollte kein TL-Zweizeichenbefehl in tlwort enthalten oder tlwort leer sein, wird NULL zurückgegeben.

### Funktionsbeschreibung

Ziel der Funktion ist das Extrahieren von TL-Zweizeichenbefehlen aus einem Wort. Das Wort besteht ausschließlich aus Buchstaben und (ggf.) aus TL-Zweizeichenbefehlen. Es wird nicht geprüft, ob die extrahierten TL-Zweizeichenbefehle im System vorhanden sind.

## Beispiele

```
-- Es wird NULL zurückgegeben
SELECT * FROM tl.gib__Liste_TLZZBefehl( NULL );

-- Es wird NULL zurückgegeben
SELECT * FROM tl.gib__Liste_TLZZBefehl( ' ' );

-- Es wird NULL zurückgegeben
```

```

SELECT * FROM tl.gib__Liste_TLZZBefehl( 'Änderung' );

-- Es wird Ä zurückgegeben
SELECT * FROM tl.gib__Liste_TLZZBefehl( 'Änderung' );

-- Es wird Ä zurückgegeben
SELECT * FROM tl.gib__Liste_TLZZBefehl( E'Änderung' );

-- Es wird zurückgegeben
SELECT * FROM tl.gib__Liste_TLZZBefehl( E'Änderung' );

```

## 10.12 tl.ist\_\_tlwort

### Funktionsname

```
tl.ist__tlwort( IN tlwort varchar, OUT isttlwort bool )
```

### Funktionsparameter/Rückgabewert(e)

tlwort Es ist das zu testende Wort anzugeben. ACHTUNG ! Das '\ ' ist doppelt anzugeben (zu maskieren). Bsp.: tl.ist\_\_TLWort( 'Änderung' )

isttlwort Wenn istTLWort = false dann ist tlwort kein TL-Wort. Wenn istTLWort = true dann ist ein TL-Wort

### Funktionsbeschreibung

Ziel der Funktion ist das Prüfen, ob die übergebene Zeichenkette tlwort als TL-Wort interpretiert werden kann.

TL-Worte bestehen ausschließlich aus AbcZk in Kombination mit mindestens einem TL-Zweizeichenbefehl (Definition und Gültigkeit siehe tl.gib\_\_ERE\_TLWort() unter der Bedingung, daß intern.ist\_\_abczk( tlwort ) = false ist).

### Beispiele

```

-- Es wird false zurückgegeben
SELECT * FROM tl.ist__TLWort( '' );

-- Es wird false zurückgegeben
SELECT * FROM tl.ist__TLWort( NULL );

-- Es wird false zurückgegeben
SELECT * FROM tl.ist__TLWort( 'Hosenbein' );

-- Es wird false zurückgegeben
SELECT * FROM tl.ist__TLWort( 'Änderung__' );

-- Es wird true zurückgegeben

```

```

SELECT * FROM tl.ist__TLWort( 'Änderung' );

-- Es wird true zurückgegeben
SELECT * FROM tl.ist__TLWort( 'Änderung' );

-- Es wird true zurückgegeben
SELECT * FROM tl.ist__TLWort( 'Hosenbein' );

```

### 10.13 tl.nimm\_\_datei

#### Funktionsname

tl.nimm\_\_datei( IN dateiinhalte varchar, IN dateiname varchar, OUT tldateiid varchar )

#### Funktionsparameter/Rückgabewert(e)

dateiinhalte Vollständiger Inhalt der einzulesenden TL-Datei (auch Zusatzmodule wie `bibtex`)

dateiname Vollständiger Dateiname (einschl. Dateierweiterung) der einzulesenden TL-Datei

tldateiid Es wird die Identifikationsnummer des Namens der Datei zurückgegeben, deren Inhalt eingelesen wurde.

#### Funktionsbeschreibung

Ziel der Funktion ist das Einlesen des Inhalts der mittels `dateiname` spezifizierten  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Quelltextdatei (TL-Datei). Auch Quelltextdateien von TL-Zusatzpaketen (z.B. `bibtex`) werden als TL-Dateien interpretiert. Die Funktion extrahiert aus dem übergebenen TL-Dateiinhalte die enthaltenen TL-Objekte (siehe `tl.nimm_erkannte_Liste_TLObjekte(...)`) und liest sie in `tl.pos` ein.

Das Konzept zum Extrahieren der TL-Objekte kann der Beschreibung zu `tl.nimm_erkannte_Liste_TLObjekte` entnommen werden

#### Hinweise zur Benutzung:

Um nicht auf eine separate Einleseroutine angewiesen zu sein, wird folgende (in `psql` auszuführende Handlungsweise empfohlen):

1. Es ist sichergestellt, daß die einzulesende TL-Datei im  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  fehlerfrei kompiliert werden kann
2. Es ist eine Kopie der einzulesenden TL-Datei anzulegen
3. In der Kopie sind alle `\` mittels `\\` und alle `'` mittels `''` zu ersetzen

4. Dem ersten Zeichen in der Kopie ist der SQL-Teilbefehl  
`SELECT * FROM tl.nimm__datei( E'`  
 voranzustellen
5. Dem letzten Zeichen in der Kopie mit dem Dateinamen `dateiname.tex` ist der SQL-Teilbefehl  
`', 'dateiname.tex');`  
 hinzuzufügen
6. Im Postgres-Kommandozeilenprogramm `psql` ist mittels des Befehls  
`\i dateiname.tex`  
 die in den beschriebenen Schritten modifizierte Kopie einzulesen

## Beispiele

```
-- Fehlermeldung. Der Parameter dateinhalt darf nicht leer oder NULL sein.
SELECT * FROM tl.nimm__datei( '', 'dateiname.tex' );

-- Fehlermeldung. Der Parameter dateiname darf nicht leer oder NULL sein.
SELECT * FROM tl.nimm__datei( 'ttt', NULL );

-- Fehlermeldung. Der Dateiname muß im System vorhanden sein.
SELECT * FROM tl.nimm__datei( 'ttt', 'Unbekannter Dateiname.tex' );

-- Dateiinhalt einlesen und wieder ausgeben
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['abczk', 'befname'], ARRAY['Ä', '
Ä'] );
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['befname'], ARRAY[E'\-'] );
SELECT intern.nimm__zeile( 'tl.datei', ARRAY['dname'], ARRAY['dateiname.tex'] );
SELECT * FROM tl.nimm__datei(
E'Än\ -de\ -rung
Än\ -de\ -rungHo\ -sen\ -bein
'' 2006-01-01 #1 Änderung
norm-dt_01050_din_en
\befehl'
, 'dateiname.tex' );
SELECT * FROM tl.gib__datei( 'dateiname.tex' );
```

## 10.14 tl.nimm\_\_tlwort

### Funktionsname

```
tl.nimm__tlwort( IN tlwort varchar, IN tlwortid_korr varchar, OUT tlwortid varchar
)
```

### Funktionsparameter/Rückgabewert(e)

`tlwort` Einzulesendes TL-Wort. ACHTUNG ! Das `'\'` ist doppelt anzugeben (zu maskieren).

```
Bsp.: tl.nimm__TLWort( 'Ho
-sen
-bein', true )
```

`tlwortid_korr` Wird eine im System vorhandene `tl.wort.id` benannt, so wird das im System vorhandene TL-Wort überschrieben. Soll ein Wort hinzugefügt werden, so ist entweder `'0'` oder `NULL` anzugeben.

`tlwortid` Es wird die Identifikationsnummer des eingelesenen TL-Wortes zurueckgegeben.

### Funktionsbeschreibung

Ziel der Funktion ist das Einlesen des übergebenen TL-Wortes `tlwort` in die Tabelle `tl.wort`. Das gültige Textmuster eines TL-Wortes kann der Funktion `tl.ist__TLWort(...)` entnommen werden.

Die im TL-Wort enthaltenen TL-Zweizeichenbefehle werden extrahiert, deren `AbcZk`-Bedeutung aus `tl.befehl.abczk` ausgelesen und im TL-Wort an Stelle der TL-Zweizeichenbefehle eingesetzt. Die auf diese Weise erzeugte `AbcZk` wird in `intern.abczk` eingelesen und mit dem in `tl.wort` eingelesenen TL-Wort verknüpft.

Wird eine mittels `tlWortId_korr` übergebene `tl.wort.id` im System gefunden, so wird das im System vorhandene TL-Wort überschrieben. Andernfalls wird `tlwort` mit der vorgegebenen `tlWortId_korr` im System eingefügt.

### Beispiele

```
-- Fehlermeldung. Es liegt kein TL-Wort vor
```

```
SELECT * FROM tl.nimm__TLWort( '', NULL );
```

```
SELECT * FROM tl.nimm__TLWort( 'Hosenbain', NULL ); -- Fehlermeldung
```

```
-- Test
```

```
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['abczk', 'befname'], ARRAY['Ä', 'Ä']
);
```

```
SELECT intern.nimm__zeile( 'tl.befehl', ARRAY['befname'], ARRAY[E''] );
```

```
-- id automatisch zuordnen
```

```
SELECT * FROM tl.nimm__TLWort( 'Änderung', NULL );
```

```
-- neuer Eintrag mit id=99
```

```
SELECT * FROM tl.nimm__TLWort( 'Hosenbein', '99' );
```

```
-- vorhandene id=99 überschreiben
```

```
SELECT * FROM tl.nimm__TLWort( 'Hosenbeine', '99' );
```

```
-- Test
```

```
SELECT * FROM tl.wort;
```

```
SELECT * FROM intern.abczk WHERE abczk = 'Änderung';
```

```
SELECT * FROM intern.abczk WHERE abczk = 'Hosenbeine';
```

## 10.15 `tl.nimm_erkannte_liste_tlobjekte`

### Funktionsname

`tl.nimm_erkannte_liste_tlobjekte( IN dateinhalt varchar, OUT tabname varchar, OUT tabnameid varchar )`

### Funktionsparameter/Rückgabewert(e)

`dateinhalt` Vollständiger Inhalt der einzulesenden TL-Datei

`tabname` Name der Tabelle, in die das erkannte TL-Objekt eingelesen wurde

`tabnameid` id des erkannten TL-Objektes

### Funktionsbeschreibung

Ziel der Funktion ist das Zerlegen des Inhaltes `dateinhalt` einer TL-Datei in TL-Objekte mittels Erweiterter Regularer Ausdrücke (ERE). Die erkannten TL-Objekte werden bei Bedarf in die Datenbank eingelesen.

Aus dem übergebenen TL-Dateiinhalt extrahiert die Funktion

- `AbcZk` welche bei Bedarf in `intern.abczk` (Gültigkeit siehe `intern.gib_ERE_ABCZK()`),
- `RestZk` welche bei Bedarf in `intern.restzk` (Gültigkeit siehe `intern.gib_ERE_RestZK()`),
- TL-Befehle welche bei Bedarf in `tl.befehl` (Gültigkeit siehe `tl.gib_ERE_TLBefehl()`) und
- TL-Worte welche bei Bedarf in `tl.wort` (Gültigkeit siehe `tl.gib_ERE_TLWort()`) eingelesen werden.

Die im TL-Wort auftretenden TL-Zweizeichenbefehle müssen im System vorhanden sein.

Sämtliche Zeichenketten, welche nicht diesen Kategorien zuordenbar sind, werden zerlegt und dann entsprechend eingelesen. Zum Beispiel werden TL-Variablenamen, welche ausschliesslich aus Buchstaben bestehen, in `intern.abczk` und nicht in `tl.befehl` eingelesen. Die Ursache liegt in der Unfähigkeit der Funktion, eine Kontextprüfung durchzuführen.

Mit dem Verzicht des Einlesens von Wortgruppen (siehe `spr.wgruppe`) müssen mehrsprachige Texte mittels TL-Paketen gesetzt werden (z.B. `babel`). Ursache ist auch hier, die Unfähigkeit eine Kontextprüfung durchzuführen.

### Beispiele

-- Es wird keine Zeile zurückgegeben und nichts eingelesen

```
SELECT * FROM tl.nimm_erkannte_Liste_TLObjekte( ' );
```

0



```
-- Es wird die Zeichenkette analysiert, die TL-Objekte eingelesen und zurückgegeben  
SELECT * FROM tl.nimm_erkannte_Liste_TLObjecte( 'Änderung ÄnderungHosenbein 2006-01-01  
#1 Änderung norm-dt_01050_din_en \befehl' );
```



# A GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document free in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of „copyleft“, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The „**Document**“, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as „**you**“. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A „**Modified Version**“ of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A „**Secondary Section**“ is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The „**Invariant Sections**“ are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The „**Cover Texts**“ are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A „**Transparent**“ copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not „Transparent“ is called „**Opaque**“.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The „**Title Page**“ means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, „Title Page“ means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section „**Entitled XYZ**“ means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as „**Acknowledgements**“, „**Dedications**“, „**Endorsements**“, or „**History**“.) To „**Preserve the Title**“ of such a section when you modify the Document means that it remains a section „Entitled XYZ“ according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other

conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled „History“, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled „History“ in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the „History“ section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled „Acknowledgements“ or „Dedications“, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled „Endorsements“. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled „Endorsements“ or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled „Endorsements“, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled „History“ in the various original documents, forming one section Entitled „History“; likewise combine any sections Entitled „Acknowledgements“, and any sections Entitled „Dedications“. You must delete all sections Entitled „Endorsements“.

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an „aggregate“ if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled „Acknowledgements“, „Dedications“, or „History“, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## **10. FUTURE REVISIONS OF THIS LICENSE**



The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License „or any later version“ applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled „GNU Free Documentation License“.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the „with...Texts.“ line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software

# Verzeichnis der Tabellenspalten und Begriffe

Tabellenspalte / Begriff	Beschreibung	
<code>dok.doku</code>	Tabelle zur Verwaltung der Basisinformationen der Dokumente. Die Dokumente werden anhand des Eigentümers und der Dokumentennummer unterschieden.	11
<code>dok.doku.aspekt</code>	Klassifizierung des Dokumentes anhand der möglichen Aspekte unter denen es den Protokolleinträgen, Sicherheitsanforderungen, Terminpläneinträgen, ... zugeordnet wird. Die möglichen Aspekte sind <code>dok.nimm_doku()</code> zu entnehmen.	11
<code>dok.doku.intern_firma_id</code>	Identifikationsnummer der Firma, welche Eigentümer des Dokumentes ist.	11
<code>dok.doku.spr_uebvar_id_kb</code>	Identifikationsnummer der Übersetzungsvariante, welche die Kurzbeschreibung des Inhaltes (Zweckes, Zieles, ...) des Dokumentes darstellt.	11
<code>dok.dokucc__dok_doku</code>	Es darf ausschließlich BEWEIS oder QUELLE angegeben werden.	12
<b>GNU Free Documentation License</b>	<i>Der Zweck dieser Lizenz ist es, ein Handbuch, Textbuch oder ein anderes zweckdienliches und nützliches Dokument frei, im Sinne von Freiheit, zu machen; jedermann die Freiheit zu sichern, es zu kopieren und mit oder ohne Änderungen daran, sowohl kommerziell als auch nicht kommerziell weiter zu verbreiten. Weiterhin sichert diese Lizenz einem Autor oder Verleger die Möglichkeit, Anerkennung für seine Arbeit zu erhalten ohne für Änderungen durch Andere verantwortlich gemacht zu werden.</i> (siehe <a href="http://www.giese-online.de/gnufdl-de.html">http://www.giese-online.de/gnufdl-de.html</a> )	3
<code>intern.abczk</code>	Tabelle die alle Zeichenketten aufnimmt, welche ausschliesslich aus Buchstaben bestehen. Gross-/Kleinschreibung wird unterschieden.	15
<code>intern.abczk.abczk</code>	Es sind die Zeichenketten einzugeben.	15

<b>Tabellenspalte / Begriff</b>	<b>Beschreibung</b>	
<code>intern.alle</code>	Tabelle, welche die Grundstruktur für alle weiteren Tabellen bildet. Neben der Primärschlüsselspalte wird die Spalte zur Wiederverwendung freigegebener <code>id</code> und eine Spalte zur Aufnahme interner Notizen zur Verfügung gestellt.	15
<code>intern.alle.id</code>	Primärschlüsselspalte: Diese Spalte steht in allen Tabellen jedes Schemas an erster Stelle. In vielen Fällen wird auf sie mittels Fremdschlüssel referenziert.	15
<code>intern.alle.idfrei</code>	Hilfsspalte zur Id-Verwaltung: Diese Spalte folgt unmittelbar <code>id</code> . Sie wird mit dem aktuellen Wert in <code>id</code> versehen, wenn die aktuelle Zeile als gelöscht angesehen werden soll und die zugeordnete <code>id</code> für neue Einträge zur Verfügung steht, sonst wird 0 eingetragen.	15
<code>intern.alle.notizen</code>	Interne Notizen / Anmerkungen: Diese Spalte folgt unmittelbar <code>idfrei</code> . Sie dient der Aufnahme interner Notizen, die nicht in offiziellen Auswertungen Verwendung finden. Sie kann von allen Nutzern verwendet werden, die Werte in die Datenbank eingeben oder verändern.	15
<code>intern.allecc_intern_alle</code>	In die Tabelle <code>intern.alle</code> sollen keine Werte eingefügt werden.	17
<code>intern.allewlog</code>	Tabelle, welche die Grundstruktur für alle weiteren Tabellen mit Logbuchfunktion (Darstellen der Änderungshistorie) bildet. Neben den Informationen der Tabelle <code>intern.alle</code> wird die Uhrzeit des Eintrages, der Name des Benutzers, die Datenquelle (bei automatisch eingelesenen Daten) der Zustand, ob ein automatisch eingelesener Datensatz geprüft wurde und der Status, ob der Eintrag gelöscht wurde, gehalten.	15
<code>intern.allewlog.logdatuhrz</code>	Es werden Datum und Uhrzeit des aktuellen Eintrages aufgenommen (als <code>DEFAULT</code> ). Es wird die SQL-Funktion <code>timeofday()</code> verwendet welche sicherstellt, daß auch innerhalb einer Transaktion die aktuelle Zeit eingetragen wird.	15
<code>intern.allewlog.logdqok</code>	Es ist <code>true</code> anzugeben, wenn automatisch eingelesene Daten vom Spezialisten akzeptiert wurden.	15
<code>intern.allewlog.logdquelle</code>	Es ist die Datenquelle (bei automatisch eingelesenen Daten) des aktuellen Eintrags anzugeben.	15

<b>Tabellenspalte / Begriff</b>	
<code>intern.allowlog.logentfernt</code>	<b>Beschreibung</b>
<code>intern.allowlog.logname</code>	Es ist true anzugeben, wenn der aktuelle Eintrag gelöscht werden soll, sonst false. 15
<code>intern.allowlogcc...intern.allowlog</code>	Es ist der Datenbankbenutzername anzugeben, welcher den aktuellen Eintrag im System hinzugefügt hat. 15
<code>intern.firma</code>	In die Tabelle <code>intern.allowlog</code> sollen keine Werte eingefügt werden. 17
<code>intern.firma.name</code>	Tabelle zur Verwaltung der in den Projekten referenzierten Firmen. 15
<code>intern.firma.name_k</code>	Es ist der vollständige Name der Firma anzugeben. 15
<code>intern.restzk</code>	Es ist die Kurzbezeichnung des Namens der Firma anzugeben. 15
<code>intern.restzk.restzk</code>	Tabelle die alle Zeichenketten aufnimmt, welche nicht aus Buchstaben bestehen. 16
<code>prj.p.dokstrkt</code>	Es sind die Zeichenketten einzugeben. 16
<code>prj.p.dokstrkt.dok_doku_id</code>	Diese Tabelle dient der projektabhängigen Zuordnung der Strukturpunkte zu den Dokumenten. 39
<code>prj.p.dokstrkt.prj_projekte_id</code>	Identifikationsnummer des aktuellen Dokumentes. 39
<code>prj.p.dokstrkt.str_strkt_id</code>	Identifikationsnummer des aktuellen Projektes. 39
<code>prj.p.prtstrstrkt</code>	Identifikationsnummer des Strukturpunktes, welcher dem aktuellen Dokument zugeordnet ist. 39
<code>prj.p.prtstrstrkt.prj_projekte_id</code>	Diese Tabelle dient der projektabhängigen Zuordnung der Strukturpunkte zu den Protokolleinträgen / Sicherheitsanforderungen / Terminplaneinträgen. 39
<code>prj.p.prtstrstrkt.prt_protokoll_id</code>	Identifikationsnummer des aktuellen Projektes. 39
<code>prj.p.prtstrstrkt.str_strkt_id</code>	Es ist die Identifikationsnummer des PE, SF, TP anzugeben. 39
<code>prj.p.prtverbindung</code>	Identifikationsnummer des dem PE / SF / TP projektspezifisch zugeordneten Strukturpunktes. 39
<code>prj.p.prtverbindung.prj_projekte_id</code>	Diese Tabelle dient der projektabhängigen gegenseitigen Zuordnung der Protokolleinträge / Sicherheitsanforderungen / Terminplaneinträge. 39
	Identifikationsnummer des aktuellen Projektes. 39

Tabellenspalte / Begriff	Beschreibung	
<code>prj.p-prtverbindung.prt_protokoll_id_nachf</code>	Es ist die Identifikationsnummer des PE, SF, TP anzugeben, welcher dem <code>prt_protokoll_id_vorg</code> folgt.	39
<code>prj.p-prtverbindung.prt_protokoll_id_vorg</code>	Es ist die Identifikationsnummer des PE, SF, TP anzugeben, welcher der Vorgänger von <code>prt_protokoll_id_nachf</code> ist.	39
<code>prj.p-psabk</code>	Diese Tabelle dient der projektabhängigen Zuordnung der Zeichnungen der Abkürzungen.	40
<code>prj.p-psabk.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	40
<code>prj.p-psabk.ps_abk_id</code>	Identifikationsnummer der im Projekt gültigen Abkürzung.	40
<code>prj.p-psabk.spr_uebvar_id</code>	Identifikationsnummer der projektabhängig angepaßten Spalte <code>ps.abk.spr_uebvar_id</code> .	40
<code>prj.p-psabk.spr_uebvar_id_1</code>	Identifikationsnummer der projektabhängig angepaßten Spalte <code>ps.abk.spr_uebvar_id_1</code> .	40
<code>prj.p-psabk.spr_uebvar_id_zus</code>	Identifikationsnummer der projektabhängig angepaßten Spalte <code>ps.abk.spr_uebvar_id_zus</code> .	40
<code>prj.p-strverbindung</code>	Diese Tabelle dient der projektabhängigen gegenseitigen Zuordnung der Strukturpunkte.	40
<code>prj.p-strverbindung.fbtor</code>	Projektabhängig angepaßte Spalte <code>str.verbindung.fbtor</code> .	40
<code>prj.p-strverbindung.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	40
<code>prj.p-strverbindung.str_strkt_id_nachf</code>	Identifikationsnummer des Strukturpunktes, welcher auf einen weiteren Strukturpunkt verweist (diesem folgt).	40
<code>prj.p-strverbindung.str_strkt_id_vorg</code>	Identifikationsnummer des Strukturpunktes, welcher von einem weiteren Strukturpunkt referenziert wird.	40
<code>prj.p-strverbindungcc--prj_p-strverbindung</code>	Wenn <code>id</code> Null ist, müssen <code>str_strkt_id_vorg</code> und <code>str_strkt_id_nachf</code> Null sein. Sonst muß <code>str_strkt_id_vorg</code> immer größer Null sein.	50
<code>prj.personen</code>	Tabelle zur Verwaltung der in den Projekten eingebundenen Personen.	39
<code>prj.personen.intern_firma_id</code>	Es ist die Identifikationsnummer der Firma anzugeben, in welcher die Person ange stellt ist.	39
<code>prj.personen.name_n</code>	Es ist der Nachname der Person anzugeben.	39
<code>prj.personen.name_v</code>	Es ist der Vorname der Person anzugeben.	39

<b>Tabellenspalte / Begriff</b>	<b>Beschreibung</b>	
<code>prj.personen.notizen</code>	Es sind Zusatzinformationen (z.B. Position, Aufgabe, ...) zur Person anzugeben.	39
<code>prj.pl.dokausgabe</code>	Tabelle zur Verwaltung der Ausgaben (Revisionsstände) des aktuellen Dokumentes. Mittels Logbuchfunktion kann der Wechsel der Revisionsstände verfolgt werden.	41
<code>prj.pl.dokausgabe.datum</code>	Datum der aktuellen Ausgabe des Dokumentes.	41
<code>prj.pl.dokausgabe.dok_doku_id</code>	Identifikationsnummer des aktuellen Dokumentes.	41
<code>prj.pl.dokausgabe.istfrei</code>	<code>true</code> — Das Dokument ist in der aktuellen Ausgabe freigegeben (nicht mehr im Entwurfsstand).	41
<code>prj.pl.dokausgabe.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	41
<code>prj.pl.dokausgabe.stand</code>	Revisionsstand (Kennzeichen der Ausgabe) des aktuellen Dokumentes.	41
<code>prj.pl.prt_dokdoku.dok_doku_id</code>	Es ist die Datenbank-Identifikationsnummer des aktuellen Dokumentes anzugeben.	42
<code>prj.pl.prt_dokdoku.nr</code>	Identifikationsnummer (Dokumentennummer) des Dokumentes.	42
<code>prj.pl.prt_dokdoku.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	42
<code>prj.pl.prt_dokdoku.prt_protokoll_id</code>	Es ist die Identifikationsnummer des PE, SF, TP anzugeben.	42
<code>prj.pl.prt_dokdoku.spr_uebvar_id</code>	Identifikationsnummer der Übersetzungsvariante, welche den Titel des Dokumentes (Dokumententitel) darstellt.	42
<code>prj.pl.prt_dokdoku.spr_uebvar_id.zus</code>	Identifikationsnummer der Übersetzungsvariante, welche Zusatzinformationen wie Kapitelangaben, Verweise auf Abschnitte oder ähnliches darstellt.	42
<code>prj.pl.prtpsgefszenario</code>	Mittels dieser Tabelle wird die projektabhängige Logbuchfunktion für die Protokolleinträge bereitgestellt, die dem aktuellen Gefahrenszenario zugeordnet werden.	42
<code>prj.pl.prtpsgefszenario.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	42
<code>prj.pl.prtpsgefszenario.prt_protokoll_id</code>	Es ist die Identifikationsnummer des Protokolleintrages anzugeben, welcher dem aktuellen Gefahrenszenario zugeordnet werden soll.	42
<code>prj.pl.prtpsgefszenario.ps_gefszenario_id</code>	Es ist die Identifikationsnummer des aktuellen Gefahrenszenarios anzugeben.	42

<b>Tabellenspalte / Begriff</b>	<b>Beschreibung</b>	
prj.pl_psefabk	Mittels dieser Tabelle wird die projektabhängige Logbuchfunktion für die Abkürzungen bereitgestellt, die dem aktuellen Gefahrenszenario zugeordnet sind.	43
prj.pl_psefabk.prj_projekte_id	Identifikationsnummer des aktuellen Projektes.	43
prj.pl_psefabk.ps_abk_id	Identifikationsnummer der Abkürzung <b>ps.abk.id</b> .	43
prj.pl_psefabk.ps_gefszenario_id	Identifikationsnummer des Gefahrenszenarios, welchem die Abkürzung zugeordnet werden soll.	43
prj.pl_psefabk.spr_uebvar_id_zus	Identifikationsnummer der Übersetzungsvariante, die zusätzliche Informationen (z.B. Darstellung der Häufigkeit bzw. Gefahrenstufe bezogen auf das repräsentative Gefahrenszenario) enthält.	43
prj.pl_psefabk.ps_anmerkung_id	Identifikationsnummer der Anmerkung welche dem Gefahrenszenario zugeordnet und deren Text gesetzt werden soll.	43
prj.pl_psefabk.str_strukt	Mittels dieser Tabelle wird die projektabhängige Logbuchfunktion für die Strukturpunkte bereitgestellt, die dem aktuellen Gefahrenszenario zugeordnet werden.	44
prj.pl_psefabk.str_strukt_id	Identifikationsnummer des aktuellen Projektes.	44
prj.pl_psefabk.str_strukt_id	Es ist die Identifikationsnummer des aktuellen Gefahrenszenarios anzugeben.	44
prj.pl_psefabk.str_strukt_id	Es ist die Identifikationsnummer des Strukturpunktes anzugeben, welcher dem aktuellen Gefahrenszenario zugeordnet ist.	44
prj.pp_prtprotokoll	Diese Tabelle dient der projektabhängigen Zuordnung der Personen zu den Protokolleinträgen / Sicherheitsanforderungen / Terminiplaneinträgen. Weiterhin können verschiedene Eigenschaften projektabhängig umbenannt werden. Innerhalb dieser Tabelle werden drei Aspekte abgebildet: Protokolleintrag einem Projekt zurechnen, projektspezifisch modifizieren und Randbedingungen setzen; dem Protokolleintrag die beteiligten Personen zuordnen und den Protokolleintrag als Frage / Aufgabe formulieren. Datum / Uhrzeit des Entstehens des PE oder der SF. Für TP: Datum / Uhrzeit des Meilensteines bzw. Startpunkt einer Aktivität, welche über einen Zeitraum reicht.	44



<b>Tabellenspalte / Begriff</b>	<b>Beschreibung</b>	
<code>prj.pp_prtprotokoll.duanfrage</code>	Datum / Uhrzeit der Bitte um Antwort auf bzw. Stellungnahme zum aktuellen PE / SF / TP.	44
<code>prj.pp_prtprotokoll.erledigt</code>	Wenn der aktuellen PE, SF, TP erledigt ist, wird true angegeben.	44
<code>prj.pp_prtprotokoll.intern</code>	Wenn der aktuelle PE, SF, TP ausschließlich intern verwendet werden darf, ist true anzugeben.	44
<code>prj.pp_prtprotokoll.prj_personen_id</code>	Identifikationsnummer der Person, welche mit dem aktuellen PE / SF / TP in direktem Zusammenhang steht. Wurde <code>prj.ppu_prtprotokoll.teilnehmer = true</code> gesetzt, war die Person am / beim Entstehen des PE / SF / TP beteiligt.	44
<code>prj.pp_prtprotokoll.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	44
<code>prj.pp_prtprotokoll.prt_protokoll_id</code>	Identifikationsnummer des aktuellen PE / SF / TP.	44
<code>prj.pp_prtprotokoll.spr_uebvar_id</code>	Identifikationsnummer der projektabhängig angepaßten Spalte <code>prt.protokoll.spr_uebvar_id</code> .	44
<code>prj.pp_prtprotokoll.teilnehmer</code>	Wird true gesetzt, war die beteiligte Person am / beim Entstehen des PE / SF / TP beteiligt. Wird false gesetzt, wurde der aktuellen Person eine Aufgabe gestellt.	44
<code>prj.pp_prtprotokoll.termin</code>	Spätester Zeitpunkt (Datum / Uhrzeit) der Erledigung des aktuellen PE, SF, TP. Für TP: Zeitpunkt einer Aktivität, welche über einen Zeitraum reicht.	44
<code>prj.pp_psgefszenario</code>	Diese Tabelle dient der projektabhängigen Zuordnung der Beschreibung des Gefahrenszenarios einschließlich Zuordnung einer Person.	45
<code>prj.pp_psgefszenario.prj_personen_id</code>	Identifikationsnummer der im Projekt für das Gefahrenszenario verantwortlichen Person.	45
<code>prj.pp_psgefszenario.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	45
<code>prj.pp_psgefszenario.ps_gefszenario_id</code>	Identifikationsnummer des im Projekt gültigen Gefahrenszenarios.	45
<code>prj.pp_psgefszenario.spr_uebvar_id</code>	Identifikationsnummer der projektabhängig angepaßten Spalte <code>ps.gefszenario.spr_uebvar_id</code> .	45
<code>prj.pp_psgefszenario.spr_uebvar_id_reprgefszen</code>	Identifikationsnummer der projektabhängig angepaßten Spalte <code>ps.gefszenario.spr_uebvar_id_reprgefszen</code> .	45
<code>prj.pp_psgefszenario.spr_uebvar_id_ursp</code>	Identifikationsnummer der projektabhängig angepaßten Spalte <code>ps.gefszenario.spr_uebvar_id_ursp</code> .	45

Tabellenspalte / Begriff	Beschreibung	
<code>prj.pp_strstrkt</code>	Diese Tabelle dient der projektabhängigen Zuordnung der Personen zu den Strukturpunkten. Weiterhin können verschiedene Eigenschaften projektabhängig umbenannt werden.	45
<code>prj.pp_strstrkt.fbereignis</code>	Projektabhängig angepaßte Spalte <code>str.strkt.fbereignis</code> .	45
<code>prj.pp_strstrkt.prj_personen_id</code>	Identifikationsnummer der im Projekt für den Strukturpunkt verantwortlichen Person.	45
<code>prj.pp_strstrkt.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	45
<code>prj.pp_strstrkt.spr_uebvar_id</code>	Projektabhängig angepaßte Spalte <code>str.strkt.spr_uebvar_id</code> .	45
<code>prj.pp_strstrkt.str_strkt_id</code>	Identifikationsnummer des im Projekt gültigen Strukturpunktes.	45
<code>prj.ppl_dokdoku</code>	Diese Tabelle dient der projektabhängigen Zuordnung der Ansprechpartner (ggf. verantwortlichen Personen) zu den Dokumenten. Mittels Logbuchfunktion kann der Wechsel der Ansprechpartner verfolgt werden.	46
<code>prj.ppl_dokdoku.dok_doku_id</code>	Es ist die Datenbank-Identifikationsnummer des aktuellen Dokumentes anzugeben.	46
<code>prj.ppl_dokdoku.prj_personen_id</code>	Es ist die Identifikationsnummer des Ansprechpartners (ggf. der für das Dokument verantwortlichen Person) anzugeben.	46
<code>prj.ppl_dokdoku.prj_projekte_id</code>	Identifikationsnummer des aktuellen Projektes.	39
<code>prj.projekte</code>	Tabelle zur Verwaltung der projektspezifischen Informationen	39
<code>prj.projekte.makefile</code>	Es ist der Quelltext des <code>makefile</code> s anzugeben, welches der Erzeugung sämtlicher projektspezifischer Dokumente dient.	39
<code>prj.projekte.prjbeschr</code>	Es ist die Projektbeschreibung anzugeben. Hier sollen Besonderheiten des Projektes erfaßt werden.	39
<code>prj.projekte.prjname</code>	Es ist die Bezeichnung (der Name) des aktuellen Projektes anzugeben.	69
<code>prt.protokoll</code>	Tabelle zur Verwaltung der Id's der PE, SF, TP.	69
<code>prt.protokoll.prt_typ_id</code>	Identifikationsnummer der Klassifizierung des PE, SF, TP.	69
<code>prt.typ</code>	Tabelle zur Verwaltung der Typen der PE, SF, TP (Aspekte).	69
<code>prt.typ.aspekt</code>	Klassifizierung des Typs (ist er ein PE, SF, TP). Die möglichen Werte sind <code>prt.nimm_xxx()</code> zu entnehmen.	69
<code>prt.typ.spr_uebvar_id</code>	Identifikationsnummer der Übersetzungsvariante zur Darstellung des aktuellen Typs.	69

<b>Tabellenspalte / Begriff</b>	<b>Beschreibung</b>	
<code>prt.typcc_prt_typ</code>	Es darf ausschließlich NULL, PE, SF, TP angegeben werden.	70
<code>ps.abk</code>	In dieser Tabelle werden alle im Schema <code>ps</code> verwendeten Abkürzungen gehalten. Sie werden anhand ihres Typs ( <code>ps.abk.typ</code> ) klassifiziert.	74
<code>ps.abk.typ</code>	Es ist der Typ der aktuellen Abkürzung anzugeben. Die gültigen Typen können <code>ps.nimm_abk()</code> entnommen werden.	74
<code>ps.anmerkung</code>	In dieser Tabelle werden alle im Schema <code>ps</code> verwendeten Anmerkungen gehalten.	74
<code>ps.gefszenario</code>	In dieser Tabelle wird die Identifikationsnummer des Gefahrenszenarios abgelegt. In der Spalte <code>ps.gefszenario.notizen</code> wird der Sinn des Szenarios notiert. Diese Tabelle stellt die zentrale Verbindung aller Sicherheitsaspekte dar.	74
<code>spr.pos</code>	Tabelle, die die Position des Wortes in der Wortgruppe aufnimmt (die Wortgruppe bildet). Die Reihenfolge der Worte wird mittels <code>spr.pos.id</code> definiert.	78
<code>spr.pos.intern_abczk_id</code>	Es ist die Identifikationsnummer des aktuell verwendeten Wortes anzugeben.	78
<code>spr.pos.intern_restzk_id</code>	Es ist die Identifikationsnummer der Zeichenkette anzugeben, welche keine Buchstaben enthält.	78
<code>spr.pos.spr_wgruppe_id</code>	Es ist die Identifikationsnummer der Wortgruppe anzugeben, in der das aktuelle Wort enthalten ist.	78
<code>spr.poscc_spr_pos</code>	Wenn <code>intern_abczk_id</code> größer Null ist, muß <code>intern_restzk_id</code> Null sein (und umgekehrt). Sind <code>intern_abczk_id</code> oder <code>intern_restzk_id</code> größer Null, muß <code>spr_wgruppe_id</code> immer größer Null sein.	79
<code>spr.sprachen</code>	Tabelle zur Verwaltung der unterschiedlichen Sprachen.	78
<code>spr.sprachen.sprache</code>	Bezeichnung der Sprache, welche für Übersetzungen verwendet wird.	78
<code>spr.sprachen.sprache_k</code>	Kurzbezeichnung der Sprache.	78

<b>Tabellenspalte / Begriff</b>	<b>Beschreibung</b>	
<code>spr.uebersetzung</code>	Tabelle, die die Verbindung zwischen Sprache, Übersetzungsvariante und Wortgruppe herstellt. Innerhalb einer Übersetzungsvariante darf eine Sprache maximal einmal zugeordnet sein.	78
<code>spr.uebersetzung.spr_sprachen_id</code>	Es ist die Identifikationsnummer der aktuell verwendeten Sprache anzugeben.	78
<code>spr.uebersetzung.spr_uebvar_id</code>	Es ist die Identifikationsnummer der aktuell verwendeten Übersetzungsvariante anzugeben.	78
<code>spr.uebersetzung.spr_wgruppe_id</code>	Es ist die Identifikationsnummer der aktuell verwendeten Wortgruppe anzugeben.	78
<code>spr.uebvar</code>	Mittels dieser Tabelle werden die möglichen Übersetzungsvarianten bereitgestellt. Sie können von weiteren Tabellen verwendet werden.	78
<code>spr.wgruppe</code>	Tabelle zur Verwaltung der unterschiedlichen Wortgruppen.	78
<code>str.name</code>	Tabelle zur Verwaltung der Bezeichnung der unterschiedlichen verwendeten Strukturen.	84
<code>str.name.name</code>	Strukturbezeichnung (Produktgruppenstruktur, Funktionsgruppenstruktur, ...).	84
<code>str.name.name_k</code>	Strukturbezeichnung (Kurzbezeichnung).	84
<code>str.strkt</code>	Tabelle zur Verwaltung der Strukturpunkte einschließlich Klassifizierung im Sinne eines Fehlerbaumereignisses (Basis, Haus, ...).	84
<code>str.strkt.nr</code>	Nummer (Position) des aktuellen Strukturpunktes in einer Struktur. Die Nummer kann aus Zeichen, Ziffern oder einer Zeichen-/Ziffernkombination bestehen.	84
<code>str.strkt.str_name_id</code>	Identifikationsnummer der Strukturbezeichnung, welcher der aktuelle Strukturpunkt zugeordnet ist.	84
<code>tl.befehl</code>	Tabelle zum Verwalten der Bezeichnungen der verwendeten $\text{\TeX/L\TeX}$ - Befehle.	88

## Tabellenspalte / Begriff

<code>tl.befehl.abczk</code>	<b>Beschreibung</b> Es ist die Zeichenkette anzugeben, die beim generieren eines ASCII-Textes als Ersatz des TL-Befehlsnamens (siehe <code>tl.befname</code> ) verwendet werden soll. Diese Spalte ist insbesondere für das Ersetzen von Zweizeichenbefehlen zu verwenden (z.B. <code>tl.befname = A — tl.abczk = Ä</code> ).	88
<code>tl.befehl.befname</code>	Name bzw. Bezeichnung des $\TeX/\LaTeX$ Befehls.	88
<code>tl.datei</code>	Tabelle zum Verwalten der unterschiedlichen $\TeX/\LaTeX$ - Dateien.	88
<code>tl.datei.dname</code>	Dateiname der TL-Datei.	88
<code>tl.pos</code>	Tabelle zum Verwalten der Position der unterschiedlichen Zeichenketten, welche in einer TL-Datei auftreten.	88
<code>tl.pos.intern.abczk_id</code>	Es ist die Identifikationsnummer des Wortes (besteht ausschließlich aus Buchstaben) anzugeben, welches in der aktuellen TL-Datei verwendet wird.	88
<code>tl.pos.intern.restzk_id</code>	Es ist die Identifikationsnummer der Zeichenkette (enthält keine Buchstaben) anzugeben, die in der aktuellen TL-Datei verwendet wird.	88
<code>tl.pos.pos</code>	Es ist die Position des TL-Objektes in der TL-Datei anzugeben.	88
<code>tl.pos.tl.befehl_id</code>	Es ist die Identifikationsnummer des TL-Befehls anzugeben, welcher in der aktuellen TL-Datei verwendet wird.	88
<code>tl.pos.tl.datei_id</code>	Es ist die Identifikationsnummer der TL-Datei anzugeben, welche die aktuellen Zeichenketten aufnimmt.	88
<code>tl.pos.tl.wort_id</code>	Es ist die Identifikationsnummer des TL-Wortes anzugeben, welches in der aktuellen TL-Datei verwendet wird.	88
<code>tl.poscc.tl_pos</code>	Alle Fremdschlüssel dürfen Null sein. Wenn ein jedoch Fremdschlüssel größer Null ist, müssen die anderen Null sein.	89
<code>tl.wort</code>	Tabelle, die sämtliche verwendeten Wörter aller natürlichen Sprachen einschließlich $\TeX/\LaTeX$ - spezifischer Befehle aufnimmt. Als Beispiel können Trennhilfen (Rum-mel), spezielle Buchstaben einer Sprache ( $A = \text{Ä}$ ) oder sonstige in ein Wort eingebettete Befehle, welche zur korrekten Darstellung des Wortes in $\TeX/\LaTeX$ erforderlich sind, aufgenommen werden.	88

**Tabellenspalte / Begriff**  
t1.wort.intern\_abczk\_id

t1.wort.worte

### **Beschreibung**

Es ist die Identifikationsnummer zur Spezifikation des Wortes (ohne  $\text{\TeX}$ / $\text{\LaTeX}$  - spezifische Befehle) der aktuellen Sprache anzugeben.

Es sind die verwendeten Wörter aller natürlicher Sprachen einschließlich  $\text{\TeX}$ / $\text{\LaTeX}$  - spezifischer Befehle anzugeben. Sie dürfen nicht mehrfach auftreten.

88

88