

Database functies

Applicatiebouw

Op veel plekken in OpenWave kan de applicatiebeheerder SQL-statements definiëren. Bijvoorbeeld in [rapportages](#), [query's](#) op tegels en [documentsjablonen](#). Om ingewikkelde statements te vermijden heeft OpenWave zelf een aantal functies op de database gedefinieerd om veel voorkomende problemen op te lossen. Deze functies kunnen opgenomen worden in de query's op dezelfde manier als de inheemse Postgres functies.

- **fn_bedrag(p_bedrag)** retourneert een string waarbij p_bedrag (een float) is omgezet in een string met een komma voor de decimale punt en een punt voor de duizendtallen. Voorbeeld: fn_bedrag(1234.56) retourneert '1.234,56' (indien fn_bedrag(0) dan retourneert deze functie 'null' of te wel een lege waarde)
- **fn_bedragn(p_bedrag)** retourneert een string waarbij p_bedrag (een float) is omgezet in een string met een komma voor de decimale punt en een punt voor de duizendtallen. Lijkt op fn_bedrag maar zal indien de waarde van p_bedrag 0 OF null is, als resultaat **0,00** retourneren. Voorbeeld: fn_bedrag(1234.56) retourneert '1.234,56' en fn_bedrag(0) retourneert dus '0,00'
- **fn_datumplus(p_datum,p_plusmindagen)** retourneert een datum waarbij de invoerdatum (p_datum) is verhoogd/verlaagd met p_plusmindagen (een integer)
- **fn_ddmaandjjjj(p_datum)** retourneert een string waarbij de invoerdatum (p_datum) is omgezet naar een Nederlandse tekst bijvoorbeeld: '03 oktober 2017'
- **fn_isposinteger (p_string)** retourneert een integer met waarde 1 indien p_string (een gevulde string van max 20 tekens) omgezet kan worden naar een positief geheel getal. Anders is de resultwaarde 0. Indien p_string een nullwaarde heeft dan is ook het resultaat 0
- **fn_unaccent(p_string)** retourneert een string waarbij diakritische tekens van p_string (een string) zijn omgezet: ö wordt o et cetera
- **fn_vandaag(p_plusmindagen)** retourneert een datum waarbij de systeemdatum is verhoogd/verlaagd met p_plusmindagen (een integer)
- **fn_datediff(p_type,p_datumvanaf,p_datumtot)** retourneert een integer die afhankelijk van p_type het verschil in dagen, maanden of jaren weergeeft tussen de twee datums. P_type kan de waarde 'day' of 'month' of 'year' hebben
- **instr(p_domeinstring,p_zoekstring,[p_begin],[p_occurence])** retourneert een integer die de positie van p_zoekstring binnen p_domeinstring weergeeft vanaf p_begin (integer). P_begin en p_occurence zijn optioneel. Beide hebben 1 als defaultwaarde. P_occurence staat voor de nde keer dat p_zoekstring voorkomt vanaf p_begin in p_domeinstring. Voorbeelden:
 - instr('Altijd is kortjakje ziek', 's') resultaat: 9 (De positie van de eerste 's' geteld vanaf 1)
 - instr('Varen, varen over de baren','ren',4,2) resultaat: 24 (De positie van de tweede 'ren' geteld vanaf 4)
 - instr('Varen, varen over de baren','ren',-1) resultaat: 24 (De positie van de eerste 'ren' geteld vanaf -1 : van rechts naar links)
- **fn_isposintpolygoon (p_string)** retourneert een integer met waarde 1 indien p_string bestaat uit minimaal 3 paren coördinaten gescheiden door een spatie. Een coördinaatpaar bestaat uit twee positieve getallen (floats) gescheiden door een komma. Een correcte polygoonstring is bijvoorbeeld '1234,2345 3456.99,45.99 5678987,4321 4,4 1234,2345'. Is p_string niet correct dan wordt een 0 geretourneerd
- **fn_isposintlinoofpolygoon (p_string)** retourneert een integer met waarde 1 indien p_string bestaat uit minimaal 2 paren coördinaten gescheiden door een spatie. Een coördinaatpaar bestaat uit twee positieve getallen (floats) gescheiden door een komma. Een correcte polygoonstring is bijvoorbeeld '1234,2345 3456.98,45.55 5678987,4321 4,4'. Is p_string niet

correct dan wordt een 0 geretourneerd

- **fn_dec2ana(p_tijd)** retourneert een tijd in het formaat HH:MM. P_tijd is een float. Bijvoorbeeld 26.4 wordt 26:24. 1.5 wordt 1:30. 12.12345 wordt 12:07. P_tijd moet kleiner zijn dan 100
- **fn_tijdstip(p_plusmin integer, p_interval char(1))** retourneert een tijdstip op basis van het moment dat de functie wordt aangeroepen + of - een aantal uur of dagen. Het resultaat is een timestamp. p_interval kan de waarde 'D' (dagen) of 'H' (uren) hebben. Voorbeelden:
 - Stel fn_tijdstip(0,'H') = 2021-11-30 09:10:52.307227
 - dan is fn_tijdstip(-24,'H')= 2021-11-29 09:10:52.307227
 - dan is fn_tijdstip(5,'H')= 2021-11-30 14:10:52.307227
 - dan is fn_tijdstip(-1,'D')= 2021-11-29 14:10:52.307227LET OP: fn_tijdstip(-24,'H') kan een ander resultaat geven als fn_tijdstip(-1,'D'). Bij uren wordt rekening gehouden met overgang winter/zomertijd. Bij dagen niet
- **fn_rechtenkolom(p_column text,p_keyrechten integer)** retourneert 'T' of 'F' als char(1). Het resultaat is de achterliggende waarde van de parameter p_column die doorgegeven moet worden als rechtentabelnaam + '.' + rechtenkolomnaam bijvoorbeeld *tbomgrechten.dlcadvins*. De parameter p_keyrechten moet de waarde hebben van dnkeyrechten uit de medewerkerstabel bij de kaart van de betreffende inlogger. Er is een systeemquery (dvcode = *geefWaardeRechtenkolom*) die deze aanroep gebruikt; deze query geeft true of false terug en kan gebruikt worden in bijv. de schermkolomdefinitie bij tags visible of edit. Voorbeeld van hoe deze aanroep er uit zou moeten zien: *fn_rechtenkolom('tbomgrechten.dlcomgadvsb', 1)*
- **fn_rechtenkolom(p_column text,p_mwcode char(5))** retourneert 'T' of 'F' als char(1). Het resultaat is de achterliggende waarde van de parameter p_column die doorgegeven moet worden als rechtentabelnaam + '.' + rechtenkolomnaam bijvoorbeeld *tbomgrechten.dlcadvins*. De parameter p_mwcode moet de waarde hebben van een dvcode uit de medewerkerstabel bij de kaart van de betreffende inlogger. Wanneer deze functie aangeroepen wordt vanuit een opgeslagen query in tbqueries, dan kan daartoe de systeemvariable :keyaccount voor worden gebruikt). Voorbeeld van hoe deze aanroep er uit zou moeten zien:
fn_rechtenkolom('tbomgrechten.dlcomgadvsb','SYS')
- **fn_iscompartimentok(p_mwcode char(5),p_module char(1), p_dnkey integer)** retourneert 0 of 1 als integer. 0 indien de compartimentcheck niet klopt en 1 indien de compartimentscheck wel klopt.
In p_mwcode moet de medewerkerscode van de inlogger doorgegeven worden (wanneer deze functie aangeroepen wordt vanuit een opgeslagen query in tbqueries, dan kan daartoe de systeemvariabele :keyaccount voor worden gebruikt).
De parameter p_module één letter namelijk hore(C)a, mili(E)/gebruik, (H)andhaving, (I)(infoaanvragen, apv/(O)verig, (V)inrichtingen of (W) omgevingzaken.
In de parameter p_dnkey dient de primary key doorgegeven te worden van de hoofdtabel die hoort bij de module en die verwijst naar een rij waarvoor de compartimentcheck moet worden gedaan. OpenWave zoekt de dnkeywaarde op in vwfrmomgvergunningen c.q. vwfrmhandhavingen et cetera en evalueert daar of de kolom dnkeycompartiment overeenkomt met die van de medewerker.
- **fn_isLineString(p_string)** Retourneert 0 of 1 als integer. 0 indien de p_string geen LineString is vanwege onvoldoende lengte of rare coördinaten. Een coördinaatpaar bestaat uit twee positieve getallen (floats) gescheiden door een komma. Retourneert 1 indien p_string bestaat uit paren coördinaten gescheiden door een spatie, waarbij de x en y coördinaten binnen een paar gescheiden zijn door een komma EN waarbij het eerste paar ongelijk is aan het laatste paar. Bijvoorbeeld 167321,454597 167344.99,454619.87 167346.987,454613.123 167359,454617 167380,454606 167373,454592 167354,454576
- **fn_iskolomleeg(p_column text,p_dnkeykey integer)** retourneert 1 of 0 als integer. Het resultaat is 1 indien de achterliggende waarde van de parameter p_column die doorgegeven moet worden als tabelnaam + '.' + kolomnaam bijvoorbeeld *tbsoortomgverg.dvzgwuuid* leeg is.

De parameter `p_dnkey` moet de waarde hebben van een `dnkey` uit de betreffende tabel. Er zijn een systeemqueries die deze aanroep gebruiken waarbij de tweede parameter van die queryaanroep is samengesteld uit een pointer (`{id}` of `%keypointer%`) gevolgd door een `+` teken gevolgd door tabelnaam + `'.'` + kolomnaam):

- `tbquery.dvcode = beheer_setuuid` retourneert, indien de kolom leeg is, een aanroep naar de interne OpenWave-api `setcolumnvalue()` om die kolom te vullen met een gegenereerde Uuid. Aanroepvoorbeeld bij de action op een knop op een detailscherm:

```
// <action>getFlexAction(beheer_setuuid, %keypointer%+tbzgwrolypes.dvzgwuuid)</action> // (detailscherm MDDC_getSoortomgTbZGWRoltypesDetail.xml)
```
- `tbquery.dvcode = beheer_iskolomgevuld` retourneert `true` indien een bepaalde kolom gevuld is en `false` indien `null`. Aanroepvoorbeeld bij `celvisible` in scherkkolomdefinitie van een lijst:

```
//<celvisible>%query(beheer_iskolomgevuld,{id}+tbtoestemmingen.dnkeyomgvergunning)%</celvisible> // (lijstschermdl MDLC_getVwfrmlnrichtingMbaList.xml)
```

Zie voor voorbeeld met betrekking tot gebruik van de functies `fn_rechtenkolom` en `fn_iscompartimentok`: onder kopje *De kolommen van de tabel `tbsysstandardtable`* bij [Standaard Lijsten en Detailschermen](#).

Zie voor voorbeeld met betrekking tot gebruik van de functie `fn_iskolomleeg`: in het detailscherm van `tbsoortomgverg` (zaakbeheerportaal). Achter de kolom *UUID Open Zaak API* is een schermknop waarmee deze kolom gevuld kan worden met een uuid. De action bij de schermknop is: `getFlexAction(beheer_setuuid,%keypointer%+tbsoortomgverg.dvzgwuuid)`. De query `beheer_setuuid` construeert òf de aanroep naar een `showaction` met een bepaalde tekst òf de aanroep naar `setcolumnvalue`, die de kolom vult met een uuid. De query beslist dat op basis van de aanroep naar `fn_iskolomleeg()`. Indien namelijk niet leeg, dan geeft OpenWave die showtekstmelding

From:
<https://acc-doc.open-wave.nl/> - Documentatie

Permanent link:
https://acc-doc.open-wave.nl/stable/applicatiebeheer/instellen_inrichten/openwave_database-functies

Last update: 2026/07/20 10:38

