

# 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\_isposintlijnofpolygoon (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 14:10:52.307227`LET 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%+tbzgwrolytypes.dvzgwuuid)</action>` (detailscherm *MDDC\_getSoortomgTbZGWRoTypesDetail.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  
en)%</celvisible>` (lijstschermbild *MDLC\_getVwfrmInrichtingMbaList.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 Lijst- 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://doc.open-wave.nl/> - Documentatie

Permanent link:  
[https://doc.open-wave.nl/doku.php/openwave/1.34/applicatiebeheer/instellen\\_inrichten/openwave\\_database-functies](https://doc.open-wave.nl/doku.php/openwave/1.34/applicatiebeheer/instellen_inrichten/openwave_database-functies)

Last update: 2026/04/13 11:39

