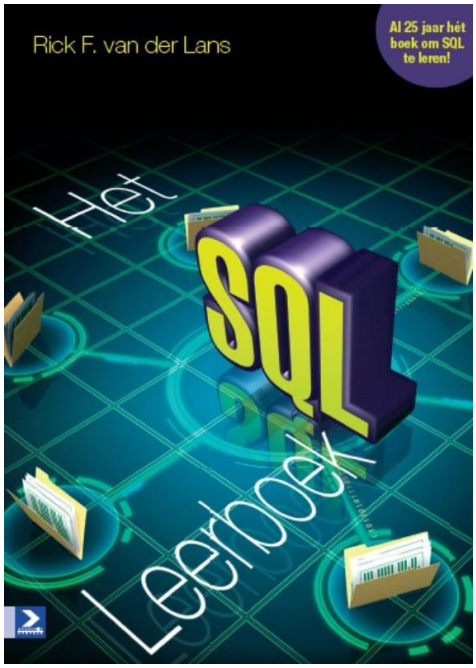
Het SQL Leerboek – zevende editie
Scalaire functies



Auteur: Rick F. van der Lans
Versie: 1.0
Datum: Februari 2012

Hoofdstuk 1

Scalaire functies

1.1 Inleiding

Beschrijving: SQL kent een groot aantal scalaire functies. In dit document geven we van de functies die door veel SQL-producten ondersteund worden de naam, een beschrijving, het datatype van het resultaat van de functie en enkele voorbeelden. De functies staan op naam gesorteerd.

Sommige functies hebben meer dan één naam. Om het zoeken te vergemakkelijken zijn ze allemaal opgenomen. Er wordt dan wel verwezen naar de functies die bij elkaar horen.

1.2 Lijst met functies

ABS(*par1*)

Beschrijving: Deze functie geeft de absolute waarde van een numerieke-expressie.
Datatype: numeriek

```
ABS(-25)    ⇨ 25
ABS(-25.89) ⇨ 25.89
```

ACOS(*par1*)

Beschrijving: Deze functie geeft in radialen de arccosinus van een hoek. De waarde van de parameter moet groter dan of gelijk aan -1 en kleiner dan of gelijk aan 1 zijn.
Datatype: numeriek

```
ACOS(0)         ⇨ 1.5707963267949
ACOS(-1) - PI() ⇨ 0
ACOS(1)         ⇨ 0
ACOS(2)         ⇨ NULL
```

ADDDATE(*par1, par2*)

Beschrijving: Deze functie voegt een interval (de tweede parameter) toe aan een datum- of timestamp-expressie (de eerste parameter). Indien de tweede parameter geen interval is maar een numeriek getal, gaat SQL ervan uit dat deze waarde een aantal dagen voorstelt.
Datatype: datum of timestamp

```
ADDDATE('2004-01-01', INTERVAL 5 MONTH) ⇨ '2004-06-01'
ADDDATE(TIMESTAMP('2004-01-01'), INTERVAL 5 MONTH)
   ⇨ '2004-06-01 00:00:00'
ADDDATE('2004-01-01 12:00:00', INTERVAL 5 DAY)
   ⇨ '2004-01-06 12:00:00'
ADDDATE('2004-01-01', 5) ⇨ '2004-01-06'
```

ADDTIME(*par1, par2*)

Beschrijving: Deze functie telt twee tijdexpressies bij elkaar op. Het resultaat is een interval bestaande uit een aantal uren, minuten en seconden. Het aantal uren kan dus groter dan 24 zijn.
Datatype: tijd

```
ADDTIME('12:59:00', '0:59:00')     ⇨ '13:58:00'
ADDTIME('12:00:00', '0:00:00.001') ⇨ '12:00:00.001000'
ADDTIME('100:00:00', '900:00:00')  ⇨ '1000:00:00'
```

ASCII(*par1*)

Beschrijving: Deze functie geeft de karaktersetpositie van het eerste teken van een alfanumerieke expressie.
Datatype: numeriek

```
ASCII('Database') ⇨ 68
ASCII('database') ⇨ 100
ASCII('')         ⇨ 0
ASCII(NULL)       ⇨ NULL
```

ASIN(*par1*)

Beschrijving: Deze functie geeft in radialen de arcsinus van een hoek. De waarde van de parameter moet groter dan of gelijk aan -1 en kleiner dan of gelijk aan 1 zijn, anders is het resultaat gelijk aan de null-waarde.
Datatype: numeriek

```
ASIN(1)    ⇨ 1.5707963267949
ASIN(0)    ⇨ 0
ASIN(NULL) ⇨ NULL
```

ATAN(*par1*)

Beschrijving: Deze functie geeft in radialen de arctangens van een hoek.
Datatype: numeriek

```
ATAN(0)   ⇨ 0
ATAN(100) ⇨ 1.56079666010823
ATAN(1)   ⇨ 0.78539816339745
```

ATAN2(*par1, par2*)

Beschrijving: Deze functie geeft in radialen de arctangens van x en y coördinaten. Dit zijn respectievelijk de eerste en de tweede parameter.
Datatype: numeriek

```
ATAN2(30,30) ⇨ 0.78539816339745
ATAN2(-1,-1) ⇨ -2.3561944901923
```

ATANH(*par1*)

Beschrijving: Deze functie geeft de hyperbolische arctangens van de parameter die in radialen gespecificeerd moet zijn.
Datatype: numeriek

```
ATANH(0.4) ⇨ 0.255412811882995
```

BIN(*par1*)

Beschrijving: Deze functie transformeert de numerieke waarde van de parameter in een binaire waarde. Deze binaire waarde bestaande uit eenen en nullen heeft het alfanumerieke datatype.
Datatype: alfanumeriek

```
BIN(7)       ⇨ '111'
BIN(1000000) ⇨ '11110100001001000000'
```

BIT_COUNT(*par1*)

Beschrijving: Deze functie geeft het aantal bits weer dat nodig is voor het weergeven van de waarde van de parameter. Er wordt gewerkt met 64-bits integers.
Datatype: numeriek

```
BIT_COUNT(3)  ⇨ 2
BIT_COUNT(-1) ⇨ 64
```

BIT_LENGTH(*par1*)

Beschrijving: Deze functie geeft de lengte in bits van een alfanumerieke waarde.
Datatype: numeriek

```
BIT_LENGTH('database')   ⇨ 64
BIT_LENGTH('100')        ⇨ 24
BIT_LENGTH(BIN(2))       ⇨ 16
```

CEILING(*par1*)

Beschrijving: Deze functie geeft het grootste gehele getal groter dan of gelijk aan de waarde van de parameter. Zie tevens de CEIL-functie.
Datatype: numeriek

```
CEILING(13.43)  ⇨ 14
CEILING(-13.43) ⇨ -13
CEILING(13)     ⇨ 13
```

CHAR(*par1*)

Beschrijving: Deze functie geeft het alfanumerieke teken van de numerieke parameter. Zie ook de CHR-functie.
Datatype: alfanumeriek

```
CHAR(80)                                  ⇨ 'P'
CHAR(82) + CHAR(105) + CHAR(99) + CHAR(107) ⇨ 'Rick'
```

CHARACTER_LENGTH(*par1*)

Beschrijving: Deze functie geeft de lengte van een alfanumerieke expressie.
Datatype: numeriek

```
CHARACTER_LENGTH('database')                    ⇨ 8
CHARACTER_LENGTH('')                            ⇨ 0
CHARACTER_LENGTH(NULL)                          ⇨ NULL
CHARACTER_LENGTH((SELECT MAX(NAAM) FROM SPELERS)) ⇨ 6
CHARACTER_LENGTH(BIN(8))                        ⇨ 4
```

CHARSET(*par1*)

Beschrijving: Deze functie geeft de naam van de karakterset van de alfanumerieke parameter.
Datatype: alfanumeriek

```
CHARSET('database')                       ⇨ 'latin1'
CHARSET((SELECT MAX(NAAM) FROM SPELERS))  ⇨ 'latin1'
CHARSET((SELECT MAX(TABLE_NAME)
         FROM   INFORMATION_SCHEMA.TABLES)) ⇨ 'utf8'
```

CHR(*par1*)

Beschrijving: Deze functie geeft het alfanumerieke teken dat hoort bij de ASCII-code gespecificeerd met de numerieke parameter.
Datatype: alfanumeriek

```
CHR(80)                                   ⇨ 'P'
CHR(82) || CHR(105) || CHR(99) || CHR(107) ⇨ 'Rick'
CHR(ASCII('D'))                           ⇨ 'D'
```

COALESCE(*par1, par2, par3, ...*)

Beschrijving: Deze functie kan een variabel aantal parameters hebben. De waarde van de functie is gelijk aan de waarde van de eerste parameter die niet gelijk is aan null.

Als E_1, E_2 en E_3 drie expressies zijn, dan is de specificatie:

```
COALESCE(E1, E2, E3)
```

gelijkwaardig aan de volgende case-expressie:

```
CASE
   WHEN E1 IS NOT NULL THEN E1
   WHEN E2 IS NOT NULL THEN E2
   WHEN E3 IS NOT NULL THEN E3
   ELSE NULL
END
```

Datatype: afhankelijk van parameters

```
COALESCE('John', 'Jim', NULL)             ⇨ 'John'
COALESCE(NULL, NULL, NULL, 'John', 'Jim') ⇨ 'John'
```

COERCIBILITY(*par1*)

Beschrijving: Deze functie bepaalt de coercibility-waarde van een expressie.
Datatype: numeriek

```
COERCIBILITY(NULL)       ⇨ 5
COERCIBILITY('Database') ⇨ 4
```

COLLATION (*par1*)

Beschrijving: Deze functie geeft de naam van de collation van de alfanumerieke parameter.
Datatype: alfanumeriek

```
COLLATION('database')
   ⇨ 'latin1_swedish_ci'
COLLATION((SELECT MAX(NAAM) FROM SPELERS))
   ⇨ 'latin1_swedish_ci'
COLLATION((SELECT MAX(TABLE_NAME)
         FROM   INFORMATION_SCHEMA.TABLES))
   ⇨ 'utf8_general_ci'
```

CONCAT(*par1, par2*)

Beschrijving: Deze functie plakt twee alfanumerieke waarden aan elkaar. Hetzelfde effect kan met de ||-operator verkregen worden.
Datatype: alfanumeriek

```
CONCAT('Data','base') ⇨ 'Database'
```

CONNECTION_ID()

Beschrijving: Deze functie geeft de numerieke identifier van de connectie.
Datatype: numeriek

```
CONNECTION_ID() ⇨ 4
```

CONV(*par1, par2, par3*)

Beschrijving: Met deze functie zetten we de waarde (eerste parameter) van het ene stelsel (tweede parameter) om naar het andere (derde parameter). De waarde van de laatste twee parameters moet liggen tussen 2 en 36, anders is het resultaat gelijk aan null. Tevens moet de waarde van de eerste parameter passen bij het stelsel van de eerste parameter, anders is het resultaat 0.
Datatype: alfanumeriek

```
CONV(1110, 2, 10)  ⇨ '14'
CONV(1110, 10, 2)  ⇨ '10001010110'
CONV(1110, 10, 8)  ⇨ '2126'
CONV(1110, 10, 16) ⇨ '456'
CONV(35, 10, 36)   ⇨ 'Z'
CONV(35, 10, 37)   ⇨ NULL
CONV(8, 2, 10)     ⇨ '0'
```

CONVERT(*par1, par2*)

Beschrijving: Deze functie converteert het datatype van de eerste parameter. De tweede parameter moet gelijk zijn aan een van de bekende datatypes, waaronder BINARY, CHAR, DATE, DATETIME, TIME, SIGNED, SIGNED INTEGER, UNSIGNED, UNSIGNED INTEGER of VARCHAR. Deze specificatie:

```
CONVERT(par1, type1)
```

is gelijkwaardig aan:

```
CAST(par1 AS type1)
```

Tevens mag de volgende formulering gebruikt worden:

```
CONVERT(par1 USING type1)
```

Datatype: afhankelijk van de tweede parameter

```
CONVERT(45, CHAR(2))              ⇨ '45'
CONVERT('2000-01-42', DATE)       ⇨ '2000-01-01'
CONVERT(12.56, UNSIGNED INTEGER)  ⇨ 13
CONVERT(-12.56, UNSIGNED INTEGER) ⇨ 18446744073709551603
```

CONVERT_TZ(*par1, par2, par3*)

Beschrijving: Deze functie berekent de timestamp-waarde van een timestamp-expressie (eerste parameter) met een bepaalde tijdzone (tweede parameter) bij een gegeven andere tijdzone (derde parameter).
Datatype: timestamp

```
CONVERT_TZ('2005-05-20 09:30:40', '+00:00', '+9:00')
   ⇨ 2005-05-20 18:30:40
```

COS(*par1*)

Beschrijving: Deze functie geeft de cosinus van een hoek in radialen. Het resultaat is een waarde die ligt tussen -1 en 1.
Datatype: numeriek

```
COS(0)      ⇨ 1
COS(PI())   ⇨ –1
COS(PI()/2) ⇨ 0
```

COT(*par1*)

Beschrijving: Deze functie geeft de cotangens van een hoek in radialen.
Datatype: numeriek

```
COT(10)     ⇨ 1.54235
COT(PI()/2) ⇨ 0
COT(NULL)   ⇨ NULL
```

CURDATE()

Beschrijving: Deze functie geeft de systeemdatum. In sommige SQL-producten wordt de functie geef-systeemdatum vervuld door de systeemvariabele SYSDATE.
Datatype: datum

```
CURDATE() ⇨ '2009-02-20'
```

CURRENT_DATE()

Beschrijving: Deze functie geeft de systeemdatum in het volgende formaat JJJJ-MM-DD. Indien de functie als een numerieke expressie wordt gezien, wordt de systeemdatum als numerieke waarde met het formaat JJJJMMDD gepresenteerd. Indien de haakjes weggelaten worden, verandert de functie in de systeemvariabele CURRENT_DATE. Zie ook de CURDATE-functie.
Datatype: datum of double

```
CURRENT_DATE()     ⇨ '2005-02-20'
CURRENT_DATE() + 0 ⇨ 20050220
CURRENT_DATE       ⇨ '2005-02-20'
```

CURRENT_TIME()

Beschrijving: Deze functie geeft de systeemtijd in het volgende formaat UU:MM:SS. De afkorting UU staat voor de uren, MM voor minuten en SS voor seconden. Indien de functie als een numerieke expressie wordt gezien, wordt de systeemtijd als numerieke waarde met het formaat UUMMSS gepresenteerd. Indien de haakjes weggelaten worden, verandert de functie in de systeemvariabele CURRENT_TIME. Zie ook de CURTIME-functie.
Datatype: tijd of double

```
CURRENT_TIME()     ⇨ '16:42:24'
CURRENT_TIME() + 0 ⇨ 164224
CURRENT_TIME       ⇨ '16:42:24'
```

CURRENT_TIMESTAMP()

Beschrijving: Deze functie geeft de systeemdatum en systeemtijd in het volgende formaat JJJJ-MM-DD UU:MM:SS. De afkorting JJJJ staat voor de jaren, de eerste MM voor de maanden, DD voor de dagen, HH voor de uren, de tweede MM voor de minuten en SS voor de seconden. Indien de functie als een numerieke expressie wordt gezien, wordt de systeemdatum en tijd als numerieke waarde met het formaat JJJJMMDDUUMMSS gepresenteerd. Indien de haakjes weggelaten worden, verandert de functie in de systeemvariabele CURRENT_TIMESTAMP.
Datatype: timestamp of double

```
CURRENT_TIMESTAMP()     ⇨ '2005-10-16 20:53:45'
CURRENT_TIMESTAMP() + 0 ⇨ 20051016205345
CURRENT_TIMESTAMP       ⇨ '2005-10-16 20:53:45'
```

CURRENT_USER()

Beschrijving: Deze functie geeft de naam van de SQL-gebruiker.
Datatype: alfanumeriek

```
CURRENT_USER() ⇨ 'root@localhost'
```

CURTIME()

Beschrijving: Deze functie geeft de systeemtijd in het volgende formaat UU:MM:SS. De afkorting UU staat voor de uren, MM voor minuten en SS voor seconden. In enkele producten heet deze functie kortweg TIME.
Datatype: tijd

```
CURTIME() ⇨ '16:42:24'
```

DATABASE()

Beschrijving: Deze functie geeft de naam van de courante database.
Datatype: alfanumeriek

```
DATABASE() ⇨ 'TENNIS'
```

DATE(*part*)

Beschrijving: Deze functie transformeert de parameter in een datumwaarde. De parameter moet wel het formaat van een correcte datum of timestamp hebben.
Datatype: datum

```
DATE('2005-12-01')          ⇨ '2005-12-01'
DATE('2005-12-01 12:13:14') ⇨ '2005-12-01'
```

DATE_ADD(*par1, par2*)

Beschrijving: Deze functie voegt een interval (de tweede parameter) toe aan een datum- of timestamp-expressie (de eerste parameter). Zie ook de ADDDATE-functie.
Datatype: datum of timestamp

```
DATE_ADD('2004-01-01', INTERVAL 5 MONTH) ⇨ '2004-06-01'
DATE_ADD('2004-01-01 12:00:00', INTERVAL 5 DAY)
   ⇨ '2004-01-06 12:00:00'
```

DATEDIFF(*par1, par2*)

Beschrijving: Deze functie berekent het aantal dagen dat ligt tussen twee datum- of timestamp-expressies.
Datatype: numeriek

```
DATEDIFF('2004-01-12', '2004-01-01')                  ⇨ 11
DATEDIFF('2004-01-01', '2004-01-12')                  ⇨ -11
DATEDIFF('2004-01-12 19:00:00', '2004-01-01') )       ⇨ 11
DATEDIFF('2004-01-12 19:00:00', '2004-01-01 01:00:00') ⇨ 11
DATEDIFF('2004-01-12', CURDATE())                     ⇨ -643
```

DATE_FORMAT(*par1, par2*)

Beschrijving: Deze functie transformeert een datum- of timestamp-expressie (de eerste parameter) naar een alfanumerieke waarde. De tweede parameter geeft het formaat aan van die alfanumerieke waarde en hierbij kunnen diverse speciale opmaakcodes gebruikt worden; zie de onderstaande tabel.

Opmaakcode	Toelichting
%a	Drieletterige Engelstalige afkorting van de weekdag (bijvoorbeeld Sun, Mon en Sat)
%b	Drieletterige Engelstalige afkorting van de maand (bijvoorbeeld Jan, Feb en Mar)
%c	Numerieke code voor de maand (0 tot en met 12)
%D	Dag van de maand met een Engelstalig achtervoegsel, zoals 0th, 1st en 2nd
%d	Tweecijferige numerieke code voor de dag van de maand (00 tot en met 31)
%e	Een- of tweecijferige numerieke code voor de dag van de maand (0 tot en met 31)
%f	Zescijferige numerieke code voor het aantal microseconden (000000 tot en met 999999)
%H	Tweecijferige numerieke code voor het uur (00 tot en met 23)
%h	Tweecijferige numerieke code voor het uur (01 tot en met 12)
%I	Tweecijferige numerieke code voor het uur (01 tot en met 12)
%i	Tweecijferige numerieke code voor het aantal minuten (00 tot en met 59)
%j	Driecijferige numerieke code voor de dag van het jaar (001 tot en met 366)
%k	Een- of tweecijferige numerieke code voor het uur (0 tot en met 23)
%l	Een- of tweecijferige numerieke code voor het uur (1 tot en met 12)
%M	Engelstalige aanduiding van de maand (bijvoorbeeld, January, February en December)
%m	Tweecijferige numerieke code voor de maand (00 tot en met 12)
%p	Aanduiding AM of PM
%r	Aanduiding van de tijd (in 12 uren) met het formaat `UU:MM:SS` gevolgd door AM of PM
%S	Tweecijferige numerieke code voor het aantal seconden (00 tot en met 59)
%s	Tweecijferige numerieke code voor het aantal seconden (00 tot en met 59)
%T	Aanduiding van de tijd (in 24 uren) met het formaat `UU:MM:SS` gevolgd door AM of PM

%U	Tweecijferige numerieke code voor de week in het jaar (00 tot en met 53), waarbij zondag als eerste dag van de week beschouwd wordt
%u	Tweecijferige numerieke code voor de week in het jaar (00 tot en met 53), waarbij maandag als eerste dag van de week beschouwd wordt
%V	Tweecijferige numerieke code voor de week in het jaar (01 tot en met 53), waarbij zondag als eerste dag van de week beschouwd wordt
%v	Tweecijferige numerieke code voor de week in het jaar (01 tot en met 53), waarbij maandag als eerste dag van de week beschouwd wordt
%W	Engelstalige aanduiding van de dag in de week (bijvoorbeeld, Sunday, Monday en Saturday)
%w	Eencijferige code voor de dag in de week (0 tot en met 6), waarbij zondag als eerste dag van de week beschouwd wordt
%X	Een vier-cijferige numerieke code die aangeeft het jaar waarin de week start behorende bij de gespecificeerde datum waarbij zondag de eerste dag van de week is
%x	Een viercijferige numerieke code die aangeeft het jaar waarin de week start behorende bij de gespecificeerde datum waarbij maandag de eerste dag van de week is
%Y	Viercijferige numerieke code voor het jaar
%y	Tweecijferige numerieke code voor het jaar
%%	Geeft het procentteken

Datatype: alfanumeriek

```
DATE_FORMAT('2005-10-16', '%a %c %b')             ⇨ 'Sun 10 Oct'
DATE_FORMAT('2005-10-06', '%d %e %D')             ⇨ '06 6 6th'
DATE_FORMAT('2005-01-16', '%j %M %m')             ⇨ '016 January 01'
DATE_FORMAT('2005-01-09', '%U %u %V %v')          ⇨ '02 01 02 01'
DATE_FORMAT('2005-12-31', '%U %u %V %v')          ⇨ '52 52 52 52'
DATE_FORMAT('2005-01-09', '%W %w')                ⇨ 'Sunday 0'
DATE_FORMAT('2005-01-02', '%X %x')                ⇨ '2005 2004'
DATE_FORMAT('2005-01-09', '%Y %y')                ⇨ '2005 05'
DATE_FORMAT('2005-01-01 12:13:14.012345', '%f')   ⇨ '012345'
DATE_FORMAT('2005-01-01 12:13:14', '%H %h %I %i') ⇨ '13 01 01 14'
DATE_FORMAT('2005-01-01 12:13:14', '%k %l %p')    ⇨ '12 12 PM'
DATE_FORMAT('2005-01-01 12:13:14', '%S %s %T')    ⇨ '14 12 12:13:14'
DATE_FORMAT('2005-01-09', 'Database')             ⇨ 'Database'
DATE_FORMAT('2005-01-09', 'Deze dag is het %W')   ⇨ 'Deze dag is het Sunday'
```

DATE_SUB(*par1, par2*)

Beschrijving: Deze functie trekt een interval (de tweede parameter) af van een datum- of timestamp-expressie (de eerste parameter). Zie ook de SUBDATE-functie.
Datatype: datum of timestamp

```
DATE_SUB('2004-01-01', INTERVAL 5 MONTH)        ⇨ '2003-08-01'
DATE_SUB('2004-01-01 12:00:00', INTERVAL 5 DAY) ⇨ '2003-12-27 12:00:00'
```

DAY(*par1*)

Beschrijving: Deze functie geeft uit een datum of timestamp-expressie het nummer van de dag van de maand. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 1, en kleiner dan of gelijk aan 31. Zie ook de DAYOFMONTH-functie.
Datatype: numeriek

```
DAY('2004-01-01')          ⇨ 1
DAY('2004-01-01 09:11:11') ⇨ 1
DAY(CURRENT_DATE())        ⇨ 17
DAY(CURRENT_TIMESTAMP())   ⇨ 17
```

DAYNAME(*parm*)

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie de Engelstalige naam van de dag van de week.
Datatype: alfanumeriek

```
DAYNAME('2005-01-01') ⇨ 'Saturday'
```

DAYOFMONTH(*parm*)

Beschrijving: Deze functie geeft uit een datum of timestamp-expressie het nummer van de dag van de maand. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 1 en kleiner dan of gelijk aan 31. Zie ook de DAY-functie.
Datatype: numeriek

```
DAYOFMONTH('2004-01-01')          ⇨ 1
DAYOFMONTH('2004-01-01 09:11:11') ⇨ 1
DAYOFMONTH(CURRENT_DATE())        ⇨ 17
DAYOFMONTH(CURRENT_TIMESTAMP())   ⇨ 17
```

DAYOFWEEK(*parm*)

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie het nummer van de dag in de week. Als parameter mag ook een alfanumerieke constante gespecificeerd worden mits deze het formaat heeft van een datumconstante. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 1, en kleiner dan of gelijk aan 7. Hierbij is zondag de eerste dag van de week.
Datatype: numeriek

```
DAYOFWEEK('2005-07-29')        ⇨ 6
DAYOFWEEK(CURRENT_TIMESTAMP()) ⇨ 3
```

DAYOFYEAR(*parm*)

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie het nummer van de dag in het jaar. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 1, en kleiner dan of gelijk aan 366.
Datatype: numeriek

```
DAYOFYEAR('2005-07-29')          ⇨ 210
DAYOFYEAR('2005-07-29 12:00:00') ⇨ 210
DAYOFYEAR(CURDATE())             ⇨ 291
```

DEFAULT()

Beschrijving: Deze functie geeft de defaultwaarde van een bepaalde kolom.
Datatype: afhankelijk van de kolom

```
DEFAULT(DATUM)  ⇨ '1990-01-01'
DEFAULT(BEDRAG) ⇨ 50.00
```

DEGREES(*parm*)

Beschrijving: Deze functie converteert een aantal graden naar een waarde in radialen.
Datatype: numeriek

```
DEGREES(1.570796) ⇨ 90
DEGREES(PI())     ⇨ 180
```

EXP(*par1*)

Beschrijving: Geeft het resultaat van het getal *e* tot de macht *x*, waarbij *x* de waarde is van de parameter en *e* het grondtal van de natuurlijke logaritmen.
Datatype: numeriek

```
EXP(1) ⇨ 2.718281828459
EXP(2) ⇨ 7.3890560989307
```

FLOOR(*par1*)

Beschrijving: Deze functie geeft het kleinste gehele getal kleiner dan of gelijk aan de waarde van de parameter.
Datatype: numeriek

```
FLOOR(13.9)   ⇨ 13
FLOOR(-130.9) ⇨ -131
```

FORMAT(*par1, par2*)

Beschrijving: Deze functie formatteert een numerieke waarde naar een patroon nn,nnn,nnn.nnn. De tweede parameter geeft het aantal decimalen achter de komma weer en moet groter dan of gelijk zijn aan nul.
Datatype: alfanumeriek

```
FORMAT(123456789.123, 2) ⇨ '123,456,789.12'
FORMAT(123456789.123, 0) ⇨ '123,456,789'
```

FOUND_ROWS()

Beschrijving: Deze functie geeft het aantal rijen in het resultaat van de voorgaande SELECT-instructie.
Datatype: numeriek

```
FOUND_ROWS() ⇨ 14
```

FROM_DAYS(*par1*)

Beschrijving: Deze functie bepaalt de datum die hoort bij een aantal dagen dat verlopen is sinds het jaar 0. De parameter vormt het aantal dagen en moet liggen tussen 366 en 3.652.424.
Datatype: datum

```
FROM_DAYS(366)                      ⇨ '0001-01-01'
FROM_DAYS(366*2000)                 ⇨ '2004-02-24'
FROM_DAYS(3652424)                  ⇨ '9999-12-31'
FROM_DAYS(3652500)                  ⇨ '0000-00-00'
FROM_DAYS(3652424) – INTERVAL 5 DAY ⇨ '9999-12-26'
```

GET_FORMAT(*par1, par2*)

Beschrijving: Deze functie geeft een formaat dat in andere functies, zoals DATE_FORMAT, TIME_FORMAT en STR_TO_DATE gebruikt kan worden. De eerste parameter geeft het datatype weer. Dit moet gelijk zijn aan DATE, TIME of DATETIME. De tweede parameter geeft het formaattype weer. Mogelijke waarden zijn 'EUR', 'INTERNAL', 'ISO', 'JIS' en 'USA'. De onderstaande voorbeelden geven alle mogelijkheden weer.
Datatype: alfanumeriek

```
GET_FORMAT(DATE, 'EUR')          ⇨ '%d.%m.%Y'
GET_FORMAT(DATE, 'INTERNAL')     ⇨ '%Y%m%d'
GET_FORMAT(DATE, 'ISO')          ⇨ '%Y-%m-%d'
GET_FORMAT(DATE, 'JIS')          ⇨ '%Y-%m-%d'
GET_FORMAT(DATE, 'USA')          ⇨ '%m.%d.%Y'
GET_FORMAT(TIME, 'EUR')          ⇨ '%H.%i.%s'
```

```
GET_FORMAT(TIME, 'INTERNAL')     ⇨ '%H%i%s'
GET_FORMAT(TIME, 'ISO')          ⇨ '%H:%i:%s'
GET_FORMAT(TIME, 'JIS')          ⇨ '%H:%i:%s'
GET_FORMAT(TIME, 'USA')          ⇨ '%h:%i:%s %p'
GET_FORMAT(DATETIME, 'EUR')      ⇨ '%Y-%m-%d %H.%i.%s'
GET_FORMAT(DATETIME, 'INTERNAL') ⇨ '%Y%m%d%H%i%s'
GET_FORMAT(DATETIME, 'ISO')      ⇨ '%Y-%m-%d %H:%i:%s'
GET_FORMAT(DATETIME, 'JIS')      ⇨ '%Y-%m-%d %H:%i:%s'
GET_FORMAT(DATETIME, 'USA')      ⇨ '%Y-%m-%d %H.%i.%s'

DATE_FORMAT('2005-01-01', GET_FORMAT(DATE, 'EUR')) ⇨ '01.01.2005'
DATE_FORMAT('2005-01-01', GET_FORMAT(DATE, 'ISO')) ⇨ '2005-01-01'
```

GREATEST(*par1, par2, ...*)

Beschrijving: Deze functie geeft de hoogste waarde uit een reeks parameters.
Datatype: afhankelijk van parameters

```
GREATEST(100, 4, 80)                            ⇨ 100
GREATEST(DATE('2005-01-01'), DATE('2005-06-12')) ⇨ '2005-06-12'
```

HEX(*par1*)

Beschrijving: Indien de parameter numeriek is geeft deze functie de hexadecimale representatie van de parameter. Indien de parameter alfanumeriek is geeft deze functie van elk teken een twee-cijferige code.
Datatype: alfanumeriek

```
HEX(11)    ⇨ 'B'
HEX(16)    ⇨ '10'
HEX(100)   ⇨ '64'
HEX(1000)  ⇨ '3E8'
HEX('3E8') ⇨ '334538'
HEX('ç')   ⇨ 'E7'
```

HOUR(*par1*)

Beschrijving: Deze functie geeft van een tijd- of timestamp-expressie de waarde van de uur-component. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 0 en kleiner dan of gelijk aan 23.
Datatype: numeriek

```
HOUR('2005-01-01 12:13:14') ⇨ 12
HOUR('12:13:14')            ⇨ 12
HOUR(CURTIME())             ⇨ 19
```

IF(*par1, par2, par3*)

Beschrijving: Indien de waarde van de eerste parameter waar is, is het resultaat van de functie gelijk aan de waarde van de tweede parameter en anders gelijk aan de waarde van de derde parameter. De specificatie

```
IF(E1, E2, E3)
```

waarbij E_1, E_2 en E_3 expressies zijn, is gelijkwaardig aan de volgende case-expressie

```
CASE
   WHEN E1 = TRUE THEN E2
   ELSE E3
END
```

Datatype: afhankelijk van de laatste twee parameters

```
IF((5>8), 'Jim', 'John') ⇨ 'John'
```

```
IF((SELECT COUNT(*) FROM SPELERS) =
   (SELECT COUNT(*) FROM BOETES), TRUE, FALSE) ⇨ 0
```

IFNULL(*par1, par2*)

Beschrijving: Indien de waarde van de eerste parameter gelijk is aan de null-waarde, is het resultaat van de functie gelijk aan de waarde van de tweede parameter en anders gelijk aan de waarde van de eerste parameter. De specificatie

```
IFNULL(E1, E2)
```

waarbij E_1 en E_2 twee expressies zijn, is gelijkwaardig aan de volgende case-expressie

```
CASE E1
   WHEN NULL THEN E2
   ELSE E1
END
```

Datatype: afhankelijk van parameters

```
IFNULL(NULL, 'John')  ⇨ 'John'
IFNULL('John', 'Jim') ⇨ 'John'
```

INSERT(*par1, par2, par3, par4*)

Beschrijving: De waarde van de vierde parameter wordt geplaatst op dat deel van de eerste parameter dat begint bij de positie aangegeven met de tweede parameter en een aantal (de derde parameter) tekens lang is.
Datatype: alfanumeriek

```
INSERT('abcdefgh',4,3,'zzz')  ⇨ 'abczzzgh'
INSERT('abcdefgh',4,2,'zzz')  ⇨ 'abczzzfgh'
INSERT('abcdefgh',4,0,'zzz')  ⇨ 'abczzzdefgh'
INSERT('abcdefgh',4,-1,'zzz') ⇨ 'abczzz'
INSERT('abcdefgh',1,5,'zzz')  ⇨ 'zzzfgh'
```

INSTR(*par1, par2*)

Beschrijving: Deze functie geeft de startpositie van de tweede alfanumerieke waarde binnen de eerste alfanumerieke waarde. De INSTR-functie heeft de waarde nul als de tweede alfanumerieke waarde niet binnen de eerste voorkomt.
Datatype: numeriek

```
INSTR('database','bas') ⇨ 5
INSTR('systeem','bas')  ⇨ 0
```

INTERVAL(*par, par2, par3, ...*)

Beschrijving: Met deze functie wordt bepaald tussen welke twee waarden in een lijst de eerste parameter voorkomt. Na de eerste parameter moeten de waarden in oplopende volgorde gespecificeerd worden.
Datatype: afhankelijk van de laatste twee parameters

```
INTERVAL(3,0,1,2,3,4,5,6,7) ⇨ 4
INTERVAL(7,0,6,11,16,21)    ⇨ 2
```

ISNULL(*par1*)

Beschrijving: De waarde van deze functie is gelijk aan 1 als de eerste parameter gelijk is aan de null-waarde, en anders gelijk aan 0. De specificatie

```
ISNULL(E1)
```

waarbij E_1 een expressie is, is gelijkwaardig aan de volgende case-expressie

```
CASE E1
   WHEN NULL THEN 1
   ELSE 0
END
```

Datatype: afhankelijk van parameters

```
ISNULL((SELECT BONDSNR FROM SPELERS WHERE SPELERSNR=27)) ⇨ 0
ISNULL((SELECT BONDSNR FROM SPELERS WHERE SPELERSNR=7))  ⇨ 1
```

LAST_DAY(*par1*)

Beschrijving: Deze functie geeft de laatste dag van de maand behorende bij een datum- of timestamp-expressie.
Datatype: datum

```
LAST_DAY('2004-02-01') ⇨ '2005-02-29'
LAST_DAY('2005-02-01') ⇨ '2005-02-28'
```

LCASE(*par1*)

Beschrijving: Deze functie zet alle hoofdletters van de waarde van de parameter om in kleine letters.
Datatype: alfanumeriek

```
LCASE('RICK') ⇨ 'rick'
```

LEAST(*par1, par2, …*)

Beschrijving: Deze functie geeft de kleinste waarde uit een reeks parameters.
Datatype: afhankelijk van parameters

```
LEAST(100, 4, 80)                            ⇨ 4
LEAST(DATE('2005-01-01'), DATE('2005-06-12')) ⇨ 2005-01-01
```

LEFT(*par1, par2*)

Beschrijving: Deze functie geeft het linkerdeel van een alfanumerieke waarde (de eerste parameter). De lengte van het deel dat gepakt wordt, wordt met de tweede parameter aangegeven.
Datatype: alfanumeriek

```
LEFT('database', 4)           ⇨ 'data'
LEFT('database', 0)           ⇨ ''
LEFT('database', 10)          ⇨ 'database'
LEFT('database', NULL)        ⇨ ''
LENGTH(LEFT('database', 0))   ⇨ 0
LENGTH(LEFT('database', 10))  ⇨ 8
LENGTH(LEFT('database', NULL)) ⇨ 0
```

LENGTH(*par1*)

Beschrijving: Deze functie geeft de lengte in bytes van een alfanumerieke waarde.
Datatype: numeriek

```
LENGTH('database')       ⇨ 8
LENGTH('data    ')       ⇨ 8
```

```
LENGTH(RTRIM('abcd    ')) ⇨ 4
LENGTH('')               ⇨ 0
LENGTH(NULL)             ⇨ NULL
```

LN(*par1*)

Beschrijving: Deze functie geeft de logaritme voor het grondtal *e* van de parameter. Zie ook de LOG-functie.
Datatype: numeriek

```
LN(50)     ⇨ 3.9120230054281
LN(EXP(3)) ⇨ 3
LN(0)      ⇨ NULL
LN(1)      ⇨ 0
```

LOCALTIME()

Beschrijving: Deze functie geeft de systeemdatum en systeemtijd. Indien de functie binnen een numerieke expressie gebruikt wordt, is het resultaat numeriek. De haakjes mogen weggelaten worden. Zie ook de NOW- en LOCALTIMESTAMP-functies.
Datatype: timestamp of double

```
LOCALTIME()     ⇨ '2005-02-20 12:26:52'
LOCALTIME() + 0 ⇨ 20050220122652
```

LOCALTIMESTAMP()

Beschrijving: Deze functie geeft de systeemdatum en systeemtijd. Indien de functie binnen een numerieke expressie gebruikt wordt, is het resultaat numeriek. De haakjes mogen weggelaten worden. Zie ook de NOW- en LOCALTIME-functies.
Datatype: timestamp of double

```
LOCALTIMESTAMP()     ⇨ '2005-02-20 12:26:52'
LOCALTIMESTAMP() + 0 ⇨ 20050220122652
```

LOCATE(*par1, par2, par3*)

Beschrijving: Deze functie geeft de startpositie van de eerste alfanumerieke waarde binnen de tweede alfanumerieke waarde. De LOCATE-functie heeft de waarde nul als de eerste alfanumerieke waarde niet binnen de tweede voorkomt. Een derde parameter mag opgegeven worden om een positie aan te geven vanaf waar gezocht moet worden.
Datatype: numeriek

```
LOCATE('bas','database')   ⇨ 5
LOCATE('bas','database',6) ⇨ 0
LOCATE('bas','systeem')    ⇨ 0
```

LOG(*par1*)

Beschrijving: Deze functie geeft de logaritme voor het grondtal *e* van de parameter. Zie ook de LN-functie.
Datatype: numeriek

```
LOG(50)     ⇨ 3.9120230054281
LOG(EXP(3)) ⇨ 3
LOG(0)      ⇨ NULL
LOG(1)      ⇨ 0
```

LOG(*par1, par2*)

Beschrijving: Deze functie geeft de logaritme van de tweede parameter waarbij de eerste parameter het grondtal vormt.
Datatype: numeriek

```
LOG(10,1000) ⇨ 3
LOG(2,64)    ⇨ 6
```

LOG10(*par1*)

Beschrijving: Deze functie geeft de logaritme voor het grondtal 10 van de parameter.
Datatype: numeriek

```
LOG10(1000)        ⇨ 3
LOG10(POWER(10,5)) ⇨ 5
```

LOG2(*par1*)

Beschrijving: Deze functie geeft de logaritme voor het grondtal 2 van de parameter.
Datatype: numeriek

```
LOG2(2)          ⇨ 1
LOG2(64)         ⇨ 6
LOG2(POWER(2,10) ⇨ 10
```

LOWER(*par1*)

Beschrijving: Deze functie zet alle hoofdletters van de waarde van de parameter om in kleine letters. Zie ook de LCASE-functie.
Datatype: alfanumeriek

```
LOWER('RICK') ⇨ 'rick'
```

LPAD(*par1, par2, par3*)

Beschrijving: De waarde van de eerste parameter wordt aan de voorkant (de linkerzijde) net zo vaak met de waarde van de derde parameter aangevuld totdat de totale lengte van de waarde gelijk is aan die van de tweede parameter. Indien de maximale lengte kleiner is dan die van de eerste parameter, wordt de eerste parameter aan de linkerkant ingekort.
Datatype: alfanumeriek

```
LPAD('data', 16, 'base') ⇨ 'basebasebasedata'
LPAD('data', 6, 'base')  ⇨ 'badata'
LPAD('data', 2, 'base')  ⇨ 'da'
```

LTRIM(*par1*)

Beschrijving: Deze functie verwijdert alle spaties die aan het begin van de parameter staan.
Datatype: alfanumeriek

```
LTRIM('   database') ⇨ 'database'
```

MAKEDATE(*par1, par2*)

Beschrijving: De tweede parameter stelt een aantal dagen voor en die worden bij de tweede parameter opgeteld. Deze tweede parameter moet een numerieke, datum- of timestamp-expressie zijn.
Datatype: datum

```
MAKEDATE(2005, 1)                 ⇨ '2005-01-01'
MAKEDATE(2005, 10)                ⇨ '2005-01-10'
MAKEDATE('2005-01-01', 1)         ⇨ '2005-01-01'
MAKEDATE('2005-01-01 12:26:52', 1) ⇨ '2005-01-01'
```

MAKETIME(*par1, par2, par3*)

Beschrijving: Deze functie creëert een tijd vanuit een aantal uren (de eerste parameter), een aantal minuten (de tweede parameter) en een aantal seconden (de derde parameter). Het aantal minuten en het aantal seconden moet liggen tussen 0 en 59, anders geeft de functie de null-waarde als resultaat.
Datatype: tijd

```
MAKETIME(12,13,14)  ⇨ '12:13:14'
MAKETIME(12,90,14)  ⇨ NULL
MAKETIME(120,13,14) ⇨ '120:13:14'
```

MICROSECOND(*par1*)

Beschrijving: Deze functie geeft uit een tijd- of timestamp-expressie het aantal microseconden. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 0 en kleiner dan of gelijk aan 999999.
Datatype: numeriek

```
MICROSECOND('2005-01-01 12:13:14.123456') ⇨ 123456
MICROSECOND('12:13:14.1')                 ⇨ 100000
```

MID(*par1, par2, par3*)

Beschrijving: Deze functie haalt een deel uit de alfanumerieke waarde van de eerste parameter. De tweede parameter geeft de beginpositie aan en de derde parameter het aantal tekens. Zie ook de SUBSTRING-functie.
Datatype: alfanumeriek

```
MID('database',5)    ⇨ 'base'
MID('database',10)   ⇨ ''
MID('database',5,2)  ⇨ 'ba'
MID('database',5,10) ⇨ 'base'
MID('database',-6)   ⇨ 'tabase'
```

MINUTE(*par1*)

Beschrijving: Deze functie geeft uit een tijd- of timestamp-expressie het aantal minuten. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 0, en kleiner dan of gelijk aan 59.
Datatype: numeriek

```
MINUTE(CURTIME())  ⇨ 52
MINUTE('12:40:33') ⇨ 40
```

MOD(*par1*)

Beschrijving: Deze functie geeft de rest van de deling van de twee parameters.
Datatype: numeriek

```
MOD(15,4)      ⇨ 3
MOD(15.4, 4.4) ⇨ 2.2
```

MONTH(*par1*)

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie het nummer van de maand. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 1, en kleiner dan of gelijk aan 12.

Datatype: numeriek

```
MONTH('1988-07-29') ⇨ 7
```

MONTHNAME

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie de Engelstalige naam van de maand.
Datatype: alfanumeriek

```
MONTHNAME('1988-05-20') ⇨ 'May'
MONTHNAME('1988-06-20') ⇨ 'June'
```

NOW()

Beschrijving: Deze functie geeft de systeemdatum en systeemtijd.
Datatype: timestamp

```
NOW() ⇨ '2005-12-20 12:26:52'
```

NULLIF(*par1, par2*)

Beschrijving: Indien de waarde van de eerste parameter ongelijk is aan die van de tweede parameter, is het resultaat van de functie gelijk aan de null-waarde, en anders gelijk aan de eerste parameter. De specificatie

```
NULLIF(E1, E2)
```

waarbij E_1 en E_2 twee expressies zijn, is gelijkwaardig aan de volgende case-expressie

```
CASE
   WHEN E1 = E2 THEN NULL
   ELSE E1
END
```

Datatype: afhankelijk van parameters

```
NULLIF(NULL, 'John')   ⇨ NULL
NULLIF('John', 'Jim')  ⇨ 'John'
NULLIF('John', 'John') ⇨ NULL
NULLIF(NULL, NULL)     ⇨ NULL
```

OCT(*par1*)

Beschrijving: Deze functie geeft de waarde volgens het octale stelsel van een numeriek getal.
Datatype: alfanumeriek

```
OCT(8)   ⇨ '10'
OCT(64)  ⇨ '100'
OCT(100) ⇨ '144'
```

OCTET_LENGTH(*par1*)

Beschrijving: Deze functie geeft de lengte in bytes van een octale waarde.
Datatype: numeriek

```
OCTET_LENGTH('100')   ⇨ 3
OCTET_LENGTH(OCT(64)) ⇨ 3
```

ORD(*par1*)

Beschrijving: Deze functie geeft de (ordinale) karakterset positie van het eerste teken van een alfanumerieke-expressie.
Datatype: numeriek

```
ORD('Database') ⇨ 68
ORD('database') ⇨ 100
ORD('')         ⇨ 0
ORD(NULL)       ⇨ NULL
```

PERIOD_ADD(*par1, par2*)

Beschrijving: Deze functie telt een aantal maanden bij een bepaalde datum op. De datum moet het formaat JJJJMM of JJMM hebben. Het formaat van het resultaat is JJJJMM. Deze functie werkt dus niet met datums.
Datatype: alfanumeriek

```
PERIOD_ADD('200508', 2)  ⇨ '200510'
PERIOD_ADD('200508', -2) ⇨ '200506'
PERIOD_ADD('200508', 12) ⇨ '200608'
```

PERIOD_DIFF(*par1, par2*)

Beschrijving: Deze functie bepaalt het aantal maanden dat ligt tussen twee datums. Beide datums moeten het formaat JJJJMM of JJMM hebben. Deze functie werkt dus niet met waarden met het datum-datatype.
Datatype: numeriek

```
PERIOD_DIFF('200508', '200510') ⇨ -2
PERIOD_DIFF('200508', '200506') ⇨ 2
PERIOD_DIFF('200508', '200608') ⇨ -12
```

PI()

Beschrijving: Deze functie geeft de waarde van het bekende getal *pi*.
Datatype: numeriek

```
PI()        ⇨ 3.141593
PI()*100000 ⇨ 314159.265359
```

POWER(*par1, par2*)

Beschrijving: De waarde van de eerste parameter wordt tot een bepaalde macht verheven. De tweede parameter geeft de macht aan.
Datatype: numeriek

```
POWER(4,3)    ⇨ 64
POWER(2.5,3)  ⇨ 15.625
POWER(4, 0.3) ⇨ 1.5157165665104
POWER(4, -2)  ⇨ 0.0625
```

QUARTER

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie het kwartaal. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 1 en kleiner dan of gelijk aan 4.
Datatype: numeriek

```
QUARTER('1988-07-29') ⇨ 3
QUARTER(CURDATE())    ⇨ 1
```

RADIANS(*par1*)

Beschrijving: Deze functie converteert een getal in graden naar een waarde in radialen.
Datatype: numeriek

```
RADIANS(90)         ⇨ 1.5707963267949
RADIANS(180) – PI() ⇨ 0
RADIANS(-360)       ⇨ -6.2831853071796
```

RAND(*par1*)

Beschrijving: Deze functie geeft een willekeurig getal (met een float-datatype) tussen 0.0 en 1.0. De parameter geeft het startpunt aan voor het berekenen van de volgende willekeurige waarde. Herhaaldelijk aanroepen van deze functie met dezelfde parameterwaarde geeft hetzelfde resultaat. Indien er geen parameter gespecificeerd wordt, wordt de volgende willekeurige waarde berekend.
Datatype: numeriek

```
RAND()                                  ⇨ 0.42908766346899
RAND(5)                                 ⇨ 0.40613597483014
CAST(RAND() * 10000 AS UNSIGNED INTEGER) ⇨ 8057
```

REPEAT(*par1, par2*)

Beschrijving: Deze functie herhaalt een alfanumerieke waarde (de eerste parameter) een bepaald aantal keer (de tweede parameter).
Datatype: alfanumeriek

```
REPEAT('bla',4) ⇨ 'blablablabla'
REPEAT('X',10)  ⇨ 'XXXXXXXXXX'
```

REPLACE(*par1, par2, par3*)

Beschrijving: Deze functie vervangt delen van de waarde van een alfanumerieke expressie door een andere waarde.
Datatype: alfanumeriek

```
REPLACE('database','a','e')        ⇨ 'detebese'
REPLACE('database','ba','warehou') ⇨ 'datawarehouse'
REPLACE('data base',' ','')        ⇨ 'database'
```

REVERSE(*par1*)

Beschrijving: Deze functie draait de tekens in een alfanumerieke waarde om.
Datatype: alfanumeriek

```
REVERSE('database') ⇨ 'esabatad'
```

RIGHT(*par1, par2*)

Beschrijving: Deze functie geeft het rechterdeel van een alfanumerieke waarde (de eerste parameter). De lengte van het deel dat gepakt wordt, wordt met de tweede parameter aangegeven.
Datatype: alfanumeriek

```
RIGHT('database', 4)           ⇨ 'base'
RIGHT('database', 0)           ⇨ ''
RIGHT('database', 10)          ⇨ 'database'
RIGHT('database', NULL)        ⇨ ''
LENGTH(RIGHT('database', 0))   ⇨ 0
LENGTH(RIGHT('database', 10))  ⇨ 8
```

```
LENGTH(RIGHT('database', NULL)) ⇨ 0
```

ROUND(*par1, par2*)

Beschrijving: Deze functie rondt getallen af bij een gegeven aantal cijfers achter de komma. Indien de tweede parameter niet gespecificeerd wordt, is dat gelijk aan het specificeren van 0.
Datatype: numeriek

```
ROUND(123.456,2)  ⇨ 123.46
ROUND(123.456,1)  ⇨ 123.5
ROUND(123.456,0)  ⇨ 123
ROUND(123.456,-1) ⇨ 120
ROUND(123.456,-2) ⇨ 100
ROUND(123.456)    ⇨ 123
```

RPAD(*par1, par2, par3*)

Beschrijving: De waarde van de eerste parameter wordt aan de voorkant (de rechterzijde) net zo vaak met de waarde van de derde parameter aangevuld tot de totale lengte van de waarde gelijk is aan die van de tweede parameter. Indien de maximale lengte kleiner is dan die van de eerste parameter, wordt de eerste parameter aan de rechterkant ingekort.
Datatype: alfanumeriek

```
RPAD('data', 16, 'base') ⇨ 'databasebasebase'
RPAD('data', 6, 'base')  ⇨ 'databa'
RPAD('data', 2, 'base')  ⇨ 'da'
```

RTRIM(*par1*)

Beschrijving: Deze functie verwijdert alle spaties die achteraan de waarde van de parameter staan.
Datatype: alfanumeriek

```
RTRIM('database     ')           ⇨ 'database'
CONCAT(RTRIM('data    '), 'base') ⇨ 'database'
```

SECOND(*par1*)

Beschrijving: Deze functie geeft uit een tijd- of timestamp-expressie het aantal seconden. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 0 en kleiner dan of gelijk aan 59.
Datatype: numeriek

```
SECOND(CURTIME())  ⇨ 6
SECOND('12:40:33') ⇨ 33
```

SEC_TO_TIME(*par1*)

Beschrijving: Deze functie transformeert een aantal seconden in een tijd.
Datatype: tijd

```
SEC_TO_TIME(1)            ⇨ '00:00:01'
SEC_TO_TIME(1000)         ⇨ '00:16:40'
SEC_TO_TIME((24*60*60)-1) ⇨ '23:59:59'
SEC_TO_TIME(24*60*60*2)   ⇨ '48:00:00'
```

SESSION_USER()

Beschrijving: Deze functie geeft de naam van de SQL-gebruiker.
Datatype: alfanumeriek

```
SESSION_USER() ⇨ 'root@localhost'
```

SIGN(*par1*)

Beschrijving: Deze functie geeft het teken van een numerieke waarde. Als de waarde van de parameter negatief is, dan is het resultaat gelijk aan -1, als het groter is dan nul, dan +1, en als de waarde gelijk is aan 0 dan is het resultaat van de functie ook gelijk aan 0.
Datatype: numeriek

```
SIGN(50)  ⇨ 1
SIGN(0)   ⇨ 0
SIGN(-50) ⇨ –1
```

SIN(*par1*)

Beschrijving: Deze functie geeft de sinus van een hoek in radialen. Het resultaat is een waarde die ligt tussen -1 en 1.
Datatype: numeriek

```
SIN(0)                  ⇨ 0
SIN(PI()/2)             ⇨ 1
SIN(PI())               ⇨ 0
SIN(1)-COS((PI()/2) - 1) ⇨ 0
```

SOUNDEX(*par1*)

Beschrijving: Deze functie geeft de SOUNDEX-code van de alfanumerieke parameter. Een SOUNDEX-code bestaat uit vier tekens. Alfanumerieke waarden die ongeveer gelijk klinken, worden omgezet naar identieke SOUNDEX-codes. De SOUNDEX-code wordt als volgt bepaald:

- Alle spaties aan het begin van de parameter worden verwijderd.
- Uit de parameter worden alle volgende letters verwijderd: a e h i o u w y, mits zij niet op de eerste positie staan.
- Aan de resterende letters worden de volgende waarden toegekend:

```
b f p v         = 1
c g j k q s x z  = 2
d t             = 3
l               = 4
m n             = 5
r               = 6
```

- Indien twee aansluitende letters dezelfde waarde hebben, wordt de tweede verwijderd.
- De code wordt afgebroken achter het vierde teken.
- Indien de resterende code uit minder dan vier tekens bestaat, wordt zij opgevuld met nullen.
- Tekens die achter een spatie staan worden overgeslagen.
- Indien de waarde van de parameter niet met een letter begint, is het resultaat gelijk aan '0000'.

Datatype: alfanumeriek

```
SOUNDEX('John')       ⇨ 'J500'
SOUNDEX('Jan')        ⇨ 'J50'
SOUNDEX('Joop')       ⇨ 'J100'
SOUNDEX(' Joop')      ⇨ 'J100'
SOUNDEX('Joop Karel') ⇨ 'J1264'
```

SPACE(*par1*)

Beschrijving: Deze functie geeft een rij met spaties. Het aantal spaties is gelijk aan de waarde van de numerieke parameter.
Datatype: alfanumeriek

```
SPACE(1)         ⇨ ' '
SPACE(5)         ⇨ '     '
LENGTH(SPACE(8)) ⇨ 8
```

SQRT(*par1*)

Beschrijving: Deze functie geeft de vierkantswortel uit de waarde van de parameter.
Datatype: numeriek

```
SQRT(225) ⇨ 15
SQRT(200) ⇨ 14.14
SQRT(-5)  ⇨ NULL
```

STRCMP(*par1, par2*)

Beschrijving: Deze functie vergelijkt de waarden van twee alfanumerieke expressies. Het resultaat is 0 indien de waarden van de parameters gelijk zijn, het resultaat is -1 als de waarde van de eerste parameter kleiner is en 1 als de rechter kleiner is.
Datatype: numeriek

```
STRCMP(1,1) ⇨ 0
STRCMP(1,2) ⇨ -1
STRCMP(2,1) ⇨ 1
```

STR_TO_DATE(*par1, par2*)

Beschrijving: Deze functie werkt tegenovergesteld aan de DATE_FORMAT-functie; een bepaalde alfanumerieke waarde wordt via een aantal opmaakcodes omgezet in een datum- of timestamp-waarde. Indien de opmaakcodes niet bij de eerste parameter passen, dan geeft de functie een null-waarde als resultaat.
Datatype: datum of timestamp

```
STR_TO_DATE('2005 Sun Oct 1st', '%Y %a %b %D')⇨ '2005-10-01'
STR_TO_DATE('2005/11/10', '%Y/%c/%d')         ⇨ '2005-11-10'
```

SUBDATE(*par1, par2*)

Beschrijving: Deze functie trekt een interval (de tweede parameter) af van een datum- of timestamp-expressie (de eerste parameter). Indien de tweede parameter geen interval is maar een numeriek getal, gaat SQL ervan uit dat deze waarde een aantal dagen voorstelt.
Datatype: datum of timestamp

```
SUBDATE('2004-01-01', INTERVAL 5 MONTH)        ⇨ '2003-08-01'
SUBDATE('2004-01-01 12:00:00', INTERVAL 5 DAY) ⇨ '2003-12-27 12:00:00'
SUBDATE('2004-01-01', 5)                       ⇨ '2003-12-27'
```

SUBSTRING(*par1, par2, par3*)

Beschrijving: Deze functie haalt een deel uit de alfanumerieke waarde van de eerste parameter. De tweede parameter geeft de beginpositie aan en de derde parameter het aantal tekens. Indien de derde parameter niet gespecificeerd wordt, wordt tot aan het laatste teken meegenomen.
Datatype: alfanumeriek

```
SUBSTRING('database',5)    ⇨ 'base'
SUBSTRING('database',10)   ⇨ ''
SUBSTRING('database',5,2)  ⇨ 'ba'
SUBSTRING('database',5,10) ⇨ 'base'
SUBSTRING('database',-6)   ⇨ 'tabase'
```

SUBSTRING(*par1* FROM *par2* FOR *par3*)

Beschrijving: Deze functie haalt een deel uit de alfanumerieke waarde van de eerste parameter. De tweede parameter geeft de beginpositie aan en de derde parameter het aantal tekens. Indien de derde parameter niet gespecificeerd wordt, wordt tot aan het laatste teken meegenomen.
Datatype: alfanumeriek

```
SUBSTRING('database' FROM 5)        ⇨ 'base'
SUBSTRING('database' FROM 10)       ⇨ ''
SUBSTRING('database' FROM 5 FOR 2)  ⇨ 'ba'
SUBSTRING('database' FROM 5 FOR 10) ⇨ 'base'
SUBSTRING('database' FROM -6)       ⇨ 'tabase'
```

SUBSTRING_INDEX(*par1, par2, par3*)

Beschrijving: Deze functie zoekt het *n*de voorkomen van een alfanumerieke waarde in de waarde van de eerste parameter. De tweede parameter geeft aan welke waarde gezocht moet worden en de derde parameter het getal *n*. Indien de derde parameter positief is, wordt vanaf links gezocht naar het *n*de voorkomen en de functie geeft dan alles links van dat voorkomen. Indien de derde parameter negatief is, wordt vanaf rechts gezocht naar het *n*de voorkomen en de functie geeft dan alles rechts van dat voorkomen.
Datatype: alfanumeriek

```
SUBSTRING_INDEX('database', 'a', 3)     ⇨ 'datab'
SUBSTRING_INDEX('database', 'a', -3)    ⇨ 'tabase'
SUBSTRING_INDEX('database', 'data', 1)  ⇨ ''
SUBSTRING_INDEX('database', 'data', -1) ⇨ 'base'
```

SUBTIME(*par1, par2*)

Beschrijving: Deze functie trekt twee tijdexpressies van elkaar af en geeft een nieuwe tijd.
Datatype: tijd

```
SUBTIME('12:59:00', '0:59:00')     ⇨ '12:00:00'
SUBTIME('12:00:00', '0:00:00.001') ⇨ '11:59:59.999000'
SUBTIME('100:00:00', '900:00:00')  ⇨ '-800:00:00'
```

SYSDATE()

Beschrijving: Geeft de systeemdatum en systeemtijd. Indien de functie binnen een numerieke expressie gebruikt wordt, is het resultaat numeriek. Zie ook de LOCALTIME- en LOCALTIMESTAMP-functies.
Datatype: timestamp of numeriek

```
SYSDATE()     ⇨ '2005-02-20 12:26:52'
SYSDATE() + 0 ⇨ 20050220122652
```

SYSTEM_USER()

Beschrijving: Deze functie geeft de naam van de SQL-gebruiker.
Datatype: alfanumeriek

```
SYSTEM_USER() ⇨ 'root@localhost'
```

TAN(*par1*)

Beschrijving: Deze functie geeft de tangens van een hoek in radialen.
Datatype: numeriek

```
TAN(0)      ⇨ 0
TAN(PI())   ⇨ 0
TAN(PI()/4) ⇨ 1
TAN(1)      ⇨ 1.5574077246549
```

TIME()

Beschrijving: Deze functie geeft het tijdsdeel van een tijd- of timestamp-expressie.
Datatype: tijd

```
TIME('2005-12-08 12:00:00') ⇨ '12:00:00'
TIME('12:13')               ⇨ '12:13:00'
```

TIMEDIFF(*par1, par2*)

Beschrijving: Deze functie geeft de hoeveelheid tijd die verstreken is tussen twee tijdexpressies.
Datatype: tijd

```
TIMEDIFF('12:00:01','12:00:00') ⇨ '00:00:01'
TIMEDIFF('12:00:00','12:00:01') ⇨ '-00:00:01'
TIMEDIFF('23:01:01','22:00:59') ⇨ '01:00:02'
```

TIME_FORMAT(*par1, par2*)

Beschrijving: Deze functie transformeert een tijd-, datum- of timestamp-expressie (de eerste parameter) naar een alfanumerieke waarde. Hierbij geeft de tweede parameter het formaat aan van die alfanumerieke waarde en hierbij kunnen diverse speciale opmaakcodes gebruikt worden; zie de onderstaande tabel. Deze functie lijkt veel op de DATE_FORMAT-functie, echter nu mogen alle tijdgerelateerde opmaakcodes gebruikt worden.

Opmaakcode	Toelichting
%f	Zescijferige numerieke code voor het aantal microseconden (000000 tot en met 999999)
%H	Tweecijferige numerieke code voor het uur (00 tot en met 23)
%h	Tweecijferige numerieke code voor het uur (01 tot en met 12)
%I	Tweecijferige numerieke code voor het uur (01 tot en met 12)
%i	Tweecijferige numerieke code voor het aantal minuten (00 tot en met 59)
%k	Een- of tweecijferige numerieke code voor het uur (0 tot en met 23)
%l	Een- of tweecijferige numerieke code voor het uur (1 tot en met 12)
%p	Aanduiding AM of PM
%r	Aanduiding van de tijd (in 12 uren) met het formaat `UU:MM:SS` gevolgd door AM of PM
%S	Tweecijferige numerieke code voor het aantal seconden (00 tot en met 59)
%s	Tweecijferige numerieke code voor het aantal seconden (00 tot en met 59)
%T	Aanduiding van de tijd (in 24 uren) met het formaat `UU:MM:SS` gevolgd door AM of PM
%%	Geeft het procentteken

Datatype: alfanumeriek

```
TIME_FORMAT('11:12:13','%h')               ⇨ '11'
TIME_FORMAT('11:12:13','%f')               ⇨ '000000'
TIME_FORMAT('12:00:00', 'Het is nu %h uur') ⇨ 'Het is nu 12 uur'
```

TIMESTAMP(*par1, par2*)

Beschrijving: Deze functie transformeert de eerste parameter naar een timestamp-waarde. Indien een tweede parameter gespecificeerd wordt, moet dat een tijdexpressie zijn die bij de waarde van de eerste parameter wordt opgeteld.
Datatype: timestamp

```
TIMESTAMP('2005-12-08')          ⇨ '2005-12-08 00:00:00'
TIMESTAMP('2005-12-08 12:00:00') ⇨ '2005-12-08 12:00:00'
TIMESTAMP('2005-12-08 12:00:00', '11:12:13')
                                 ⇨ '2005-12-08 23:12:13'
TIMESTAMP('2005-12-08 12:00:00', '-11:12:00')
                                 ⇨ '2005-12-08 00:48:00'
TIMESTAMP('2005-12-08 12:00:00', '-48:00')
                                 ⇨ '2005-12-06 12:00:00'
```

TIMESTAMPADD(*par1, par2, par3*)

Beschrijving: Met deze functie wordt een bepaald interval bij een datum- of timestamp-expressie opgeteld. De eerste parameter geeft het intervalsoort aan, zoals dagen, maanden en jaren, de tweede parameter de hoeveelheid dagen of maanden en de derde parameter is de expressie waarbij het interval opgeteld wordt. Ondersteunde intervalsoorten zijn YEAR, QUARTER, MONTH, WEEK, DAY, HOUR, MINUTE, SECOND en FRAC_SECOND.
Datatype: datum of timestamp

```
TIMESTAMPADD(DAY, 2, '2005-12-08')   ⇨ '2005-12-10'
TIMESTAMPADD(MONTH, 2, '2005-12-08') ⇨ '2006-02-08'
TIMESTAMPADD(YEAR, -2, '2005-12-08') ⇨ '2003-12-08'
TIMESTAMPADD(MINUTE, 3, '2005-12-08 12:00:00')
   ⇨ '2005-12-08 12:03:00'
TIMESTAMPADD(FRAC_SECOND, 3, '2005-12-08 12:00:00')
   ⇨ '2005-12-08 12:00:00.000003'
```

TIMESTAMPDIFF(*par1, par2, par3*)

Beschrijving: Met deze functie wordt de tijdperiode berekend die ligt tussen twee datum- of timestamp-expressies. De eerste parameter geeft het intervalsoort aan, zoals dagen, maanden en jaren, de tweede en de derde parameter vormen de twee expressies. Ondersteunde intervalsoorten zijn YEAR, QUARTER, MONTH, WEEK, DAY, HOUR, MINUTE, SECOND en FRAC_SECOND.
Datatype: numeriek

```
TIMESTAMPDIFF(DAY, '2005-12-04', '2005-12-08') ⇨ 4
TIMESTAMPDIFF(DAY, '2005-12-08', '2005-12-04') ⇨ -4
TIMESTAMPDIFF(YEAR, '1960-12-08', NOW())       ⇨ 45
TIMESTAMPDIFF(MINUTE, '2005-12-08 12:00:00',
   '2005-12-08 12:03:00') ⇨ 3
TIMESTAMPDIFF(FRAC_SECOND, '2005-12-08',
   '2005-12-08 12:00:00.000003') ⇨ 43200000003
```

TIME_TO_SEC(*par1*)

Beschrijving: Deze functie transformeert een tijd in een aantal seconden.
Datatype: numeriek

```
TIME_TO_SEC('00:00:01') ⇨ 1
TIME_TO_SEC('00:16:40') ⇨ 1000
TIME_TO_SEC('23:59:59') ⇨ 83399
TIME_TO_SEC('48:00:00') ⇨ 172800
```

TO_DAYS(*par1*)

Beschrijving: Met deze functie wordt bepaald hoeveel dagen er zijn verstreken tussen de gespecificeerde datum (de parameter) en het jaar 0.
Datatype: numeriek

```
TO_DAYS('2005-12-08') ⇨ 732653
```

TRIM(*par1*)

Beschrijving: Deze functie verwijdert alle spaties aan het begin en aan het einde van een alfanumerieke waarde (de parameter). Spaties in het midden worden niet verwijderd.
Datatype: alfanumeriek

```
TRIM('database     ') ⇨ 'database'
TRIM('    da ta    ') ⇨ 'da ta'
```

TRUNCATE(*par1, par2*)

Beschrijving: Deze functie kapt getallen (de eerste parameter) af bij een gegeven aantal cijfers (tweede parameter) achter de komma.
Datatype: numeriek

```
TRUNCATE(123.567, -1) ⇨ 120
TRUNCATE(123.567, 1)  ⇨ 123.5
TRUNCATE(123.567, 5)  ⇨ 123.56700
```

UCASE(*par1*)

Beschrijving: Deze functie zet alle kleine letters van de waarde van de parameter om in hoofdletters. Zie ook de UPPER-functie.
Datatype: alfanumeriek

```
UCASE('Database') ⇨ 'DATABASE'
```

UNHEX(*par1*)

Beschrijving: Deze functie geeft de hexadecimale representatie van de parameter. Elk tekenpaar wordt hierbij omgezet naar het corresponderende teken.
Datatype: alfanumeriek

```
UNHEX('334538')   ⇨ '3E8'
UNHEX('E7')       ⇨ 'ç'
UNHEX(HEX('SQL')) ⇨ 'SQL'
```

UPPER(*par1*)

Beschrijving: Deze functie zet alle kleine letters van de waarde van de parameter om in hoofdletters.
Datatype: alfanumeriek

```
UPPER('Database') ⇨ 'DATABASE'
```

USER()

Beschrijving: Deze functie geeft de naam van de SQL-gebruiker.
Datatype: alfanumeriek

```
USER() ⇨ 'root@localhost'
```

UTC_DATE()

Beschrijving: Deze functie geeft de werkelijke UTC-datum weer. UTC staat voor *Coordinated Universal Time*, ofwel Zulu time, of *Greenwich Mean Time* (GMT). Indien de functie een onderdeel van een numerieke expressie is, is het resultaat van de functie ook numeriek.
Datatype: datum of numeriek

```
UTC_DATE()     ⇨ '2005-01-01'
UTC_DATE() + 0 ⇨ 20050101
```

UTC_TIME()

Beschrijving: Deze functie geeft de werkelijke UTC tijd weer; zie de UTC_DATE-functie. Indien de functie een onderdeel van een numerieke expressie is, is het resultaat van de functie ook numeriek.
Datatype: datum of numeriek

```
UTC_TIME()                              ⇨ '2005-01-01'
HOUR(TIMEDIFF(UTC_TIME(), TIME(NOW()))) ⇨ 1
```

UTC_TIMESTAMP()

Beschrijving: Deze functie geeft de werkelijke UTC-datum en -tijd weer; zie de UTC_DATE-functie. Indien de functie een onderdeel van een numerieke expressie is, is het resultaat van de functie ook numeriek.
Datatype: datum of numeriek

```
UTC_TIMESTAMP() ⇨ '2005-01-01 13:56:12'
```

UUID()

Beschrijving: Deze functie genereert een 18-bytes brede unieke code. De afkorting UUID staat voor *Universal Unique Identifier*. De eerste drie delen van deze code worden afgeleid van de systeemtijd. Het vierde deel zorgt dat de codes uniek zijn voor het geval dat er door tijdzones anders dubbele waarden kunnen ontstaan. Het vijfde deel identificeert op een bepaalde wijze de server. Het genereren van unieke waarden is niet gegarandeerd, maar het is hoogst onwaarschijnlijk dat dubbele voorkomen.
Datatype: alfanumeriek

```
UUID() ⇨ '2bf2aaec-bc90-1028-b6bf-cc62846e9cc5'
UUID() ⇨ '390341e3-bc90-1028-b6bf-cc62846e9cc5'
```

VERSION()

Beschrijving: Deze functie geeft een identificatie van het versienummer van MySQL.
Datatype: alfanumeriek

```
VERSION() ⇨ '5.0.7-beta-nt'
VERSION() ⇨ '5.0.3-alpha-log'
```

WEEK(*part*)

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie de week. De waarde van het resultaat is altijd een geheel getal groter dan of gelijk aan 1 en kleiner dan of gelijk aan 53.
Datatype: numeriek

```
WEEK('1988-07-29') ⇨ 30
WEEK('1997-01-01') ⇨ 1
WEEK('2000-12-31') ⇨ 53
WEEK(CURDATE())    ⇨ 7
```

WEEKDAY(*par1*)

Beschrijving: Deze functie geeft het nummer van de dag in de week. Het resultaat is een getal dat ligt tussen 0 (maandag) en 6 (zondag).
Datatype: numeriek

```
WEEKDAY('2005-01-01') ⇨ 5
```

WEEKOFYEAR(*par1*)

Beschrijving: Deze functie geeft het weeknummer behorende bij een bepaalde datumexpressie. Het resultaat is een getal tussen 1 en 53.
Datatype: numeriek

```
WEEKOFYEAR('2005-01-01') ⇨ 53
WEEKOFYEAR('2005-01-03') ⇨ 1
```

YEAR(*par1*)

Beschrijving: Deze functie geeft uit een datum- of timestamp-expressie het nummer van het jaar (jaartal). Het resultaat is altijd een getal groter dan 0.
Datatype: numeriek

```
YEAR(NOW()) ⇨ 1998
```

YEARWEEK(par1, par2)

Beschrijving: Indien er slechts één parameter gespecificeerd wordt, geeft deze functie uit een datum- of timestamp-expressie het jaartal gevolgd door het weeknummer in het formaat JJJJWW. Het weeknummer loopt van 01 tot en met 53. Er wordt van uitgegaan dat een week op zondag start. Als er een tweede parameter gespecificeerd wordt, moet dat dezelfde code zijn als die gebruikt wordt bij de WEEK-functie.
Datatype: numeriek

```
YEARWEEK('2005-12-03')   ⇨ 200548
YEARWEEK('2005-12-03',0) ⇨ 200548
YEARWEEK('2005-01-02',0) ⇨ 200501
YEARWEEK('2005-01-02',1) ⇨ 200453
```

De Auteur

Rick F. van der Lans is auteur van vele boeken over SQL. Naast dit SQL Leerboek dat in diverse talen vertaald is, waaronder Engels, Duits, Chinees enItaliaans, heeft hij SQL boeken geschreven voor producten als MySQL, Oracle, SQLite, Ingres en Pervasive PSQL.

Hij is onafhankelijk adviesur, auteur en docent gespecialiseerd in databasetechnologie, datawarehousing en applicatie-integratie. Hij is oprichter en directeur van R20/Consultancy. Door de jaren heen heeft hij veel organisaties geadviseerd op het gebied van IT-architecturen.

Als spreker op conferenties en seminars wordt hij internationaal gerespecteerd. Al meer dan vijfentwintig jaar geeft hij over de gehele wereld lezingen, inclusief in de meeste Europese landen, Noord- en Zuid-Amerika en Australië. Hij is voorzitter van het jaarlijkse European Data Warehouse and Business Intelligence Conference. Hij schrijft een column voor Database Magazine en voor het internationale Beye-Network.com. Zeven jaar lang was hij lid van de Nederlandse ISO commissie verantwoordelijk voor ISO SQL Standaard.

Rick kan via de volgende kanalen bereikt worden:

Email: rick@r20.nl
Twitter: http://twitter.com/Rick_vanderlans
LinkedIn: http://www.linkedin.com/pub/rick-van-der-lans/9/207/223

Cursussen over de volgende onderwerpen kunnen door Rick F. van der Lans verzorgd worden

- Database-ontwerp en informatiemodellering
- De basis van SQL
- Het ontwikkelen van geavanceerde SQL queries
- Datawarehousing en business intelligence
- Data virtualisatie

Andere boeken geschreven door Rick F. van der Lans