**Inhoud**

[1 Het selecteren van gegevens (uitbreiding kolommen) 3](#_Toc48665957)

[1.1 Inleiding 3](#_Toc48665958)

[1.2 Eenvoudige bewerkingen op kolommen 3](#_Toc48665959)

[1.3 Functie TRUNC: afronden van getallen 4](#_Toc48665960)

[3.4 Functie CONCAT: samenvoegen van tekst 5](#_Toc48665965)

[3.5 Meer over functies 6](#_Toc48665966)

# Het selecteren van gegevens (uitbreiding kolommen)

## Inleiding

In een eerdere les hebben we al een begin gemaakt met het SELECT statement. We gaan deze kennis nu uitbreiden.

Na bestudering van dit hoofdstuk dient de gebruiker tot het volgende in staat te zijn.

* + Het kunnen uitvoeren van bewerkingen met kolommen.
  + Kunnen werken met eenvoudige functies

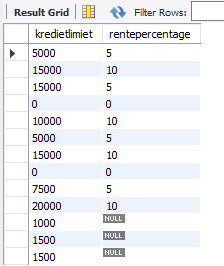
## Eenvoudige bewerkingen op kolommen

Tot nu toe hebben we steeds alleen de inhoud van kolommen getoond. Maar je kan ook bewerkingen uitvoeren op kolommen.

 Voer de volgende query uit:

***select kredietlimiet, rentepercentage***

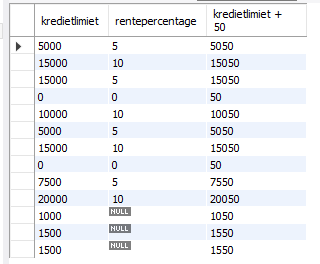
***from tblKlant;***



 De volgende query toont hoeveel de kredietlimiet wordt als elke klant 50 euro meer krediet heeft:

***select kredietlimiet, rentepercentage, kredietlimiet + 50***

***from tblKlant;***

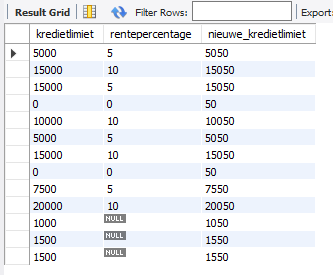


Je niet nu dat de berekening die wij uitgevoerd hebt in de output de kop  krijgt. Dat is niet zo mooi.

 Pas nu de vorige query aan en voer deze uit:

***select kredietlimiet, rentepercentage, kredietlimiet + 50 as nieuwe\_kredietlimiet***

***from tblKlant;***



Nu staat in de output een duidelijke naam voor de kolom. Door het toevoegen van **as nieuwe\_kredietlimiet** is een zogenaamde *kolomalias* gemaakt. Let op dat de naam die je achter **as** opgeeft geen spaties mag hebben en begint met een letter.

Als je een berekening uitvoert, maak je gebruik van zogenaamde operatoren. Er zijn verschillende operatoren die je kan gebruiken. De meest voorkomende worden hier weergegeven:

|  |  |
| --- | --- |
| **Operator** | **Betekenis** |
| **\*** | Vermenigvuldigen |
| **/** | Delen |
| **+** | Optellen |
| - | Aftrekken |
| () | Met haakjes bepaal je de volgorde van uitvoeren. |

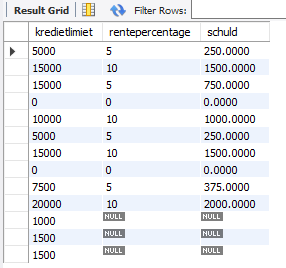
## Functie TRUNC: afronden van getallen

Stel nu dat je wil weten hoeveel schuld de klant heeft als hij de volledige limiet heeft gebruikt. Dat kun je dan berekenen door de kredietlimiet te vermednigvuldigen met het rentepercentage gedeeld door 100:

*SELECT kredietlimiet, rentepercentage, kredietlimiet \* rentepercentage/100 as schuld*

*FROM tblKlant;*

 Voer deze query uit. Je krijgt dan de volgende output:



Je ziet dat er nu een nieuw getal uitkomt met 4 decimalen. Dat kun je veranderen door gebruiken te maken van de functie ROUND. De syntax hiervoor is:

*SELECT ROUND(<kolom of berekening>,n)*

*FROM tabel;*

Met ,n geef wordt het aantal decimalen aangegeven waarop je wil afronden. Als je deze weglaat wordt er op 0 decimalen afgerond.

 Voer de volgende query uit:

***select kredietlimiet, rentepercentage, round(kredietlimiet \* rentepercentage/100, 2) as schuld***

***from tblKlant;***



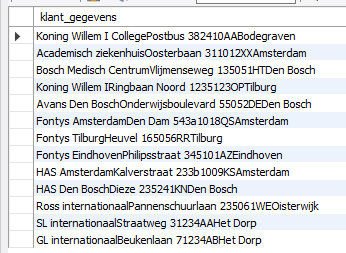
## Functie CONCAT: samenvoegen van tekst

In de database worden de NAW-gegevens (naam, adres, woonplaats) meestal in aparte kolommen opgeslagen. Soms wil je deze wel eens achter elkaar tonen. Dat kun je doen door gebruik te maken van de CONCAT functie.

 Voer de volgende query uit:

***select concat(naam, adres, postcode, plaats) as klant\_gegevens***

***from tblKlant;***

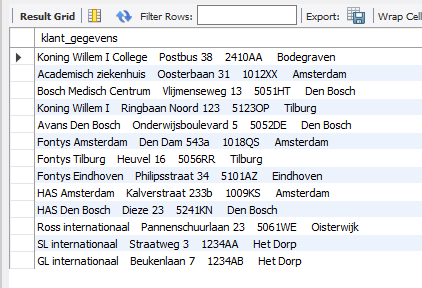


Je ziet dat alle kolommen aan elkaar geplakt zijn zonder spatie tussen bijvoorbeeld naam en adres. Dit los je op door spaties toe te voegen.

 Voer de volgende query uit (voor dit voorbeeld zijn meerdere spaties toegevoegd):

***select concat(naam, ' ', adres, ' ', postcode, ' ', plaats) as klant\_gegevens***

***from tblKlant;***



Je kan in plaats van spaties ook met stukjes tekst (of andere tekens) werken.

 Voer de volgende query uit:

***select concat('Bedrijf ', naam, ' is gevestigd in ', plaats) as dit\_kan\_ook***

***from tblKlant;***



## Meer over functies

Je hebt nu 2 functies geoefend in de SELECT clausule, maar je mag ze ook gebruiken in een UPDATE statement of de WHERE clausule.

Er zijn nog veel meer functies die je kan gebruiken. Deze zijn op internet terug te vinden in de documentatie van MySQL. Je hebt nu 2 voorbeelden gezien. Als je een andere functie wil gebruiken moet je goed de syntax lezen en kun je deze op dezelfde manier gebruiken.