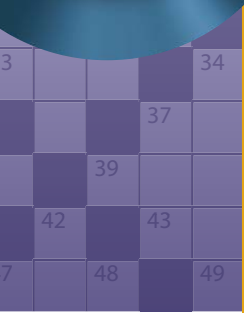


# Puzzelen met SQL



*Komkommertijd is aangebroken. Letterlijk. Nu heb ik zelf geen groene vingers, gelukkig maar anders wordt mijn toetsenbord zo vies, maar mijn vrouw wel. Sinds kort is zij helemaal op de 'biologische toer'. Als het niet biologisch is, dan komt het bij ons niet meer in huis. En wat is er nu het meest biologisch van allemaal? Juist, zelf planten, zelf oogsten en zelf opeten.*

Op de verschillende zakjes met zaden staat wanneer je deze moet inplanten, of althans bij benadering. Na het inzaaien moet je een bepaalde tijd wachten, voordat de kleine plantjes moeten worden verplaatst. Als de plantjes te dicht op elkaar staan kunnen ze niet echt lekker groeien en worden ze nooit groot. Na verloop van tijd moet er natuurlijk ook geoogst worden. Nu is het natuurlijk allemaal wel op een A4-tje te krijgen, het hele schema, maar we hebben niet voor niets een Oracle-database tot onze beschikking. Het plant-, verplant- en oogstschema gaan we dit keer met SQL bepalen. Het opeten van de groenten vind echter wel buiten de database plaats.

De lijst met de groenten die we willen gaan planten krijgen we als CSV (Comma Separated Values) aangeleverd. Maar de delimiter die gebruikt wordt is geen komma, maar een puntkomma. Deze gegevens willen we graag in een tabel plaatsen. Een manier om de CSV te splitsen zou handig zijn.

In Oracle 10g zijn Reguliere Expressies geïntroduceerd, en daar gaan we nu gebruik van maken.

De gegevens zien er zo uit:

```
Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage
Slabonen;Spinazie
```

Zoals je kunt zien zijn de verschillende groente namen gescheiden door een puntkomma. Sommige groenten bevatten ook spaties, zoals de 'Biet of Kroot' of de 'Lage Slabonen' - er zullen ook wel 'Hoge Slabonen' zijn. We gaan dit als volgt aanpakken: Eerst zorgen we dat er net zoveel rijen zijn als dat er verschillende groenten zijn. Vervolgens halen we de afzonderlijke groenten er met een Reguliere Expressie uit.

## Komkommertijd

### Stap 1: Net zoveel rijen als er groenten zijn

Maar hoeveel groenten zijn er eigenlijk in deze regel? Met Reguliere Expressies kun je op zoek gaan naar patronen in tekst. Hoewel Oracle 10g Reguliere Expressies introduceerde, is er een uitbereiding in Oracle 11g die het tellen een bepaald patroon ondersteunt, de REGEXP\_COUNT. De functie hebben we gevonden, maar welke expressie gaan we nu gebruiken? Om maar vast met de deur in huis te vallen, dit is de expressie:

```
'[^;]+'
```

Wat betekent dit? 'Geef me één of meerdere karakters die niet in de lijst staan'. De 'lijst' in dit geval zijn alle karakters die tussen de rechthoekige haken staan. De '^' (circonflexe) staat in dit geval voor 'niet' of 'behalve'. De '+' betekent: één of meerdere. Laten we het eens een keertje testen:

```
SQL> with groenten as
 2   (select 'Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage
 3     Slabonen;Spinazie' groente
 4     from dual
 5   )
 6   select regexp_count (groente, '[^;]+')
 7   from groenten
 8 /

REGEXP_COUNT(GROENTE, '[^;]+')
-----
6
```

Nu weten we hoeveel soorten groenten er in de tekst zitten en kunnen nu net zoveel rijen genereren. Er zijn verschillende manieren om dit te doen. Een gemakkelijke manier is om een grote tabel te kiezen, en deze met de ROWNUM te beperken.

Zoals in dit voorbeeld:

```
SQL> with groenten as
 2   (select 'Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage
 3     Slabonen;Spinazie' groente
 4     from dual
 5   )
```

```

5 , rijen as
6 (select l r
7   from all_objects
8 )
9 select groente
10  from groenten
11   , rijen
12  where rownum <= regexp_count (groente, '[^;]+')
13 /

GROENTE
-----
Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage Slabonen;Spinazie
Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage Slabonen;Spinazie
Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage Slabonen;Spinazie
Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage Slabonen;Spinazie
Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage Slabonen;Spinazie
Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage Slabonen;Spinazie

6 rows selected.
```

```

Biet of Kroot
Boerenkool
Knollenselderie
Lage Slabonen
Spinazie

6 rows selected.
```

En zo hebben we in ieder geval de verschillende groenten die we gaan planten.

De beschrijvingen van de verschillende groenten worden ook aangeleverd en wel in het volgende formaat:

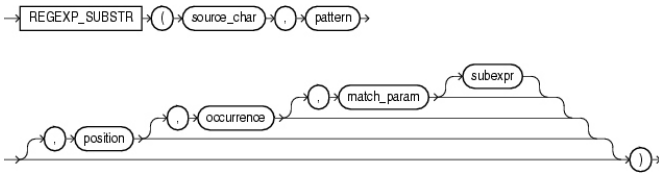
```

Andijvie#Cichorium Endivia: Juni-juli op zaaibed in de volleggrond
en na verspenen uitplanten of direct in regels zaaien en uitdunnen#Biet
of Kroot#Vanaf half ...
```

Zoals je ziet bouwen we eerst een cartetisch produkt op, waarna we het aantal rijen beperken. Nu hebben we net zoveel rijen als dat we groenten hebben, tijd voor stap 2.

**Stap 2: Met een reguliere expressie de verschillende groenten uit de rijen halen.**

Dit kunnen we natuurlijk met een gewone SUBSTR en INSTR doen, maar dat doen we niet. We maken gebruik van de Reguliere Expressie variant van de SUBSTR; REGEXP\_SUBSTR. De parameters die deze functie verwacht zijn als volgt:



Aangezien we de Reguliere Expressie al hebben, hoeven we nu alleen nog maar de overige argumenten mee te geven aan de functie en dan zijn we met dit deel klaar:

```

SQL> with groenten as
2 (select 'Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage
Slabonen;Spinazie' groente
3   from dual
4 )
5 , rijen as
6 (select l r
7   from all_objects
8 )
9 select regexp_substr (groente, '[^;]+', 1, rownum)
10  from groenten
11   , rijen
12  where rownum <= regexp_count (groente, '[^;]+')
13 /

REGEXP_SUBSTR(GROENTE, '[^;]+',1,ROWNUM)
-----
Andijvie
```

Alle informatie die je nodig hebt om de zaden op het juiste moment te kunnen zaaien staat beschreven na de plantnaam, gevolgd door een '#'. De gehele beschrijving van de plant loopt tot aan de volgende '#'. We krijgen dus één regel aangeleverd met daarin één kolom voor de verschillende plantnamen en één kolom met zowel de plantnaam als diens beschrijving. Om de juiste gegevens uit de tweede kolom te halen gaan we ook gebruik maken van Reguliere Expressies. Niet omdat het niet zou kunnen met een 'gewone' INSTR en SUBSTR, maar om te laten zien dat Reguliere Expressies een krachtig hulpmiddel kan zijn. Hier merken we bij op dat Reguliere Expressies nogal CPU-intensieve operaties kunnen zijn.

We willen nu een Reguliere Expressie gebruiken die de beschrijving van de groente erbij zoekt. Aangezien we nu een lijstje met zes groenten hebben, willen we wel de juiste beschrijving bij de juiste groente hebben. Per rij gaan we op zoek naar de positie waar de groente staat. Als we die gevonden hebben, dan willen we alle tekst zien die na het eerst volgende '#' staat, tot aan de dan volgende '#'.

Om er voor te zorgen dat niet de afzonderlijke letters van de groente gebruikt worden om de positie van de groente in de beschrijvingsregel te vinden, gebruiken we haakjes. Door de haakjes wordt de groente als een subexpressie behandeld.

```
'(.*)('||currgroente||')#(.*)#'
```

De huidige groente, gevolgd door een '#', dan nul of meerdere karakters, wederom gevolgd door een "#". Voor de huidige groente kunnen nul of meerdere karakters staan, vandaar de expressie:

```
(.*)
```

Dit levert natuurlijk veel te veel informatie op, als je bijvoorbeeld 'Boerenkool' hebt gevonden vind je alle beschrijvingen die na die van de Boerenkool staan, zelfs die van de Spinazie, en

dat is teveel informatie. Eigenlijk willen we alleen die beschrijving die bij de Boerenkool hoort, in ons geval is dit de derde beschrijving. De overige beschrijvingen willen we eigenlijk wegpoetsen. Dit gaan we doen met de Reguliere Expressie variant van de REPLACE functie.

```
regexp_replace ( beschrijving
, '(.*)('|currgroente|')#(.*?)#'
, '\3'
)
```

Deze expressie bevat drie subexpressies, deze staan namelijk tussen de haakjes. Omdat we niet geïnteresseerd zijn in de huidige groente, die weten we al, maar wel in de beschrijving gebruiken we '\3'. De derde expressie is de enige die overblijft. Het resultaat van de Reguliere Expressie is dat de beschrijving van de groente en alle groenten die nog volgen in de beschrijving achterblijven. En dat is nog steeds teveel van het goede. Om dit eenvoudig op te lossen gaan we deze expressie nesten in een andere Reguliere Expressie die erg veel lijkt op de bovenstaande.

```
regexp_replace(
  regexp_replace (beschrijving
, '(.*)('|currgroente|')#(.*?)#'
, '\3'
)
, '([^\#]*)#([^\#]*)#'
, '\1'
)
```

Van alle beschrijvingen die de vorige expressie opleverde willen we eigenlijk alleen maar de eerste hebben. De expressie die we nu gebruiken haalt alle karakters uit de beschrijvingen die geen scheidingstekens zijn, gevolgd door een scheidingsteken, en dan nog meer karakters die geen scheidingstekens zijn. Nu hebben we dan eindelijk de beschrijving te pakken. Het leek zo eenvoudig, maar wordt toch al gauw complex.

```
SQL> with groenten as
2 (select 'Andijvie;Biet of Kroot;Boerenkool;Knollenselderie;Lage
Slabonen;Spinazie' groente
3 , 'Andijvie#Cichorium Endivia: Juni-juli op zaaibed in de
vollegrond en na verspenen uitplanten of direct in regels zaaien en
uitdunnen#Biet of Kroot#Vanaf half april tot half juli in de vollegrond
in regels. In de regels eventueel iets uitdunnen#Boerenkool#Vanaf eind
april tot juli in de vollegrond op zaaibed en ca. 5-6 weken later uitpl
anten#Knollenselderie#Vanaf maart onder glas, verspenen en in mei buite
n planten, of vanaf half april buiten zaaien, verspenen en
uitplanten#Lage Slabonen#Vanaf half april onder glas of vanaf half mei
tot half juli buiten zaaien. De onder glas gezaaide bonen kunnen in de
bak worden gekweekt of buiten worden uitgeplant. Vroeg ras met donker-
groene, lange slanke, extra fijne bonen#Spinazie#Zodra het weer het
toelaat kan men onder glas zaaien, breedwerpig en flink dik. Vanaf
begin maart tot eind april en vanaf eind juli to eind augustus in de
vollegrond en in regels. Securo is zeer geschikt voor voorjaars- en
najaarsteelt#' beschrijving
4 from dual
5 )
6 , rijen as
```

```
7 (select 1 r
8 from all_objects
9 )
10 , t as
11 ( select regexp_substr (groente, '[^;]+', 1, rownum) currgroente
12 from groenten
13 , rijen
14 where rownum <= regexp_count (groente, '[^;]+')
15 )
16 select currgroente
17 , regexp_replace(
18 regexp_replace(beschrijving
19 , '(.*)('|currgroen-
te|')#(.*?)#'
20 , '\3'
21 )
22 , '([^\#]*)#([^\#]*)#'
23 , '\1'
24 ) beschrijving
25 from t
26 , groenten
27 ;
```

CURRGROENTE

BESCHRIJVING

Andijvie

Cichorium Endivia: Juni-juli op zaaibed in de vollegrond en na verspenen uitplanten of direct in regnen

Biet of Kroot

Vanaf half april tot half juli in de vollegrond in regels. In de regels eventueel iets uitdunnen

Boerenkool

Vanaf eind april tot juli in de vollegrond op zaaibed en ca. 5-6 weken later uitplanten

Knollenselderie

Vanaf maart onder glas, verspenen en in mei buiten planten, of vanaf half april buiten zaaien, versp

Lage Slabonen

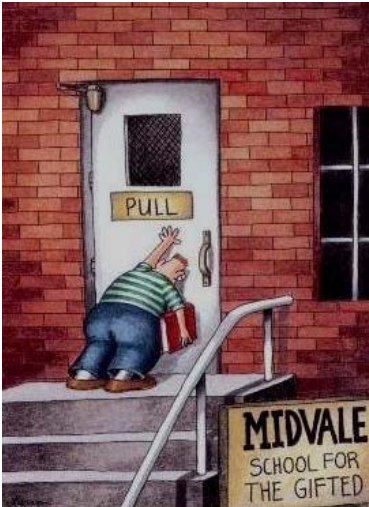
Vanaf half april onder glas of vanaf half mei tot half juli buiten zaaien. De onder glas gezaaide bonen worden gekweekt of buiten worden uitgeplant. Vroeg ras met donkergroene, lange slanke, extra fijne

Spinazie

Zodra het weer het toelaat kan men onder glas zaaien, breedwerpig en flink dik. Vanaf begin maart tot f eind juli to eind augustus in de vollegrond en in regels. Securo is zeer geschikt voor voorjaars-

6 rows selected.

Als laatste zullen we uit de verkregen beschrijvingen dan bepalen of de groente in de volle grond moeten komen te staan of dat deze onder glas dienen te worden gekweekt.



Met de Reguliere Expressie REGEXP\_INSTR kunnen we kijken of een bepaald woord in een string voorkomt. Dit kan natuurlijk ook met een 'gewone' INSTR, maar de Reguliere Expressie biedt de mogelijkheid om verschillende variaties van spelling in één expressie op te nemen. Zo kan er in de beschrijving voorkomen 'volle grond', maar ook 'vollegrond'.

Met of zonder spatie tussen 'volle' en "grond", bij beide wordt hetzelfde bedoeld. Stel dat er in een beschrijving staat: 'Deze plant draagt volle trossen. Ze dient onder glas in de grond te worden gepland'. 'Volle' en 'grond' staan erin, in de juiste volgorde, maar er staat teveel tussen. Hier wordt niet bedoeld dat er in de vollegrond moet worden gepland. Tenminste daar gaan we hier even van uit. Zoals ik al eerder schreef, ik heb geen groene vingers.

De Expressie die we gaan gebruiken is:

```
volle( |)grond
```

De betekenis van deze expressie is: eerst de karakters 'volle' gevolgd door een spatie of niets, dan 'grond'. Andere varianten worden niet toegestaan.

Om niet de gehele query hier te herhalen, maak ik gebruik van een drietal verkorte beschrijvingen.

```
SQL> with test as
  2 (select 'verdere beschrijving vollegrond verdere beschrijving' str
  from dual union all
  3 select 'verdere beschrijving volle grond verdere beschrijving'
  str from dual union all
  4 select 'verdere beschrijving volle onder glas grond verdere
  beschrijving' str from dual
  5 )
  6 select regexp_instr (str, 'volle( |)grond')
  7 from test
  8 /
```

```
REGEXP_INSTR(STR, 'VOLLE(|)GROND')
-----
                22
                22
                0
```

In bovenstaande resultaat is te zien dat de varianten 'volle grond' en 'vollegrond' wel worden gevonden, maar dat de laatste niet wordt gevonden - de REGEXP\_INSTR is nul.

Als laatste zou je met een CASE expressie kunnen bepalen of het vollegrond is, of dat er in de kas moet worden gepland.

```
SQL> with test as
  2 (select 'verdere beschrijving vollegrond verdere beschrijving' str
  from dual union all
  3 select 'verdere beschrijving volle grond verdere beschrijving'
  str from dual union all
  4 select 'verdere beschrijving volle onder glas grond verdere
  beschrijving' str from dual
  5 )
  6 select case
  7     when regexp_instr (str, 'volle( |)grond') > 0
  8     then 'Vollegrond'

  9     else 'Onder Glas'
  10 end
  11 from test
  12 /
```

```
CASEWHENRE
-----
Vollegrond
Vollegrond
Onder Glas
```

Het lijkt er soms op met Reguliere Expressies dat je andersom moet denken. In het eerste voorbeeld waar we de verschillende groenten uit de lijst splitsen, zeggen we niet 'toon alle karakters die tussen het scheidingsteken staan', maar zeggen we eigenlijk 'toon alle karakters die geen scheidingsteken zijn'. En deze manier van denken is toch even wennen. Misschien is het je al opgevallen, de titel is 'Komkommertijd' maar toch komen er geen komkommers voor in dit verhaal. Ook staat er niets over het nuttigen van de gekweekte groente, maar dat gebeurt natuurlijk buiten de Database.

Hopelijk hebben we je in dit artikel een tipje van de sluier van Reguliere Expressies kunnen oplichten en ga je een keer zelf aan de slag met deze zeer krachtige functionaliteit.

## Referenties

- [http://www.oracle.com/technology/oramag/webcolumns/2003/techarticles/rischert\\_regexp\\_pt1.html](http://www.oracle.com/technology/oramag/webcolumns/2003/techarticles/rischert_regexp_pt1.html) (en deel 2 ook)
- [tahiti.oracle.com](http://tahiti.oracle.com)



**Patrick Barel** is consultant bij AMIS Services. Hij schrijft op het blog van AMIS (<http://technology.amis.nl/blog>) en op zijn eigen blog (<http://blog.bar-solutions.com>).



**Alex Nuijten** is Oracle-consultant bij AMIS Services. Hij schrijft op het blog van AMIS (<http://technology.amis.nl/blog>) en op zijn eigen blog (<http://nuijten.blogspot.com>).



CONSPECT

ICT diensten

# Wij zoeken Oracle Consultants ...



*Wij bieden nieuwe collega's, die meer verwachten van een ICT werkgever, naast carrièreperspectief natuurlijk ook uitstekende primaire en secundaire arbeidsvoorwaarden zoals:*

*Conspect Coaching Program, seminars, spiegelsessies, Conspect Talent Support, optieplannen, materiële zaken zoals auto, telefoon, laptop etc. en regelmatig informele bijeenkomsten.*

*Voor meer informatie:  
Conspect ICT diensten  
Robin van der Beek  
Transistorstraat 71 j  
1322 CK Almere  
T (036) 538 72 92  
E [post@conspect.nl](mailto:post@conspect.nl)  
of kijk op [www.conspect.nl](http://www.conspect.nl)*

Wij zijn een consultancybedrijf met een gevestigde reputatie in de Oracle en Java wereld. Conspect kun je het beste omschrijven als een organisatie waar zowel resultaatgerichtheid, dynamiek, pret als creativiteit voorkomt.

Gezamenlijk bepalen wij in een open en directe cultuur visie en beleid. Het centraal stellen van onze goed opgeleide en gemotiveerde medewerkers is voor ons dan ook één van de succesfactoren van onze organisatie.

ORACLE CERTIFIED PARTNER

# Tijd voor verandering?



Bij Truston zoeken we altijd de balans tussen werk en privé. Simpelweg omdat wij vinden dat mensen werken om te leven en niet leven om te werken. Daarom streven we dan ook altijd lange termijnrelaties na. Bouwen we dag in dag uit aan de toekomst van onze klanten en onze mensen. Om nu en in de toekomst aan onze hoge kwaliteitsstandaard te blijven voldoen, zoeken wij voortdurend specialisten om ons team te versterken. Bezoek voor meer informatie en de openstaande vacatures [www.truston.com](http://www.truston.com) en laat je verrassen.