

Relationele gegevensbanken

Bert De Saffel

2017-2018

Inhoudsopgave

I	Theorie	2
1	Enkelvoudige tabellen	3
1.1	Eenvoudige select opdrachten	3
1.1.1	Bewerkingen op velden	4
1.1.2	De where-clausule	5
1.1.3	De order by-clausule	7
1.2	De case-expressie	8
1.2.1	Case-expressies in een select-clausule	9

Deel I

Theorie

Hoofdstuk 1

Enkelvoudige tabellen

1.1 Eenvoudige select opdrachten

- **SELECT** : De enige opdracht waarmee gegevens opgevraagd kunnen worden.
- **FROM** : specificeert de tabel waarin gezocht moet worden.
- ***** : Verkorte notatie voor alle kolommen in de tabel.
- **DISTINCT**: Elimineren van meervoudige gelijke rijen. Rij_1 is gelijk aan Rij_2 indien alle velden gelijk zijn.
- **AS**: Een kolomalias. Gebruik dit als je de naam van een kolom wilt veranderen in de output.

```
SELECT *  
FROM   Ranking
```

geeft alle kolommen van de tabel 'Ranking'

```
SELECT name, season, discipline, points  
FROM   Ranking
```

selecteert respectievelijk de 4 kolommen van de tabel 'Ranking'

```
SELECT DISTINCT discipline
FROM    Races
```

Toont alle disciplines van de tabel 'Races'. Indien er meerdere van dezelfde disciplines zijn worden deze niet getoond.

```
SELECT name AS plaats,
        nation AS land
from    Resorts
```

Toont de name als de kolom plaats, en nation als de kolom land van alle rijen in de tabel 'Resorts'

1.1.1 Bewerkingen op velden

- CAST : Expliciet omzetten van een waarde van het ene datatype naar een waarde van een ander datatype

Je kan tabellen selecteren als waarde van een wiskundige bewerking

```
SELECT    22. /      7 PIV1,
          335. /   113 PIV2,
          52163. / 16604 PIV3
```

Heeft als output:

PIV1	PIV2	PIV3
3.142857	3.141592	3.141592

Bij sommige databanken moet je casten

```
SELECT
CAST(CAST( 22 AS NUMERIC(19, 7))/ 7 AS NUMERIC(9,7)) PIV1,
CAST(CAST( 355 AS NUMERIC(19, 7))/ 113 AS NUMERIC(9,7)) PIV2,
CAST(CAST(52163 AS NUMERIC(19, 7))/16604 AS NUMERIC(9,7)) PIV3,
```

Heeft als output:

PIV1	PIV2	PIV3
3.142857	3.141592	3.141592

De 1ste parameter bij NUMERIC staat voor het aantal digits VOOR de komma. De 2de parameter staat voor het aantal digits NA de komma.

1.1.2 De where-clausule

- WHERE : Rijen selecteren die aan een specifieke voorwaarde voldoen. WHERE heeft altijd betrekking tot één of meerdere kolommen.
- LIKE/NOT LIKE: Rijen selecteren aan de hand van een bepaald patroon.
- IS NULL/IS NOT NULL: Rijen selecteren op het feit dat een bepaalde kolom **null** of **niet null** is.
- BETWEEN ... AND .../NOT BETWEEN ... AND ...: Selectie op basis van een interval inclusief de intervalgrenzen.
- AND/OR/NOT: Samenstellen van predicaten.
- IN/NOT IN: Vereenvoudiging van de *AND* en de *OR* operator.

```
SELECT name, capital
FROM   [Regio's]
WHERE  parent = 'FR'
```

Selecteert de *name* en *capital* van de tabel *Regio's* waarvan het *parent* veld gelijk is aan **FR**.

```
SELECT name, parent
FROM   [Regio's]
WHERE  parent LIKE 'BE.[OM]V'
```

Selecteert *name* en *parent* van de tabel *Regio's* waarvan het *parent* veld **BE.OV** of **BE.MV** is.

```
SELECT name, population, area
FROM   [Regio's]
WHERE  parent IS NULL
```

Selecteert de *name*, *population* en *area* veld van de tabel *Regio's* waarvan het *parent* veld **NULL** is.

```
SELECT name, longitude, latitude
FROM   [Regio's]
WHERE  latitude BETWEEN 51.05 AND 51.06
```

Selecteert de *name*, *longitude* en *latitude* veld van de tabel *Regio's* waarvan het *latitude* veld tussen **51.05** en **51.06** ligt.

```
SELECT name, capital, parent
FROM   [Regio's]
WHERE  parent = 'BE.VL'
       OR parent = 'BE.WA'
       OR ( parent = 'BE'
           AND capital IS NOT NULL)
```

Selecteert het *name*, *capital* en *parent* veld van de tabel *Regio's* waarvan het *parent* veld gelijk is aan **BE.VL** OF aan **BE.WA** OF aan **BE**. Indien het gelijk is aan **BE** worden enkel de records geselecteerd waarvan het *capital* veld niet gelijk is aan **NULL**.

```
SELECT name, capital
FROM   [Regio's]
WHERE  hasc IN ( 'NL', 'LU', 'FR', 'DE', 'GB' )
```

Selecteert het *name* en *capital* veld van de tabel *Regio's* waarvan het *hasc* veld één van de volgende waarden heeft: **NL**, **LU**, **FR**, **DE** of **GB**.

```
SELECT name, nation
from   Resorts
WHERE  nation NOT IN ( 'ARG', 'USA', 'CAN', 'AUS',
                      'NZE', 'JPN', 'KOR' )
```

Selecteert het *name* en *nation* veld van de tabel *Resorts* waarvan het *nation* veld niet gelijk is aan één van de volgende waarden: **ARG**, **USA**, **CAN**, **AUS**, **NZE**, **JPN** of **KOR**.

```
SELECT name, capital, geo_distance(latitude, longitude, 51.03, 3.703)
FROM   [Regio's]
WHERE  parent = 'EUR'
AND    geo_distance(latitude, longitude, 51.03, 3.703) < 600
```

1.1.3 De order by-clausule

- ORDER BY : Sorteren op een bepaalde kolom.
- ASC/DESC : In oplopende of aflopende zin sorteren.

Vanaf hier wordt de SELECT en FROM niet meer uitgelegd.

```
SELECT    season, discipline, name, points
FROM      Ranking
WHERE     gender = 'L'
ORDER BY  season DESC, discipline, points DESC
```

Het *gender* veld met gelijk zijn aan **L**. Eerst wordt het *season* veld in aflopende zin gesorteerd, dan het *discipline* veld in oplopende zin en dan het *points* veld in aflopende zin.

```
SELECT name, capital,
        CAST(population / area AS NUMERIC(9, 0)) dichtheid
FROM    [Regio's]
WHERE   parent = 'EUR'
        AND area > 0
ORDER BY dichtheid DESC
```

In de select clausule wordt de waarde van het *population* veld gedeeld door de waarde van het *area* veld. Deze bewerking komt terecht in de kolomalias *dichtheid*. Het *parent* veld moet gelijk zijn aan **EUR** en het *area* veld moet groter zijn dan **0**. Het resultaat wordt gesorteerd op de kolomalias *dichtheid* in aflopende zin.

```
SELECT    hasc, name
FROM      [Regio's]
WHERE     level = 4
ORDER BY  SUBSTRING(hasc, LEN(hasc) - 4, 5)
```

Het *level* veld moet gelijk zijn aan 4. Het resultaat wordt gesorteerd op een de laatste 5 getallen of letters uit de *hasc* kolom in oplopende zin.

1.2 De case-expressie

- CASE : Het begin van een case-expressie.
- END : Het einde van een case-expressie.
- WHEN ... THEN ... : Afhankelijk van een bepaalde waarde voer je een bepaalde actie uit.
- ELSE : Optionele bewerking indien geen enkele van de WHEN ... THEN ... clauses waar is.

```
SELECT date, resort,  
       CASE discipline  
         WHEN 'DH' THEN 'Afdaling'  
         WHEN 'SG' THEN 'Super-G'  
         WHEN 'GS' THEN 'Reuzenslalom'  
         WHEN 'SL' THEN 'Slalom'  
         WHEN 'KB' THEN 'Combinatie'  
       END  
FROM Races  
ORDER BY rid
```

Het *discipline* veld wordt vergeleken met de 5 waardes uit de case expressie. Wanneer discipline gelijk is aan **DH**, zal de kolom 'Afdaling' bevatten, analoog voor de andere gevallen. Het resultaat wordt gesorteerd in oplopende zin op het veld *rid*.

```
SELECT name, latitude,  
       CASE  
         WHEN latitude BETWEEN -18 AND 18 THEN 'tropisch'  
         WHEN latitude BETWEEN -36 AND 36 THEN 'subtropisch'  
         WHEN latitude BETWEEN -54 AND 54 THEN 'gematigd'  
         WHEN latitude BETWEEN -72 AND 72 THEN 'subpolair'  
         ELSE 'polair'  
       END klimaatzone  
FROM   [Regio's]  
WHERE  parent IN ('NAM', 'SAM')
```

```

        AND latitude IS NOT NULL
ORDER BY latitude

```

De case-expressie vergelijkt elke keer het *latitude* veld en afhankelijk van de conditie van dit veld wordt er een bepaalde waarde ingevuld in de kolomalias *klimaatzone*. Het *parent* veld moet ofwel **NAM** of **SAM** zijn en het *latitude* veld mag niet **NULL** zijn. Het resultaat wordt gesorteerd op de waarde van het *latitude* veld in oplopende zin.

1.2.1 Case-expressies in een select-clausule

```

SELECT name,
       CASE WHEN ABS(latitude) < 18 THEN 'X'
            ELSE '' END tropisch,
       CASE WHEN ABS(latitude) BETWEEN 18 AND 36 THEN 'X'
            ELSE '' END subtropisch,
       CASE WHEN ABS(latitude) BETWEEN 36 AND 54 THEN 'X'
            ELSE '' END gematigd,
       CASE WHEN ABS(latitude) BETWEEN 54 AND 72 THEN 'X'
            ELSE '' END subpolair,
       CASE WHEN ABS(latitude) > 72 THEN 'X'
            ELSE '' END polair
FROM [Regio's]
WHERE parent = 'EUR'
      AND latitude IS NOT NULL
ORDER BY name

```

Er worden 5 case-expressies aangemaakt dat de absolute waarde van het *latitude* veld bekijkt. Bij elke case wordt er een kolomalias gemaakt. Deze methode zorgt ervoor dat je een typische resultaat tabel in de volgende vorm krijgt:

name	tropisch	subtropisch	gematigd	subpolair	polair
Cyprus		X			
Zweden			X		
Zwitserland				X	