

# Subqueries 1

Niet-Gecorreleerde Subqueries

[wim.bertels@ucll.be](mailto:wim.bertels@ucll.be)

Naamsvermelding-NietCommercieel-GelijkDelen 4.0  
Unported Licentie

Hoe zou je de output van een SELECT statement beschrijven?

# Wat is een subquery?

- Een tabellexpressie binnen een tabellexpressie
- Resultaat wordt doorgegeven aan aanroepende tabellexpressie
- Subqueries mogen genest zijn

# Waarom gebruiken we subquery's?

- Query opsplitsen in deelproblemen die je kan oplossen en de output ervan verder gebruiken
- Zoals bij programmeren: een complexere methode opsplitsen in eenvoudigere (atomaire) taken

# Soorten subquery's

- Scalaire subquery: output = 1 rij, 1 kolom (dus 1 waarde)
- Rij-subquery: output = 1 rij
- Kolom-subquery: output = meerdere rijen met elk 1 waarde
- Tabel-subquery: output = meerdere rijen en kolommen

# Scalaire subquery (1 rij, 1 kolom)

Voorbeeld:

Geef voor elke planeet hoeveel groter of kleiner deze is dan de zon.

# Scalaire subquery (1 rij, 1 kolom)

Voorbeeld:

Geef voor elke planeet hoeveel groter of kleiner deze is dan de zon.

```
1 SELECT objectnaam, diameter - (SELECT diameter FROM hemelobjecten WHERE objectnaam = 'Zon') AS verschil
2 FROM hemelobjecten
3 WHERE satellietvan = 'Zon';
4
```

Data Output Explain Messages Notifications Query History

|   | diameter<br>numeric (7) |
|---|-------------------------|
| 1 | 1393000                 |

Output van subquery

# Scalaire subquery (1 rij, 1 kolom)

SELECT objectnaam, diameter –

(SELECT diameter  
FROM hemelobjecten  
WHERE objectnaam = 'Zon')

AS verschil

FROM hemelobjecten

WHERE satellietvan = 'Zon';



# Scalaire subquery (1 rij, 1 kolom)

Voorbeeld 2:

Geef de hemellichamen met een diameter groter dan Venus.

# Scalaire subquery (1 rij, 1 kolom)

Voorbeeld 2:

Geef de hemellichamen met een diameter groter dan Venus.

```
1 SELECT objectnaam, diameter
2 FROM hemelobjecten
3 WHERE diameter >
4 (SELECT diameter
5 FROM hemelobjecten
6 WHERE objectnaam = 'Venus');
7
```

| Data Output | Explain                 | Messages | Notifications | Q |
|-------------|-------------------------|----------|---------------|---|
|             | diameter<br>numeric (7) |          |               |   |
| 1           | 12104                   |          |               |   |

Output van subquery

# Scalaire subquery (1 rij, 1 kolom)

```
SELECT  objectnaam, diameter  
FROM    hemelobjecten  
WHERE   diameter >
```

```
(SELECT diameter  
FROM    hemelobjecten  
WHERE   objectnaam = 'Venus');
```

# Rij-subquery (1 rij)

Voorbeeld:

Geef alle spelers met hetzelfde geslacht en dezelfde woonplaats als de speler met nummer 7.

# Rij-subquery (1 rij)

Voorbeeld:

Geef alle spelers met hetzelfde geslacht en dezelfde woonplaats als de speler met nummer 7.

```
1 SELECT spelersnr, naam, plaats
2 FROM spelers
3 WHERE (plaats, geslacht) =
4 (SELECT plaats, geslacht
5 FROM spelers
6 WHERE spelersnr = 7);
7
```

Output van subquery

|   | plaats<br>character varying (30) | geslacht<br>character (1) |
|---|----------------------------------|---------------------------|
| 1 | Den Haag                         | M                         |

# Rij-subquery (1 rij)

```
SELECT  spelersnr, naam, plaats  
FROM    spelers  
WHERE   (plaats, geslacht) =  
        (SELECT  plaats, geslacht  
         FROM    spelers  
         WHERE   spelersnr = 7);
```

# Kolom-subquery (meerdere rijen, elk 1 waarde)

Voorbeeld:  
Geef alle manen.

Tijd voor een micropauze

# Kolom-subquery (meerdere rijen, elk 1 waarde)

Voorbeeld:  
Geef alle manen.

```
1 SELECT objectnaam
2 FROM hemelobjecten
3 WHERE satellietvan IN
4 (SELECT objectnaam
5  FROM hemelobjecten
6  WHERE satellietvan = 'Zon');
```

|   | objectnaam<br>character varying (10) |
|---|--------------------------------------|
| 1 | Mercurius                            |
| 2 | Venus                                |
| 3 | Aarde                                |
| 4 | Mars                                 |
| 5 | Jupiter                              |
| 6 | Saturnus                             |
| 7 | Uranus                               |
| 8 | Neptunus                             |
| 9 | Pluto                                |

Output van subquery



# Kolom-subquery (meerdere rijen, elk 1 waarde)

```
SELECT  objectnaam  
FROM    hemelobjecten  
WHERE   satellietvan IN  
        (SELECT  objectnaam  
         FROM    hemelobjecten  
         WHERE   satellietvan = 'Zon');
```

# Tabel subquery (meerdere rijen en kolommen)

- Geeft een tijdelijk resultaat
- Subquery moet een pseudoniem krijgen als de subquery in de from staat van de originele query

# Tabel subquery (meerdere rijen en kolommen)

Voorbeeld:

Geef de reizen die een hemelobject bezoeken dat over alle reizen heen minstens 5 keer bezocht wordt.

# Tabel subquery (meerdere rijen en kolommen)

Voorbeeld:

Geef de reizen die een hemelobject bezoeken dat over alle reizen heen minstens 5 keer bezocht wordt.

Tussenstap:

```
SELECT objectnaam
```

```
FROM
```

```
bezoeken
```

```
GROUP BY objectnaam
```

```
HAVING COUNT(*) >= 5;
```

# Tabel subquery (meerdere rijen en kolommen)

Voorbeeld:

Geef de reizen die een hemelobject bezoeken dat over alle reizen heen minstens 5 keer bezocht wordt.

```
1 SELECT reizen.reisnr, reizen.vertrekdatum
2 FROM reizen
3 INNER JOIN bezoeken b using (reisnr)
4 INNER JOIN (
5     SELECT objectnaam
6     FROM bezoeken
7     GROUP BY objectnaam
8     HAVING COUNT(*) >= 5
9 ) AS veelbez ON b.objectnaam = veelbez.objectnaam
10 GROUP BY reizen.reisnr, reizen.vertrekdatum;
```

| Data Output                          | Explain | Messages | Notifications | Query History |
|--------------------------------------|---------|----------|---------------|---------------|
| objectnaam<br>character varying (10) |         |          |               |               |
| 1 Maan                               |         |          |               |               |

Output van  
subquery

# Tabel subquery (meerdere rijen en kolommen)

```
SELECT  reizen.reisnr, reizen.vertrekdatum
FROM    reizen INNER JOIN bezoeken b USING (reisnr)
        INNER JOIN
        (SELECT  objectnaam
         FROM    bezoeken
         GROUP BY objectnaam
         HAVING   COUNT(*) >= 5)
        AS veelbez
        ON b.objectnaam = veelbez.objectnaam
GROUP BY reizen.reisnr, reizen.vertrekdatum;
```

# Nog iets klein

Geef alle reizen die geen bezoek hebben gebracht aan de maan.

# Nog iets klein

Geef alle reizen die geen bezoek hebben gebracht aan de maan.

```
SELECT  reizen.reisnr  
FROM    reizen INNER JOIN bezoeken USING (reisnr)  
WHERE   bezoeken.objectnaam <> 'Maan'  
GROUP BY reizen.reisnr;
```

?



# Nog iets klein

Geef alle reizen die geen bezoek hebben gebracht aan de maan.

```
SELECT reizen.reisnr  
FROM reizen INNER JOIN bezoeken USING (reisnr)  
WHERE bezoeken.objectnaam <> 'Maan'  
GROUP BY reizen.reisnr;
```

Waarom fout?

Check:  
Alle reizen

```
SELECT reisnr  
FROM reizen;
```

# Oplossing (tussenstap)

Alle reizen die WEL de maan hebben bezocht:

```
SELECT  reizen.reisnr  
FROM    reizen INNER JOIN bezoeken USING (reisnr)  
WHERE   bezoeken.objectnaam = 'Maan'  
GROUP BY reizen.reisnr;
```

# Oplossing

Alle reizen die GEEN bezoek hebben gebracht aan de maan.

```
SELECT  reisnr  
FROM    reizen  
WHERE   reisnr NOT IN  
        (SELECT  reisnr  
         FROM    bezoeken  
         WHERE   objectnaam = 'Maan');
```

# Uitdaging

Probeer deze zonder subquery te schrijven (op een regenachtige dag.., als je echt teveel tijd hebt..):

```
SELECT  avg(totaal)
FROM
    (SELECT  spelersnr, sum(bedrag) as total
    FROM    boetes
    GROUP BY spelersnr) as totalen
```

Wat toont deze query?

Wim Bertels (CC)BY-SA-NC

Referenties:

Slides subqueries deel 1 sql 2012-13, K. Beheydt

SQL Leerboek, R. Van der lans