

Subqueries 1

Niet-Gecorreleerde Subqueries

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 tabelexpressie binnen een tabelexpressie
- Resultaat wordt doorgegeven aan aanroepende tabelexpressie
- 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:

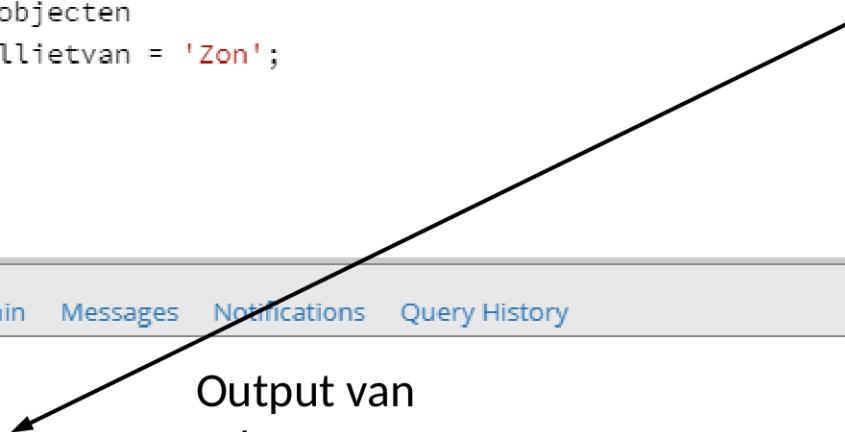
Geef voor elke planeet hoeveel groter of kleiner deze is dan de zon.

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	
	diameter 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 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
Explain subquery

	plaats	geslacht
▼	character varying (30)	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');
```

Data Output		Explain	Messages	Notifications
	objectnaam 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				
<table border="1"><thead><tr><th></th><th>objectnaam</th></tr></thead><tbody><tr><td>1</td><td>Maan</td></tr></tbody></table>			objectnaam	1	Maan	Output van subquery			
	objectnaam								
1	Maan								

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

SUBQUERIES 2

GECORRELEERDE SUBQUERIES

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0
Unported Licentie

Soorten subqueries

■ Scalaire subquery:

- *1 rij , 1 kolom => 1 waarde*

■ Rij-subquery:

- *1 rij, meerdere kolommen*

■ Kolom-subquery:

- *Meerdere rijën, 1 kolom*

■ Tabel-subquery:

- *Meerdere rijën, meerdere kolommen*

Subquery in WHERE

Scalaire subquery

- =
- >
- <
- ...

Kolom-subquery

- IN()
- >= ALL()
- ...

Subquery in FROM

- Geef voor alle hemelobjecten die minstens 5 keer bezocht zijn alle reizen die dat hemelobject bezocht hebben. Toon reisnr en objectnaam.

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

Hoofdqueries en subqueries

- De hoofdquery krijgt enkel de output van de subquery en weet niets over details zoals: gebruikte tabellen, berekeningen, ...
- Alleen de SELECT wordt dus doorgegeven aan de hoofdquery
- De subquery weet alles over de hoofdquery tot in detail en kan alle gegevens van de hoofdquery gebruiken

Gecorreleerde subqueries

Subquery waarin een kolom wordt
gebruikt die tot een tabel behoort uit een
ander select-blok.

Dus een gecorreleerde subquery kan niet
autonomoos uitgevoerd worden.

Oefening 1

■ Geef voor iedere reis
het bezoek met de
langste verblijfsduur

Oplossing 1

- SELECT b.reisnr, b.objectnaam
FROM bezoeken AS **b**
WHERE b.verblijfsduur =
 (SELECT MAX(verblijfsduur)
 FROM bezoeken AS **allebezoeken**
 WHERE **allebezoeken.reisnr = b.reisnr**);
- Geef voor iedere reis het bezoek met de langste verblijfsduur

Oefening 2

- Geef de spelers die meer keer bestuurslid zijn geweest dan dat ze wedstrijden hebben gespeeld. Toon spelersnr.

Oplossing 2

■ SELECT spelersnr

FROM bestuursleden AS b

GROUP BY spelersnr

HAVING COUNT(*) >

(SELECT COUNT(*)

FROM wedstijden AS w

WHERE w.spelersnr = b.spelersnr):

■ Geef de spelers die meer keer bestuurslid zijn geweest dan dat ze wedstrijden hebben gespeeld.
Toon spelersnr.

EXISTS operator (is er iet, of nie?)

- Kijk of er output BESTAAT voor een subquery
- TRUE of FALSE
- Wat er in de SELECT staat maakt niet uit:
 - *Iets* = TRUE
 - *Niets / empty* = FALSE

EXISTS operator oefening

- Geef alle reizen met een bezoek aan Jupiter. Of:
- = Geef alle reizen waarbij er een bezoek aan Jupiter BESTAAT.
- Toon reisnr en vertrekdatum.

EXISTS operator oplossing

■ SELECT reisnr, vertrekdatum

FROM **reizen**

WHERE EXISTS

(SELECT * , 'is erietofnie'

FROM **bezoeken AS b**

WHERE b.objectnaam = 'Jupiter'

AND **b.reisnr = reizen.reisnr**):

EXISTS operator oplossing

■ SELECT reisnr, vertrekdatum

FROM **reizen**

WHERE EXISTS

```
(SELECT    reisnr  
  FROM      bezoeken AS b  
 WHERE     b.objectnaam = 'Jupiter'  
 AND      b.reisnr = reizen.reisnr);
```

NOT EXISTS operator (als er niks is .. dan..)

- Tegenovergestelde van EXISTS
- Geef alle hemelobjecten doe nog nooit bezocht zijn:
- SELECT h.objectnaam

FROM hemelobjecten AS h

WHERE NOT EXISTS

(SELECT reisnr

FROM bezoeken AS b

WHERE b.objectnaam = h.objectnaam);

En Deze?:

- SELECT objectnaam
FROM hemelobjecten
WHERE NOT EXISTS (
 SELECT reisnr
 FROM bezoeken b
);
- --..

ANY en ALL operatoren

■ Deze operatoren verwachten een rij uitdrukking, om te vergelijken met 1 of meerdere waarden (ALL is de 'voor alle' en ANY is de 'er bestaat' uit de wiskunde)

- $> ALL$ $> ANY$
- $\geq ALL$ $\geq ANY$
- $< ...$
- $\leq ...$

ANY en ALL operator oefening

- Geef de langste reis. Of:
- = Geef de reis waarbij de reisduur groter of gelijk is aan alle reizen.
- Toon reisnr

ANY en ALL operator oplossing

■ Geef de langste reis:

■ SELECT reisnr

FROM reizen

WHERE reisduur >= ALL

(SELECT reisduur

FROM reizen);

ANY en ALL operator, andere oplossing?

■ Geef de langste reis

■ SELECT reisnr

FROM **reizen**

WHERE reisduur > ALL

(SELECT reisduur

FROM **reizen AS anderen**

WHERE **anderen.reisnr <> reizen.reisnr**);

■ -- WAT ALS 2 REIZEN DEZELFDE REISDUUR HEBBEN ???

ANY en ALL operator oefening 2

- Geef alle reizen, behalve de langste reis. Of:
- = Er is minstens 1 reis met een langere reisduur
- Toon reisnr

ANY en ALL operator oplossing 2

■ Geef alle reizen, behalve de langste reis:

■ SELECT reisnr

FROM reizen

WHERE reisduur < ANY

(SELECT reisduur

FROM reizen);

ANY en ALL operator: OPGELET!

- Geef een lijst van alle planeten die groter zijn dan al hun satellieten

- SELECT objectnaam

```
FROM    hemelobjecten AS h
WHERE   satellietvan = 'Zon'
AND     diameter > ALL
       (SELECT diameter
        FROM    hemelobjecten AS maan
        WHERE   maan.satellietvan = h.objectnaam);
```

- -- Klopt dit?

objectnaam
Mercurius
Venus
Aarde
Mars
Jupiter
Saturnus
Uranus
Neptunus
Pluto
(9 rows)

ANY en ALL operator: OPGELET!

- Geef een lijst van alle planeten die kleiner zijn dan al hun satellieten

```
■ SELECT objectnaam  
      FROM hemelobjecten AS h  
     WHERE satellietvan = 'Zon'  
    AND   diameter < ALL  
          (SELECT diameter  
              FROM hemelobjecten AS maan  
             WHERE maan.satellietvan = h.objectnaam);
```

- -- Daarnet groter, nu kleiner, .. ?

objectnaam

Mercurius
Venus
(2 rows)

ANY en ALL operator: OPGELET!

```
■ SELECT diameter  
      FROM hemelobjecten AS manen  
     WHERE manen.satellietvan IN ('Mercurius', 'Venus');
```

diameter

(0 rows)

(ANY en ALL operator) vs NULL

- Vergelijken met NULL: vaak opnieuw NULL (onbekend) tenzij het niet uitmaakt wat deze null waarde ook zou zijn, of het duidelijk is, of ..
- NULL is een geval apart!
- Wanneer de subquery geen output (NULL) heeft dan:
 - *Geeft ALL: waar / TRUE;*
 - *Geeft ANY: onwaar / FALSE;*

UNIQUE operator

■ Geef de spelers voor wie precies één boete betaald werd.

■ SELECT spelersnr

FROM boetes AS BT

WHERE UNIQUE

(SELECT B.spelersnr

FROM boetes AS B

WHERE **B.spelersnr = BT.spelersnr**);

■ -- Niet geïmplementeerd in PostgreSQL, kan vervangen worden door
HAVING

OVERLAPS operator

■ Geef de spelers en hun functie die in het bestuur zaten van 1 januari 1991 tot en met 31 december 1993

■ SELECT spelersnr, functie

FROM bestuursleden

WHERE (begin_datum, eind_datum)

OVERLAPS ('1991-01-01', '1993-12-31');

Combinatie oefening 1

- Geef de klanten die op een reis zijn meegegaan waar ook klant met klantnr 126 op meegegaan is.
- Toon klantnr

Combinatie oefening 1: oplossing

■ Geef de klanten die op een reis zijn meegegaan waar ook klant met klantnr 126 op meegegaan is.

■ SELECT d.klantnr

FROM klanten AS k INNER JOIN deelnames AS **d** USING(klantnr)

WHERE EXISTS

(SELECT *

FROM deelnames AS **andereDeelnames**

WHERE klantnr = 126

AND **andereDeelnames.reisnr = d.reisnr**)

GROUP BY d.klantnr;

Combinatie oefening 2

- Geef de planeten die bezocht zijn op een reis waar klantrn 126 niet op meeging.
- Toon alle gegevens van de hemelobjecten

Extra oefening 2: oplossing

■ Geef de planeten uit ons zonnestelsel die bezocht zijn op een reis waar klantnr 126 niet op meeging.

■ SELECT *

```
FROM hemelobjecten AS h
```

```
WHERE EXISTS
```

```
(SELECT *
```

```
FROM bezoeken AS b
```

```
WHERE NOT EXISTS
```

```
(SELECT *
```

```
FROM deelnames AS d
```

```
WHERE klantnr = 126 AND d.reisnr = b.reisnr)
```

```
AND h.objectnaam = b.objectnaam)
```

```
AND satellietvan = 'Zon';
```

Uitdaging: Wat doet deze query ?

```
SELECT spelersnr
FROM spelers AS s
WHERE NOT EXISTS
  (SELECT *
   FROM wedstrijden AS w1
   WHERE spelersnr = 57
   AND NOT EXISTS
     (SELECT *
      FROM wedstrijden AS w2
      WHERE w1.teamnr = w2.teamnr
      AND s.spelersnr = w2.spelersnr))
AND spelersnr NOT IN
  (SELECT spelersnr
   FROM wedstrijden
   WHERE teamnr IN
     (SELECT teamnr
      FROM teams
      WHERE teamnr NOT IN
        (SELECT teamnr
         FROM wedstrijden
         WHERE spelersnr = 57))));
```

Wim Bertels (CC)BY-SA-NC

Referenties:

- Slides subqueries deel 1 sql 2012-13, K. Beheydt
- Slides Databanken, H. Martens
- SQL Leerboek, R. Van der Lans