

Joins

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0 Unported
Licentie

FROM

- Bevat tabelspecificaties
- Kunnen subqueries zijn (alias niet vergeten)
- Impliciete joins > expliciete joins

Impliciete JOIN =

```
SELECT S.spelersnr  
FROM Spelers as S, anderschema.Woonplaatsen as Stad  
WHERE S.plaats = Stad.plaatsnaam
```

Tabellen uit andere schema's moeten gekwalificeerd worden

Of we doen dit via search_path

search_path

```
SHOW search_path;
```

```
SET      search_path = public, myschema;
```

- Opgepast voor tabellen met dezelfde naam!
- Oplossing?

Expliciete INNER JOIN

```
SELECT    s.spelersnr  
FROM      spelers AS s INNER JOIN anderschema_woonplaatsen AS stad  
WHERE     s.plaats = stad.plaatsnaam;
```

```
SELECT    s.spelersnr  
FROM      spelers as s INNER JOIN anderschema_woonplaatsen AS stad  
          ON (s.plaats = stad.plaatsnaam)
```

- **INNER JOIN ... ON (.. operator ..)**
- Doorsnede, wat we gewoon zijn

Expliciete FULL OUTER JOIN

```
SELECT spelers.spelersnr, naam, bedrag  
FROM spelers FULL OUTER JOIN boetes  
      USING (spelersnr);
```

	spelersnr integer	naam character (15)	bedrag numeric (7,2)
1	6	Permentier	100.00
2	44	Bakker, de	75.00
3	27	Cools	100.00
4	104	Moerman	50.00
5	44	Bakker, de	25.00
6	8	Niewenburg	25.00
7	44	Bakker, de	30.00
8	27	Cools	75.00
9	39	Bischoff	[null]
10	57	Bohemen, van	[null]
11	2	Elfring	[null]
12	83	Hofland	[null]
13	112	Baalen, van	[null]
14	100	Permentier	[null]
15	28	Cools	[null]
16	95	Meuleman	[null]
17	7	Wijers	[null]

Expliecate FULL OUTER JOIN

```
SELECT spelers.spelersnr, naam, bedrag  
FROM spelers FULL OUTER JOIN boetes  
      USING (spelersnr);
```

- USING veronderstelt gelijke kolomnamen
- Alle rijen uit beide tabellen worden weerhouden (getoond)
- Verschil met ON: er blijft maar 1 rij met spelersnr over
- ON: 2 rijen met spelersnr (uit beide tabellen)

- Alle spelers met hun eventuele boetes?
- Met de naam van de spelers in HOOFDLETTERS

Expliecate LEFT OUTER JOIN

```
SELECT spelers.spelersnr, naam, bedrag  
FROM spelers LEFT OUTER JOIN boetes  
USING (spelersnr)
```

- Alle rijen uit de LINKSE tabel spelers worden weerhouden (getoond)
- HOOFDLETTERS?

	spelersnr integer	naam character (15)	bedrag numeric (7,2)
1	6	Permentier	100.00
2	44	Bakker, de	75.00
3	27	Cools	100.00
4	104	Moerman	50.00
5	44	Bakker, de	25.00
6	8	Nieuwenburg	25.00
7	44	Bakker, de	30.00
8	27	Cools	75.00
9	39	Bischoff	[null]
10	57	Bohemens, van	[null]
11	2	Elfring	[null]
12	83	Hofland	[null]
13	112	Baalen, van	[null]
14	100	Permentier	[null]
15	28	Cools	[null]
16	95	Meuleman	[null]
17	7	Wijers	[null]

Documentatie

- www.postgresql.org
- Daarna pas via een alternatieve bronnen

```
1
```

```
SELECT version();
```

Explicitie LEFT OUTER JOIN

```
SELECT spelers.spelersnr, UPPER(naam), bedrag  
FROM spelers LEFT OUTER JOIN boetes  
USING (spelersnr)
```

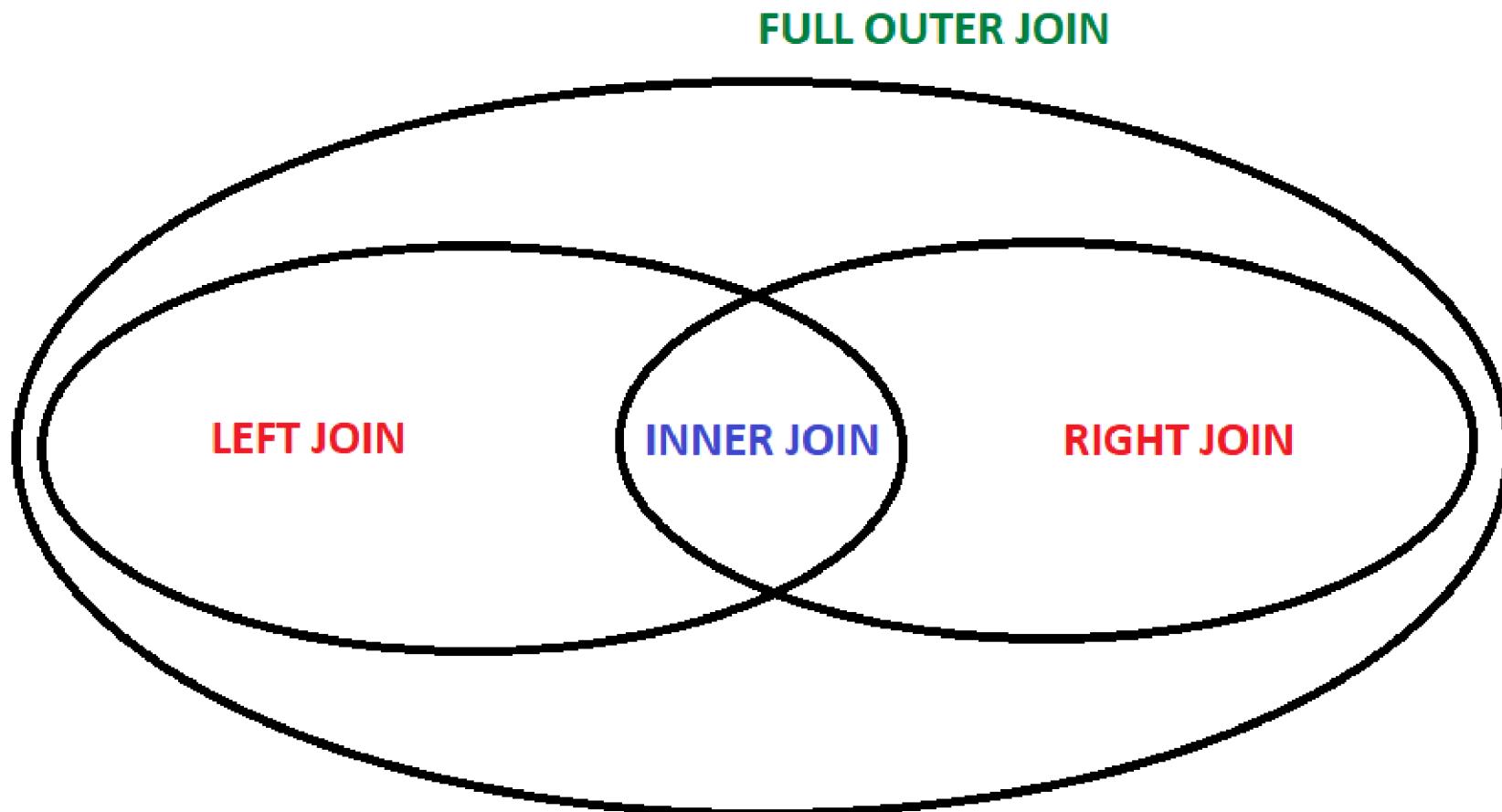
	spelers integer	upper text	bedrag numeric (7,2)
1	6	PERMENTIER	100.00
2	44	BAKKER, DE	75.00
3	27	COOLS	100.00
4	104	MOERMAN	50.00
5	44	BAKKER, DE	25.00
6	8	NIEWENBURG	25.00
7	44	BAKKER, DE	30.00
8	27	COOLS	75.00
9	39	BISCHOFF	[null]
10	57	BOHEMEN, VAN	[null]
11	2	ELFRING	[null]
12	83	HOFLAND	[null]
13	112	BAALEN, VAN	[null]
14	100	PERMENTIER	[null]
15	28	COOLS	[null]
16	95	MEULEMAN	[null]
17	7	WIJERS	[null]

- <http://www.postgresql.org/docs/current/interactive/functions-string.html>
- Andere JOINS?

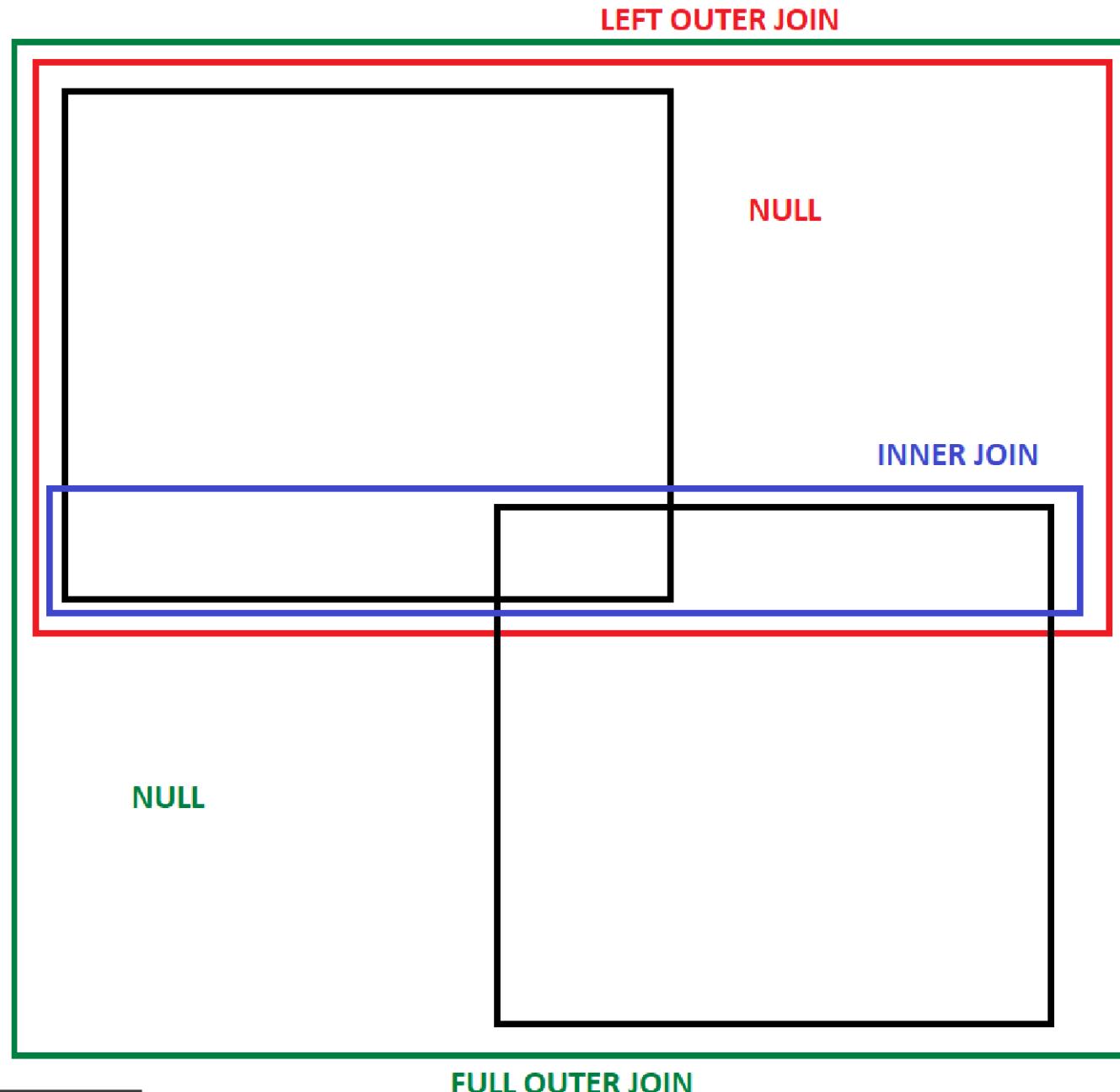
OUTER JOINS

- LEFT OUTER JOIN:
 - Alle rijen uit linker tabel met eventuele bijhorende gegevens uit rechter tabel, anders NULL waardes
- RIGHT OUTER JOIN:
 - Alle rijen uit rechter tabel met eventueel bijhorende gegevens uit linker tabel, anders NULL waardes
- FULL OUTER JOIN:
 - Alle rijen uit linker tabel en alle rijen uit rechter tabel met eventueel bijhorende gegevens uit andere tabel, anders NULL waardes

Verzamelingen



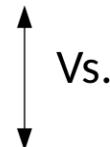
Tabellen



Condities FROM vs. WHERE

- Extra voorwaarden mogen
- Let op: verschil met de WHERE !

```
SELECT teams.spelersnr, teams.teamnr, betalingsnr  
FROM teams LEFT OUTER JOIN boetes  
    ON (teams.spelersnr = boetes.spelersnr)  
WHERE divisie = 'tweede'
```



Vs.

```
SELECT teams.spelersnr, teams.teamnr, betalingsnr  
FROM teams LEFT OUTER JOIN boetes  
    ON (teams.spelersnr = boetes.spelersnr) AND divisie = 'tweede'
```

Query vraag

Geef alle spelers die nog nooit een boete hebben gehad van 50 euro.

Oplossing 1

```
SELECT  *
FROM    spelers
WHERE   spelersnr NOT IN
        (SELECT spelersnr
         FROM   boetes
         WHERE  bedrag = 50);
```

- Niet gecorreleerde oplossing

Oplossing 1 (bis)

	spelersnr integer	naam character (15)	voorletters character (3)	geb_datum date	geslacht character (1)	jaartoe smallint	straat character varying (30)	huisnr character (4)	postcode character (6)	plaats character varying (30)	telefoon character (13)	bondsnr character (4)
1	2	Elfring	R	1948-09-01	M	1975	Steden	43	3575NH	Den Haag	070-237893	2411
2	6	Permentier	R	1964-06-25	M	1977	Hazensteinln	80	1234KK	Den Haag	070-476537	8467
3	7	Wijers	GWS	1963-05-11	M	1981	Erasmusweg	39	9758VB	Den Haag	070-347689	[null]
4	8	Niewenburg	B	1962-07-08	V	1980	Spoorlaan	4	6584WO	Rijswijk	070-458458	2983
5	27	Cools	DD	1964-12-28	V	1983	Liespad	804	8457DK	Zoetermeer	079-234857	2513
6	28	Cools	C	1963-06-22	V	1983	Oudegracht	10	1294QK	Leiden	010-659599	[null]
7	39	Bischoff	D	1956-10-29	M	1980	Ericaplein	78	9629CD	Den Haag	070-393435	[null]
8	44	Bakker, de	E	1963-01-09	M	1980	Lawaalstraat	23	4444LJ	Rijswijk	070-368753	1124
9	57	Bohemen, van	M	1971-08-17	M	1985	Erasmusweg	16	4377CB	Den Haag	070-473458	6409
10	83	Hofland	PK	1956-11-11	M	1982	Mariakade	16a	1812UP	Den Haag	070-353548	1608
11	95	Meuleman	P	1963-05-14	M	1972	Hoofdweg	33a	5746OP	Voorburg	070-867564	[null]
12	100	Permentier	P	1963-02-28	M	1979	Hazensteinln	80	6494SG	Den Haag	070-494593	6524
13	112	Baalen, van	IP	1963-10-01	V	1984	Vosseweg	8	6392LK	Rotterdam	010-548745	1319

Oplossing 2

```
SELECT  *
FROM    spelers s LEFT OUTER JOIN boetes b
        ON (s.spelersnr = b.spelersnr AND b.bedrag = 50)
WHERE   b.spelersnr IS NULL
```

- Oplossing met JOIN

Oplossing 2 (bis)

	spelersnr integer	naam character (15)	voorletters character (3)	geb_datum date	geslacht character (1)	jaartoe smallint	straat character varying (30)	huisnr character (4)	postcode character (6)	plaats character varying (30)	telefoon character (13)	bondsnr character (4)	betelingsnr integer	spelersnr integer	datum date	bedrag numeric (7,2)
1	2	Elfring	R	1948-09-01	M	1975	Steden	43	3575NH	Den Haag	070-237893	2411	[null]	[null]	[null]	[null]
2	6	Permentier	R	1964-06-25	M	1977	Hazensteinln	80	1234KK	Den Haag	070-476537	8467	[null]	[null]	[null]	[null]
3	7	Wijers	GWS	1963-05-11	M	1981	Erasmusweg	39	9758VB	Den Haag	070-347689	[null]	[null]	[null]	[null]	[null]
4	8	Nieuwenburg	B	1962-07-08	V	1980	Spoorlaan	4	6584WO	Rijswijk	070-458458	2983	[null]	[null]	[null]	[null]
5	27	Cools	DD	1964-12-28	V	1983	Liespad	804	8457DK	Zoetermeer	079-234857	2513	[null]	[null]	[null]	[null]
6	28	Cools	C	1963-06-22	V	1983	Oudegracht	10	1294QK	Leiden	010-659599	[null]	[null]	[null]	[null]	[null]
7	39	Bischoff	D	1956-10-29	M	1980	Ericaplein	78	9629CD	Den Haag	070-393435	[null]	[null]	[null]	[null]	[null]
8	44	Bakker, de	E	1963-01-09	M	1980	Lawaaistraat	23	4444LJ	Rijswijk	070-368753	1124	[null]	[null]	[null]	[null]
9	57	Bohemens, van	M	1971-08-17	M	1985	Erasmusweg	16	4377CB	Den Haag	070-473458	6409	[null]	[null]	[null]	[null]
10	83	Hofland	PK	1956-11-11	M	1982	Mariakade	16a	1812UP	Den Haag	070-353548	1608	[null]	[null]	[null]	[null]
11	95	Meuleman	P	1963-05-14	M	1972	Hoofdweg	33a	5746OP	Voorburg	070-867564	[null]	[null]	[null]	[null]	[null]
12	100	Permentier	P	1963-02-28	M	1979	Hazensteinln	80	6494SG	Den Haag	070-494593	6524	[null]	[null]	[null]	[null]
13	112	Baalen, van	IP	1963-10-01	V	1984	Vosseweg	8	6392LK	Rotterdam	010-548745	1319	[null]	[null]	[null]	[null]

CROSS JOIN

- = expliciet cartesisch product

```
SELECT *
FROM teams CROSS JOIN boetes;
```

- Leesbaarheid

UNION JOIN*

- Uit de standaard verwijderd sinds SQL2003
- UNION JOIN: elke rij van elke tabel wordt 1 maal opgenomen en aangevuld met null-waardes voor de kolommen uit de andere tabel

```
SELECT  *
FROM    teams UNION JOIN boetes;
-- (niet ondersteund door postgresql)
```

NATURAL JOIN

- Natuurlijke join, lexicografisch

```
SELECT  *
FROM    teams NATURAL INNER JOIN boetes
WHERE   divisie = 'ere';
```

- hetzelfde als een INNER JOIN USING?
- werkt ook voor andere JOINS (vb. NATURAL RIGHT OUTER JOIN)

EQUI/THETA JOIN

- EQUI JOIN: vergelijking met =
 - THETA JOIN: vergelijking met een andere vergelijkingsoperator
-
- tel hoeveel spelers er zijn met een ander nummer?
 - tel en toon hoeveel spelers er zijn met een even lange naam?
 - we willen telkens de spelersnr, naam en aantal

THETA JOIN

- Tel hoeveel spelers er zijn met een ander nummer
- We willen telkens de spelersnr, naam en aantal

```
SELECT s.spelersnr, s.naam, count(sp.spelersnr)
FROM spelers s INNER JOIN spelers sp
    ON (s.spelersnr <> sp.spelersnr)
GROUP BY s.spelersnr, s.naam
```

	spelersnr integer	naam character (15)	count bigint
1	112	Baalen, van	13
2	83	Hofland	13
3	28	Cools	13
4	7	Wijers	13
5	104	Moerman	13
6	100	Permentier	13
7	2	Elfring	13
8	6	Permentier	13
9	27	Cools	13
10	95	Meuleman	13
11	39	Bischoff	13
12	57	Bohemen, van	13
13	44	Bakker, de	13
14	8	Niewenburg	13

THETA JOIN

- Tel hoeveel spelers er zijn met een even lange naam
- We willen telkens de spelersnr, naam en aantal

THETA JOIN

```
SELECT s.spelersnr, s.naam, count(sp.spelersnr)
FROM spelers s INNER JOIN spelers sp
ON (s.spelersnr <> sp.spelersnr)
WHERE length(s.naam) = length(sp.naam)
GROUP BY s.spelersnr, s.naam
```

```
SELECT s.spelersnr, s.naam, count(sp.spelersnr)
FROM spelers s INNER JOIN spelers sp
ON (s.spelersnr <> sp.spelersnr)
AND length(s.naam) = length(sp.naam)
GROUP BY s.spelersnr, s.naam
```

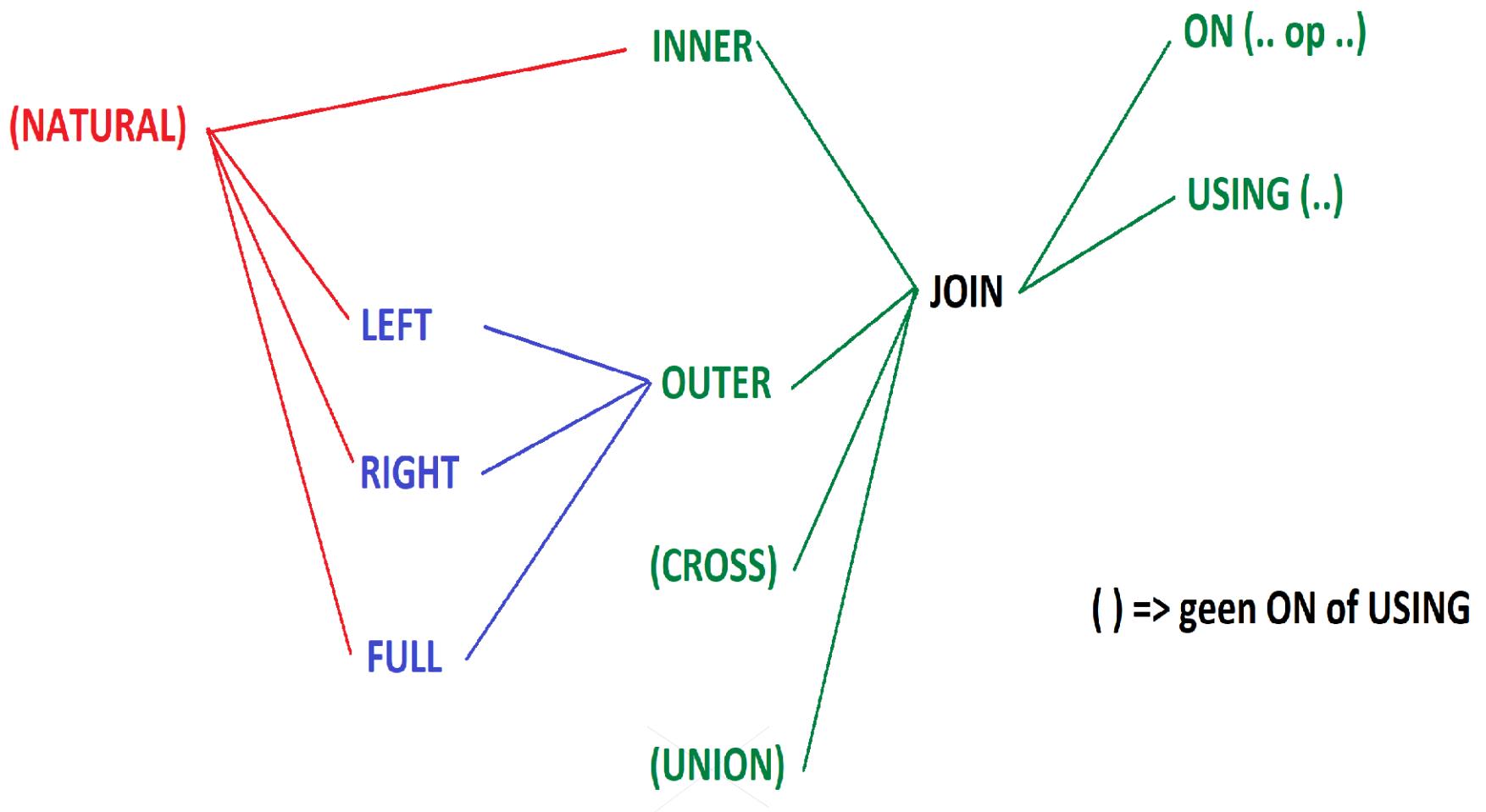
	spelersnr integer	naam character (15)	count bigint
1	104	Moerman	2
2	100	Permentier	3
3	2	Elfring	2
4	6	Permentier	3
5	27	Cools	1
6	95	Meuleman	1
7	83	Hofland	2
8	28	Cools	1
9	39	Bischoff	1
10	44	Bakker, de	3
11	8	Nieuwenburg	3

THETA JOIN

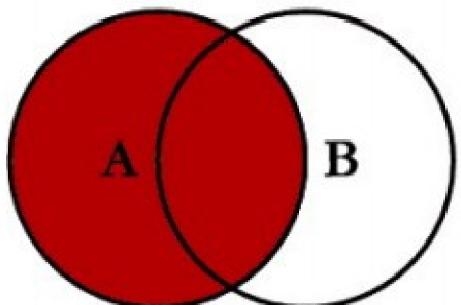
```
1 | SELECT      s.spelersnr, s.naam, count(sp.spelersnr), array_agg(sp.naam)
2 | FROM        spelers s INNER JOIN spelers sp
3 |          ON (s.spelersnr <> sp.spelersnr)
4 | WHERE       length(s.naam) = length(sp.naam)
5 | GROUP BY    s.spelersnr, s.naam
6 | ORDER BY    3
```

	Data Output	Explain	Messages	Notifications	Query History
	spelersnr integer	naam character (15)	count bigint	array_agg character[]	
1	27	Cools	1	{"Cools "}	
2	95	Meuleman	1	{"Bischoff "}	
3	39	Bischoff	1	{"Meuleman "}	
4	28	Cools	1	{"Cools "}	
5	104	Moerman	2	{"Hofland ","Elfring "}	
6	2	Elfring	2	{"Moerman ","Hofland "}	
7	83	Hofland	2	{"Moerman ","Elfring "}	
8	6	Permentier	3	{"Permentier ","Bakker, de ","Nieuwenburg "}	
9	100	Permentier	3	{"Bakker, de ","Nieuwenburg ","Permentier "}	
10	44	Bakker, de	3	{"Permentier ","Nieuwenburg ","Permentier "}	
11	8	Nieuwenburg	3	{"Permentier ","Bakker, de ","Permentier "}	

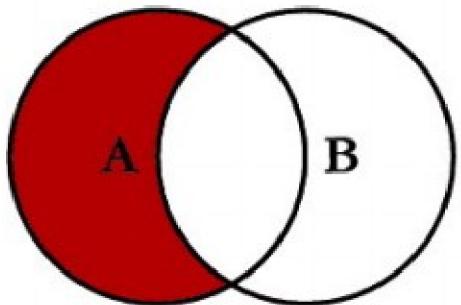
Notedop



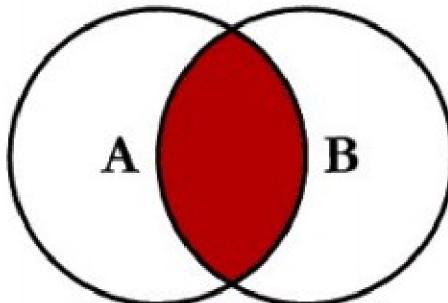
SQL JOINS



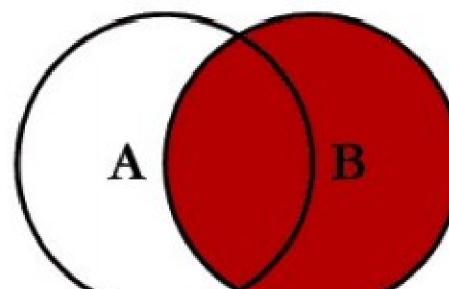
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



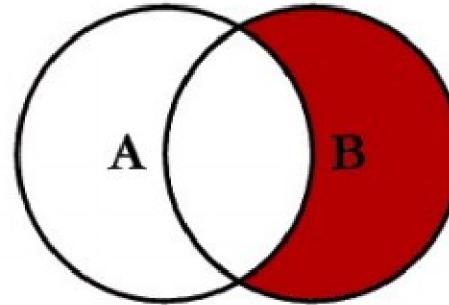
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



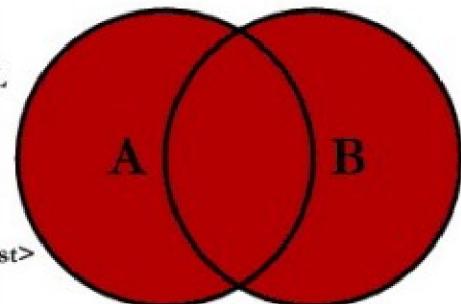
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



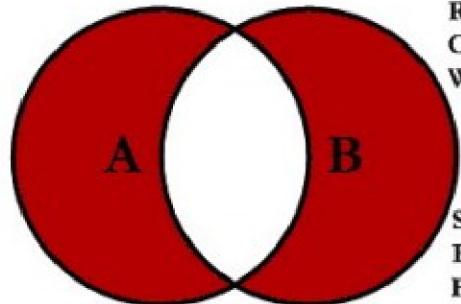
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Wim Bertels (CC)BY-SA-NC

Referenties:

- SQL Leerboek, R. Van der Lans
- Slides Databanken, H. Martens
- C. L. Moffatt, 2008
- <http://users.atw.hu/sqlnut/sqlnut2-chp-1-sect-2.html>

Lateral Joins

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0 Unported
Licentie

Inleiding

```
CREATE TEMPORARY TABLE numbers AS  
SELECT generate_series(1,3) AS max_num;
```

```
SELECT *  
FROM numbers;
```

-- Tijdelijke tabellen worden periodiek opgeruimd

max_num

1
2
3
(3 rows)

Probleem?

```
SELECT  *
FROM    numbers,
        (SELECT generate_series(1,max_num)) AS max_lijst;
```

Probleem?

```
SELECT *
FROM   numbers,
       (SELECT generate_series(1,max_num)) AS max_lijst;
```

```
/*
psql:04_2_LATERAL.sql:11: ERROR:  column "max_num" does not exist
LINE 3:   (SELECT generate_series(1,max_num)) AS max_lijst;

HINT:  There is a column named "max_num" in table "numbers",
but it cannot be referenced from this part of the query.
*/
```

LATERAL

Subqueries die verschijnen in FROM kunnen voorafgegaan worden door het sleutelwoord LATERAL.

Hierdoor kunnen ze verwijzen naar kolommen uit voorafgaande FROM-items.

(Zonder LATERAL, wordt elke subquery onafhankelijk geëvalueerd en kan deze dus niet verwijzen naar een ander FROM item).

Oftewel: gecorreleerde subqueries in de FROM-komponent

LATERAL Detail

Via LATERAL kunnen we verwijzen naar een eerdere referentie uit de FROM-komponent:

- naar een eerdere tabelreferentie
- naar een eerdere subquery
- naar een eerdere functie die een verzameling teruggeeft (SRF)
 - (In dit geval wordt een LATERAL gedrag door de standaard bepaalt, je kan dus LATERAL weglaten in dit geval)

Oplossing

```
SELECT *
FROM   nummers,
       (SELECT generate_series(1,max_num) AS max_lijst;
```

```
/*
psql:04_2_LATERAL.sql:11: ERROR:  column "max_num" does not exist
LINE 3:  (SELECT generate_series(1,max_num)) AS max_lijst;
HINT:  There is a column named "max_num" in table "nummers",
but it cannot be referenced from this part of the query.
*/
```

```
SELECT *
FROM   nummers, LATERAL
       (SELECT generate_series(1,max_num) AS max_lijst;
```

Uitvoer

```
SELECT *
FROM   numbers, LATERAL
       (SELECT generate_series(1,max_num) AS max_lijt;
```

max_num
1
2
3
(3 rows)

max_num	generate_series
1	1
2	1
2	2
3	1
3	2
3	3
(6 rows)	

PostgreSQL uitbreiding

```
SELECT      *
FROM        numbers, LATERAL generate_series(1,max_num);
```

```
SELECT      *
FROM        numbers, generate_series(1,max_num);
-- SELECT voor functie is optioneel in dit geval
```

Voorbeeld

```
SELECT      *
FROM        klanten k
LEFT JOIN LATERAL
(SELECT    age(k.geboortedatum) as leeftijd) AS l
ON (leeftijd=age(k.geboortedatum));
```

Voorbeeld

```
SELECT      *
FROM        klanten k
LEFT JOIN LATERAL
(SELECT    age(k.geboortedatum) as leeftijd) AS l
ON        (leeftijd=age(k.geboortedatum));
```

-- toon per klant de leeftijd

klantnr	naam	vnaam	geboortedatum	leeftijd
121	Hassoui	Sjeik	1975-06-12	43 years 8 mons 27 days
122	Martens	Hedwigh	1978-06-30	40 years 8 mons 9 days
123	Ellison	Larry	1975-10-10	43 years 4 mons 30 days
124	Van Rossem	Jean-Pierre	1975-01-12	44 years 1 mon 28 days
125	Frimout	Dirk	1980-11-29	38 years 3 mons 10 days
126	Gates	Bill	1982-12-25	36 years 2 mons 15 days

JOIN Conditie

- Is de join conditie nodig?
 - De subquery is gecorreleerd.
- Wat is het effect van het cartesisch produkt in dit geval?

Voorbeeld

```
SELECT      *
FROM        klanten k
LEFT JOIN LATERAL
(SELECT    age(k.geboortedatum) as leeftijd) AS l
ON (leeftijd=age(k.geboortedatum));
```

```
SELECT      *
FROM        klanten k
LEFT JOIN LATERAL
(SELECT    age(k.geboortedatum) as leeftijd) AS l
ON true;
```

JOIN Conditie

When a FROM item contains LATERAL cross-references, evaluation proceeds as follows:

for each row of the FROM item providing the cross-referenced column(s), or set of rows of multiple FROM items providing the columns, the LATERAL item is evaluated using that row or row set's values of the columns. The resulting row(s) are joined as usual with the rows they were computed from. This is repeated for each row or set of rows from the column source table(s).

> Cartesisch produkt gedraagt zich hier als een foreach lus

JOIN Conditie

Welk soort JOIN?

- CROSS, INNER, LEFT
- Niet: RIGHT of FULL!
- De subquery is gecorreleerd.

Gedrag LEFT blijft tov INNER

Een geval

Geef voor elke reis
de twee kleinste objecten die bezocht worden

- Ter vergelijking:

```
SELECT *
```

```
FROM bezoeken NATURAL INNER JOIN hemelobjecten
```

```
WHERE reisnr = 32
```

```
ORDER BY diameter;
```

objectnaam	reisnr	volgnr	verblijfsduur	satellietvan	afstand	diameter
Deimos	32	3		0	Mars	23.400
Phobos	32	2		1	Mars	9.270
Maan	32	5		2	Aarde	384.400
Maan	32	1		0	Aarde	384.400
Mars	32	4		3	Zon	227900.000

(5 rows)

Een geval

Geef voor elke reis
de twee kleinste objecten die bezocht worden

```
SELECT *
FROM reizen r LEFT JOIN LATERAL
  (SELECT *
   FROM bezoeken b NATURAL INNER JOIN hemelobjecten h
   WHERE b.reisnr=r.reisnr
   ORDER BY h.diameter
   FETCH FIRST 2 ROWS ONLY) AS I
ON (true);
```

Wim Bertels (CC)BY-SA-NC

Referenties:

- <https://www.postgresql.org/docs/current/sql-select.html>
- <https://www.postgresql.org/docs/current/static/queries-table-expressions.html>
- <https://blog.2ndquadrant.com/join-lateral/>
- <https://modern-sql.com/slides>
- Becoming A SQL Guru, Stella Nisenbaum