

5 Geavanceerde Raadpleging



5.1 De Join

5.2 Groepsfuncties

5.3 Gecorreleerde subqueries

5.4 Verzamelingoperatoren

5.5 Oefeningen

5.1.1 De equi-join (inner join)

- Geef de namen van alle medewerkers en de naam van de afdeling waar ze werken

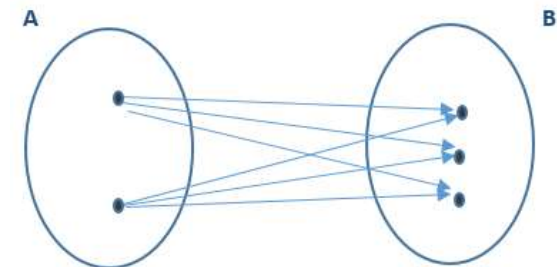
```
SELECT      m.naam, a.naam "afdeling"  
FROM        afdelingen a  
,          medewerkers m;
```

NAAM	AFDELING
CASPERS	OPLEIDINGEN
ALLARD	VERKOOP
DEFOUR	VERKOOP
JACOBS	OPLEIDINGEN
MARTENS	VERKOOP

- FROM-component: meerdere tabellen aanspreken → info uit verschillende tabellen opvragen
- Medewerkers **m** → tabelalias
- Resultaat: cartesiaans product (productverzameling) van beide tabellen

→ 56 rijen nl. 14 medewerkers X 4 afdelingen

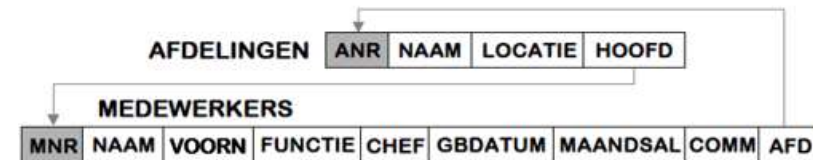
Vermelding tabellen van klein naar groot!



5.1.1 De equi-join (inner join)

□ WHERE-component: juiste rij-combinaties verkrijgen

```
□ SELECT    m.voorn
           ,    m.naam
           ,    a.naam
FROM        afdelingen a
           ,    medewerkers m
WHERE       m.afd = a.anr;
```



□ Opmerking:

□ SQL-commando zo overzichtelijk mogelijk maken

■ Leesbaarheid

■ Onderhoud baarheid

→ Commando zoveel mogelijk over verschillende regels verdelen

5.1.2 De auto-join

- Tabel met zichzelf joinen (geen essentieel verschil met equi-join)
- Geef een overzicht van alle medewerkers met de naam van hun chef

□ SELECT m.naam as medewerker
 c.naam as chef
 medewerkers m
 medewerkers c
WHERE m.chef = c.mnr;

MEDEWERKER		CHEF
-----		-----
7902	DE COOMAN	7566 JACOBS
7788	SWINNEN	7566 JACOBS
7900	JACOBS	7698 BRIERS
7844	DEN RUYTER	7698 BRIERS
7654	MARTENS	7698 BRIERS
7521	DEFOUR	7698 BRIERS
7499	ALLARD	7698 BRIERS
7934	WOUTERS	7782 CLERCKX
7876	SLECHTEN	7788 SWINNEN
7782	CLERCKX	7839 DE KONING
7698	BRIERS	7839 DE KONING
7566	JACOBS	7839 DE KONING
7369	CASPERS	7902 DE COOMAN

5.1.3 De outer-join

mnr	naam	afd
7001	Thommis	10
7002	Schepers	20
7003	Claesen	30
7004	Brams	20

anr	naam
10	Boekhouden
20	Expeditie
30	Personeelszaken
40	Verkoop

5.1.3 De outerjoin

- `SELECT m.mnr, a.anr, a.naam`
`FROM afdelingen a`
`, medewerkers m`
- Resultaat:

mnr	naam	afd
7001	Thommis	10
7002	Schepers	20
7003	Claesen	30
7004	Brams	20

anr	naam
10	Boekhouden
20	Expeditie
30	Personeelszaken
40	Verkoop

mnr	anr	naam
7001	10	Boekhouden
7001	20	Expeditie
7001	30	Personeelszaken
7001	40	Verkoop
7002	10	Boekhouden
7002	20	Expeditie
7002	30	Personeelszaken
7002	40	Verkoop
7003	10	Boekhouden
7003	20	Expeditie
7003	30	Personeelszaken
7003	40	Verkoop
7004	10	Boekhouden
7004	20	Expeditie
7004	30	Personeelszaken
7004	40	Verkoop

5.1.3 De outerjoin

❑ `SELECT m.mnr, a.anr, a.naam`
`FROM afdelingen a`
`, medewerkers m`
`WHERE m.afd = a.anr`

❑ Resultaat:

MNR	ANR	NAAM
7369	20	OPLEIDINGEN
7499	30	VERKOOP
7521	30	VERKOOP
7566	20	OPLEIDINGEN
7654	30	VERKOOP
7698	30	VERKOOP
7782	10	HOOFDKANTOOR
7788	20	OPLEIDINGEN
7839	10	HOOFDKANTOOR
7844	30	VERKOOP
7876	20	OPLEIDINGEN
7900	30	VERKOOP
7902	20	OPLEIDINGEN
7934	10	HOOFDKANTOOR

5.1.3 De outerjoin

```
□ SELECT m.mnr, a.anr, a.naam
   FROM medewerkers m
   ,   afdelingen a
  WHERE m.afd(+) = a.anr
```

(+) *aan kant met matching Null*
OF tegenkant van tabel met alle rijen

MNR	ANR	NAAM
7369	20	OPLEIDINGEN
7499	30	VERKOOP
7521	30	VERKOOP
7566	20	OPLEIDINGEN
7654	30	VERKOOP
7698	30	VERKOOP
7782	10	HOOFDKANTOOR
7788	20	OPLEIDINGEN
7839	10	HOOFDKANTOOR
7844	30	VERKOOP
7876	20	OPLEIDINGEN
7900	30	VERKOOP
7902	20	OPLEIDINGEN
7934	10	HOOFDKANTOOR
	40	PERSONEELSZAKEN

5.1.3 De outerjoin

□ Hoe bekom je dit resultaat?

□

```
SELECT m.naam as medewerker
      , c.naam as chef
FROM   medewerkers m
      , medewerkers c
WHERE  m.chef = c.mnr(+)
```

MEDEWERKER	CHEF
CASPERS	DE COOMAN
ALLARD	BRIERS
DEFOUR	BRIERS
JACOBS	DE KONING
MARTENS	BRIERS
BRIERS	DE KONING
CLERCKX	DE KONING
SWINNEN	JACOBS
DE KONING	
DEN RUYTER	BRIERS
SLECHTEN	SWINNEN
JACOBS	BRIERS
DE COOMAN	JACOBS
WOUTERS	CLERCKX

14 rows selected.

5.1.4 ANSI/ISO SQL-Standaard !!!

□ ANSI/ISO SQL-Standaard : **JOIN** of **INNER JOIN**

□ SELECT m.naam
 ,a.naam "afdeling"

FROM afdelingen a
 INNER JOIN medewerkers m
 ON m.afd = a.anr;

```
SELECT      m.voorn  
,          m.naam  
,          a.naam  
FROM        afdelingen a  
,          medewerkers m  
WHERE       m.afd = a.anr;
```

JOIN ON: join-conditions

□ SELECT m.naam "medewerker"
 mm.naam "chef"
 ,
FROM medewerkers m
 INNER JOIN medewerkers mm
 ON m.chef = mm.mnr

WHERE m.gbdatum > date '1985-01-01'
ORDER BY m.naam;

WHERE: niet-join conditions

5.1.4 ANSI/ISO SQL-Standaard

- Voorbeeld: **RIGHT JOIN** of **RIGHT OUTER JOIN** (geeft alle rijen!)

```
SELECT  a.anr, a.locatie, m.naam
        FROM  medewerkers m RIGHT OUTER JOIN  afdelingen a
              ON  m.afd = a.anr;
```

Of

```
SELECT  a.anr, a.locatie, m.naam
        FROM  afdelingen a LEFT OUTER JOIN medewerkers m
              ON  m.afd = a.anr;
```

<u>mnr</u>	anr	naam
7001	10	Boekhouden
7002	20	Expeditie
7003	30	Personeelszaken
7004	20	Expeditie
NULL	40	Verkoop

Opmerking:

- LEFT OUTER JOIN (alle rijen van de linker tabel)
- RIGHT OUTER JOIN (alle rijen van de rechter tabel)
- FULL OUTER JOIN (alle rijen van beide tabellen)

5.1.4 ANSI/ISO SQL-Standaard

□ USING-component

- SELECT i.cursist, u.locatie, i.cursus
FROM inschrijvingen i
INNER JOIN uitvoeringen u
ON i.cursus=u.cursus AND i.begindatum = u.begindatum;

- SELECT i.cursist, u.locatie, cursus (! i.cursus of u.cursus gaat hier niet)
FROM inschrijvingen i
INNER JOIN uitvoeringen u
USING (cursus,begindatum);

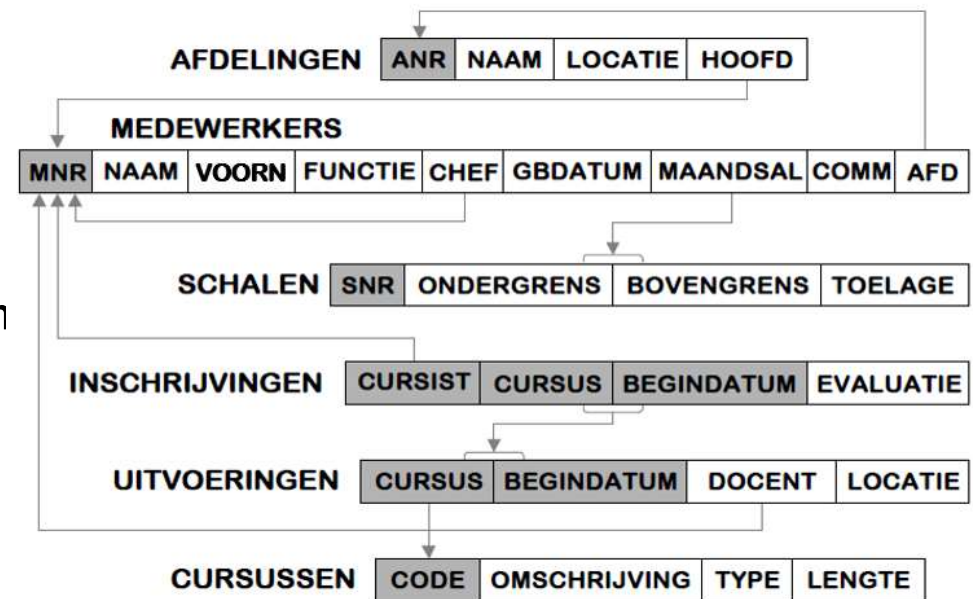
USING-component gebruiken wanneer in beide tabellen de JOIN-restrictor **dezelfde kolomnamen** gebruiken

5.1.4 ANSI/ISO SQL-Standaard

- Vb. Geef de namen van de medewerkers die een cursus volgen in één van de bedrijfsafdelingen (dus geen cursuslocaties).

```
SELECT distinct m.naam, m.voorn, u.locatie
FROM medewerkers m
  INNER JOIN inschrijvingen i
    ON m.mnr = i.cursist
  INNER JOIN uitvoeringen u
    ON u.begindatum = i.begindatum
    AND u.cursus = i.cursus
  INNER JOIN afdelingen a
    ON a.locatie = u.locatie;
```

NAAM	VOORN	LOCATIE
JACOBS	SIMON	HASSELT
DE COOMAN	DORIEN	HASSELT
JACOBS	SIMON	GENK
JACOBS	EMMA	HASSELT...



5.1.5 Break

Met behulp van een BREAK-commando kunnen we in een rapport kolomwaarden onderdrukken, een regel overslaan of een extra regel toevoegen.

□ Voorbeeld 1

SQL> BREAK ON afd SKIP 1 *(onderbreking op afdeling en 1 lege regel toevoegen)*

SQL> select afd, functie, mnr, naam, maandsal, comm
from medewerkers
order by afd,functie;

AFD	FUNCTIE	MNR	NAAM	MAANDSAL	COMM
10	BOEKHOUDER	7934	WOUTERS	2300	
	DIRECTEUR	7839	DE KONING	7000	
	MANAGER	7782	CLERCKX	3450	
20	MANAGER	7566	JACOBS	4975	
	TRAINER	7369	CASPERS	1800	
	TRAINER	7902	DE COOMAN	4000	
	TRAINER	7876	SLECHTEN	2700	
	TRAINER	7788	SWINNEN	4000	
30	BOEKHOUDER	7900	JACOBS	2800	
	MANAGER	7698	BRIERS	5850	
	VERKOPER	7521	DEFOR	2250	5000
	VERKOPER	7844	DEN RUYTER	2500	0
	VERKOPER	7499	ALLARD	1600	3000
	VERKOPER	7654	MARTENS	2250	3400

Onderbreking op afdeling met pauzestop

SQL> BREAK ON afd PAGE

SQL> SET PAUSE ON

5.1.5 Break

□ Voorbeeld 2

SQL> BREAK ON afdeling SKIP 1 ON functie;

```
select a. naam "afdeling"  
      ,      m.functie  
      ,      m. naam "medewerker"  
from medewerkers m  
      right outer join  
      afdelingen a  
      on a.anr=m.afd  
order by a. naam, m.functie;
```

□ Break-instellingen verwijderen.

SQL> clear breaks

AFDELING	FUNCTIE	MEDEWERKER
HOOFDKANTOOR	BOEKHOUDER	WOUTERS
	DIRECTEUR	DE KONING
	MANAGER	CLERCKX
OPLEIDINGEN	MANAGER	JACOBS
	TRAINER	CASPERS
		SWINNEN
		DE COOMAN
		SLECHTEN
PERSONEELSZAKEN		
VERKOOP	BOEKHOUDER	JACOBS
	MANAGER	BRIERS
	VERKOPER	DEFOUR
		DEN RUYTER
		ALLARD
		MARTENS