

4 Raadpleging

4.1 Overzicht van de SELECT-component

4.2 Subquery's

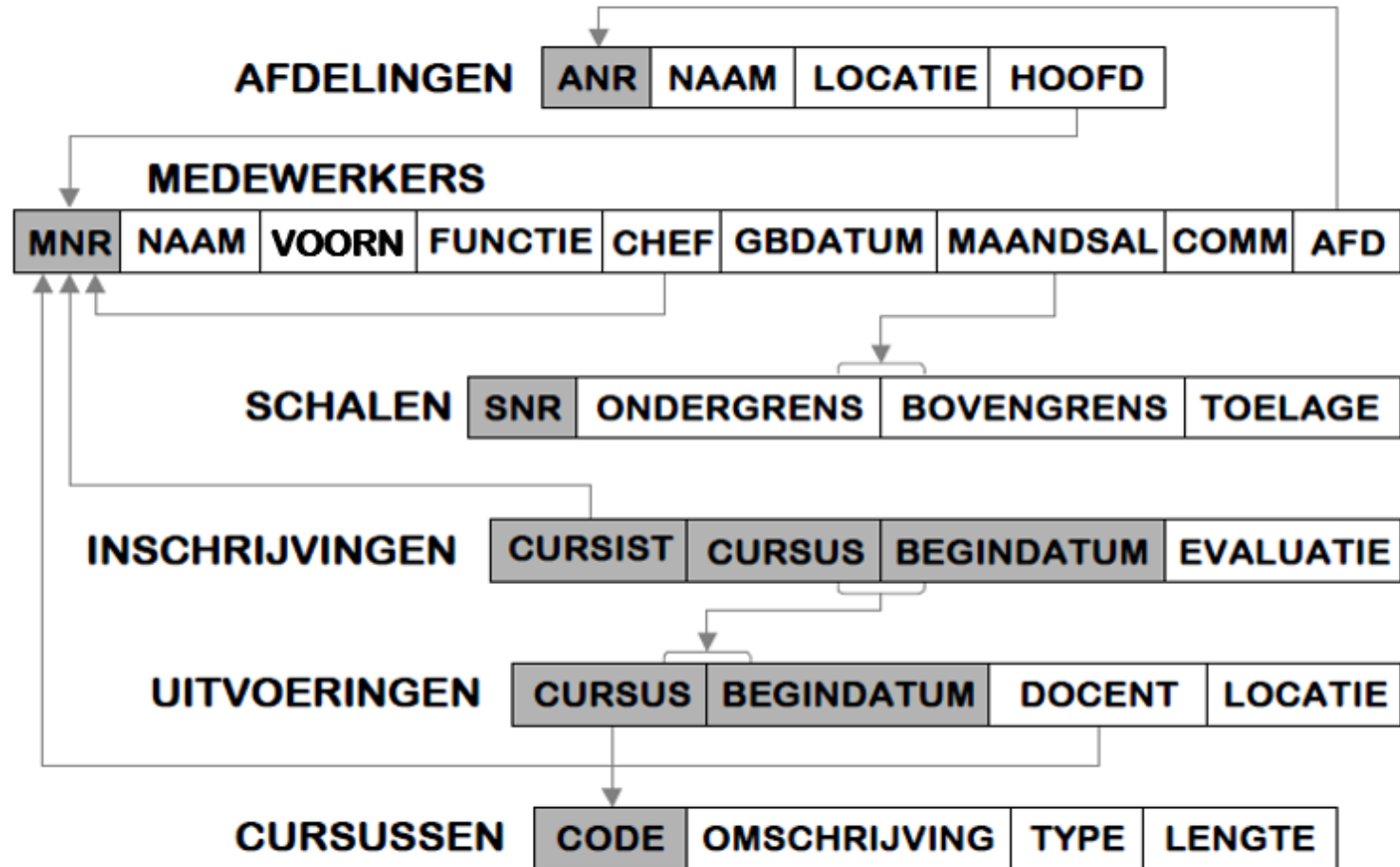
4.3 Functies

4.4 Oefeningen

4.1 Overzicht SELECT-Component

```
SELECT [DISTINCT] select_expressie  
FROM tabel_expressie  
[WHERE conditie]  
[GROUP BY expressie] [HAVING conditie]  
[ORDER BY {expressie} [ASC | DESC]]
```

dB Casus



4.1.1 De SELECT-component

- Alle gegevens van alle medewerkers
 - ▣ SELECT *
FROM medewerkers;
- Geef de naam en het afdelingsnummer van alle medewerkers;
 - ▣ SELECT naam, afd
FROM medewerkers;

4.1.1 De SELECT-component

- Verfraai de kolomtitels (kolom van vorig voorbeeld)
 - ▣ SELECT naam "Medewerkers uit ", afd as afdeling
FROM Medewerkers
- Geef de afdelingen waar de medewerkers werken
 - ▣ SELECT DISTINCT afd
FROM medewerkers

4.1.1 De SELECT-component

- Geef naam en afdelingsnummer van alle medewerkers en verfraai
 - ▣ SELECT naam, 'is werkzaam in afdeling', afd
FROM medewerkers;
- Werk in vorig voorbeeld de tabulators weg
 - ▣ SELECT naam || ' is werkzaam in afdeling ' || afd
FROM medewerkers;

4.1.1 De SELECT-component

- Geef de huidige datum
 - ▣ SELECT sysdate
FROM dual
- Geef het jaarsalaris (excl. commissie) van iedere medewerkers
 - ▣ SELECT voorn, naam, 12*maandsal as jaarsalaris
FROM medewerkers;

4.1.2 De WHERE-Component

- Met de **relationele restrictie-operator WHERE** kunnen we een **voorwaarde of conditie** specificeren
- Enkelvoudige condities
 - ▣ Vergelijkingsoperatoren ($>$, $<$, ...) zie syllabus p 9
- Samengestelde condities
 - ▣ Logische operatoren (AND, OR en NOT)

		AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

4.1.1.1 Enkelvoudige conditie

- Geef de medewerkers van afdeling 30
 - ▣ SELECT voorn, naam, afd
FROM medewerkers
WHERE afd=30
- Geef de werknemers geboren na 1 mei 1983
 - ▣ SELECT voorn, naam, gbdatum
FROM medewerkers
WHERE gbdatum > DATE '1983-05-01';

standaard datumformat: '01-05-1983 '

4.1.1.2 Operatoren: Between, In, Like

- Geef de medewerkers die werken in afdeling 20 t.e.m. afdeling 30
 - ▣ SELECT voorn, naam
FROM medewerkers
WHERE afd **between** 20 AND 30
- Geef de medewerkers van afdeling 10 of 30
 - ▣ SELECT voorn, naam
FROM medewerkers
WHERE afd **in** (30,10)

4.1.1.2 Operatoren Between, In, Like

- Geef alle medewerkers in wiens naam een 'A' voorkomt
 - ▣ SELECT voorn, naam
FROM medewerkers
WHERE naam like '%A%' *Caspers, Allard, Martens,...*
- Betekenis wildcards
 - ▣ Onderlijningsteken (_): precies 1 willekeurig karakter
 - ▣ %-teken: nul, 1 of meer willekeurige karakters

4.1.1.3 Samengestelde conditie

- Geef de medewerkers die niet in afdeling 30 werken

- ▣ SELECT voorn, naam
FROM medewerkers
WHERE NOT afd = 30

of

WHERE afd <> 30

4.1.1.3 Samengestelde conditie

- Geef de medewerkers van afdeling 10 of afdeling 30
 - ▣ SELECT voorn, naam
FROM medewerkers
WHERE afd = 30 OR afd = 10
of
 - ▣ SELECT voorn, naam
FROM medewerkers
WHERE afd in (30,10)

4.1.1.3 Samengestelde conditie

- Geef alle medewerkers behalve Briers of de medewerkers van afdeling 30

- `SELECT *`

- `FROM medewerkers`

- `WHERE NOT (naam = 'BRIERS' OR afd = 30)`

of

- `WHERE naam <> 'BRIERS' AND afd <> 30`

JACOBS	20	TRUE
JACOBS	30	FALSE
BRIERS	20	FALSE
BRIERS	30	FALSE

4.1.1.3 Samengestelde conditie

- Geef alle cursussen die geen goede waardering (goed: 3,4,5) hebben behaald

- ▣ SELECT *
 - FROM inschrijvingen
 - WHERE evaluatie NOT IN (3,4,5)

of

WHERE NOT evaluatie IN (3,4,5)

WHERE NOT (evaluatie = 3 OR evaluatie = 4 OR evaluatie = 5)

WHERE evaluatie <> 3 AND evaluatie <> 4 AND evaluatie <> 5

4.1.1.3 Samengestelde conditie

□ Let op!

- Samengestelde conditie: aangeraden om haakjes te gebruiken → volgorde afdwingen

- `where naam = 'BRIERS' OR naam like 'D%' AND afd = 20`

- `where naam = 'BRIERS' OR (naam like 'D%' AND afd = 20)`

- `where (naam = 'BRIERS' OR naam like 'D%') AND afd = 20`



- **! AND heeft hogere prioriteit dan OR (1ste en 2de where hetzelfde)**

Oefeningen 4 -18 (behalve oef 14 – werken met NULL)

4.1.3 De ORDER BY-component

- Geef de gegevens van alle werknemers in volgorde van personeelsnummer
 - ▣ SELECT *
FROM medewerkers
ORDER BY mnr DESC
- Volgorde in functie van kolomnummer 2
 - ▣ SELECT *
FROM medewerkers
ORDER BY 2

4.1.3 De ORDER BY-component

- *Volgorde in stijgende volgorde van commissie en dalend voornaam.*

- ▣ SELECT *

- FROM medewerkers

- ORDER BY comm asc, naam desc;

- of** ORDER BY comm, naam desc

4.1.4 CASE-expressie

- Vb1: Geef een beoordeling voor de cursus SQL afh. van de evaluatie (1 = slecht, 2 = matig,..... 5=zeer goed)

```
SELECT cursist, begindatum  
, CASE evaluatie  
    WHEN 1 THEN 'Slecht'  
    WHEN 2 THEN 'Matig'  
    WHEN 3 THEN 'Ok'  
    WHEN 4 THEN 'Goed'  
    WHEN 5 THEN 'Zeer goed'  
    ELSE 'Niet ingevuld'  
END As Beoordeling  
FROM inschrijvingen  
WHERE cursus = 'SQL'
```

CURSIST	BEGINDATU	BEOORDELING
7499	16-APR-15	Goed
7934	16-APR-15	Zeer goed
7698	16-APR-15	Goed
7876	16-APR-15	Matig
7788	08-OCT-15	Niet ingevuld
7839	08-OCT-15	Ok
7902	08-OCT-15	Goed
7902	17-DEC-15	Niet ingevuld
7698	17-DEC-15	Niet ingevuld

9 rows selected.

4.1.4 CASE-expressie

□ Vb2:

```
SELECT cursist, begindatum  
, CASE  
    WHEN evaluatie = 0 THEN 'Slecht'  
    WHEN evaluatie < 2 THEN 'Matig'  
    WHEN evaluatie < 5 THEN 'Ok'  
    WHEN evaluatie < 7 THEN 'Goed'  
    WHEN evaluatie < 8 THEN 'Zeer goed'  
    ELSE 'Uitstekend'  
END Beoordeling  
FROM inschrijvingen  
WHERE cursus = 'SQL'
```

4.1.4 CASE-expressie

- Complexe procedurele problemen aanpakken
- Syntax:
 - ▣ **CASE** *expressie*
 WHEN *waarde* **THEN** *resultaat*
 [ELSE] *resultaat*
 END
- *expressie* en **WHEN**-*waarde*: hetzelfde datatype
- **THEN**- en **ELSE**-*resultaat*: hetzelfde datatype

4.1.5 NULL-waarden

- Geef de medewerkers waarvan de commissie niet van toepassing is.
 - ▣ SELECT voorn, naam, comm
FROM medewerkers
WHERE comm IS NULL
- Stelling!
 - ▣ NULL = NULL is onbeslist
 - ▣ NULL IS NULL is waar

4.1.5 NULL-waarden

- Indien kolom geen waarde = NULL waarde → informatie ontbreekt
- Weergave van NULL: kan op kolomniveau worden geregeld met commando Column
 - ▣ Show NULL
 - ▣ SET NULL "NULL"
 - ▣ Col comm NULL "Onbekend" *zie dia 5.7 (ook volgend dia 24)*

4.1.5 NULL-waarden

- Sortering in stijgende volgorde van commissie en dalende voor naam
 - ▣ `SELECT *`
`FROM medewerkers`
`ORDER BY comm ASC, naam DESC;`
 - OF***
 - `ORDER BY comm, naam DESC`
- Let op! Bij dalend sorteren NULL-waarde bovenaan en bij stijgend sorteren onderaan (hoogste waarde)

4.1.5.1 NVL-functie (Null Value)

- Geef de maandsalarissen en de commissie (inclusief verhoging van 10% op comm)

- ▣ SELECT maandsal, maandsal+NVL(comm,0)*1.1
FROM medewerkers

OF

SELECT maandsal, maandsal+NVL2(comm,comm,0) * 1.1
FROM medewerkers

- NVL: flexibel met NULL-waarde omgaan
 - ▣ NVL(x,y): geeft y als x NULL is, anders x zelf
 - ▣ NVL2(x,y,z): geeft y als x niet NULL is, anders z