

# 4 Raadpleging

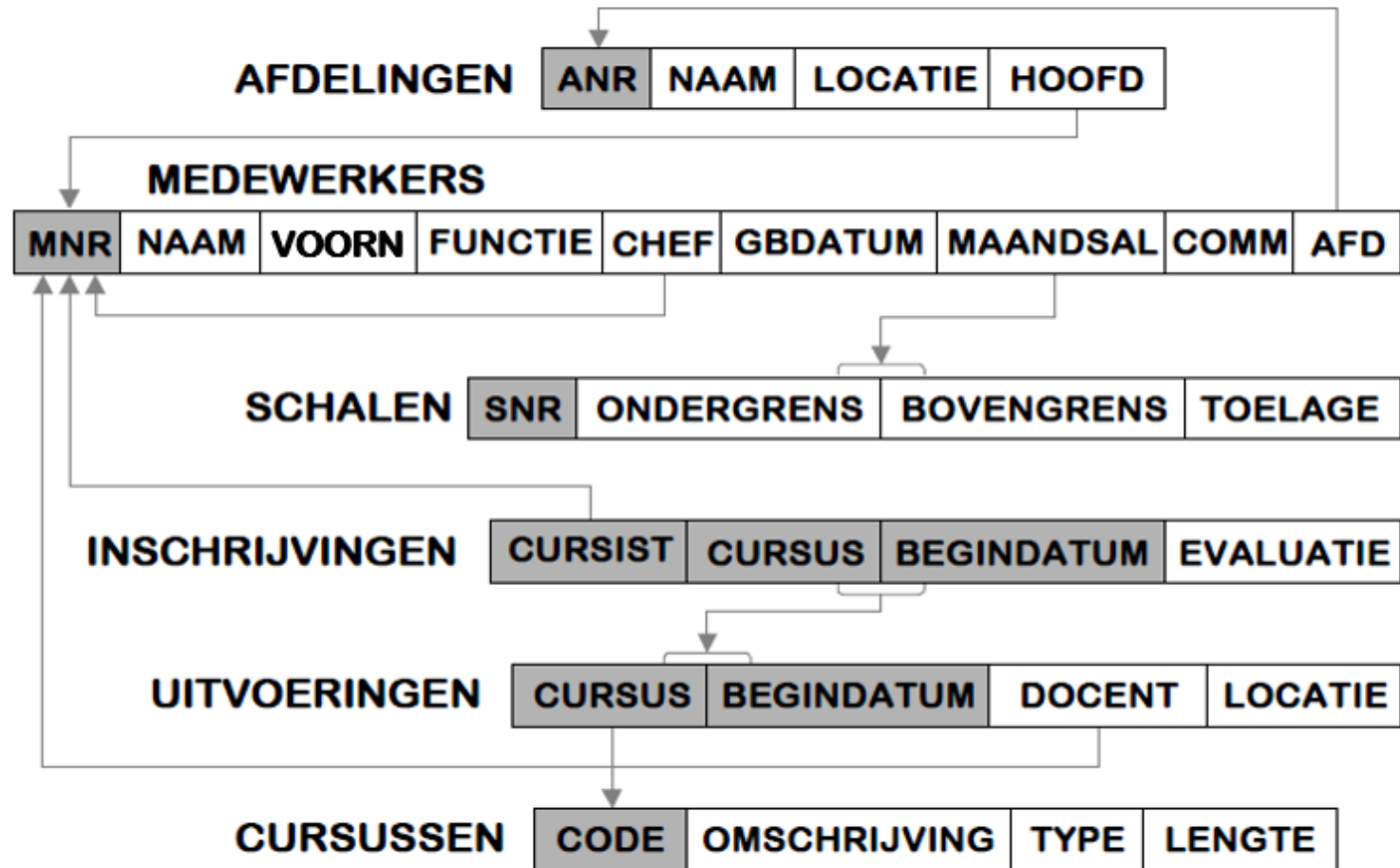
---

- 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 "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 "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)**



## 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 "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"

## 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

## 4.2 Subquery's

- Geef alle medewerkers die meer verdienen dan medewerker Slechten.

- ```
SELECT    voorn, naam, maandsal
FROM      medewerkers
WHERE     maandsal > ( SELECT maandsal
                        FROM      medewerkers
                        WHERE     naam = 'SLECHTEN');
```

## 4.2 Subquery's

- Geef de unieke cursistenummers van cursisten die de bouw cursus type BLD gevolgd hebben.

■ `SELECT DISTINCT cursist`  
`FROM inschrijvingen`  
`WHERE cursus in ( SELECT code`  
`FROM cursussen`  
`WHERE type='BLD')`

WEB, CIS, WIN

| CURSUSSEN |                               |     |
|-----------|-------------------------------|-----|
| CODE      | OMSCHRIJVING                  | TYP |
| SQL       | Introductie SQL en databanken | ALG |
| ORG       | IT Organisatie                | ALG |
| WEB       | Ontwikkeling website          | BLD |
| CIS       | Cisco CCNA                    | BLD |
| WIN       | Windows Server                | BLD |
| LIN       | Linux OS                      | DSG |
| PR2       | Programmeren2 in Visual C#    | DSG |
| WBA       | Web applicaties               | DSG |
| PR1       | Programmeren1 Visual C#       | DSG |
| NET       | Netwerkbeheer                 | DSG |

| INSCHRIJVINGEN |      |             |
|----------------|------|-------------|
| CURSIST        | CURS | BEGINDATUM  |
| 7499           | SQL  | 16-APR-2015 |
| 7934           | SQL  | 16-APR-2015 |
| 7698           | SQL  | 16-APR-2015 |
| 7876           | SQL  | 16-APR-2015 |
| 7788           | SQL  | 08-OKT-2015 |
| 7839           | SQL  | 08-OKT-2015 |
| 7902           | SQL  | 08-OKT-2015 |
| 7902           | SQL  | 17-DEC-2015 |
| 7698           | SQL  | 17-DEC-2015 |
| 7521           | ORG  | 10-AUG-2015 |
| 7900           | ORG  | 10-AUG-2015 |
| 7902           | ORG  | 10-AUG-2015 |
| 7844           | ORG  | 27-SEP-2016 |
| 7499           | WEB  | 17-DEC-2015 |
| 7782           | WEB  | 17-DEC-2015 |
| 7876           | WEB  | 17-DEC-2015 |
| 7788           | WEB  | 17-DEC-2015 |
| 7839           | WEB  | 17-DEC-2015 |
| 7566           | WEB  | 05-FEB-2016 |
| 7788           | WEB  | 05-FEB-2016 |
| 7698           | WEB  | 05-FEB-2016 |
| 7900           | WIN  | 04-FEB-2016 |
| 7499           | WIN  | 04-FEB-2016 |
| 7566           | CIS  | 11-SEP-2016 |
| 7499           | CIS  | 11-SEP-2016 |
| 7876           | CIS  | 11-SEP-2016 |

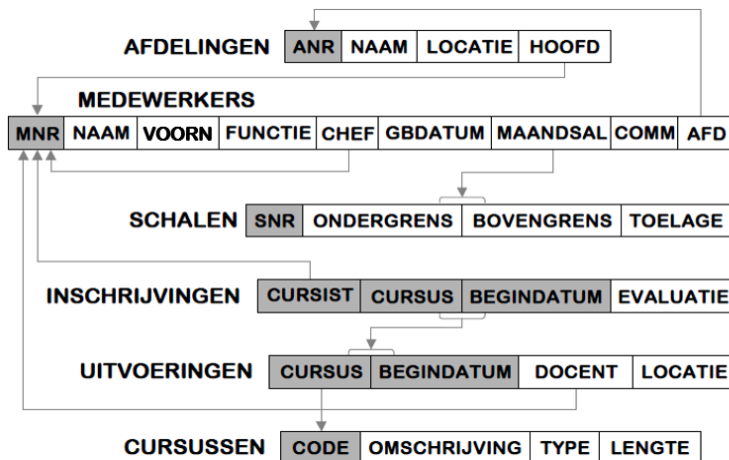
## 4.2 Subquery's

- Geef alle namen van medewerkers die de bouw cursus (type BLD) gevolgd hebben.

- SELECT voorn, naam  
FROM medewerkers

- WHERE mnr IN ( SELECT cursist  
FROM inschrijvingen

- WHERE cursus IN ( SELECT code  
FROM cursussen  
WHERE type = 'BLD' );



## 4.3 Functies

- Functie: naam gevolgd door () met een of meerdere argumenten
- Vrijwel overal toepasbaar in query
- 6 groepen
  - ▣ Numerieke functies                      toepasbaar op numerieke gegevens
  - ▣ Alfnumerieke functies                  toepasbaar op alfanumerieke gegevens
  - ▣ Datumfuncties                            toepasbaar op datum
  - ▣ Algemene functies                        toepasbaar op ieder datatype
  - ▣ Conversiefuncties                        conversie naar een ander datatype
  - ▣ Groepsfuncties                            toepasbaar op een groep gegevens

## 4.3.1 Rekenfuncties

Belangrijkste functies om met cijfers te werken:

- `ROUND(n[,m])`      rondt  $n$  af op  $m$  decimale posities
- `CEIL(n)`      rondt  $n$  naar boven af op een geheel getal
- `FLOOR(n)`      rondt  $n$  naar beneden af op een geheel getal
- `TRUNC(n[,m])`      kapt  $n$  af op  $m$  decimale posities
- `ABS(n)`      de absolute waarde van  $n$
- `SIGN(n)`      -1, 0 of 1 als  $n$  negatief, nul of positief is
- `SQRT(n)`      vierkantswortel uit  $n$  (*square root*)
- `POWER(n,m)`       $n$  tot de  $m$ -de macht
- `MOD(n,m)`      rest na deling van  $n$  door  $m$

## 4.3.1 Rekenfuncties

- `SELECT ROUND (345.678) FROM dual`
    - ▣ 346 (standaard: geen cijfers na de komma)
  - `SELECT CEIL(345.678), CEIL(345.278) FROM dual`
    - ▣ 346 346
  - `SELECT FLOOR (345.578) FROM dual`
    - ▣ 345
  - `SELECT ROUND(345.678, 2) FROM dual`
    - ▣ 345.68
  - `SELECT ROUND (345.678, -1) FROM dual`
    - ▣ 350
- (-1=tiental, -2=honderdtal, -3=duizendtal,...)

## 4.3.1 Rekenfuncties

- `SELECT TRUNC(345.678, 2) FROM dual`
  - ▣ 345.67
- `SELECT ABS(-123) , ABS(0), ABS(456) FROM dual`
  - ▣ 123/0/456
- `SELECT SIGN(-13) , SIGN(0), SIGN(456) FROM dual`
  - ▣ -1/0/1
- `SELECT SQRT(16), SQRT(8), SQRT(4) FROM dual`
  - ▣ 4/2.8284271/2
- `SELECT POWER(2, 3), POWER(-2,3) FROM dual`
  - ▣  $2^3=8$     $-2^3=-8$



## 4.3.1 Rekenfuncties

- `SELECT MOD(8,3) FROM dual` → 2
- `SELECT SIGN(maandsal-1600)`  
`FROM medewerkers`
  - ▣ 1/0/-1/... (welke groter, kleiner dan 1600)
- `SELECT *`  
`FROM medewerkers`  
`WHERE MOD (mnr, 2 ) = 0` → *even nummers....*
- `SELECT naam`  
`, sysdate-gbdatum "leeftijdInDagen"`  
`, floor ((sysdate-gbdatum)/7) "aantalWekenOud"`  
`, floor (mod(sysdate-gbdatum,7)) "aantalDagenOud"`  
`from medewerkers`

*vb. Caspers is 1922 weken en 1 dag oud*

## 4.3.2 Tekstfuncties

- **LENGTH(t)**      aantal karakters (lengte) van *t*  
*select LENGTH('hi all') from dual; → 6*
- **ASCII(t)**      ascii-waarde eerste karakter van *t*  
*select ascii('A') from dual; → 65*
- **CHR(n)**      karakter met ascii-waarde *n*  
*select chr(65) from dual; → A*
- **UPPER(t)**      *t* in hoofdletters  
*select UPPER('Hi all') from dual; → HI ALL*
- **LOWER(t)**      *t* in kleine letters  
*select LOWER('Hi ALL') from dual; → hi all*
- **INITCAP(t)**      elke woord in *t* met beginhoofdletter  
*select INITCAP('Hi jean-marie') from dual → Hi Jean-Marie*

## 4.3.2 Tekstfuncties

- **LTRIM(t,k)**      verwijdt links van *t* de *k*-karakters, t.e.m. het eerste karakter niet in *k*  
*stopt als het teken niet gevonden is: ltrim('marteressa', 'ram') → teressa*
- **RTRIM(t,k)**      verwijdt rechts van *t* de *k*-karakters , na het laatste karakter niet in *k*  
*stopt als een teken niet gevonden is: rtrim('marteress', 'sen') → marter*
- **LPAD(t,n)**      vult *t* links uit met spaties tot lengte *n*
- **LPAD(t,n,k)**      idem, met *k*-karakter  
*select lpad('\*', 10, '-') from dual; → -----\**
- **RPAD(t,n)**      vult *t* rechts aan met spaties tot lengte *n*
- **RPAD(t,n,k)**      idem, met *k*-karakter  
*select rpad('\*', 10, '-') from dual; → \*-----*

## 4.3.2 Tekstfuncties

- SUBSTR(*t*,*n*)  
geeft deel van *t* vanaf positie *n* tot het einde  
*select substr('kelder', 2) from dual; → elder*
- SUBSTR(*t*,*n*,*m*)  
geeft deel van *t* vanaf positie *n*, *m*-karakters lang  
*select substr('kelder', 2, 3) from dual; → eld*
- INSTR(*t*,*k*)  
positie eerste voorkomen van *k* in *t*  
*select instr('kelder', 'e') from dual; → 2*
- INSTR(*t*,*k*,*n*)  
idem, vanaf de *n*-de positie in *t*  
*select instr('kelder', 'e', 3) from dual; → 5*
- INSTR(*t*,*k*,*n*,*m*)  
het *m*-de voorkomen van *k*, vanaf de *n*-de positie in *t*  
*select instr('kelderen', 'e', 3, 2) from dual; → 7*

## 4.3.2 Tekstfuncties

- REPLACE(t,v)      verwijdt uit *t* elk voorkomen van *v* (woorden)  
*select replace('hello world', 'l') from dual; → heo word*
- REPLACE(t,v,w)      vervangt in *t* elk voorkomen van string *v* in *w* (woorden)  
*select replace('hello world', 'l', '\*') from dual;*  
*→ he\*\*o wor\*d*
- TRANSLATE(t,v,w)      vervangt alle **karakters** uit *v* die in *t* voorkomen door het corresponderende karakter uit *w*  
*select translate('veer', 've', 'bo') from dual;*  
*→ boor*
- CONCAT(t1,t2)      voegt *t1* en *t2* samen (equivalent met ||)  
*select concat('boor', 'machine') from dual;*  
*→ boormachine*

## 4.3.2 Tekstfuncties

- `SELECT UPPER(omschrijving), LOWER(type)`  
`FROM cursussen`
  - ▣ `INTRODUCTIE... alg`
  
- `SELECT anr, naam, INITCAP(locatie)`  
`FROM afdelingen ORDER BY LENGTH(naam)`
  
- `SELECT *`  
`FROM medewerkers`  
`WHERE LOWER (functie) = 'trainer'`
  - ▣ `SWINNEN`                      `TRAINER ...`

| ANR | NAAM            | INITCAP(LOCATIE) |
|-----|-----------------|------------------|
| 30  | VERKOOP         | Genk             |
| 20  | OPLEIDINGEN     | Hasselt          |
| 10  | HOOFDKANTOOR    | Maasmechelen     |
| 40  | PERSONEELSZAKEN | Leuven           |

## 4.3.2 Tekstfuncties

- `SELECT ASCII( 'a'), ASCII( 'z'), chr(77) FROM dual`
  - ▣ 97/122/M
  
- `SELECT SUBSTR(naam,4), SUBSTR(naam,4,3) FROM afdelingen`
  - ▣ hoofdkantoor → fdkantoor    fdk
  - ▣ Personeelszaken → soneelszaken    son

## 4.3.2 Tekstfuncties

□ SELECT naam,  
INSTR(naam,'A')  
FROM medewerkers

□ ALLARD:

■ 1

□ JACOBS:

■ 2

□ DE COOMAN

■ 8

□ SELECT naam,  
INSTR(naam,'A',3)  
FROM medewerkers

□ ALLARD:

■ 4

□ JACOBS:

■ 0

□ DE COOMAN

■ 8

□ SELECT naam,  
INSTR(naam,'A',1,2)  
FROM medewerkers

□ ALLARD:

■ 4

□ JACOBS:

■ 0

□ DE COOMAN

■ 0



## 4.3.2 Tekstfuncties

### □ SELECT

**LTRIM**(naam,'SDAER')  
FROM medewerkers

□ CASPERS

■ CASPERS

□ ALLARD

■ LLARD

□ DEN RUYTER

■ N RUYTER

### □ SELECT

**RTRIM**(naam,'SDAER')  
FROM medewerkers

□ CASPERS

■ CASP

□ ALLARD

■ ALL

□ DEN RUYTER

■ DEN RUYT

## 4.3.2 Tekstfuncties

- `SELECT LPAD(naam,8,'@'), RPAD (naam,10,'=')`  
`FROM medewerkers`
  - ▣ `@@JACOBS`                      `JACOBS=====`
  
- `SELECT LPAD (maandsal,4) || ' ' || RPAD ('=',maandsal/100,'=')`  
`FROM medewerkers`
  - ▣ `1800 =====`                      *(18 =-tekens)*
  - ▣ `1300 =====`                      *(13 =-tekens)*
  - ▣ `4000 =====`
  - ▣ `2500 =====`

## 4.3.2 Tekstfuncties

- `SELECT TRANSLATE(code, 'AESOL', '12345') FROM cursussen`
    - ▣ `SQL → 3Q5`
  
  - `SELECT REPLACE (omschrijving, 'SQL', 'Visual C#') FROM cursussen`
    - ▣ `INTRODUCTIE SQL → INTRODUCTIE Visual C#`
-

## 4.3.3 Algemene functies

- **GREATEST(a,b,...)** grootste waarde uit de argumenten
  - ▣ `select GREATEST(5,7,2,9,0,-5,8) from dual; → 9`
  - ▣ `select GREATEST('karel', 'karen') from dual; → karen`
- **LEAST(a,b, ...)** kleinste waarde uit de argumenten
  - ▣ `select LEAST(5,7,2,9,0,-5,8) from dual; → -5`
  - ▣ `select LEAST(voorn, naam) from medewerkers;`
- **NULLIF(a,b)** geef NULL als a=b, anders a
  - ▣ `select NULLIF('aap', 'beer') from dual; → aap`
  - ▣ `select NULLIF(5,abs(-5)) from dual; → null`
- **NVL(x,y)** geeft y als x NULL is, anders x zelf
  - ▣ `select NVL(5,9) from dual; → 5`
  - ▣ `select NVL(null,9) from dual; → 9`

## 4.3.3 Algemene functies

- **NVL2(x,y,z)** geeft y als x niet NULL is, anders z
  - `select NVL2(7,9,8) from dual;` → 9
  - `select NVL2(null,9,8) from dual;` → 8
  - `select comm, nvl2(comm, 'wel comm', 'geen comm') from medewerkers;`
  
- **DECODE( x ,a1, b1 ,a2, b2 ... ,an, bn [,y] )** geeft : *b1* als *x = a1*  
*b2* als *x = a2*  
...  
*bn* als *x = an*  
en anders *y* (of default: NULL)
  - `select DECODE ('HELLO', 'HOI', 'Er staat Hoi', 'HELLO', 'Er staat Hello', 'BLA', 'erstaat bla') from dual;`  
→ Er staat Hello  
*select naam, decode(naam, 'CASPER', 'casperke', 'JACOBS', 'jacobke') from medewerkers;*
  - `select DECODE ('HOERA', 'HOI', 'Er staat Hoi', 'HELLO', 'Er staat Hello', 'Er staat iets anders') from dual;`  
→ Er staat iets anders  
*select naam, decode(naam, 'CASPER', 'casperke', 'JACOBS', 'jacobke', naam) from medewerkers;*

## 4.3.3 Algemene functies

- COALESCE(a,b,...)      Retourneer het eerste argument dat niet NULL is (en NULL als ze allemaal NULL zijn)
- ▣ SELECT COALESCE(NULL, NULL, 'element3', 'element4') from dual;  
→ *element3*

N.B. COALESCE kan je ook zien als een specifieke Case expression:

CASE

WHEN (expression1 is NOT NULL) THEN expression1

WHEN (expression2 is NOT NULL) THEN expression2

....

ELSE expression n

END

## 4.3.3 Algemene functies

- Bepaal het jaarlijks inkomen van elke medewerker

```
SELECT naam, maandsal * 12 + NVL(comm,0) as salaris  
FROM medewerkers
```

Of

```
SELECT naam, maandsal * 12 + NVL2(comm,comm,0) as salaris  
FROM medewerkers
```

Of

```
SELECT naam, maandsal * 12 + COALESCE(comm,0) as salaris  
FROM medewerkers
```

*Nvl: geeft y als x NULL is, anders x zelf*

*Nvl2: geeft y als x niet NULL is, anders z*

*COALESCE: retourneer het eerste argument dat niet NULL is*

## 4.3.3 Algemene functies

- Sorteer de medewerkers op basis van de functie (directeur, manager, verkoper, trainer, boekhouder)

```
SELECT naam, functie
FROM medewerkers
ORDER BY DECODE (functie , 'DIRECTEUR', 1
                  , 'MANAGER', 2
                  , 'VERKOPER', 3
                  , 'TRAINER', 4
                  , 5
                  )
```

! Hetzelfde datatypes gebruiken

```
SQL> SELECT naam, functie
2  FROM medewerkers
3  ORDER BY case functie
4             when 'DIRECTEUR' then 1
5             when 'MANAGER' then 2
6             when 'VERKOPER' then 3
7             when 'TRAINER' then 4
8             end;
```



## 4.3.4 Datumfuncties

- Vb. Geboren na 1980

```
SQL> SELECT naam, gbdatum  
2   FROM medewerkers  
3   WHERE gbdatum > DATE '1980-01-01';
```

- Conversie alfanum. string → vertrouwen op impliciete conversie door Oracle
- Slagen of falen hangt af van de NLS\_DATE\_FORMAT instellingen
- Datumformaat, taal en munteenheid kan via parameters van het National Language Support (NLS) aangepast worden
- `SELECT * FROM nls_session_parameters;`
  - alter session  
set nls\_date\_format='dd-mm-yyyy' of 'DD-MON-YYYY HH24:MI:SS'  
( 05-11-2017 05-NOV-2017 13:24:08)

## 4.3.4 Datumfuncties

### Tijdsgerelateerde constanten

- DATE 'yyyy-mm-dd'
- TIMESTAMP 'yyyy-mm-dd hh24:mi:ss.ff'
- INTERVAL 'expr' <qualifier>

```
SQL> select date '2019-11-23' from dual;  
  
DATE '2019-  
-----  
23-11-2019
```

### Voorbeelden

- DATE '2018-09-25'
- TIMESTAMP '2018-09-25 23:59:59.99999' AT TIME ZONE 'CET' (13u CET=11u UTC)
- INTERVAL '1' YEAR (zomer+2u /winter+1u op UTC- vroeger GMT)
- INTERVAL '1 2:3' DAY TO MINUTE -- 1 dag, 2 uur, 3 min
- INTERVAL '1-3' YEAR TO MONTH -- 1 jaar, 3 maanden
- SELECT interval '1' YEAR + gbdatum -- 1 jaar later / geboren voor 20/01/1984  
FROM medewerkers  
WHERE gbdatum < DATE '1984-01-20';

Nanoseconden (miljardste  
van een seconde)

## 4.3.4 Datumfuncties

### DATE voorbeeld

```
SQL> select date '2020-10-25' from dual;  
  
DATE '2020-  
-----  
25-10-2020
```

### INTERVAL voorbeeld (add 1 jaar en 3 maanden) bij)

```
SQL> SELECT gbdatum, interval '1-3' YEAR TO MONTH + gbdatum from medew  
  
GBDATUM      INTERVAL '1  
-----  
17-12-1985  17-03-1987  
20-02-1981  20-05-1982  
22-02-1982  22-05-1983  
02-04-1987  02-07-1988  
28-09-1976  28-12-1977  
01-11-1983  01-02-1985  
09-06-1985  09-09-1986  
26-11-1979  26-02-1981  
17-11-1972  17-02-1974  
28-09-1988  28-12-1989  
30-12-1986  30-03-1988  
03-12-1989  03-03-1991  
13-02-1979  13-05-1980  
23-01-1982  23-04-1983
```

### INTERVAL (add 1 dag, 2 uur en 3 minuten)

```
SQL> select systimestamp, INTERVAL '1 2:3' DAY TO MINUTE + systimestamp from dual;  
  
SYSTIMESTAMP                                INTERVAL '12:3'DAYTOMINUTE+SYSTIMESTAMP  
-----  
22/10/20 19:49:05,732000 +02:00              23/10/20 21:52:05,732000000 +02:00  
  
SQL>
```

# 4.3.4 Datumfuncties

- **ADD\_MONTHS(d,n)** *datum d plus n maanden*
  - `select sysdate, add_months(sysdate, 4) from dual; → 18-10-2020 / 18-02-2021`
- **MONTHS\_BETWEEN(d,e)** *maanden verschil tussen d en e*
  - `select sysdate, months_between(sysdate, date '2020-03-01') from dual; → 7,5628644`
- **LAST\_DAY(d)** *laatste dag van de maand waarin d valt*
  - `select sysdate, last_day(sysdate) from dual; → 18-10-2020 / 31-10-2020`
- **NEXT\_DAY(d,wkdag)** *de eerste weekdag (ma, di, ...) na d*
  - `select sysdate, next_day(sysdate, 'MAANDAG') from dual; → 18-10-2020 / 19-10-2020`
- **NEW\_TIME (d,z1,z2)** *converteer datum/tijd van tijdzone1 naar tijdzone2*
- **ROUND(d[,fmt])** *d afgerond op fmt (default middernacht)*
  - `select round(date '2020-03-18'), round(date '2020-07-18') from dual; → 18-03-2020 / 18-07-2020`
  - `select round(date '2020-03-18', 'YYYY'), round(date '2020-07-18', 'YEAR') from dual → 01-01-2020 / 01-01-2021`
- **TRUNC(d[,fmt])** *d afgekapt op fmt (default middernacht)*
  - `SELECT TRUNC(date '2020-10-27', 'YEAR') "New Year" FROM DUAL; → 01-01-2020 (truncate tot op het jaar)`
- **EXTRACT(c FROM d)** *extraheert component c uit expressie d*
  - `select extract (YEAR FROM DATE '2015-03-17') from dual; → 2015`
  - `select extract (DAY FROM DATE '2015-03-17') from dual; → 17`

## 4.3.4 Datumfuncties

- `SELECT naam,  
months_between(sysdate,gbdatum) FROM medewerkers`
  - ▣ `CLERCKX      388.273368`
  
- `SELECT gbdatum,  
add_months(gbdatum,13) ,  
add_months(gbdatum,-3) FROM medewerkers`
  - ▣ `17-DEC-1985 → 17-JAN-1987 17-SEP-1985`
  
- `SELECT add_months(date '2015-01-29',1) ,  
add_months(date '2016-01-29',1) FROM dual`
  - ▣ `28-FEB-2015`
  - ▣ `29-FEB-2016` *Schrikkeljaar*

## 4.3.4 Datumfuncties

- `SELECT sysdate, next_day(sysdate,'zat'), last_day(sysdate) FROM dual`
  - ▣ 25-10-2021 / 30-10-2021 / 31-10-2021
  
- `SELECT extract(year from gbdatum)  
          , extract(month from gbdatum)  
          , extract(day from gbdatum)  
          FROM medewerkers WHERE naam = 'DE KONING'`
  - ▣ 1972 / 11 / 17
  
- `SELECT SYSTIMESTAMP FROM DUAL;`
  - ▣ 18/10/20 11:00:47,257000 +02:00    *datum uur-min-sec-aantal sec-tijdzone UTC*

## 4.3.4 Datumfuncties

- `SELECT round(date '2018-04-26', 'MONTH'),  
trunc (date '2018-04-16', 'MONTH') FROM dual`
  - ▣ 01-05-2018 / 01-04-2018

- `SELECT round(date '2018-08-15', 'YEAR'),  
trunc (date '2018-08-15', 'YEAR') FROM dual`
  - ▣ 01-01-2019 / 01-01-2018

`SELECT round(date '2068-04-16', 'CC'),  
trunc (date '2068-04-16', 'CC') FROM dual`

- ▣ 01-01-2101 / 01-01-2001  
>2050 afronden naar 2101 / < 2050 afronden naar 2001

## 4.3.4 Datumfuncties

- Datumformaten (fmt) ondersteund door ROUND en TRUNC
  - CC,SCC eeuw, met of zonder minteken (BC)
  - [S]YYYY,[S]Year,YYY,YY,Y jaar (in alle gedaantes)
  - IYYY,IYY,IY,I ISO jaar (van maandag tot zondag)
  - Q kwartaal
  - MONTH,MON,MM,RM maand(voluit,afgekort,getal,Romeins getal)
  - IW,WW (ISO) weeknummer (weken volgens ISO jaar)
  - W dag van de week
  - DDD,DD,J dag(jaar/maand/Juliaans)
  - DAY,DY,D dichtstbijzijnde zondag
  - HH,HH12,HH24 uur
  - MI minuut



## 4.3.5 Conversiefuncties

- `TO_CHAR(n[,fmt])` zet getal *n* om naar een string
- `TO_CHAR(d[,fmt])` zet datum *d* om naar een string
- `TO_NUMBER(t)` zet string *t* om naar een getal
- `TO_DATE(t[,fmt])` zet *t* om naar een datum
  
- Opmerking:
  - ▣ DATE: op papier en scherm → enkel weergegeven als string
  - ▣ Invoer via toetsenbord → enkel invoer als string

## 4.3.5 Conversiefuncties

- SELECT sysdate

, to\_char(sysdate,'hh24:mi:ss')

, to\_char(DATE '2020-10-30', ' "valt op" Day')

FROM dual;

- ▣ 22-10-2020 / 20:09:35 / valt op Vrijdag

```
alter session
set  nls_date_format = "dd-mm-yyyy"
     nls_language=Dutch
```

- SELECT to\_char (sysdate,'yyyy')

, to\_char (sysdate, 'yy')

, to\_char (sysdate,'y')

, to\_char (sysdate,'year') FROM dual;

- ▣ 2019 / 19 / 9 /twenty nineteen → *geeft altijd in Engels ongeacht alter session*

- SELECT to\_char (sysdate,'Q') FROM dual; → *kwartaal*

- ▣ 4

## 4.3.5 Conversiefuncties

□ SELECT sysdate

, to\_char (sysdate,'mm')

, to\_char (sysdate,'month')

, to\_char (sysdate,'mon')

FROM dual

▣ 12-11-2019 / 11 / november / nov

□ SELECT to\_char (date '2019-11-13', 'ddd')

, to\_char (date '2019-11-13', 'dd')

, to\_char (date '2019-11-13', 'd')

, to\_char (date '2019-11-13', 'day')

, to\_char (date '2019-05-1', 'Dy')

, to\_char (date '2019-05-1', 'dy')

FROM dual;

→ nde dag v/h jaar

→ dagNr v/d maand

→ dagNr v/d week

→ dagnaam

→ dagnaam (afk.)

→ dagnaam (afk.)

▣ 317 / 13 / 3 / woensdag / Wo / wo

## 4.3.5 Conversiefuncties

- `SELECT to_char (sysdate, 'hh:mi:ss AM')`  
`, to_char (sysdate, 'hh24:mi:ss')`  
`, to_char (sysdate, 'sssss') FROM dual;`
  - ▣ 01:19:15 PM / 13:19:15 / 47955 (*sec na middernacht*)
- `SELECT to_char(1203920,'9G999G999D00L')` FROM dual;
  - ▣ 120392000 EUR

```
ALTER session
SET   nls_numeric_characters = ", "
      nls_currency=' EUR';
```
- `SELECT to_char(sysdate, 'Day: Dd Month yyyy')` from dual;
  - ▣ Dinsdag: 12 November 2019

## 4.3.5 Conversiefuncties

- *Vraag de weekday waarop je geboren bent.*
  - ▣ `select decode (to_char(to_date('17 02 1977','dd mm yyyy'),'d')  
                  , '7', 'zondag'  
                  , '1', 'maandag'  
                  , '2', 'dinsdag'  
                  , '3', 'woensdag'  
                  , '4', 'donderdag'  
                  , '5', 'vrijdag'  
                  , '6', 'zaterdag') geboortedag`

`FROM dual;`

→ *donderdag*

### 4.3.5 Conversiefuncties

## □ Formaten voor conversiefuncties to\_char en to\_date

- |   |             |                                               |                                                                                    |
|---|-------------|-----------------------------------------------|------------------------------------------------------------------------------------|
| □ | [S]CC       | eeuw, S voor het minteken (BC)                | <i>select to_char(to_date('-4711/01/01','yyyy/mm/dd'), 'scc') from dual; → -48</i> |
| □ | [S]YYYY     | jaar, met of zonder minteken                  |                                                                                    |
| □ | [S]Year     | jaartal uitgespeld, met minteken (S)          | <i>select to_char(date '2019-05-21', 'year') from dual; → twenty nineteen</i>      |
| □ | YYY,YY,Y    | jaar (laatste 3,2 of 1 getal)                 |                                                                                    |
| □ | BC,AD       | BC/AD indicator                               |                                                                                    |
| □ | Q           | kwartaal(1,2,3,4)                             |                                                                                    |
| □ | MM          | maand (01–12)                                 |                                                                                    |
| □ | MONTH       | maandnaam, met spaties uitgevuld tot lengte 9 |                                                                                    |
| □ | MON         | maand(afgekort)                               |                                                                                    |
| □ | IW,WW (ISO) | weeknummer (01-52)                            |                                                                                    |
| □ | W           | weeknummer van de maand (1-5)                 |                                                                                    |

## 4.3.5 Conversiefuncties

- DDD dagnummer van het jaar (1-366)
- DD dagnummer van de maand (1-31)
- D dagnummer van de week (1-7)
- J Juliaans datum, dagnummer sinds 01/01/4712 BC
- DAY dagnaam, met spaties uitgevuld tot lengte 9
- DY afkorting van de dag
- AM,PM AM/PM indicator (Ante meridiem/Post meridiem)
- HH[1 2] uur van de dag (01-12)
- HH24 uur van de dag (00-23)
- MI minuut (00-59)
- SS seconden
- SSSSS seconden na middernacht (0-86399)
- /., deze leestekens letterlijk in de datum
- "... " string wordt eveneens weergegeven

## 4.3.5 Conversiefuncties

- Verder zijn er nog enkele toevoegingen mogelijk

- TH                      ordinaal getal (4th)

*select to\_char(date '2019-05-21', 'ddth') from dual; → 21st*

- SP                      uitgespeld getal (four)

- THSP, SPTH           uitgespeld ordinaalgetal (fourth)

- FM                      voornoopnullen en spaties onderdrukken – met FM  
fillmechanisme in- en uitschakelen

- `SELECT to_char(sysdate, 'fmDay, ddth "of" fmmonth yyyysp') from dual;`  
**Zondag, 24th of oktober two thousand twenty-one**

- `SELECT to_char(sysdate, 'fmDay: Dd Month yyyy') from dual;`  
**Zondag: 24 Oktober 2021**

---