

Samenvatting Data

Hoofdstuk 1: Inleiding van basisbegrippen

Wat is een databank?

Een digital opgeslagen archief, ingericht met oog op het makkelijk terugvinden van informatie, en een flexibel gebruik.

Aan welke voorwaarden moet de databank voldoen om echt te kunne spreken van een databank?

- Gegevens moeten eenvoudig worden opgeslagen
- Gegevens moeten eenvoudig worden op of doorgezocht
- Gegevens moeten gewijzigd kunnen worden
- Gegevens moeten verwijderd kunnen worden (zonder dat dit de werking van de db beïnvloed)

Wat is een index?

Een index is de data die uniek is en er voor zorgt dat alle gegevens optimaal doorzoekbaar zijn.

Wat is een datamodel?

Hetgene dat beschrijft hoe de gegevens in een database of informatiesysteem gestructureerd zijn.

Welke soorten datamodellen ken je?

- Platte datamodel (1^{ste} generatie)
- Hierargische model (1^{ste} generatie)
- Het netwerkmodel (2^{de} generatie)
- Rationele databank (3^{de} generatie)
 - Deze gebruiken wij. Het is een verzameling van tabellen die relaties hebben met elkaar. Het verband tussen de tabellen wordt niet expliciet gedefinieerd maar aangeduid door eigenschappnen tussen verschillende tabellen.
- Objectgeoriënteerde databank (4^{de} generatie)

Wat is het verschil tussen informatie en data?

Informatie	Data
Gestructureerde en betekenisvolle relatie met data	Betekenisloze gegevens zonder informatie

Uit welke componenten staat een gegevensverzameling?

- Groep (Tabel)
- Gegevens (Kolommen in de tabel)
- Waarden (De data die in de kolommen staat)
- Occurence/Voorkomen (1 rij in de tabel)

Hoofdstuk 2: Structured Query Language

2.1 Kennismaking met SQL

Structured Query Language SQL

Is een query taal om een groep gegevens af te halen of te wijzigen. Deze wordt toegepast op 2 manieren. Interactief (sql commando rechtstreeks ingeven en het resultaat krijgen) en embedded (sql commando binnen een programmeertaal)

De onderverdelingen in SQL:

- Gegevensdefinitie (DDL -> Data Definition Language)
 - Tabel maken, wijzigen of verwijderen (Create table, Alter table, Drop table)
- Gegevensmanipulatie (DML -> Data Manipulation Language)
 - Groepen rijen invoegen, bijwerken en verwijderen (Delete, insert, update, commit, rollback)
- Raadpleging of Query
 - Raadplegingen van databasegegevens (selectiequery) op tabelniveau -> select
- Beveiliging of DCL (Data Control Language)
 - Toegang tot de database met een wachtwoord
 - Gevolg -> Privileges (toegang tot gegevens in detail regelen)
 - Objectprivileges -> recht om een bepaald object te benadeen bv insert, select, delete, update table
 - Systeemprivileges -> Recht om bepaalde acties te ondernemen bv aanloggen, tabel maken,
 - Create user, alter user, drop user, grant(privileges regelen) en revoke (privileges ontnemen)

2.2 Basisbegrippen

De verschillende basisbegrippen

Constante	Numeriek, alfanumeriek, datum
Variabelen	Kolomvariabelen en systeemvariabelen
Operatoren	Rekenkundige operatoren (+, -, *, /)
Alfanumerieke operator	-> bv 'Patricia' 'Briers' -> 'Patricia Briers'
Vergelijkingsoperatoren	<, >, =, <=, >=, !=
Logische operatoren	AND, OR, NOT

De voorrangsregel bij de operatoren:

Rekenkundig > Vergelijking > Logisch

Naamgeving van tabellen en kolommen:

Beginnen met een letter, max 30 char, geen verschil tussen kleine en hoofdletters

/* commentaar */ -- verhelderend voorbeeld

Hoofdstuk 3: SQL* Plus

3.2 De SQL-Buffer

Deze bevat de laatst ingegeven SQL-Commando.

Select * from cat; -> uitgevoerd en in de buffer

Select * from cat -> zit in de buffer en wordt nog niet uitgevoerd

Een commando in de buffer uitvoeren doe je met /

3.4 SQL* Plus – Editor

Dit is een regeeditor, die 1 regel actueel wijzigd gemarkeerd met *

De SQL*Plus commandos :

- A(ppend) name -> toevoegen aan de huidige regel
- C(hange) /van /naar -> wijzigen
- Del -> Verwijdert alle regels
- Del 1 -> verwijdert regel 1
- Del 1 5 -> verwijdert regel 1 tot en met 5
- I(nput) -> toevoegen van een nieuwe regel
- L(ist) -> Geeft alle regels in de buffer
- L 5 of 5 -> geeft regel 5
- L5 10 -> geeft regel 5 tot en met 10
- L* Geeft de huidige regels

3.5 Commando's bewaren

Inhoud van de buffer bewaren in een txt bestand kan je op verschillende manieren

- Via buffer (afiedt.buf)
- Via Commando (Save met optie append /replace bv save "c:\oracle\oefening append")
 - Terug ophalen in buffer -> get "c:\oracle\oefening"
 - Onmiddellijk uitvoeren -> start "c:\oracle\oefening"
 - Commando "Spool" -> Volledige schermuitvoer bewaren -> spool test -> get test.lst opent de opgeslagen scherm.

3.6 SQL*Plus – Instellingen

Instellingen voor de omgeving te wijzigen

- | | |
|--------------------|---|
| ▣ Set pagesize 100 | doseert uitvoer naar beeldscherm per pagina |
| ▣ Set numwidth 10 | standaardbreedte voor kolommen |
| ▣ Set linesize 120 | breedte van uitvoer |
| ▣ Set space 2 | aantal spaties tussen de kolommen |
| ▣ Set sqlprompt | oef: |
| ▣ Set pause on | dosering per pagina |
| ▣ Set pause | "Druk op een toets..." |
| ▣ Show pause | geeft instelling van pause |
| ▣ Show all | geeft alle instellingen |

Naast SET-commando ook ALTER SESSION

bv: NLS (National Language Support) parameters regelen

■ Alter session

set nls_date_format = "dd-mm-yyyy"	standaard datumformaat
nls_language=Dutch	taal voor boodschappen
nls_currency='EUR'	symbool voor geldbedrag
nls_numeric_characters=', '	' ' decimale scheiding en ' ' scheiding duizendtal

■ Aangepaste wijzigingen: SQL> start login.sql

-- Datumformaat wijzigen

SQL> alter session

SQL> set nls_date_format = "dd-mm-yyyy"

■ Opvragen: SQL> select * from V\$NLS_PARAMETERS

3.7 Overige SQL*Plus – Commando's

Column

■ Indien lay-out niet aangepast → kolomkoppen beetje versmallen/vergroten

- Col omschrijving format a32 32 karakters breed
- Col toeilage format 999.99 3 cijfers.2cijfers
- Col naam justify L(ef), C(enter), R(ight) uitlijning kolomkop
- Col afd format 999 heading afdeling benaming kolom
- Col naam clear verwijder kolominstellingen
- Col comm NULL "Onbekend" Onbekend ipv nulpositie

Describe

- Overzicht van de tabelstructuur
- Bv: desc medewerkers

Buffer en scherm leegmaken

- Cl(ear) scr(een)
- Cl(ear) buffer

Hoofdstuk 4: Raadpleging

4.1 Het select component

Werk in vorig voorbeeld de tabulators weg

```
SELECT naam || ' is werkzaam in afdeling ' || afd  
FROM medewerkers;
```

Geef de huidige datum

```
SELECT sysdate  
FROM dual
```

```
SELECT voorn, naam  
FROM medewerkers  
WHERE afd between 20 AND 30
```

```
SELECT voorn, naam  
FROM medewerkers  
WHERE naam like '%A%'
```

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
JACOBS	30
BRIERS	20
BRIERS	30

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

```
SELECT voorn, naam, gbdatum  
FROM medewerkers  
WHERE gbdatum > DATE '1983-05-01';
```

Geef de medewerkers van afdeling 10 of 30

```
SELECT voorn, naam  
FROM medewerkers  
WHERE afd in (30,10)
```

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
```

Volgorde in functie van kolomnummer 2

```
SELECT *  
FROM medewerkers  
ORDER BY 2
```

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	BEGIN DATUM	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.

```
CASE expressie  
WHEN waarde THEN resultaat  
[ELSE] resultaat  
END
```

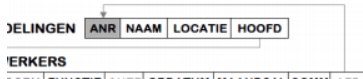
NVL: flexibel met NULL-waarde omgaan

- `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 Subqueries

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

```
SELECT voorn, naam
FROM medewerkers
WHERE mnr IN ( SELECT distinct cursist
               FROM inschrijvingen
               WHERE cursus IN ( SELECT code
                                FROM cursussen
                                WHERE type = 'BLD'))
```



4.3 Functies

4.3.1 Rekenfuncties

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

```
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,...)

4.3.2 Tekstfuncties

LENGTH(t)	aantal karakters (lengte) van <i>t</i> <i>select LENGTH('hi all') from dual; → 6</i>	REPLACE(t,v)	verwijdert uit <i>t</i> elk voorkomen van <i>v</i> (woorden) <i>select replace('hello world', 'l') from dual; → heo word</i>
ASCII(t)	ascii-waarde eerste karakter van <i>t</i>	REPLACE(t,v,w)	vervangt in <i>t</i> elk voorkomen van string <i>v</i> in <i>w</i> (woorden) <i>select replace('hello world', 'l', '*') from dual;</i> <i>→ he**o wor*d</i>
CHR(n)	karakter met ascii-waarde <i>n</i>	TRANSLATE(t,v,w)	vervangt alle karakters uit <i>v</i> die in <i>t</i> voorkomen door het corresponderende karakter uit <i>w</i> <i>select translate('veer', 've', 'bo') from dual;</i> <i>→ boor</i>
UPPER(t)	<i>t</i> in hoofdletters <i>select UPPER('Hi all') from dual; → HI ALL</i>	CONCAT(t1,t2)	voegt <i>t1</i> en <i>t2</i> samen (equivalent met) <i>select concat('boor', 'machine') from dual;</i> <i>→ boormachine</i>
LOWER(t)	<i>t</i> in kleine letters <i>select UPPER('Hi ALL') from dual; → hi all</i>		
INITCAP(t)	elke woord in <i>t</i> met beginhoofdletter <i>select INITCAP('Hi jean-marie') from dual → Hi Jean-Marie</i>		
LTRIM(t,k)	verwijdert links van <i>t</i> de <i>k</i> -karakters, t.e.m. het eerste karakter niet in <i>k</i> <i>stopt als het teken niet gevonden is: ltrim('marteress', 'ram') → teress</i>		
RTRIM(t,k)	verwijdert rechts van <i>t</i> de <i>k</i> -karakters, na het laatste karakter niet in <i>k</i> <i>stopt als een teken niet gevonden is: rtrim('marteress', 'sen') → marter</i>		
LPAD(t,n)	vult <i>t</i> links uit met spaties tot lengte <i>n</i>		
LPAD(t,n,k)	idem, met <i>k</i> -karakter <i>select lpad('*', 10, '-') from dual; → -----*</i>		
RPAD(t,n)	vult <i>t</i> rechts aan met spaties tot lengte <i>n</i>		
RPAD(t,n,k)	idem, met <i>k</i> -karakter <i>select rpad('*', 10, '-') from dual; → *-----</i>		
SUBSTR(t,n)	geeft deel van <i>t</i> vanaf positie <i>n</i> tot het einde <i>select substr('kelder', 2) from dual; → elder</i>		
SUBSTR(t,n,m)	geeft deel van <i>t</i> vanaf positie <i>n</i> , <i>m</i> -karakters lang <i>select substr('kelder', 2, 3) from dual; → eld</i>		
INSTR(t,k)	positie eerste voorkomen van <i>k</i> in <i>t</i> <i>select instr('kelder', 'e') from dual; → 2</i>		
INSTR(t,k,n)	idem, op of na de <i>n</i> -de positie in <i>t</i> <i>select instr('kelder', 'e', 3) from dual; → 5</i>		
INSTR(t,k,n,m)	het <i>m</i> -de voorkomen van <i>k</i> , vanaf de <i>n</i> -de positie in <i>t</i> <i>select instr('kelderen', 'e', 3, 2) from dual; → 7</i>		

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

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

DECODE(x	,a1, b1	geeft : b1 als x = a1
	,a2, b2	b2 als x = a2

	,an, bn	bn als x = an
	[,y])	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 DECODE ('HOERA', 'HOI', 'Er staat Hoi', 'HELLO', 'Er staat Hello', 'Er staat iets anders') from dual;
→ Er staat iets anders

COALESCE(a,b,...) Retourneer het eerste argument dat niet NULL is (en NULL als ze allemaal NULL zijn)

- ❑ SELECT COALESCE(NULL, NULL, 'element3', 'element4');
→ 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

4.3.4 Datumfuncties

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)

```
SQL> SELECT gbdatum, interval '1-3' YEAR TO MONTH + gbdatum from medev
```

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

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

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 <i>n</i> om naar een string
TO_CHAR(d[,fmt])	zet datum <i>d</i> om naar een string
TO_NUMBER(t)	zet string <i>t</i> om naar een getal
TO_DATE(t[,fmt])	zet <i>t</i> om naar een datum

Opmerking:

- DATE: op papier en scherm → enkel weergegeven als string
- Invoer via toetsenbord → enkel invoer als string

Formaten voor conversiefuncties to_char en to_date

□	[S]CC	eeuw, S voor het minteken (BC) <small><i>select to_char(to_date('-4711/01/01','syyy/mm/dd'),'scc') from dual; → -48</i></small>
□	[S]YYYY	jaar, met of zonder minteken
□	[S]Year	jaartal uitgespeld, met minteken (S) <small><i>select to_char(date '2019-05-21','year') from dual; → twenty nineteen</i></small>
□	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)

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[12]	uur van de dag (01-12)
HH24	uur van de dag (00-23)
MI	minuut (00-59)
SS	seconden
SSSS	seconden na middernacht (0-86399)
/.,	deze leestekens letterlijk in de datum
"..."	string wordt eveneens weergegeven

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 fill mode (voornoopnullen en spaties
 onderdrukken – met FM fillmechanisme in- en
 uitschakelen)

SELECT to_char(sysdate,'fmDay, ddth "of" fmmonth yyyysp') from dual;

Saturday, 13th of may two thousand seventeen

SELECT to_char(sysdate, 'fmDay: Dd Month yyyy') from dual;

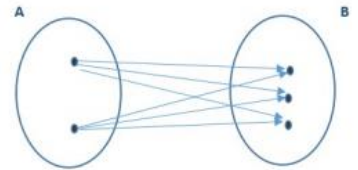
Zondag: 20 Januari 2019

5 Raadpleging Deel 2

5.1 De Join

5.1.1 De Equi-join (Inner join)

Als je deze join doet ga je meerdere tabellen aanspreken na de "FROM"
Select m.naam, a.naam as afdeling from medewerkers m, afdelingen a;



5.1.2 De auto-join

Een tabel met zichzelf joinen (er is niet persee een fel verschil met de innerjoin)

Select m.naam as medewerker, c.naam as chef
from medewerkers m, medewerkers c where m.chef = c.mnr;

5.1.3 De outer-join

Joint een andere tabel op de eerste tabel.

+ → Zorgt ervoor dat de gegevens die een nullwaarde in de gejoinde tabel hebben, ook getoond worden

5.1.4 SQL-Standaard: Inner join (of gewoon join)

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

5.1.4.1 Right outer join (of right join)

Geeft alle rijen van de rechtertabel inclusief de rijen waar de initiele tabel geen waardes voor heeft

5.1.4.2 Left outer join (of left join)

Geeft alle rijen van de linkertabel inclusief de rijen waar de initiele tabel geen waardes voor heeft

5.1.4.3 Full outer join (of full join)

Geeft rijen van beiden tabellen

5.1.4.4 Using component

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

```
SELECT      i.cursist, u.locatie, cursus (! gaat niet: i.cursus en u.cursus in USING)
FROM        inschrijvingen i
            JOIN uitvoeringen u
            USING (cursus,begindatum);
```

5.4.5 Break

Hiermee kunnen we de kolomwaarden onderdrukken en een regel overslaan of toevoegen

Break on afd skip 1 (skip 1 lijn op elke afdeling)

Break on afd page set (streepjes bij elke nieuwe afdeling)

pause on (nieuwe pagina bij elke afdeling)

clear breaks (breaks weg doen)

5.2 Groepsfuncties

5.2.1 Group by-component

Groepeert de table aande hand van de parameter die mee is gegeven.

```
SELECT
to_char (gbdatum, 'YYYY') geboortejaar,
count(mnr) aantalMedewerkersInJaar
FROM medewerkers
GROUP BY to_char(gbdatum, 'YYYY')
```

Hierboven worden alle geboortejaren geselecteerd samen met het aantal werknemers dat in dit jaar geboren werd

5.2.2 Groepsfuncties

COUNT ()	geeft aantal waarden	alle datatypes
-----------	----------------------	----------------

▣ *Select count(mnr) from medewerkers; → 14*

SUM ()	som van de waarden	numeriek
---------	--------------------	----------

▣ *Select sum(mnr) from medewerkers; → 108172*

AVG ()	gemiddelde waarden	numeriek
---------	--------------------	----------

▣ *Select avg(mnr) from medewerkers; → 7726,57143 (=108172 / 14)*

MIN ()	minimumwaarde	alle datatypes
---------	---------------	----------------

▣ *Select min(m.gbdatum) from medewerkers m; → 17/11/1972*

MAX ()	maximumwaarde	alle datatypes
---------	---------------	----------------

▣ *Select max(m.gbdatum) from medewerkers m; → 03/12/1989*

STDEV ()	standaarddeviatie	numeriek ⁽¹⁾
-----------	-------------------	-------------------------

VARIANCE ()	variantie	numeriek ⁽²⁾
--------------	-----------	-------------------------

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

5.2.3 Having component

Het verschil met where:

Where → op niveau van rijen

Having → op niveau van de geselecteerde waarden

5.2.4 Break and Compute commando

Break on afd skip 1

COMPUTE number label "AANTAL MEDEWERKERS" of functie on afd

Er wordt een total aantal medewerkers berekend van het aantal functies binnen elke afdeling

De COMPUTE clause wordt samen met de select statement gebruikt om ee samengevatte data te berekenen van het resultaat van een groep data.

De toegestane functies van COMPUTE zijn:

AVG	gemiddelde
COUNT	het aantal not null-waarden in een kolom
MAX	maximum
MIN	minimum
NUMBER	aantal rijen
STD	standaardafwijking
SUM	som
VAR	variantie

Compute-instellingen verwijderen.

SQL> Clear Computes

Met rollup(...,...) kan je ook per groep een overzicht krijgen

SQL> break on afd skip 1

SQL> select afd, functie
, count (mnr) aantal
from medewerkers
group by ROLLUP(afd, functie);

AFD	FUNCTIE	AANTAL
10	MANAGER	1
	DIRECTEUR	1
	BOEKHOUDER	1
		3
20	MANAGER	1
	TRAINER	4
		5
30	MANAGER	1
	VERKOPER	4
	BOEKHOUDER	1
		6
		14

5.3 Gecorreleerde subqueries

Alle medewerkers met hoger salaris dan het gemiddelde salaris van hun afdeling (cfr oef 11)

```
SELECT m.naam, m.voorn, m.maandsal
FROM medewerkers m
WHERE m.maandsal > (SELECT AVG(n.maandsal)
                     FROM medewerkers n
                     WHERE n.afd = m.afd)
```

NAAM	VOORN	MAANDSAL
JACOBS	EMMA	4975
BRIERS	ANDREA	5850
SWINNEN	CHRIS	4000
DE KONING	LIEVE	7000
DE COOMAN	DORIEN	4000

```
SELECT m.naam, m.maandsal, mm.gem
FROM medewerkers m
INNER JOIN
  (SELECT afd, AVG(maandsal) gem
   FROM medewerkers
   GROUP BY afd) mm
ON mm.afd = m.afd
AND m.maandsal > mm.gem
```

```
SELECT m.naam, m.maandsal, mm.gem
FROM medewerkers m
, (SELECT afd, AVG(n.maandsal) gem
   FROM medewerkers
   GROUP BY afd) mm
WHERE m.afd = mm.afd
AND m.maandsal > mm.gem
```

□ Wie is de 4^{de} jongste medewerker

```
SELECT m.naam, m.voorn, m.gbdatum
FROM medewerkers m JOIN medewerkers mm ON m.gbdatum <= mm.gbdatum
GROUP BY m.naam, m.voorn, m.gbdatum HAVING count(m.mnr) = 4;
```

Medewerkers m

NAAM	GBDATUM
LIEVE DE KONING	17-11-1972
RAF MARTENS	28-09-1976
DORIEN DE COOMAN	13-02-1979
CHRIS SWINNEN	26-11-1979
NELE ALLARD	20-02-1981
SVEN WOUTERS	23-01-1982
THOMAS DEFOUR	22-02-1982
ANDREA BRIERS	01-11-1983
AN CLERCKX	09-06-1985
JANA CASPERS	17-12-1985
TOM SLECHTEN	30-12-1986
EMMA JACOBS	02-04-1987
JOACHIM DEN RUYTER	28-09-1988
SIMON JACOBS	03-12-1989

<=

Medewerkers mm

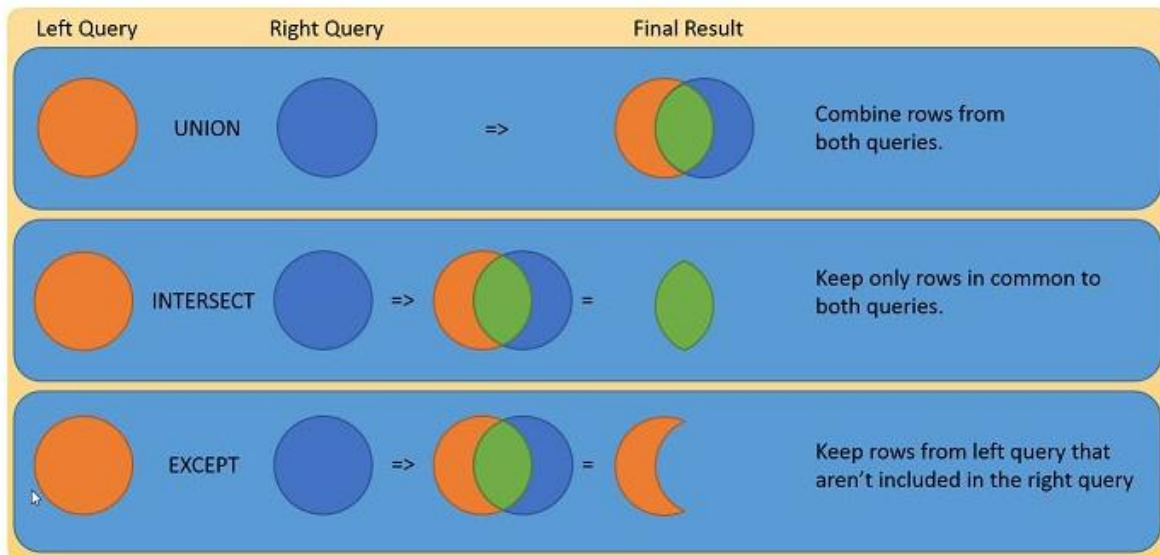
NAAM	GBDATUM
LIEVE DE KONING	17-11-1972
RAF MARTENS	28-09-1976
DORIEN DE COOMAN	13-02-1979
CHRIS SWINNEN	26-11-1979
NELE ALLARD	20-02-1981
SVEN WOUTERS	23-01-1982
THOMAS DEFOUR	22-02-1982
ANDREA BRIERS	01-11-1983
AN CLERCKX	09-06-1985
JANA CASPERS	17-12-1985
TOM SLECHTEN	30-12-1986
EMMA JACOBS	02-04-1987
JOACHIM DEN RUYTER	28-09-1988
SIMON JACOBS	03-12-1989

→ 4
→ 3
→ 2
→ 1

5.4 Verzalemingoperatoren

De 3 verzamelingsoperatoren:

- Union
- Minus
- Intersect



Het verschil tussen union en union all

Union → combineert de rijen van beide queries, en voert automatisch een distinct uit

Union all → Geeft ook de dubbele waarden

Belangrijk:

- De resultaatstabel neemt de aliassen over van de Q1
- Q1 en Q2 kunnen aparte voorwaarden hebben en deze zet je voor of na de Q (bv select .. where .. intersect select ... where)
- De select parameter mag maar 1 resultaat column hebben
- De datatypes van beide selects moeten hetzelfde zijn
- Verzamelingsoperatoren werken NIET met order by in het eerste deel
- De order by telt voor het hele geheel en wordt op het einde van de query gezet, je gebruikt de waarde van de eerste select

5.5 Views (table maken, opgesteld uit andere tables via een join)

```
CREATE view myFirstView as
  select m.naam,a.naam as afdeling, a.locatie,
  h.naam hoofd
  from medewerkers m
  INNER JOIN afdelingen a
  ON a.anr = m.afd
  INNER JOIN medewerkers h
  ON h.mnr= a.hoofd
```

View created.

select * from tab; -> laat je alle views (tables) zien
desc MYFIRSTVIEW. -> laat alle kolomnamen zien, descending by name(??)

Views in de datadictionary

```
SQL> set long 999 -- LONG kolommen op 999 karakters zetten
SQL> col text format a40 word_wrapped -- tekstterugloop
regelen
```

```
select view_name, text
from user_views
where view_name = 'MYFIRSTVIEW'
```

VIEW_NAME	TEXT
MYFIRSTVIEW	select m.naam,a.naam as afdeling, a.locatie, h.naam hoofd from medewerkers m JOIN afdelingen a ON a.anr = m.afd JOIN medewerkers h ON h. mnr= a.hoofd

Views met select *

```
create view afd20 as
select *
from medewerkers
where afd=20
```

```
SQL> select view_name, text from user_views;
```

VIEW_NAME	TEXT
AFD20	select "MNR","NAAM","VOORN","FUNCTIE","CHEF","G BDATUM","MAANDSAL","COMM","AFD" from medewerkers where afd=20

```
SQL> select * from afd20;
```

MNR	NAAM	VOORN	FUNCTIE	CHEF	GBDATUM	MAANDSAL	COMM	AFD
7369	CASPERS	JANA	TRAINER	7902	17-12-1985	1800		20
7566	JACOBS	EMMA	MANAGER	7839	02-04-1987	4975		20
7788	SWINNEN	CHRIS	TRAINER	7566	26-11-1979	4000		20
7876	SLECHTEN	TOM	TRAINER	7788	30-12-1986	2700		20
7902	DE COOMAN	DORIEN	TRAINER	7566	13-02-1979	4000		20

Drop view afd20 -> verwijder view

Save common queries in a view ->

```
create view VW_Provincielimburg =
select gemeente from provinces where provinviennaam = 'Limburg';
```

Hoofdstuk 6: Data Manipulation Language

Dit is een taal die gebruikt wordt om datamanipulatie te doen vanuit applicaties zoals forms.

6.1 Gegevens invoeren

```
INSERT INTO tabelnaam [(kolomnaam,...)]  
VALUES (expressie,...)
```

Values -> nieuwe values

```
INSERT INTO tabelnaam [(kolomnaam,...)]  
SELECT [kolomnaam,...]  
FROM tabelnaam
```

Vanuit een bestaande tabel
copieeren

Fysieke kolomvolgorde

```
INSERT INTO medewerkers  
VALUES ('NIJS', 7955, 'PETER', 'TRAINER', 7566, date '1967-02-16', 2660, NULL,  
20);
```

Willekeurige volgorde (onvolledig)

```
INSERT INTO medewerkers (naam,voorn,mnr,maandsal,gbdatum)  
VALUES ('VOS', 'GEERT', 7956, 2660, date '1963-05-22');
```

Op basis van een query worden 4 rijen toegevoegd

```
INSERT INTO schalen  
SELECT snr+5, ondergrens + 2300, bovengrens + 2300, 500  
FROM schalen  
WHERE snr <= 4;
```

SNR	ONDERGRENS	BOVENGRENS	TOELAGE
1	1700	2200	0
2	2201	2400	50
3	2401	4000	100
4	4001	5000	200
5	5001	9999	500
6	4000	4500	500
7	4501	4700	500
8	4701	6300	500
9	6301	7300	500

SNR	ONDERGRENS	BOVENGRENS	TOELAGE
1	1700	2200	0
2	2201	2400	50
3	2401	4000	100
4	4001	5000	200
5	5001	9999	500

Rijen 6 tem 9 worden toegevoegd

6.2 Gegevens wijzigen

```
UPDATE medewerkers  
SET functie = 'TRAINER'  
, maandsal = 5030  
, afd = 60  
WHERE mnr = 7666
```

```
UPDATE medewerkers  
SET maandsal = maandsal * 1.1;
```

```
UPDATE medewerkers  
SET maandsal = maandsal * 1.1  
WHERE afd = ( SELECT anr  
              FROM afdelingen  
              WHERE naam = 'HOOFDKANTOOR');
```

```
UPDATE tabelnaam  
SET kolomnaam = waarde, ....  
WHERE voorwaarde
```

6.3 Gegevens verwijderen

```
DELETE FROM medewerkers  
WHERE mnr = 7499;
```

6.4 Transactieverwerking

Wordt gebruikt om veranderingen te beheren. Na een edit, alter, delete, worden deze vaak gebruikt om een veilige manier van datamanipulatie te doen

Commit → Veranderingen toepassen

Rollback → wijzigingen ongedaan maken

Savepoint → markeringspunten maken tussen verschillende aanpassingen

DML-commando 1

Commit

mutatie 1 wordt definitief

DML-commando 2

DML-commando 3

Savepoint EEN

DML-commando 4

DML-commando 5

➤ **Savepoint TWEE**

DML-commando 6

DML-commando 7

– **Rollback to savepoint TWEE**

mutaties 6 en 7 worden ongedaan gemaakt

Commit

mutaties 2, 3, 4 en 5 worden definitief

Data	Opgeslagen gegevens
Datamodel	De wijze waarop de gegevens opgeslagen zijn
Database Management Systeem (DBMS)	De software waar de db mee aangemaakt of benaderd wordt
Gegevensdictionary	Dit bevat hoe gegevens in een informatiesysteem gestructureerd zijn. Het geeft info over de gegevens.