**4.4.1 Select-operator: selecteren van rijen**

1. **Geef het nummer en de naam van de medewerkers.**

select mnr, naam

from medewerkers;

1. **In welke locaties zijn er afdelingen gevestigd?**

select locatie

from afdelingen;

1. **Geef de code en de omschrijving van elke cursus.**

select code, omschrijving

from cursussen;

1. **Geef de naam en functie van alle medewerkers die verkoper zijn.**

select naam, functie

from medewerkers

where functie = 'VERKOPER';

1. **Geef de naam en functie van alle medewerkers die geen boekhouders zijn.**

select naam, functie

from medewerkers

where functie != 'BOEKHOUDER';

select naam, functie

from medewerkers

where not functie = 'BOEKHOUDER';

select naam, functie

from medewerkers

where functie <> 'BOEKHOUDER';

1. **Geef alle medewerkers die meer dan € 5000 verdienen en geen manager zijn.**

select \*

from medewerkers

where maandsal > 5000 and functie != 'MANAGER';

1. **Welke medewerkers verdienen minder dan € 2500?**

select \*

from medewerkers

where maandsal < 2500;

1. **Geef de naam van de afdeling waarvan medewerker 7698 de baas is.**

select naam

from afdelingen

where hoofd = 7698;

1. **Geef de verschillende functies van de medewerkers in de onderneming.**

select distinct functie

from medewerkers;

1. **Geef het nummer en de naam van elke medewerker die trainer is.**

select mnr, naam

from medewerkers

where functie = 'TRAINER';

1. **Geef de naam, de functie en het salaris van elke medewerker die in afdeling 30 werkt en meer dan € 2500 verdient.**

select naam, functie, maandsal

from medewerkers

where afd = 30 and maandsal > 2500;

1. **Geef de locatie van de afdeling die door medewerker 7566 geleid wordt?**

select locatie

from afdelingen

where hoofd = 7566;

1. **Geef de namen van de medewerkers die in afdeling 10 of 20 werken.**

select naam

from medewerkers

where afd in (10, 20);

select naam

from medewerkers

where afd = 10 or afd = 20;

1. **Geef van elke medewerker de naam en jaarsalaris (incl. commissie).**

select naam, maandsal\*12 + nvl(comm, 0) as jaarsalaris

from medewerkers;

1. **Geef de namen van de medewerkers wiens naam eindigt op 'N'.**

select naam

from medewerkers

where naam like '%N';

1. **Geef de namen van alle medewerkers waarin een dubbele 'o' voorkomt.**

select naam

from medewerkers

where naam like '%OO%';

1. **Alle medewerkers krijgen een opslag van 10%. Geef een lijst waarop de naam van de medewerkers staat, zijn huidig salaris en zijn toekomstig salaris. Zorg voor verzorgde attribuutkoppen in een realistisch formaat!**

select naam, maandsal, maandsal \* 1.1 as "TOEKOMSTIG SALARIS"

from medewerkers;

1. **Geef de namen en het salaris van de medewerkers van groot naar klein gesorteerd op** **salaris.**

select naam, maandsal

from medewerkers

order by maandsal desc;

1. **Geef de namen en het salaris van alle medewerkers die tussen de € 2000 en € 4000** **verdienen, maar niet gelijk is aan € 3850, noch aan € 2600. Druk het resultaat volgens dalend salaris af.**

select naam, maandsal

from medewerkers

where maandsal between 2000 and 4000

and maandsal not in (2600, 3850)

order by maandsal desc;

select naam, maandsal

from medewerkers

where maandsal between 2000 and 4000

and maandsal != 3850 and maandsal != 2600

order by maandsal desc;

select naam, maandsal

from medewerkers

where maandsal >= 2000 and maandsal <= 4000

and maandsal != 3850 and maandsal != 2600

order by maandsal desc;

1. **Geef de namen en het salaris van de medewerkers, gesorteerd op functie en binnen de** **functie op naam.**

select naam, functie

from medewerkers

order by functie, naam;

1. **Bepaal het jaarlijks inkomen van elke medewerker. Los de null-problematiek op dmv de** **CASE-expressie.**

select naam, maandsal\*12 +

(case

when comm is NULL then 0

else comm

end) as jaarsalaris

from medewerkers;

select naam, maandsal\*12 + nvl(comm, 0) as jaarsalaris

from medewerkers;

**4.4.2 Subquery**

Tip: werk van binnen naar buiten toe (eerst de subquery, dan de omhullende query)

Of omgekeerd en naar elkaar toe.

1. **Geef de naam en het salaris van de medewerkers die meer verdienen dan Clerckx.**

SELECT voorn, naam, maandsal

FROM medewerkers

WHERE maandsal >

(SELECT maandsal

FROM medewerkers

WHERE naam = 'CLERCKX');

1. **Geef de namen van de medewerkers die in dezelfde afdeling werken als Allard.**

SELECT voorn, naam

FROM medewerkers

WHERE naam != 'ALLARD' AND afd IN

(SELECT afd

FROM medewerkers

WHERE naam = 'ALLARD');

Let op: IN in plaats van = (probeer eens where naam = 'JACOBS'...)

1. **Geef de naam van alle medewerkers die zich voor een cursus hebben ingeschreven (in volgorde van de naam).**

SELECT voorn, naam

FROM medewerkers

WHERE mnr IN

(SELECT DISTINCT cursist

FROM inschrijvingen)

ORDER BY naam ASC;

1. **Geef de medewerkers (namen) die jonger zijn dan hun collega E Jacobs.**

SELECT voorn, naam

FROM medewerkers

WHERE gbdatum >

(SELECT gbdatum

FROM medewerkers

WHERE naam = 'JACOBS' AND voorn LIKE 'E%');

1. **Geef de cursisten die een cursus in Maaseik gevolgd hebben.**

(vanaf hier best casus tabellen overzicht erbij nemen...)

SELECT (voorn || ' ' || naam) AS "Cursist Maaseik"

FROM medewerkers

WHERE mnr IN

(SELECT cursist

FROM inschrijvingen

WHERE (cursus, begindatum) IN

(SELECT cursus, begindatum

FROM uitvoeringen

WHERE locatie = 'MAASEIK'));

1. **Geef de cursusnamen (en het type) die de medewerkers van de afdeling Verkoop hebben** **gevolgd.**

SELECT omschrijving, type

FROM cursussen

WHERE code IN

(SELECT cursus

FROM inschrijvingen

WHERE cursist IN

(SELECT mnr

FROM medewerkers

WHERE afd IN

(SELECT anr

FROM afdelingen

WHERE naam = 'VERKOOP')));

1. **Geef naam en voornaam van iedereen die ooit bij J. Caspers een cursus heeft gevolgd.**

SELECT naam, voorn

FROM medewerkers

WHERE mnr IN

(SELECT cursist

FROM inschrijvingen

WHERE (cursus,begindatum) IN

(SELECT cursus, begindatum

FROM uitvoeringen

WHERE docent IN

(SELECT mnr

FROM medewerkers

WHERE naam = 'CASPERS' AND voorn LIKE 'J%')));

1. **Geef de namen van de cursussen waarvan de docent niet gekend is.**

SELECT omschrijving

FROM cursussen

WHERE code IN

(SELECT cursus

FROM uitvoeringen

WHERE docent IS NULL);

1. **Geef de namen van de medewerkers van de afdeling verkoop wiens commissie niet gekend is.**

SELECT voorn, naam

FROM medewerkers

WHERE afd IN

(SELECT anr

FROM afdelingen

WHERE naam = 'VERKOOP')

AND comm IS NULL;

1. **Geef de namen van de medewerkers die zich inschreven voor een cursus waarvan de docent niet gekend is.**

SELECT DISTINCT voorn, naam

FROM medewerkers

WHERE mnr IN

(SELECT cursist

FROM inschrijvingen

WHERE (cursus, begindatum) IN

(SELECT cursus, begindatum

FROM uitvoeringen

WHERE docent IS NULL));

**4.4.3 Functies**

Extra weetje: uur uit datum halen => EXTRACT(hour FROM CAST(gbdatum AS TIMESTAMP))

1. **Maak een lijst met achter elke medewerker het precieze uurloon, het uurloon dat op twee decimalen is afgerond, en het uurloon dat op twee decimalen is afgekapt. Het uurloon wordt berekend op basis van een kwartaal: (maandsalaris x 3) / (38 x 13)**

select initcap(voorn) "Voornaam", initcap(naam) "Naam",

((maandsal \* 3) / (38 \* 13)) "Precieze uurloon",

round((maandsal \* 3) / (38 \* 13), 2) "Afgerond",

trunc((maandsal \* 3) / (38 \* 13), 2) "Afgekapt"

from medewerkers;

1. **Maak een lijst waarin de verschillende waarden van FUNCTIE elk op drie manieren wordt weergegeven: met alleen aan het begin een hoofdletter, volledig in hoofdletters en volledig in kleine letters.**

select distinct

upper(functie) "FUNCTIE",

initcap(functie) "Functie",

lower(functie) "functie"

from medewerkers;

**-- Je mag ook AS zetten, maar moet niet**

SELECT DISTINCT

UPPER(functie) AS "FUNCTIE",

INITCAP(functie) AS "Functie",

LOWER(functie) AS "functie"

FROM medewerkers;

1. **Geef van elke cursus de naam en de lengte van de naam.**

select omschrijving "Cursusnaam", length(omschrijving) "Lengte cursusnaam"

from cursussen;

1. **Doorzoek de namen van de medewerkers op de letter 'e'. Wordt de letter 'e' gevonden dan drukt men in het resultaat het positienummer af. Laat op drie manieren zoeken: zoek vanaf de eerste positie naar de eerste letter 'e'. zoek vanaf de vierde positie naar de eerste letter 'e' zoek vanaf de eerste positie naar de tweede letter 'e'.**

select

naam "Naam",

instr(naam, 'E') "Manier 1",

instr(naam, 'E', 4) "Manier 2",

instr(naam, 'E', 1, 2) "Manier 3"

from medewerkers;

1. **Maak een histogram met als maatstaf de lengte van de namen van de medewerkers. Zorg dat de kolom histogram 25 karakters breed is.**

-- NAAM LENGTE HISTOGRAM

-- BRIERS 6 \*\*\*\*\*\*

-- DE KONING 9 \*\*\*\*\*\*\*\*\*

col histogram format a25;

select

naam "Naam",

length(naam) "Lengte",

rpad('\*', length(naam), '\*') "Histogram"

from medewerkers;

1. **Maak een lijst met de namen van de medewerkers, met hun namen zonder vooraan de letters 'de' en met hun namen zonder achteraan de letters 'sen' indien ze voorkomen.**

select ltrim(naam, 'DE'), rtrim(naam, 'NES')

from medewerkers;

1. **Geef een lijst met de namen van de afdelingen, de nummers en de nummers voorafgegaan door de letters 'AF'.**

select naam "Naam", anr "Nummer", concat('AF ', anr) "AF-Nummer"

from afdelingen;

select naam "Naam", anr "Nummer", 'AF ' || anr "AF-Nummer"

from afdelingen;

1. **Geef een lijst met de namen van alle cursussen waarbij de letters 'WC#' vervangen worden door 'ZK-'. Zowel de hoofdletters als de kleine letters vervangen.**

select initcap(translate(omschrijving, 'WCwc#', 'ZKzk-')) "Gekke naam"

from cursussen;

1. **Sorteer de namen van de medewerkers zonder rekening te houden met eventuele blanco's.**

select naam "Naam", initcap(replace(naam, ' ')) "Sorteernaam"

from medewerkers

order by replace(naam, ' ') ASC;

select naam "Naam", initcap(replace(naam, ' ', '')) "Sorteernaam"

from medewerkers

order by replace(naam, ' ', '') asc;

1. **Druk enkel de medewerkers af waarvan de naam uit 2 delen bestaat. Het eerste gedeelte wordt in kleine letters en tussen haakjes na het tweede gedeelte (in hoofdletters) van de naam afgedrukt.  
   DE KONING   
   DEKONING  
   Vb. KONING (de)**

select

substr(naam, instr(naam, ' ')) ||

' (' ||

lower(substr(naam, 1, instr(naam, ' ') - 1)) ||

')' "Naam"

from medewerkers

where naam like '% %';

SELECT

(UPPER(SUBSTR(naam, INSTR(naam, ' '))) || ' (' ||

LOWER(RTRIM(SUBSTR(naam, 1, INSTR(naam, ' ')), ' ')) || ')') "Naam"

FROM medewerkers

WHERE (LENGTH(naam) - LENGTH(REPLACE(naam, ' ', ''))) = 1;

1. **Bepaal voor elke medewerker wiens naam een 'e' bevat wat groter is:   
   zijn salaris\*2, zijn MNR of het MNR van zijn chef?   
   Naast de naam en het grootste element wordt aangegeven welk element de grootste waarde heeft (chef, salaris of mnr). Gebruik hiervoor de DECODE()**

select naam, greatest(maandsal\*2, mnr, nvl(chef, mnr)) "Grootste waarde",

decode(greatest(maandsal\*2, mnr, nvl(chef, mnr)),

maandsal\*2, 'Salaris',

mnr, 'MNR',

'Chef') "Type"

from medewerkers

where upper(naam) like '%E%';

DECODE(input,

a, x,

b, y,

c, z,

w)

CASE input

WHEN a THEN x

WHEN b THEN y

WHEN c THEN z

ELSE w

END

CASE

WHEN input = a THEN x

WHEN input = b THEN y

WHEN input = c THEN z

ELSE w

END

select

naam "Naam",

greatest(maandsal \* 2, mnr, nvl(chef, mnr)) "Grootste waarde",

decode(greatest(maandsal \* 2, mnr, nvl(chef, 0)),

maandsal \* 2, 'Salaris',

mnr, 'MNR',

'Chef') "Type"

from medewerkers

where lower(naam) like '%e%';

1. **Bepaal voor elke medewerker wiens naam een 'e' bevat wat kleiner is: zijn salaris\*2, zijn MNR of het MNR van zijn chef? Naast de naam en het kleinste element wordt aangegeven welk element de kleinste waarde heeft (chef, salaris of mnr). Gebruik hiervoor de case.**

select naam "Naam",

least(maandsal \* 2, mnr, nvl(chef, mnr)) "Kleinste waarde",

case least(maandsal \* 2, mnr, nvl(chef, mnr))

when maandsal \* 2 then 'Salaris'

when mnr then 'MNR'

else 'Chef'

end "Type"

from medewerkers

where lower(naam) like '%e%';

1. **Geef een lijst van de medewerkers waarbij de functie 'TRAINER' voorgesteld wordt als 'LERAAR' en 'DIRECTEUR' als 'HOOFD'. Alle overige functies worden als 'MEDEWERKER' voorgesteld. Los op met decode() en de case.**

select naam,

case functie

when 'TRAINER' then 'LERAAR'

when 'DIRECTEUR' then 'HOOFD'

else 'MEDEWERKER'

end as functie

from medewerkers;

select naam,

decode(functie,

'TRAINER', 'LERAAR',

'DIRECTEUR', 'HOOFD',

'MEDEWERKER') as functie

from medewerkers;

1. **Welke waarde is het grootst: het MNR-2000 of de 2de macht van de eerste letterwaarde van de medewerkernaam? Geef aan welke waarde het grootste is.**

col "Grootste waarde" format a30;

select

naam "Naam",

case greatest(mnr - 2000, power(ascii(naam), 2))

when (mnr - 2000) then 'mnr - 2000'

else 'letterwaarde'

end "Grootste waarde"

from medewerkers;

select

naam "Naam",

decode(greatest(mnr - 2000, power(ascii(naam), 2)),

mnr - 2000, 'mnr - 2000',

'letterwaarde') "Grootste waarde"

from medewerkers;

1. **Geef een lijst van de medewerkers waarbij de kolommen NAAM, FUNCTIE en OVERZICHT worden afgedrukt. In de kolom OVERZICHT wordt voor een trainer zijn/haar salaris afgedrukt, voor een verkoper wordt zijn/haar commissie afgedrukt en in alle andere gevallen drukt men 'onbelangrijk' af.**

select naam, functie,

case functie

when 'TRAINER' then to\_char(maandsal)

when 'VERKOPER' then to\_char(comm)

else 'onbelangrijk'

end as overzicht

from medewerkers;

select naam, functie,

decode(functie,

'TRAINER', to\_char(maandsal),

'VERKOPER', to\_char(comm),

'onbelangrijk') as overzicht

from medewerkers;

1. **Geef voor alle managers de dag van de week, de dag, de maand en het jaar in vier cijfers waarop ze geboren zijn.  
   Vb. JACOBS donderdag 02 april 1987**

select

naam "Naam",

to\_char(gbdatum, 'day') "Weekdag",

to\_char(extract(day from gbdatum), '09') "Dag",

to\_char(gbdatum, 'month') "Maand",

extract(year from gbdatum) "Jaar"

from medewerkers

where functie = 'MANAGER';

1. **Hoeveel dagen oud ben je?**

select trunc(sysdate - DATE '1992-01-12') from dual;

select trunc(sysdate - to\_date('12-01-1992')) from dual;

1. **Geef voor alle trainers de geboortedatum voluit geschreven.  
   NAAM GEBOORTEDATUM VISSER The Twenty-fifth of December One Thousand Nine Hunderd Sixty-five**

alter session set nls\_language=English;

SELECT naam, TO\_CHAR(gbdatum, '"The" Ddsp "of" fmMonth Yyyysp') "GEBOORTEDATUM"

FROM medewerkers

WHERE functie = 'TRAINER';

alter session set nls\_language=Dutch;

1. **Geef het weeknummer, het kwartaal en het Juliaans dagnummer van de geboortedata van Wouters en Swinnen?**

SELECT

naam "Naam",

TO\_CHAR(gbdatum, 'W') "Weeknummer",

TO\_CHAR(gbdatum, 'Q') "Kwartaal",

TO\_CHAR(gbdatum, 'J') "Juliaans dagnummer"

FROM medewerkers

WHERE naam IN ('WOUTERS', 'SWINNEN');

1. **Maak een lijst waarin voor elke medewerker aangegeven staat op welk uur van de dag hij/zij geboren is. De tijd die in het voorbeeld wordt afgedrukt is de tijd die in het attribuut GBDATUM werd pgeslagen. Standaard is deze tijd 12:00 AM omdat het niet meegedeeld is aan de databank.**

SELECT voorn AS "Voornaam", naam AS "Naam", EXTRACT(hour FROM CAST(gbdatum AS TIMESTAMP)) AS "Uur geboorte"

FROM medewerkers;

1. **Geef voor alle medewerkers de geboortedatum en de datum 6 maanden later.**

SELECT

voorn "Voornaam",

naam "Naam",

gbdatum "Geboortedatum",

ADD\_MONTHS(gbdatum, 6) "6 maanden later"

FROM medewerkers;

1. **Geef naam, de vierkantswortel van het salaris en het aantal jaren (afgerond op 2 cijfers na de komma) dat de medewerker oud is.**

select

naam "Naam",

sqrt(maandsal) "Vierkantswortel",

round(months\_between(sysdate, gbdatum) / 12, 2) "Jaren"

from medewerkers;

1. **Druk de leeftijd af in maanden, dagen en jaren (zonder cijfers na de komma) voor elke medewerker ouder dan 35 jaar.**

select

voorn "Voornaam",

naam "Naam",

trunc(months\_between(sysdate, gbdatum)) "Maanden",

trunc(sysdate - gbdatum) "Dagen",

trunc(months\_between(sysdate, gbdatum) / 12) "Jaren"

from medewerkers

where trunc(months\_between(sysdate, gbdatum) / 12) > 35;

of gebruik floor() in plaats van trunc()

1. **Druk voor alle medewerkers de laatste dag van de maand af waarop zij geboren zijn.**

select voorn "Voornaam", naam "Naam", last\_day(gbdatum) "Laatste dag"

from medewerkers

1. **Druk van elke medewerker de datum af van de eerste vrijdag volgend op de 3000 dagen na de geboortedatum.**

select voorn "Voornaam", naam "Naam", next\_day((gbdatum + 3000), 'vri') "Datum"

from medewerkers;

1. **Geef de functie en naam van alle medewerkers die voor 1984 geboren zijn. Vermeld per medewerker de inkomensklasse. Wanneer het maandelijks salaris kleiner is dan 2500 wordt "goedkoop" vermeld anders is het "duur". Sorteer de resultaattabel op basis van de functie. Eerst de directeur, dan de managers, vervolgens de verkopers en tenslotte de trainer en boekhouder.**

select functie, naam,

case

when maandsal < 2500 then 'goedkoop'

else 'duur'

end as klasse

from medewerkers

where extract(year from gbdatum) < 1984

order by

case functie

when 'DIRECTEUR' then 1

when 'MANAGER' then 2

when 'VERKOPER' then 3

when 'TRAINER' then 4

else 5

end asc;

select functie, naam,

case

when maandsal < 2500 then 'goedkoop'

else 'duur'

end as klasse

from medewerkers

where extract(year from gbdatum) < 1984

order by

case

when functie = 'DIRECTEUR' then 1

when functie = 'MANAGER' then 2

when functie = 'VERKOPER' then 3

when functie = 'TRAINER' then 4

else 5

end asc;

select functie, naam,

case

when maandsal < 2500 then 'goedkoop'

else 'duur'

end as klasse

from medewerkers

where extract(year from gbdatum) < 1984

order by

decode(functie,

'DIRECTEUR', 1,

'MANAGER', 2,

'VERKOPER', 3,

'TRAINER', 4,

5) asc;

**5.7.1 Join-Operator**

**1) Geef van elke medewerker het nummer en de naam van de afdeling waarvoor hij/zij werkt.**

SELECT medewerkers.naam, medewerkers.afd, afdelingen.naam

FROM medewerkers

INNER JOIN afdelingen

ON medewerkers.afd = afdelingen.anr;

SELECT m.naam, m.afd, a.naam

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr;

**2)Geef de naam en het salaris van elke medewerker die meer verdient dan Clerckx.**

select naam, maandsal

from medewerkers

where maandsal >

(select maandsal

from medewerkers

where naam = 'CLERCKX');

select m.naam, m.maandsal

from medewerkers m

join medewerkers c

on c.naam = 'CLERCKX'

where m.maandsal > c.maandsal;

**3) Geef de namen van de medewerkers die in dezelfde afdeling werken als Wouters. Via auto-join:**

SELECT m.naam

FROM medewerkers m

JOIN medewerkers mm

ON m.afd = mm.afd

WHERE mm.naam = 'WOUTERS' AND m.naam != 'WOUTERS';

-- Via subquery:

SELECT m.naam

FROM medewerkers m

WHERE m.naam != 'WOUTERS' AND m.afd IN

(SELECT afd

FROM medewerkers

WHERE naam = 'WOUTERS');

**4) Geef het nummer van elke medewerker die in de afdeling verkoop werkt.**

SELECT m.mnr

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr

WHERE a.naam = 'VERKOOP';

**5) Geef het nummer van elke medewerker gevolgd door de naam van zijn of haar baas. Als je wil dat een medewerker ook getoond wordt zelfs als hij geen chef heeft => LEFT (OUTER) JOIN:**

SELECT m.mnr, baas.naam AS baas

FROM medewerkers m

LEFT JOIN medewerkers baas

ON m.chef = baas.mnr;

**Anders als je dit niet wil tonen (INNER) JOIN:**

SELECT m.mnr, baas.naam AS baas

FROM medewerkers m

JOIN medewerkers baas

ON m.chef = baas.mnr;

**6) Geef van elke medewerker het medewerkersnummer en de naam van de afdeling waarvoor hij of zij werkt en maak een overzicht per afdeling mogelijk.**

AFDELING MEDEWERKERS

--------------------------------------------------

HOOFDKANTOOR 7782 AN CLERCKX

...

OPLEIDINGEN 7566 EMMA JACOBS

7902 DORIEN DE COOMAN

...

VERKOOP 7521 THOMAS DEFOUR

...

BREAK ON afdeling;

SELECT

a.naam afdeling,

m.mnr || ' ' || m.voorn || ' ' || m.naam medewerkers

FROM afdelingen a

JOIN medewerkers m

ON a.anr = m.afd

ORDER BY a.naam, m.mnr;

CLEAR BREAKS;

**7) Welke medewerkers verdienen meer dan hun baas wanneer het salaris van de baas met 1000 EUR verminderd wordt?**

SELECT m.naam

FROM medewerkers m

JOIN medewerkers baas

ON m.chef = baas.mnr

WHERE m.maandsal > (baas.maandsal - 1000);

**8) Geef de namen van de chef (ook van De Koning) en hun ondergeschikten. Voorzie een overzicht per chef.**

BAAS ONDERGESCHIKTEN

------------------------------

BRIERS DEN RUYTER

JACOBS

DEFOUR

MARTENS

ALLARD

...

BREAK ON baas SKIP 1 ON ondergeschikten;

SELECT DISTINCT b.naam baas, o.naam ondergeschikten

FROM medewerkers o

JOIN medewerkers b

ON o.chef = b.mnr

ORDER BY b.naam, o.naam;

CLEAR BREAKS;

**9) Geef de namen van alle afdelingen en de naam van het afdelingshoofd. Doe LEFT (OUTER) JOIN als je ook afdelingen zonder afdelingshoofd wil tonen.**

SELECT a.naam afdeling, m.naam afdelingshoofd

FROM afdelingen a

LEFT JOIN medewerkers m

ON a.hoofd = m.mnr;

**Doe (INNER) JOIN als je geen afdelingen zonder afdelingshoofd**

**wil tonen.**

SELECT afdelingen.naam afdeling, afdelingshoofd.naam afdelingshoofd

FROM afdelingen

JOIN medewerkers afdelingshoofd

ON afdelingen.hoofd = afdelingshoofd.mnr;

**10) Geef de naam en voornaam van de medewerkers die een cursus volgen waar een bedrijfsafdeling gevestigd is.**

SELECT DISTINCT m.naam, m.voorn

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr

JOIN inschrijvingen i

ON m.mnr = i.cursist

JOIN uitvoeringen u

ON i.cursus = u.cursus

WHERE u.locatie = a.locatie;

**11) Geef de naam en voornaam van de medewerkers die een cursus volgen waar hun eigen afdeling gevestigd is.**

SELECT DISTINCT m.naam, m.voorn

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr

JOIN inschrijvingen i

ON m.mnr = i.cursist

JOIN uitvoeringen u

ON i.cursus = u.cursus

WHERE u.locatie = a.locatie;

**12) Geef de namen van de medewerkers die een cursus gevolgd hebben die hun chef ook gevolgd heeft.**

SELECT DISTINCT m.naam

FROM medewerkers m

JOIN inschrijvingen i

ON i.cursist = m.mnr

JOIN inschrijvingen ii

ON ii.cursist = m.chef

WHERE i.cursus = ii.cursus;

**13) Geef de namen van docenten die cursussen SQL, ORG of WEB doceren.**

SELECT DISTINCT m.naam

FROM medewerkers m

JOIN uitvoeringen u

ON m.mnr = u.docent

WHERE u.cursus IN ('SQL', 'ORG', 'WEB');

**14) Geef van alle inschrijvingen de naam van de docent, de naam van de cursus en de naam van de cursist. Geef een overzicht per docent en per omschrijving.**

break on docent on omschrijving skip 1;

select docent.naam as docent, c.omschrijving as omschrijving, deelnemer.naam

from uitvoeringen u

join medewerkers docent

on u.docent = docent.mnr

join cursussen c

on u.cursus = c.code

full outer join inschrijvingen i

on u.cursus = i.cursus and u.begindatum = i.begindatum

join medewerkers deelnemer

on i.cursist = deelnemer.mnr

order by docent.naam, c.omschrijving, deelnemer.naam;

clear breaks;

***waarom die full outer join?***

***\* Toon ook docenten die geen studenten hebben***

***\* Toon ook studenten die geen docent hebben.***

**15) Geef de namen van alle cursussen samen met de docent die deze cursussen doceren, evenals de begindata. De kolom omschrijving mag maar 35 karakters breed zijn en bovendien moet je voor een lege waarde bij docent '==onbekend==' weergeven. Geef een overzicht per cursusnaam. Opmerking: werkt enkel als je de kolommen in de break aanspreekt via hun alias!**

break on omschrijving skip 1 on begindatum;

col omschrijving format a35;

select c.omschrijving as omschrijving,

u.begindatum as begindatum,

nvl(docent.naam, '==onbekend==') as docent

from cursussen c

join uitvoeringen u

on c.code = u.cursus

left join medewerkers docent

on u.docent = docent.mnr

order by omschrijving;

clear breaks;

**16) Geef van alle cursisten hun naam, de naam van hun chef, de naam van de afdeling waar ze werken, de namen van de cursussen waarvoor ze zich eventueel inschreven en de namen van de docenten die deze cursussen geven. Opmerking: werkt enkel als je de kolommen in de break aanspreekt via hun alias!**

break on deelnemer on chef on afdeling skip 1;

select deelnemer.naam || ',' || deelnemer.voorn as deelnemer,

chef.naam as chef,

a.naam as afdeling,

c.omschrijving as omschrijving,

docent.voorn || ' ' || docent.naam as docent

from inschrijvingen i

join medewerkers deelnemer

on i.cursist = deelnemer.mnr

join medewerkers chef

on deelnemer.chef = chef.mnr

join afdelingen a

on deelnemer.afd = a.anr

join uitvoeringen u

on i.cursus = u.cursus and i.begindatum = u.begindatum

join cursussen c

on u.cursus = c.code

join medewerkers docent

on u.docent = docent.mnr

order by deelnemer, chef, afdeling, omschrijving, docent;

clear breaks;

**5.7.2 Groepsfuncties**

**1) Geef het aantal medewerkers per afdeling.**

SELECT afd, COUNT(mnr) aantal

FROM medewerkers

GROUP BY afd;

**-- Of als je echt de naam van de afdeling wilt tonen:**

SELECT a.naam afdeling, COUNT(m.mnr) aantal

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr

GROUP BY a.naam;

**2) Geef het gemiddelde salaris per afdeling.**

SELECT afd, AVG(maandsal)

FROM medewerkers

GROUP BY afd;

**Of als je echt de naam van de afdeling wilt tonen:**

SELECT a.naam "afdeling", AVG(m.maandsal) "gemiddeld salaris"

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr

GROUP BY a.naam;

**3) Geef het aantal medewerkers per afdeling, maar enkel voor de afdelingen waar in de afdelingsnaam een "OO" voorkomt.**

SELECT a.naam afdeling, COUNT(m.mnr) aantal

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr

WHERE a.naam LIKE '%OO%'

GROUP BY a.naam;

select m.afd, count(m.mnr)

from medewerkers m

where m.afd in

(select a.anr

from afdelingen a

where naam like '%OO%')

group by m.afd;

**4) Geef het aantal medewerkers voor alle afdelingen behalve de afdeling 10.**

SELECT afd, COUNT(mnr) aantal

FROM medewerkers

WHERE afd != 10

GROUP BY afd;

**5) Geef het maximum en het minimum salaris en de afdelingsnaam per afdeling.**

SELECT a.naam, MAX(m.maandsal), MIN(m.maandsal)

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr

GROUP BY a.naam;

**6) Geef de naam en het salaris van de medewerker die het meest verdient.**

SELECT naam, maandsal

FROM medewerkers

WHERE maandsal = (SELECT MAX(maandsal) FROM medewerkers);

**7) Geef per afdeling en per functie het aantal medewerkers en het gemiddelde salaris.**

SELECT afd, functie, COUNT(mnr), AVG(maandsal)

FROM medewerkers

GROUP BY afd, functie

ORDER BY afd, functie;

**8) Geef per afdeling het gemiddelde salaris voor die afdelingen waar meer dan 3 medewerkers werken.**

SELECT afd, AVG(maandsal)

FROM medewerkers

GROUP BY afd

HAVING COUNT(mnr) > 3;

**9) Geef per afdeling de naam van de afdeling en de naam van de medewerker(s) die het meest verdient.**

select a.naam, m.naam

from medewerkers m

join afdelingen a

on a.anr = m.afd

and m.maandsal in

(select max(mm.maandsal)

from medewerkers mm

where m.afd = mm.afd)

group by a.naam;

**10) Geef de namen van de medewerkers die een hoger salaris hebben dan het gemiddelde salaris.**

SELECT naam

FROM medewerkers

WHERE maandsal > (SELECT AVG(maandsal) FROM medewerkers);

**11) Geef de namen van de medewerkers die een hoger salaris hebben dan het gemiddelde salaris van hun afdeling.**

SELECT m.naam

FROM medewerkers m

WHERE m.maandsal >

(SELECT AVG(mm.maandsal)

FROM medewerkers mm

WHERE m.afd = mm.afd

GROUP BY mm.afd);

**12) Geef het aantal cursussen die elke medewerker gevolgd heeft (stijgend gesorteerd).**

SELECT m.naam, COUNT(i.cursus) aantal

FROM medewerkers m

LEFT JOIN inschrijvingen i

ON i.cursist = m.mnr

GROUP BY m.naam

ORDER BY aantal ASC;

**We doen een LEFT JOIN voor ook de medewerkers die geen cursussen volgden!**

**13) Wat is het gemiddelde salaris van de medewerkers die in dezelfde afdeling werken als 'Den Ruyter'?**

SELECT AVG(m.maandsal) gemiddelde

FROM medewerkers m

WHERE m.afd IN

(SELECT mm.afd

FROM medewerkers mm

WHERE mm.naam = 'DEN RUYTER');

**14) Hoeveel medewerkers verdienen minder dan het gemiddelde salaris plus 200?**

SELECT COUNT(m.mnr)

FROM medewerkers m

WHERE m.maandsal < (SELECT AVG(mm.maandsal) + 200 FROM medewerkers mm);

**15) Hoe heten de drie hoogst betaalde medewerkers?**

SELECT m.naam

FROM medewerkers m

WHERE 3 >=

(SELECT COUNT(mm.mnr)

FROM medewerkers mm

WHERE mm.maandsal >= m.maandsal);

**-- Even tussendoor, in MySQL of PostgreSQL kan dit echter vÃ©Ã©l korter:**

SELECT naam

FROM medewerkers

ORDER BY maandsal DESC

LIMIT 3;

**Of vanaf Oracle 12.1 en later volgens de ANSI/ISO standaard: (Stel dat je zit op Oracle 11, werkt dit niet...)**

SELECT naam

FROM medewerkers

ORDER BY maandsal DESC

FETCH NEXT 3 ROWS ONLY;

**16) Wat zijn de namen en de salarissen van de vijf laagst betaalde medewerkers?**

SELECT m.naam, m.maandsal

FROM medewerkers m

WHERE 5 >=

(SELECT COUNT(mm.mnr)

FROM medewerkers mm

WHERE mm.maandsal <= m.maandsal);

**17) Idem vraag 16, maar sorteer het resultaat dalend op salaris.**

SELECT m.naam, m.maandsal

FROM medewerkers m

WHERE 5 >=

(SELECT COUNT(mm.mnr)

FROM medewerkers mm

WHERE mm.maandsal <= m.maandsal)

ORDER BY m.maandsal DESC;

**18) Hoe heten de 3 meest verdienende verkopers?**

SELECT m.naam

FROM medewerkers m

WHERE

m.functie = 'VERKOPER' AND

3 >=

(SELECT COUNT(mm.mnr)

FROM medewerkers mm

WHERE mm.functie = 'VERKOPER' AND mm.maandsal >= m.maandsal);

**19) Welke is van alle gemiddelde maandsalarissen per afdeling het hoogste gemiddelde maandsalaris?**

SELECT MAX(AVG(maandsal))

FROM medewerkers

GROUP BY afd;

SELECT MAX(sub.gemiddelde) AS gemiddelde

FROM

(SELECT AVG(maandsal) gemiddelde

FROM medewerkers

GROUP BY afd) sub;

**20) Welke afdeling (naam) heeft het hoogste gemiddelde maandsalaris?**

SELECT a.naam

FROM afdelingen a

JOIN medewerkers m

ON a.anr = m.afd

GROUP BY a.naam

HAVING

AVG(m.maandsal) =

(SELECT MAX(AVG(mm.maandsal))

FROM medewerkers mm

GROUP BY mm.afd);

**21) Op welke datum is voor het laatst een cursus gepland?**

SELECT MAX(begindatum)

FROM uitvoeringen;

Of meer informatie tonen:

SELECT cursus, begindatum, locatie

FROM uitvoeringen

JOIN

(SELECT MAX(begindatum) laatste

FROM uitvoeringen) u

ON begindatum = u.laatste;

**22) Geef de resultaatqueries zoals beschreven in de syllabus.**

break on afd skip 1;

compute avg label 'GEMIDDELDE' of avg(maandsal) on afd;

compute count label 'AANTAL' of count(mnr) on afd;

select afd, functie, count(mnr), avg(maandsal)

from medewerkers

group by afd, functie

order by afd, functie;

clear computes;

clear breaks;

**5.7.3 Union, Difference, Intersect-operator**

**1) Geef een lijst van de medewerkersnamen en de afdelingsnamen.**

SELECT m.naam

FROM medewerkers m

UNION

SELECT a.naam

FROM afdelingen a;

**2) Geef een lijst met namen van afdelingen en namen van cursussen die respectievelijk in Hasselt gevestigd zijn of daar doorgaan.**

SELECT a.naam

FROM afdelingen a

WHERE a.locatie = 'HASSELT'

UNION

SELECT c.omschrijving

FROM uitvoeringen u

JOIN cursussen c

ON u.cursus = c.code

WHERE u.locatie = 'HASSELT';

**3) Geef de medewerkers die geen manager zijn.**

SELECT naam, functie

FROM medewerkers

WHERE functie != 'MANAGER';

**4) Geef de locaties waar wel cursussen doorgaan maar waar geen afdelingen gevestigd zijn.**

SELECT u.locatie

FROM uitvoeringen u

MINUS

SELECT a.locatie

FROM afdelingen a;

**5) Geef de naam en begindatum van elke cursus die alleen door de medewerker 7844 gevolgd is.**

select cursus, begindatum

from inschrijvingen

where cursist = 7844;

select cursus, begindatum

from inschrijvingen

intersect

select cursus, begindatum

from inschrijvingen

where cursist = 7844;

select cursus, begindatum

from inschrijvingen

minus

select cursus, begindatum

from inschrijvingen

where cursist != 7844;

select distinct c.omschrijving, i.begindatum

from inschrijvingen i

join uitvoeringen u

on i.cursus = u.cursus

join cursussen c

on u.cursus = c.code

where i.cursist = 7844;

**6) Geef de cursuscode van de cursussen die E. Jacobs niet gevolg heeft.**

SELECT i.cursus

FROM inschrijvingen i

MINUS

SELECT i.cursus

FROM inschrijvingen i

JOIN medewerkers m

ON i.cursist = m.mnr

WHERE m.naam = 'JACOBS' AND m.voorn LIKE 'E%';

**7) Geef de namen van de medewerkers die geen cursussen gevolgd hebben.**

SELECT m.naam

FROM medewerkers m

MINUS

SELECT DISTINCT m.naam

FROM medewerkers m

JOIN inschrijvingen i

ON m.mnr = i.cursist;

**8) In welke plaatsen is minstens 1 afdeling gevestigd en wordt minstens 1 cursus gepland?**

SELECT a.locatie

FROM afdelingen a

INTERSECT

SELECT u.locatie

FROM uitvoeringen u;

**9) Geef de nummers van de medewerkers die de cursus 'Windows Server' gevolgd hebben en in de afdeling Verkoop werken.**

select i.cursist

from inschrijvingen i

join uitvoeringen u

on i.cursus = u.cursus and i.begindatum = u.begindatum

join cursussen c

on u.cursus = c.code

where c.omschrijving = 'Windows Server'

intersect

select m.mnr

from medewerkers m

join afdelingen a

on m.afd = a.anr

where a.naam = 'VERKOOP';

select i.cursist

from inschrijvingen i

join uitvoeringen u

on i.cursus = u.cursus and i.begindatum = u.begindatum

join cursussen c

on u.cursus = c.code

where c.omschrijving = 'Windows Server'

intersect

select mnr

from medewerkers

where afd in (select anr from afdelingen where naam = 'VERKOOP');

**5.7.4 View**

Opmerking voor als je geen plan hebt van database:

\* Overzicht van tabellen tonen: SELECT \* FROM tab;

\* Beschrijving datatypes en structuur van een tabel: DESC medewerkers;

1. **Maak een view BAAS dat het nummer van elke medewerker geeft, gevolgd door de naam van zijn of haar baas.**

create or replace view BAAS as

select m.mnr, nvl(baas.naam, m.naam) as baas

from medewerkers m

left join medewerkers baas

on m.chef = baas.mnr;

SELECT \* FROM tab;

DESC baas;

SELECT \* FROM baas;

1. **Wijzig de view zodat de gegevens stijgend gesorteerd op de naam van de baas worden weergegeven.**

create or replace view BAAS as

select m.mnr, nvl(baas.naam, m.naam) as baas

from medewerkers m

left join medewerkers baas

on m.chef = baas.mnr

order by baas;

1. **Zoek de view op in de datadictionary en geef de viewdefinitie.**

set long 999;

SELECT view\_name, text

FROM user\_views;

**-- extra) Verwijder de view BAAS:**

DROP VIEW baas;

1. **Maak een view JAARSAL dat de voornamen, de namen, afdelingsnaam en het jaarsalaris (incl. commissie) van alle medewerkers berekent.**

CREATE VIEW jaarsal AS

SELECT

m.voorn,

m.naam,

a.naam afdeling,

(m.maandsal \* 12 + NVL(m.comm, 0)) jaarsalaris

FROM medewerkers m

JOIN afdelingen a

ON m.afd = a.anr;

1. **Gebruik de view om enkel de medewerkers van de afdeling opleidingen te selecteren. Verwijder je laatste view.**

DESC jaarsal;

SELECT voorn, naam

FROM jaarsal

WHERE afdeling = 'OPLEIDINGEN';

DROP VIEW jaarsal;

**6.5.1 script**

--------------------------------------------------------------------------------

Voer script CreCaseDML.sql uit

--------------------------------------------------------------------------------

Onderstaande wijzigingen mogen NIET definitief doorgevoerd worden!!!

DUS DOE GEEN COMMIT!!!

1. **Wijzig de naam van medewerker met mnr 7876 in Boonen.**

UPDATE medewerkers

SET naam = 'BOONEN'

WHERE mnr = 7876;

**maak eventueel ongedaan met commando:**

ROLLBACK;

1. **Verander de locatie van afdeling 10 in Tongeren.**

UPDATE afdelingen

SET locatie = 'TONGEREN'

WHERE anr = 10;

**maak eventueel ongedaan met commando:**

ROLLBACK;

1. **Verwijder cursus LIN.**

DELETE FROM cursussen

WHERE code = 'LIN';

**maak eventueel ongedaan met commando:**

ROLLBACK;

1. **Alle medewerkers van de afdeling Verkoop krijgen 10% opslag.**

UPDATE medewerkers

SET maandsal = maandsal \* 1.1

WHERE afd =

(SELECT a.anr

FROM afdelingen a

WHERE a.naam = 'VERKOOP');

**maak eventueel ongedaan met commando:**

ROLLBACK;

1. **Voeg aan de tabel MEDEWERKERS de gegevens toe van een nieuwe medewerker: 7999, Willem** **Revis, 21/01/1983, boekhouder, salaris €2950, chef 7782.**

INSERT INTO medewerkers (mnr, voorn, naam, gbdatum, functie, maandsal, chef)

VALUES (7999, 'Willem', 'Revis', date '1983-05-22', 'BOEKHOUDER', 2950, 7782);

**maak eventueel ongedaan met commando:**

ROLLBACK;

1. **Voer de gegevens in van nog een medewerker: Polien Dox, 7989, trainer, chef 7902 en geboren op de 350ste dag van 1980, om 3:30 's nachts.**

***Let op: ik moet hier wel een maandsal meegeven, want dit mag niet NULL zijn!***

INSERT INTO medewerkers (voorn, naam, mnr, functie, chef, gbdatum, maandsal)

VALUES ('PAULIEN', 'DOX', 7989, 'TRAINER', 7902,

to\_date('01-01-1980 03:30:00', 'dd-mm-yyyy hh24:mi:ss') + 349, 0);

**maak eventueel ongedaan met commando:**

ROLLBACK;