

4 Berekeningen en functies in SQL

Applicatie die in dit hoofdstuk gebruikt wordt: Oracle Apex Database die in dit hoofdstuk gebruikt wordt: HR

In vorige hoofdstukken hebben we in de select-lijst eenvoudige kolommen opgehaald. In de where clause hebben eenvoudige kolommen vergeleken met eenvoudige waarden. In dit hoofdstuk zullen we in de select-lijst en in de where-clause functies en berekeningen gebruiken.

4.1 Berekende waarden selecteren

Rekenkundige operatoren

optelling	+
verschil	-
vermenigvuldiging	*
deling	/

De rekenkundige operatoren kunnen gecombineerd worden met haakjes.

Vraag: geef een lijst met het nummer van de werknemer, het loon, het commissiepercentage, de commissie als ze 10000 euro zouden verkopen, de som van het loon en deze berekende commissie. Geef enkel die werknemers die een loon hebben die groter is dan 10000 euro. Het resultaat moet gesorteerd zijn op het nummer van de werknemer.

```
SELECT    EMPLOYEE_ID, SALARY, COMMISSION_PCT,  
            COMMISSION_PCT * 10000,  
            SALARY + (COMMISSION_PCT * 10000)
```

```
FROM      EMPLOYEES
```

```
WHERE     SALARY > 10000
```

```
ORDER BY EMPLOYEE_ID
```

Resultaat:

EMPLOYEE_ID	SALARY	COMMISSION_PCT	COMMISSION_PCT*10000	SALARY+(COMMISSION_PCT*10000)
100	24000	-	-	-
101	17000	-	-	-
102	17000	-	-	-
149	10500	.2	2000	12500

174	11000	.3	3000	14000
201	13000	-	-	-
205	12000	-	-	-

4.2 Kolommen in de resultatentabel benoemen

In de resultatentabel hebben de kolommen de naam van de kolommen uit de onderliggende tabel. De berekende kolommen kunnen dus geen naam hebben. Dit wordt opgelost met de **AS**.

Voorbeeld:

```

SELECT  EMPLOYEE_ID, SALARY, COMMISSION_PCT,
           COMMISSION_PCT * 10000 AS COMMISSIE,
           SALARY + (COMMISSION_PCT * 10000) AS EINDLOON

FROM    EMPLOYEES

WHERE   SALARY > 10000

ORDER BY EINDLOON

```

Resultaat:

EMPLOYEE_ID	SALARY	COMMISSION_PCT	COMMISSIE	EINDLOON
149	10500	.2	2000	12500
174	11000	.3	3000	14000
102	17000	-	-	-
201	13000	-	-	-
101	17000	-	-	-
100	24000	-	-	-
205	12000	-	-	-

Opmerking: de labels (na de AS) kunnen terug gebruikt worden in de ORDER BY clause, maar niet in de WHERE-clause

4.3 Bewerkingen met NULL-waarden

Vraag: geef alle werknemers met een nummer tussen 170 en 180. Geef hun nummer, familienaam, voornaam en het nummer van het departement. Is er geen nummer plaats dan de tekst 'ongekend'. Het resultaat moet gesorteerd zijn op het nummer van de werknemer.

```

SELECT  EMPLOYEE_ID, LAST_NAME, FIRST_NAME,
           COALESCE (TO_CHAR(DEPARTMENT_ID), 'ongekend') AS
           DEPARTEMENT

```

```

FROM      EMPLOYEES

WHERE     EMPLOYEE_ID BETWEEN 170 AND 180

ORDER BY EMPLOYEE_ID

```

Resultaat:

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTEMENT
174	Abel	Ellen	80
176	Taylor	Jonathon	80
178	Grant	Kimberely	ongekend

Betekenis van de functie COALESCE: geef de eerste niet-NULL die komt in de lijst die volgt.

Opmerking: in de vorige query hadden we SALARY + (COMMISSION_PCT * 10000). Dit geeft echter een fout als het COMMISSION_PCT NULL is. Van zodra er een NULL is ergens in de berekening dan is het eindresultaat NULL. COMMISSION_PCT * 10000 → als de inhoud van COMMISSION_PCT NULL is dan is COMMISSION_PCT * 10000 ook NULL. Dus SALARY + (COMMISSION_PCT * 10000) is dan ook NULL, wat echter niet juist is. Die werknemer werkt niet op commissie maar heeft wel een loon. En toch zal zijn eindloon niet zichtbaar zijn. Het zal echter wel juist zijn als we die NULL kunnen omzetten naar het cijfer 0.

Voorbeeld:

```

SELECT    EMPLOYEE_ID, SALARY, COMMISSION_PCT,
            COMMISSION_PCT * 10000 AS COMMISSIE,
            SALARY + COALESCE((COMMISSION_PCT * 10000),0) AS
                                                    EINDLOON

FROM      EMPLOYEES

ORDER BY EMPLOYEE_ID

```

of

```

SELECT    EMPLOYEE_ID, SALARY, COMMISSION_PCT,
            COMMISSION_PCT * 10000 AS COMMISSIE,
            SALARY + (COALESCE(COMMISSION_PCT,0) * 10000) AS
                                                    EINDLOON

FROM      EMPLOYEES

ORDER BY EMPLOYEE_ID

```

Resultaat:

EMPLOYEE_ID	SALARY	COMMISSION_PCT	COMMISSIE	EINDLOON
100	24000	-	-	24000
101	17000	-	-	17000
102	17000	-	-	17000
103	9000	-	-	9000
104	6000	-	-	6000
107	4200	-	-	4200
124	5800	-	-	5800
141	3500	-	-	3500
142	3100	-	-	3100
143	2600	-	-	2600
144	2500	-	-	2500
149	10500	.2	2000	12500
174	11000	.3	3000	14000
176	8600	.2	2000	10600
178	7000	.15	1500	8500
200	4400	-	-	4400
201	13000	-	-	13000
202	6000	-	-	6000
205	12000	-	-	12000
206	8300	-	-	8300

NULL gerelateerde functies

naam van de functie	standard SQL	omschrijving
COALESCE(expr1,expr2)	x	het eerste niet-null argument
LNNVL(conditie)		True als de conditie False of unknown oplevert, anders false
NULLIF(expr1,expr2)	x	null als expr1=expr2 anders expr1
NVL(expr1,expr2)		expr2 als expr1 null is anders expr1
NVL2(expr1,expr2,expr3)		expr2 als expr 1 niet null is en anderst expr3

4.4 Numerieke functies

Voorbeeld:

```
SELECT POWER(10,2) AS HONDERD
FROM DUAL
```

Resultaat:

Numerieke functies

naam van de functie	standard SQL	omschrijving
ABS(n)	x	absolute waarde van n
ACOS(n)		arc cosinus van n
ASIN(n)		arc sinus van n
ATAN(n)		arc tangens van n
BITAND(n,m)		logische AND over de binaire representatie van n en m
CEIL(n)	x	kleinste geheel getal groter dan of gelijk aan n
COS(n)		cosinus van n
COSH(n)		cosinus hyperbolicus van n
EXP(n)	x	e = 2.71828183 to de n-de macht
FLOOR(n)	x	grootste geheel getal kleiner dan of gelijk aan n
LN(n)	x	natuurlijke logaritme van n
LOG(n,m)		de n logaritme van m
MOD(m,n)	x	rest na deling van m door n (afgekapt)
NANVL(n,m)		als n geen getal is wordt m teruggegeven
POWER(m,n)	x	m tot de nde macht
REMAINDER(m,n)		rest na deling van m door n (afgerond)
ROUND (n,m)		n afgerond op m posities
SIGN(n)		teken : -1,0 of 1
SIN(n)		sinus van n
SINH(n)		sinus hyperbolicus van n
SQRT(n)	x	vierkantwortel n
TAN(n)		tangens van n
TANH(n)		arc tangens van n
TRUNC (n,m)		n afgekapt op m posities
WIDTH_BUCKET(e,min,max,nb)	x	histogram: bucket waarin e valt als we het interval van min tot max verdelen in nb buckets

4.5 Alfnumerieke functies

Vraag: geef van alle werknemers het werknemernummer, familienaam en voornaam in de volgende vorm: familienaam v met v de eerste letter van de voornaam. Sorteert de lijst op familienaam.

```
SELECT EMPLOYEE_ID, LAST_NAME || ' ' ||
SUBSTR(FIRST_NAME,1,1) AS NAAM
```

```
FROM EMPLOYEES
```

```
ORDER BY LAST_NAME
```

Resultaat (eerste 10 records):

EMPLOYEE_ID	NAAM
174	Abel E
142	Davies C
102	De Haan L
104	Ernst B
202	Fay P
206	Gietz W
178	Grant K
201	Hartstein M
205	Higgins S
103	Hunold A

Alfanumerieke functies

naam van de functie	standard SQL	omschrijving
CHR(n)		karakter met ascii waarde n
CONCAT(c1,c2)		concateneer c1 en c2
INITCAP(char)		ieder woord beginnend met een hoofdletter
LOWER(char)	x	alle letters omgezet naar kleine letters
LPAD(c1,n,c2)		c1 wordt links aangevuld tot lengte n met karakters van c2
LTRIM(char,set)		karakters worden verwijderd tot het eerste karakter dat niet in set voorkomt
REGEXP_REPLACE(c1,p)		zoek in c1 naar de expressie p; geef de positie in c1 terug
REGEXP_SUBSTR(c1,p)		zoek in c1 naar de expressie p; geef de substr terug
REPLACE(char,s,r)		ieder voorkomen van sin char vervangen door r; zonder r worden alle voorkomens van s verwijderd
RPAD(c1,n,c2)		c1 rechts aangevuld tot lengte met de karakters van c2

RTRIM(char,set)		karakters na het laatste karakter dat niet in set voorkomt worden verwijderd
SUBSTR(char,m,n)	x	substring van char vanaf positie m, n karakters
TRANSLATE(char,f,t)	x	char vertaald van karakterset f naar karakterset t
TRIM(..from) src)	x	verwijder karakters aan het begin en/of einde van src
UPPER(char)	x	alle letters omgezet naar hoofdletters
LENGTH(char)	x	Lengte van karakterveld

Voorbeeld:

```
SELECT    LAST_NAME, FIRST_NAME, LAST_NAME || FIRST_NAME AS
           STREEPJES, CONCAT(LAST_NAME, FIRST_NAME) AS FUNCTIE

FROM      EMPLOYEES

ORDER BY  LASTNAME
```

Resultaat (eerste 10 records):

LAST_NAME	FIRST_NAME	STREEPJES	FUNCTIE
Abel	Ellen	AbelEllen	AbelEllen
Davies	Curtis	DaviesCurtis	DaviesCurtis
De Haan	Lex	De HaanLex	De HaanLex
Ernst	Bruce	ErnstBruce	ErnstBruce
Fay	Pat	FayPat	FayPat
Gietz	William	GietzWilliam	GietzWilliam
Grant	Kimberely	GrantKimberely	GrantKimberely
Hartstein	Michael	HartsteinMichael	HartsteinMichael
Higgins	Shelley	HigginsShelley	HigginsShelley
Hunold	Alexander	HunoldAlexander	HunoldAlexander

4.6 Berekeningen met datum waarden

Vraag: geef van alle werknemers: voornaam, familienaam, aantal dienstjaren, aanwervingjaar, aanwervingmaand, aanwervingdag. Sorteert de lijst dalend op het aantal dienstjaren.

```
SELECT    FIRST_NAME, LAST_NAME,
           EXTRACT (YEAR FROM CURRENT_DATE) - EXTRACT (YEAR
           FROM HIRE_DATE) AS anc,
           EXTRACT (YEAR FROM HIRE_DATE) AS startjaar,
```

EXTRACT (MONTH FROM HIRE_DATE) AS startmaand,
EXTRACT(DAY FROM HIRE_DATE) AS startdag

FROM EMPLOYEES

ORDER BY anc **DESC**

Resultaat (eerste 10 records):

FIRST_NAME	LAST_NAME	ANC	STARTJAAR	STARTMAAND	STARTDAG
Jennifer	Whalen	27	1987	9	17
Steven	King	27	1987	6	17
Neena	Kochhar	25	1989	9	21
Alexander	Hunold	24	1990	1	3
Bruce	Ernst	23	1991	5	21
Lex	De Haan	21	1993	1	13
William	Gietz	20	1994	6	7
Shelley	Higgins	20	1994	6	7
Trenna	Rajs	19	1995	10	17
Ellen	Abel	18	1996	5	11

Datumfuncties

naam van de functie	standard SQL	omschrijving
ADD_MONTHS(d,n)		datum d plus n maanden
CURRENT_DATE	x	huidige datum, met tijdzone info
CURRENT_TIMESTAMP	x	huidig tijdstip met tijdzone info
EXTRACT (c from d)		haal de component c uit een datum
LAST_DAY(d)		de laatste dag van de maand waarin d valt
LOCALTIMESTAMP	x	huidig tijdstip zonder tijdzone info
MONTHS_BETWEEN(d,e)	x	verschil in maanden tussen datum d en e
NEXT_DAY(d,char)		datum van de eerste weekdag die gelijk is of later valt dan
SESSIONTIMEZONE		tijdzone van huidige sessie
SYSDATE		huidige systeemdatum
SYSTIMESTAMP		huidig systeemtijdstip
TO_CHAR (datetime)		omzetting naar karakter

Opmerking: als het verschil gemaakt wordt tussen 2 datumvelden dan is het resultaat een numeriek met het aantal dagen tussen de 2 datumvelden.

4.7 Conversie functies

Voorbeeld:

```
SELECT   SYSDATE AS VANDAAG,  
          TO_CHAR(SYSDATE, 'DAY')|| EXTRACT(DAY FROM SYSDATE)|| ' '  
          ||TO_CHAR(SYSDATE,'MONTH') AS OMSCHRIJVING  
  
FROM DUAL
```

Resultaat:

VANDAAG	OMSCHRIJVING
10/13/2014	MONDAY 13 OCTOBER

Conversiefuncties

naam van de functie	standard SQL	omschrijving
TO_CHAR		Zet een datum of getal om naar karakters
TO_NUMBER		Zet een string om naar een getal
TO_DATE		Zet een string om naar een datum

4.8 Berekeningen in de WHERE-clause

Vraag: geef van alle werknemers waarvan het aantal dienstjaren groter of gelijk is aan 20: voornaam, familienaam, aantal dienstjaren, aanwervingjaar, aanwervingmaand, aanwervingstag. Sorteert de lijst dalend op het aantal dienstjaren.

```
SELECT   FIRST_NAME, LAST_NAME,  
          EXTRACT (YEAR FROM CURRENT_DATE) - EXTRACT (YEAR  
                FROM HIRE_DATE) AS anc,  
          EXTRACT (YEAR FROM HIRE_DATE) AS startjaar,  
          EXTRACT (MONTH FROM HIRE_DATE) AS startmaand,  
          EXTRACT( DAY FROM HIRE_DATE) AS startdag  
  
FROM     EMPLOYEES  
  
WHERE    EXTRACT (YEAR FROM CURRENT_DATE) - EXTRACT (YEAR  
                FROM HIRE_DATE) >= 20  
  
ORDER BY anc DESC
```

Resultaat:

FIRST_NAME	LAST_NAME	ANC	STARTJAAR	STARTMAAND	STARTDAG
Steven	King	27	1987	6	17
Jennifer	Whalen	27	1987	9	17
Neena	Kochhar	25	1989	9	21
Alexander	Hunold	24	1990	1	3
Bruce	Ernst	23	1991	5	21
Lex	De Haan	21	1993	1	13
Shelley	Higgins	20	1994	6	7
William	Gietz	20	1994	6	7

Vraag: toon alle jobs waarbij de beschrijving iets te maken heeft met 'manager'. Geef alle kolommen

```
SELECT *  
FROM JOBS  
WHERE UPPER(JOB_TITLE) LIKE '%MANAGER%'
```

Resultaat:

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AC_MGR	Accounting Manager	8200	16000
SA_MAN	Sales Manager	10000	20000
ST_MAN	Stock Manager	5500	8500
MK_MAN	Marketing Manager	9000	15000

Alle berekeningen en functies die gebruikt werden in de SELECT clause mogen ook gebruikt worden in de WHERE clause.

4.9 De case-expressie in de SELECT-clause

Vraag: geef van alle bedienden, de familienaam, voornaam en hun 'verticale dienst'. Sorteert de lijst op familienaam. De bedienden werken in verschillende departementen over verschillende locaties. Op de verschillende afdelingen kunnen er meerdere jobs uitgevoerd worden. Zo zijn er bijvoorbeeld op verschillende afdelingen mensen die zorgen voor de marketing. Nu willen we niet het departement zien, maar een algemene omschrijving van het soort job dat ze uitvoeren. Dit is terug te vinden in de eerste 2 letters van hun job_id: IT:informatica, AC=boekhouding, AD=administratie, MK=marketing, SA=verkoop, ST= magazijn.

```
SELECT LAST_NAME, FIRST_NAME,  
CASE UPPER(SUBSTR(JOB_ID,1,2))  
WHEN 'IT' THEN 'informatica'  
WHEN 'AC' THEN 'boekhouding'
```

```

WHEN 'AD' THEN 'administratie'
WHEN 'MK' THEN 'marketing'
WHEN 'SA' THEN 'verkoop'
WHEN 'ST' THEN 'magazijn'
ELSE JOB_ID
END AS VERT_DIENST

```

FROM EMPLOYEES

ORDER BY LAST_NAME

Resultaat (eerste 10 records):

LAST_NAME	FIRST_NAME	VERT_DIENST
Abel	Ellen	verkoop
Davies	Curtis	magazijn
De Haan	Lex	administratie
Ernst	Bruce	informatica
Fay	Pat	marketing
Gietz	William	boekhouding
Grant	Kimberely	verkoop
Hartstein	Michael	marketing
Higgins	Shelley	boekhouding
Hunold	Alexander	informatica

Dit is de eenvoudige (simple) CASE: een kolomwaarde wordt vergeleken met vaste waarden. De gelijkheid wordt nagekeken, "=" wordt als vergelijkingsoperator gebruikt. De kolomwaarde die vergeleken wordt, wordt vast gehouden naast het sleutelwoord CASE.

In het onderstaand voorbeeld wordt een andere vergelijkingsoperator gebruikt. De kolomwaarden worden in iedere WHEN opnieuw geschreven. Iedere mogelijke expressie mag daar geschreven worden.

Het resultaat van de CASE komt als volgt tot stand:

- de WHEN's worden geëvalueerd in de volgorde zoals ze voorkomen. Als een conditie waar is, wordt niet meer verder gezocht
- voldoet er geen enkele WHEN, dan wordt de ELSE uitgevoerd
- is er geen ELSE, dan wordt een NULL-waarde terug gegeven



De verschillende waarden die kunnen gekozen worden in de CASE moeten allemaal van hetzelfde datatype zijn.

Vraag: geef van alle werknemers, familienaam, voornaam en hun loon. Als het loon echter hoger is dan 20000 dan mag dit niet weergegeven worden. Op deze plaats moet dan de tekst komen 'too high to show'.

```

SELECT LAST_NAME, FIRST_NAME,

```

```

CASE
  WHEN (SALARY > 20000) THEN 'TOO HIGH TO SHOW'
  ELSE TO_CHAR(SALARY)
END AS SAL

```

FROM EMPLOYEES

ORDER BY SALARY **DESC**

Resultaat (eerste 10 records):

LAST_NAME	FIRST_NAME	SAL
King	Steven	TOO HIGH TO SHOW
Kochhar	Neena	17000
De Haan	Lex	17000
Hartstein	Michael	13000
Higgins	Shelley	12000
Abel	Ellen	11000
Zlotkey	Eleni	10500
Hunold	Alexander	9000
Taylor	Jonathon	8600
Gietz	William	8300

Vraag: geef van alle werknemers, familienaam, voornaam, loon, aanwervingdatum, jobnummer en hun mogelijke opslag. Hebben de werknemers een loon kleiner dan 5000 dan krijgen ze 10% opslag, indien dit niet zo is maar is het aantal dienstjaren kleiner dan 5 dan krijgen ze 8%, indien dit niet zo is maar zijn ze werkzaam in het magazijn (eerste 2 letters van jobnummer = ST) dan krijgen ze 5%, de rest krijgt 3%.

```

SELECT LAST_NAME, FIRST_NAME, SALARY, HIRE_DATE, JOB_ID,
CASE
  WHEN (SALARY < 5000) THEN '10%'
  WHEN ((EXTRACT(YEAR FROM CURRENT_DATE) -
    EXTRACT(YEAR FROM HIRE_DATE)) < 5) THEN '8%'
  WHEN (UPPER(SUBSTR(JOB_ID,1,2)) = 'ST') THEN '5%'
  ELSE '3%'
END AS OPSLAG

```

FROM EMPLOYEES

ORDER BY SALARY, HIRE_DATE, JOB_ID

Resultaat (eerste 10 records):

LAST_NAME	FIRST_NAME	SALARY	HIRE_DATE	JOB_ID	OPSLAG
Vargas	Peter	2500	07/09/1998	ST_CLERK	10%
Matos	Randall	2600	03/15/1998	ST_CLERK	10%
Davies	Curtis	3100	01/29/1997	ST_CLERK	10%

Rajs	Trenna	3500	10/17/1995	ST_CLERK	10%
Lorentz	Diana	4200	02/07/1999	IT_PROG	10%
Whalen	Jennifer	4400	09/17/1987	AD_ASST	10%
Mourgos	Kevin	5800	11/16/1999	ST_MAN	5%
Ernst	Bruce	6000	05/21/1991	IT_PROG	3%
Fay	Pat	6000	08/17/1997	MK_REP	3%
Grant	Kimberely	7000	05/24/1999	SA_REP	3%

4.10 De case-expressie in de WHERE-clause

SELECT *

FROM TABLE

WHERE COL1 = **CASE**

WHEN CONDITION1 **THEN** VALUE1

WHEN CONDITION2 **THEN** VALUE2

ELSE VALUE3

END

Voorbeeld:

SELECT LAST_NAME, FIRST_NAME, JOB_ID, SALARY

FROM EMPLOYEES

WHERE SALARY > **CASE**

WHEN UPPER(SUBSTR(JOB_ID,1,2))='AC' **THEN** 10000

ELSE 15000

END

Resultaat:

LAST_NAME	FIRST_NAME	JOB_ID	SALARY
King	Steven	AD_PRES	24000
Kochhar	Neena	AD_VP	17000
De Haan	Lex	AD_VP	17000
Higgins	Shelley	AC_MGR	12000

Deze query kan herschreven worden naar:

SELECT LAST_NAME, FIRST_NAME, JOB_ID, SALARY

FROM EMPLOYEES

WHERE (SALARY > 1000 AND UPPER(SUBSTR(JOB_ID,1,2))='AC')

OR SALARY > 15000)

OEFENINGEN

- 1 Voer volgende query uit: `SELECT * FROM DUAL`
- 2 Voer de volgende functies uit op het getal 345.678: round, ceil, Floor. Bekijk het resultaat en ontdek de verschillen. (gebruik de tabel dual). Maak 1 query.
- 3 Neem het getal 345.678 rond af: op 2 decimalen, op 1 decimaal, op eenheid, op tiental, op honderdtal. Maak 1 query.
- 4 Bepaal de absolute waarde van en het teken van -123. Maak 1 query.
- 5 Maak een query om te bepalen hoe oud je bent (in dagen).
- 6 Maak een query op de tabel dual. Toon de datum 29-jan-1996, deze datum + 1 maand, 29-jan-1997, deze datum + 1 maand, huidige datum - 3 maanden
- 7 Maak een lijst met alle gegevens van de werknemers die een oneven werknemersnummer hebben.
- 8 Maak een lijst van alle werknemers met het nummer van de werknemer en de familienaam. Sorteert op familienaam, zodat de kortste naam eerst staat.
- 9 Maak een lijst van alle werknemers met hun familienaam, voornaam, aanwervingsdatum. De lijst moet gesorteerd zijn op het aanwervingsjaar, vervolgens op naam.
- 10 Maak een lijst van alle werknemers behalve van King.S. Geef de familienaam.v waarbij v de eerste letter is van de voornaam.
- 11 Herschrijf onderstaande query met een CASE:
SELECT EMPLOYEE_ID, SALARY, COMMISSION_PCT, COMMISSION_PCT * 10000 **AS** COMMISSIE, SALARY + **COALESCE**((COMMISSION_PCT * 10000),0) **AS** EINDLOON
FROM EMPLOYEES
ORDER BY EMPLOYEE_ID
- 12 Maak een lijst van alle werknemers met hun familienaam, voornaam en als hun loon lager is dan 5000 dan 'zeer laag loon', is het tussen de 5000 en 10000 dan 'laag loon', is het tussen 10000 en 15000 'gemiddeld loon', is het tussen 15000 en 20000 dan 'hoog loon', is het hoger dan 20000 dan 'zeer hoog loon'.