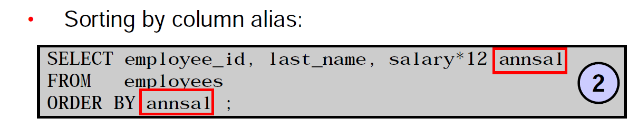
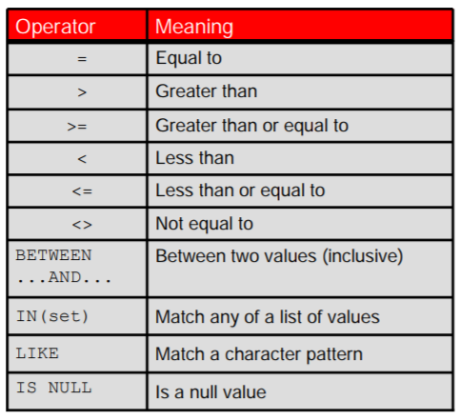
DATA

select theatre\_performance, actor\_id as acteurs from performance\_actors where season like '2018% ;

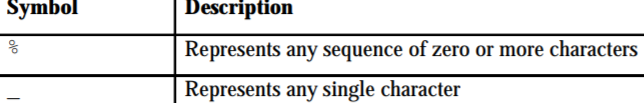
Select = query opstellen from subject;  
desc (employees ,department…) = geeft alle velden van het gegeven object

Berekeningen in SQL = + \* / bv jaarsalaris = salary \* 12  
Haken gebruiken!  
Null = bestaat niet, nog niet gevonden, nog niet ingevuld… null heeft meerdere betekenissen  
**kolom aliassen** = select last\_name AS name, commission\_pct commissie; (is hoofdletter!)  
kleine letters ipv caps voor alias = select last\_name “Naam”;  
concatenatie van 2 gegevens = || tussen 2 objecten kan ook tekst toevoegen ‘ ‘ singles! Tussen 2 ||  
DISTINCT(col) = hoeveel verschillende objecten van iets, 3 maal 90 komt maar 1 keer voor  
Oef:  
select last\_name, (salary \* 12) \* commission\_pct from employees; jaarlijks commissie berekenen

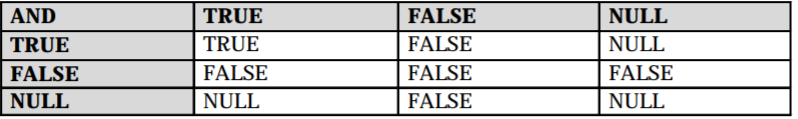
  
***H2***  
where = komt na de from, zoekt welke objecten een gegeven opzoekt. Bv from jobs where job\_id = AC;  
dag van vandaag = select sysdate from dual

Te gebruiken bij Where:

Gebruiken bij Like: tussen single quotes bv ‘%isa%’



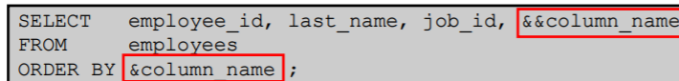
!= niet gelijk aan

And, or, not operatoren werken in sql ook



Order by = op einde van de select, zal ordenen op hetgene wat opgegeven word, bv order by hire\_date; of met desc of asc erachter. Kan ook aliassen selecteren, of kolomnr. De order by hoeft niet in de uitvoer te komen wanneer niet nodig

Om user input in te laten geven: WHERE of order by of from gebruiken met object = &naamdatopschermkomttestaan

Kan ook && gebruiken om de invoer nogmaals te gebruiken zonder opnieuw invoer te vragen. Bv:  
Order by &column\_name; dan kan er een lijn terug &&column\_name gebruikt worden  


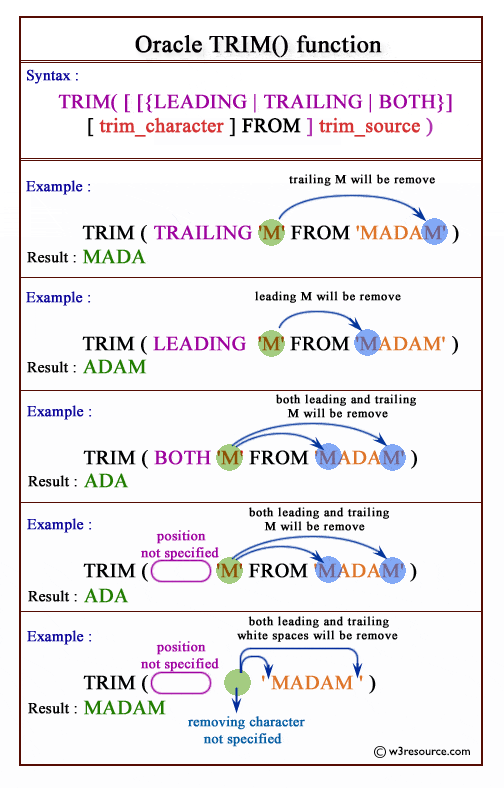
zet rond de ‘&object’ zodat t meteen een string word, en er geen error komt na de ingave zonder ‘ ’

Define = vooraf een waarde geven, de naam van de variabele gebruik je na de &, blijven enkel bestaan tijdens sessie  
undefine = verwijder een variabale

Verify= SET VERIFY ON: geeft wanneer we een && gebruiken een verificatie van welke invoer er is ingegeven. Ma best set verify off want das dikke bullshit

***H3 datums ingeven altijd ’01-SEP-98’ singlesss!!***

Character functions = lower, upper, initcap, concat, substr, length,instr,lpad,rpad,trim,replace

Initcap= elk woord begint met hoofdletter  
length = geeft aantal characters in een expressie terug  
instr= geeft plaats van gegeven string in nummers terug  
concat = voegt 2 expressies aan elkaar toe zoals ||  
substr = (string, start, length) start kan negatief zijn, en begint dan achteraan. Geef enkel gekoezen deel weer, bv (vippy, 2, 1) print dan i, (vippy, 3, 2) print dan pp.  
  
RPAD of LPAD (*text-exp* , *length* [, *pad-exp*]) = beginwoord, lengte van string, deel wat word toegevoegd tot de lengte van beginwoord evenlang als gegeven lengte is)  
bv rpad(vippy,9,ok) wordt vippyokok (9 chars als ik kan tellen anders, u kno wtf i mean)  
dit kan ook gebruikt worden om woorden in te korten, door een lengte korter in te geven dan de string is. Nav Lpad of Rpad word er shit links of rechts weggeknipt.  
  
Trim = dis shit is stupid

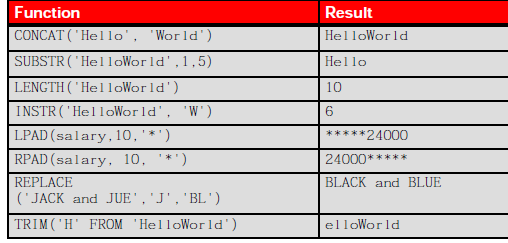
SQL> SELECT TRIM(LEADING '0' FROM '000123') FROM DUAL;

TRI

---

123

SQL> SELECT TRIM(TRAILING '0' FROM '123000') FROM DUAL;

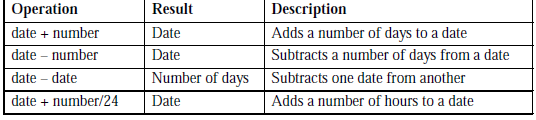
TRI

---

123

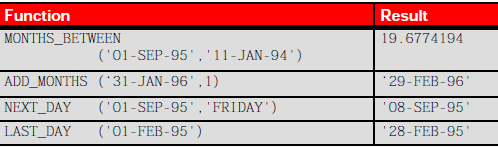
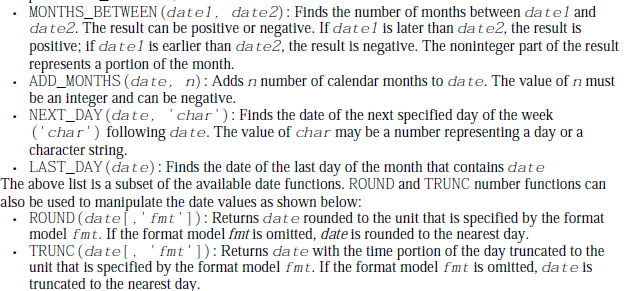
Round(n1, n2) = n1 gegeven getal, n2 welke plaats wordt afgerond. 0 is voor de komma, 1 en hoger is achter de komma obv

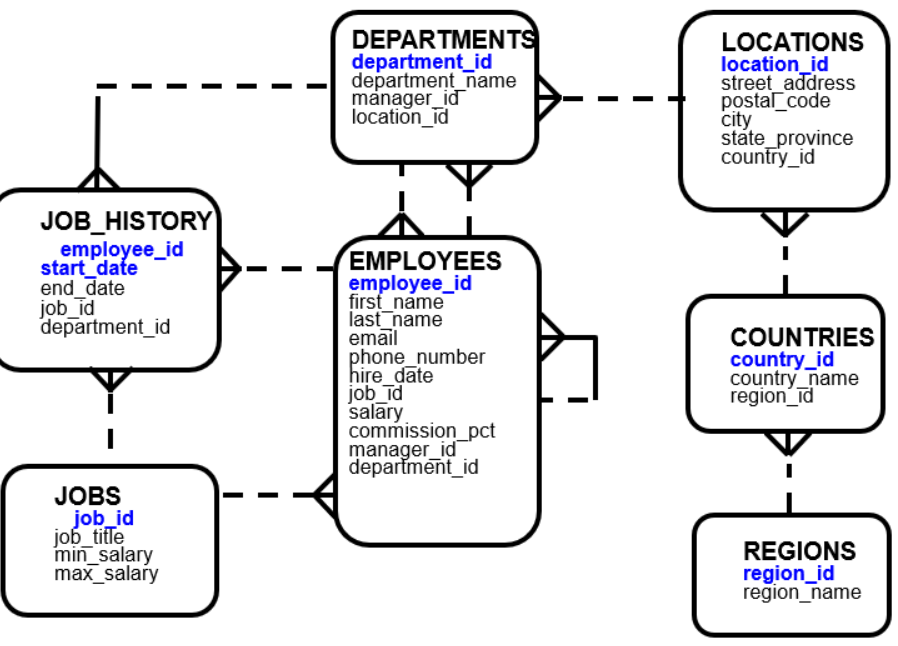
Trunc(n1, n2)= n1 gegeven getal, n2 welke plaats na de komma moet worden weegegeven, trunc(2.465, 1) geeft dan 2.4. als er een negatief getal in n2 word gegeven gebeurt de trunc voor de komma: bv ( 142.256, -2) geef dan 100, deze functie rond niet af maar plaatst eigl 0’en waar trunc wordt gebruikt. Jeez zo ne tekst u better understand it  
mod(n1, n2) = Modulo van getal n1 door n2

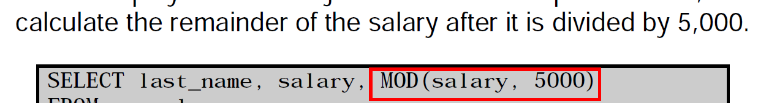
dit ^^ was alles van nummers, behold, below the bullshit of dates:

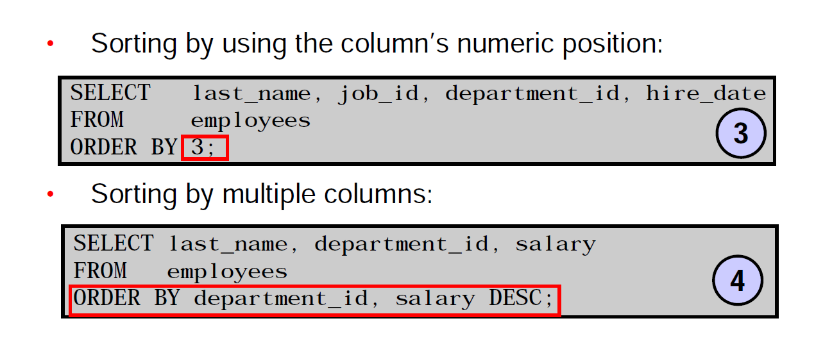
datum syntax = DD-MON-YR



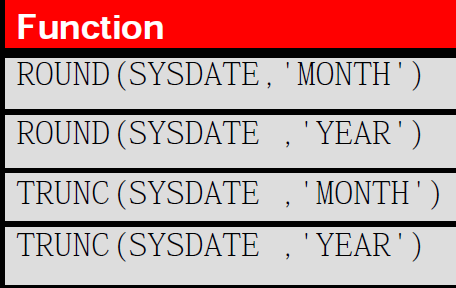
Select sysdate from dual; datum van vandaag  
trunc en round werken ook op datums

 Bij nextday(date, char) kan als char gwn ‘FRIDAY of MONDAY etc’ ingevoerd worden tussen single quotes

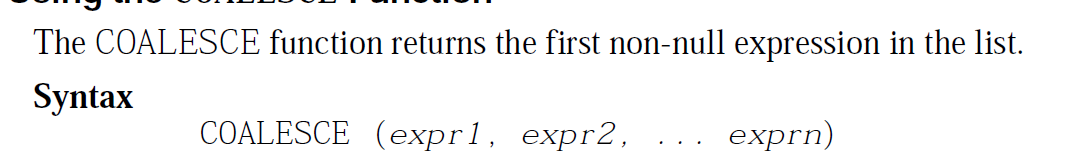


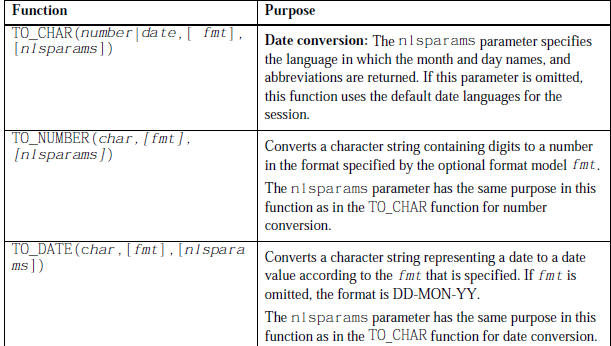


SORTEREN

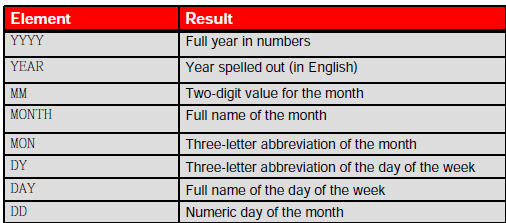
 Trunc = naar de eerste maand, dag wahtever

H4

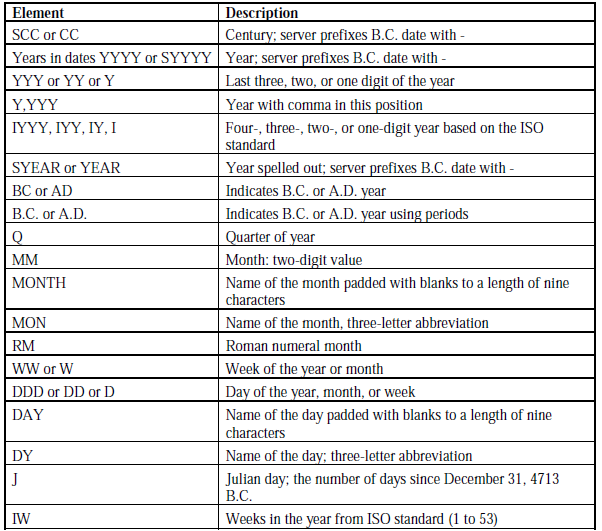


to\_char ‘month’ in hoofdletters of kleine letters  
case sensitive  
  
‘fmDdspth’  
om ranggetal te formateren  
bv eerste, tweede…. Voluit geschreven

select wijzigt nooit iets in een tabel  
 maar geeft ze anders weer  
  
om van 25929 naar $25,929  
to\_char(factuurbedrag,’$999,999.99’)

om de .00 te tonen vervang je de .99   
naar .00 in de format, een 9 geeft  
enkel alle getallen weer dat geen 0 is  
om bv 0.25 in te geven word t 90.00

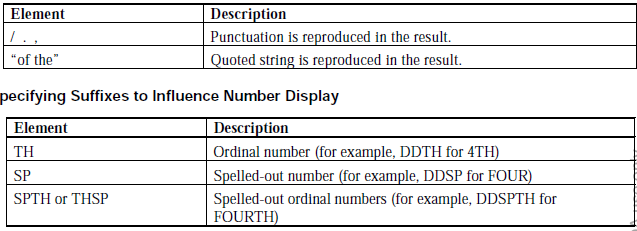
hire\_date is een datum dus om spef dat  
te zoeken, gebruiken we   
hire\_date = To\_date(‘may 25, 1999’  
,’monthDD year)

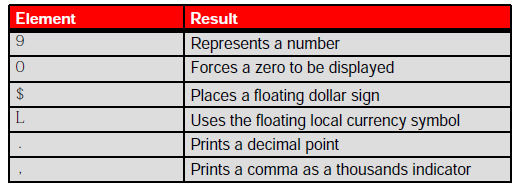
Fm = voluitgeschreven bv dag  
ddth =ordinaal getal (2de dag of 1ste)  
MONTH= zal de maand in caps weergeven  
Month = maand met hoofdletter  
month= maand zonder hoofdletter

Alter (wijzigen)  
session = 1 enkele sessie (verdwijnen na exit)  
set nls\_territory = ‘BELGIUM’;  
set nls\_language = ‘DUTCH’;

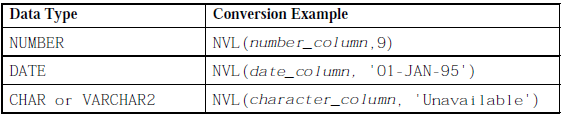
Hiermee verandert de sysdate naar  
europese normen

als ge group functions gebruikt moet ge werken met having ipv where





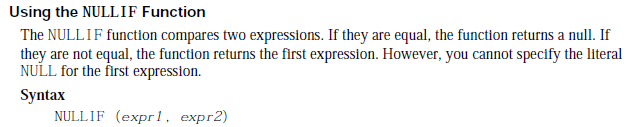
Nesting Functies  
  
functie3(functie2(functie1, 1, 2), 1), 1) , sql behandelt functies altijd vanuit het ‘diepste’ deel

Als er lege vakken worden getoond :

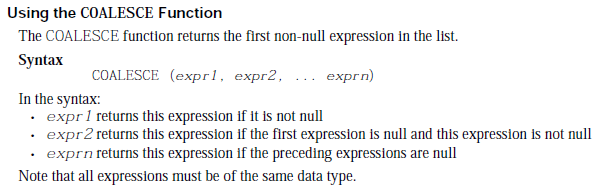
NVL1

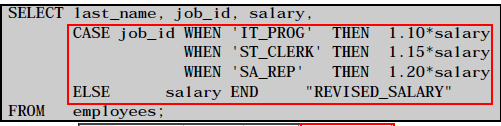
NVL2

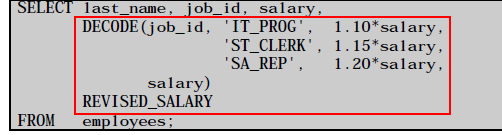
NVL3

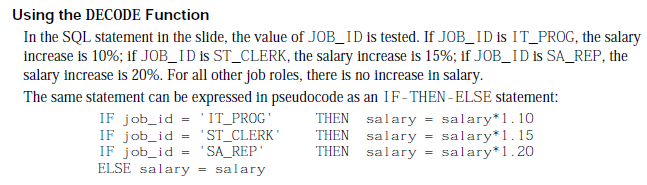


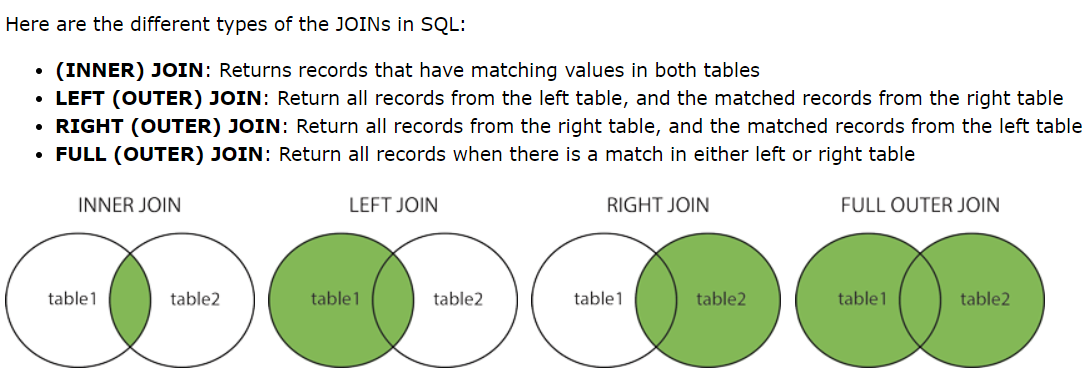








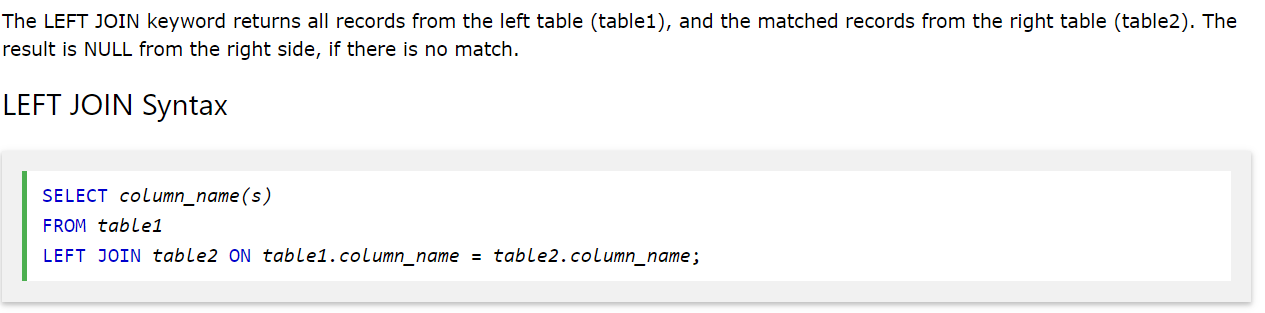


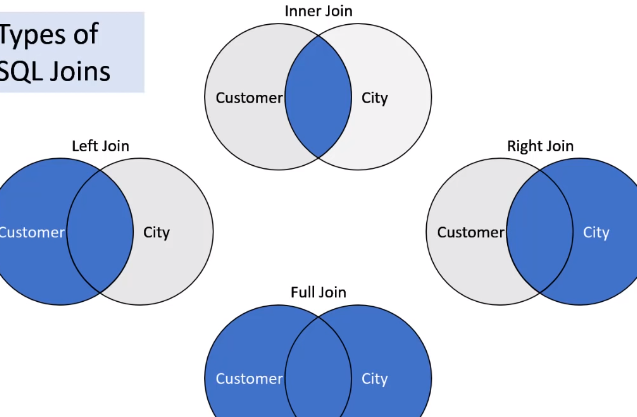
Data uit verschillende tabellen:

Syntax = select “data\_ID“, “data\_date “ from “1ste tabel “   
 JOIN 1 bovenstaande “2de tabel” ON tabel1.column\_name = table2.column\_name

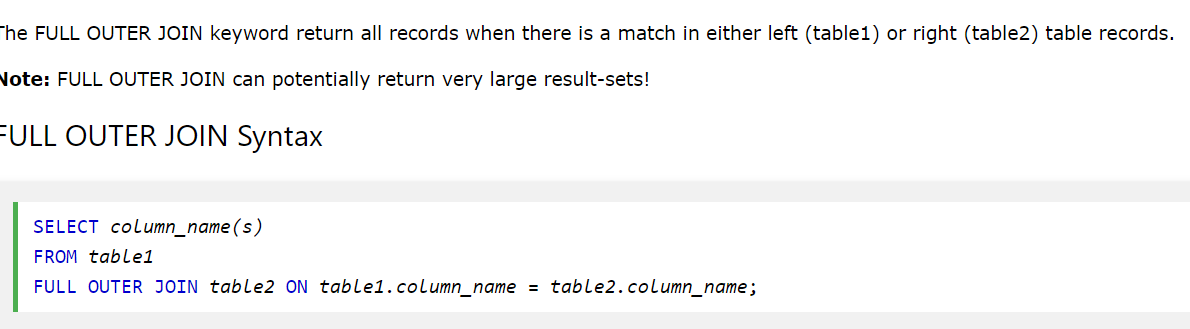
select alle kolommen, die we nodig hebben uit meerdere tabellen from tabelnaam1  
JOIN tabelnaam2 on tabelnaam1.primarykey = tabelnaam2.primarykey;  
tablenamen na ON zijn hetzelfde als de namen die we na de from en voor de JOIN zetten  
kolomnamen zullen een mix zijn van data uit de 2 tabellen

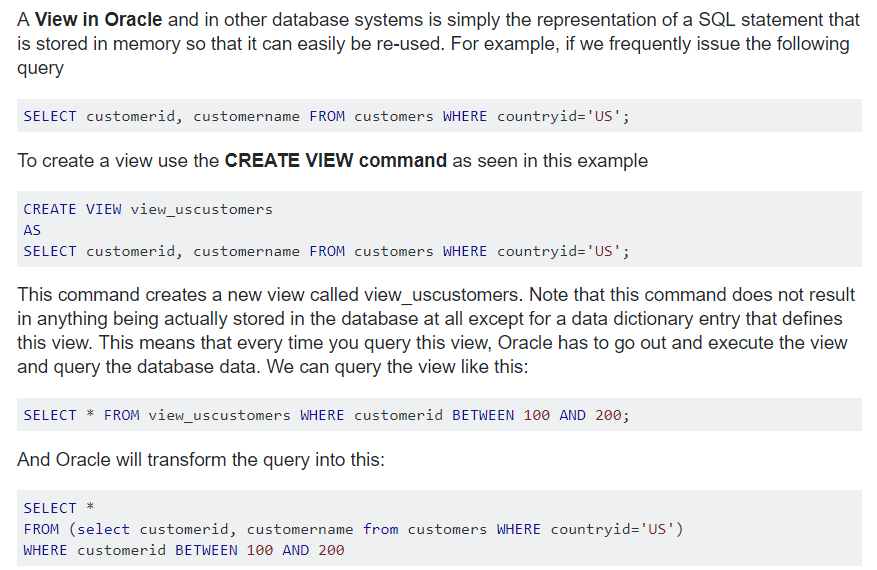
**select first\_name, end\_date, start\_date from employees   
JOIN job\_history on employees.employee\_id = job\_history.employee\_id;**

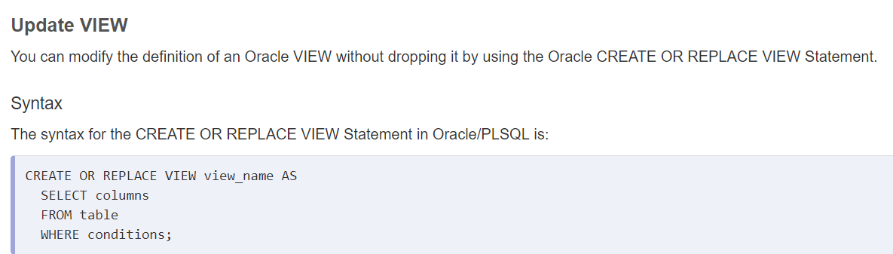


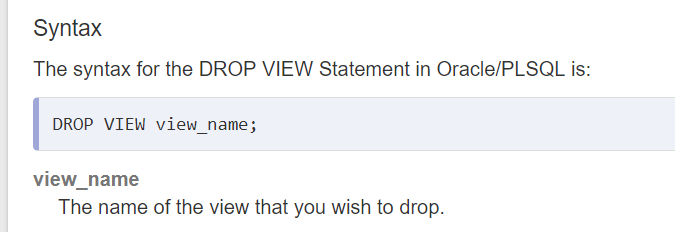


H6



VIEWS

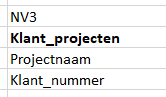




**1NF** Splits de herhalingsstructuren af.

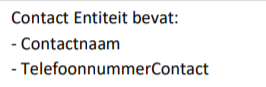
**2NF** Splits alle niet van de volledige sleutel afhankelijke gegevens af:  
**3NF** Splits alle transitieve afhankelijkheden af.

ERD

Entity-relationship-diagram – diagram dat grafisch een reeks gegevens voorstelt  
We werken met de Martin relatie, for some reason.

Entiteit word afgeleid uit een tabel na normalisatie ->

Entiteiten en entiteitsinstanties:



Contact is hier de entiteitsklasse en naam en telnr is hier de instantie

**Relaties, aka de leuke lijntjes tussen onze entiteitjes**

1 : 1 relatie  
1 instantie van deze entiteti is gerelateerd met 1 instantie van de andere entiteit, in dit voorbeeld heeft elke werkenemer MAX 1 auto.

1:N relatie of gwn 1 op veel relatie  
1 instantie staat in verband met vele instanties uit de andere entiteit. In dit voorbeeld wonen veel studenten in 1 studentenhuis maar een bepaalde student woont ofc maar in 1 studentenhuis

M:N relatie of veel op veel relatie  
dit type mag normaal niet meer voorkomen na het normaliseren, omdat er hier teveel mogelijkheden zijn, in dit voorbeeld kan een student lid worden van meerdere clubs maar 1 student kan ook lid zijn van meerdere clubs

Optionele shit

Hier moet minstens 1 student in een studentenhuis wonen.



Hier kan een student in geen of hoogstens 1 studentenhuis wonen, en dat er minstens 1 student moet zijn die in een studentenhuis woont



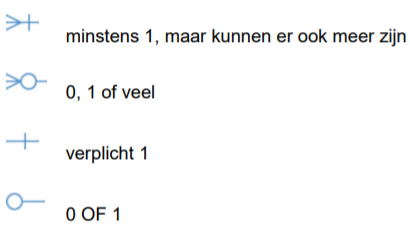
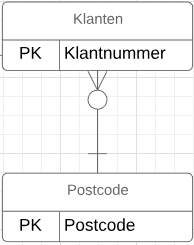
Hier is elke student gekoppeld aan minstens 1 studentenhuis, maar dat elk studentenhuis geen of meerdere studenten heeft.



Hier zijn alle werknemers als instantie, dat betekent ALLE werknemers, ook met hogere functies. Waardoor een werknemer een andere werknemer als baas KAN hebben en geen of meerdere werknemers onder zich kan hebben staan werken.

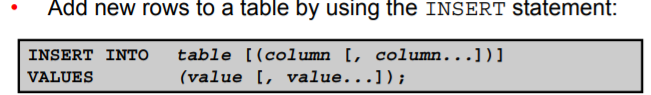


Hier zijn zowel bestuurder als begeleiders werknemers van een trein, maar een trein word maar door 1 bestuurder besteerd, maar moet exact 1 of meerder begeleiders hebben, we mogen onze relaties dan benoemen.



De relatie tussen postcode en gemeentes is zo goed als altijd zoals onderstaande, dit omdat dit lijsten zijn die vooraf opgesteld zijn en meestal onveranderlijk zijn.

Hier moet dus elke klant als die er is, sowieso 1 postcode hebben

H9: scripts

Enkel 1 rij word met deze syntax geinstert

Examen structuur:

