



**Vlaamse Dienst voor Arbeidsbemiddeling en
Beroepsopleiding**

SQL OPLOSSINGEN VAN DE OEFENINGEN

Inhoudsopgave

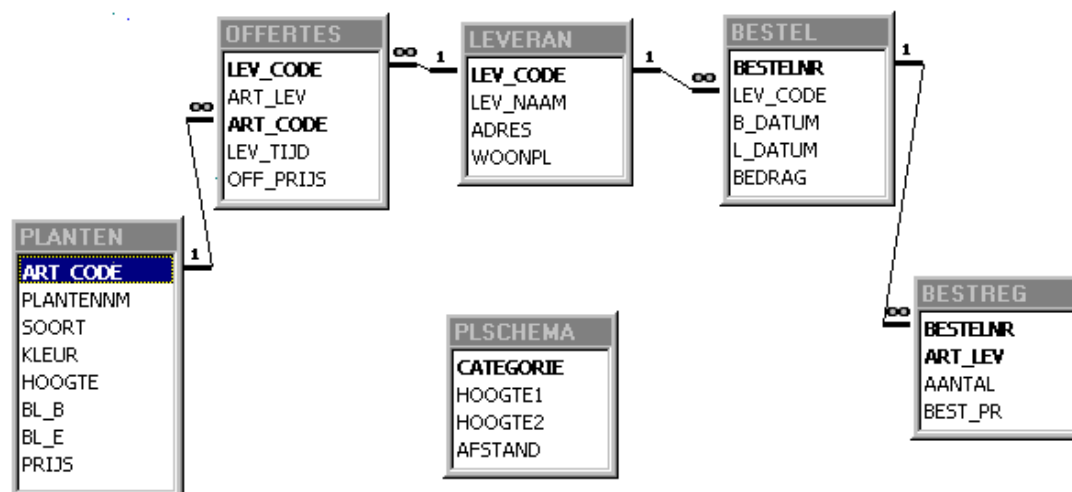
1	DE DATABASE PLANTV	3
2	GEGEVENS SELECTEREN	4
2.1.1	Oefeningen SELECT deel 1	4
2.1.2	Oefening SELECT deel 2	7
2.1.3	Oefeningen SELECT deel 3	8
2.1.4	Oefeningen SELECT deel 4	10
2.1.5	Oefeningen SELECT deel 5	12
3	RECORDS TOEVOEGEN	14
4	GEGEVENS WIJZIGEN.....	15
5	GEGEVENS VERWIJDEREN.....	16
6	TABELLEN	17
6.1	Oefeningen CREATE TABLE	17
6.2	Oefeningen ALTER.....	17
6.3	Oefeningen CONSTRAINT	17
7	INDEXEN	18
8	VIEWS	19
9	EINDOEFENING.....	20
10	COLOFON	22

1 DE DATABASE PLANTV

De database plantv bestaat uit 6 tabellen, nl:

Planten	Bevat de gegevens van de planten waarbij het veld art_code de primary key, plantennm bevat de naam van de plant, bl_b en bl_e bevat de maanden wanneer de bloeiperiode van de plant begint en eindigt.
Offertes	Bevat de catalogusgegevens van onze leveranciers. lev_code en art_code vormen samen de primary key. art_lev is de code die de leveranciers gegeven aan de plant.
Leveran	Bevat de gegevens van de leveranciers. Lev_code is de primary key.
Bestel	Bevat de bestellingen die wij geplaatst hebben bij de leveranciers. B_datum is de besteldatum en l_datum is de gewenste leveringsdatum.
Bestreg	Bevat de detaillijnen van de bestelling. De primary key wordt bepaald door de velden bestelnr en art_lev.
Plschema	Bevat de gegevens van de verschillende soorten categorieën van planten.

De prijzen in deze tabel zijn uitgedrukt in Euro. De hoogte is uitgedrukt in cm.



Deze database wordt gebruikt in de oefeningen.

2 GEGEVENS SELECTEREN

2.1.1 Oefeningen SELECT deel 1

1. Select * from leveran where woonpl='AALSMEER'
(resultaat : 4 rijen)
2. Select art_code, plantennm, prijs from planten order by plantennm
(resultaat : 119 rijen)
3. Select art_code, plantennm, bl_b from planten where bl_b=3
(resultaat : 3 rijen)
4. Select art_code, art_lev, lev_code from offertes order by art_code, art_lev
(resultaat : 218 rijen)
5. Select * from planten where soort='water' order by hoogte desc
(resultaat : 9 rijen)
6. Select distinct kleur from planten
(resultaat : 13 rijen)
7. Select * from planten where kleur is Null
(resultaat : 11 rijen)
8. Select distinct soort from planten
(resultaat : 10 rijen)
9. Select art_code, plantennm, hoogte, bl_b from planten where kleur='geel' and
soort='vast'
(resultaat : 3 rijen)
10. Select * from planten where prijs > 10 and soort<>'boom'
(resultaat : 8 rijen)
11. Select * from planten where (bl_b=6 and kleur='geel') or (bl_b=8 and
kleur='rood')
(resultaat : 2 rijen)
12. Select art_code, plantennm, hoogte from planten where kleur='gemengd' and
hoogte<=60
(resultaat : 13 rijen)
13. Select * from leveran where woonpl<>'Hillegom'
(resultaat : 9 rijen)
14. SELECT art_code, plantennm, kleur, hoogte FROM planten WHERE kleur IS
NULL AND hoogte IS NULL
(resultaat : geen rijen)

15. `SELECT art_code, plantennm FROM planten WHERE bl_e >= 10 AND bl_b <=8`
(resultaat : 6 rijen)
16. `Select art_code, plantennm from planten where bl_e >= 9 and bl_b <=9`
(resultaat : 26 rijen)
17. `Select * from planten where soort='vast' and prijs between 3 and 5`
(resultaat : 17 rijen)
18. `Select * from planten where bl_b in (3,4,9,10)`
(resultaat : 15 rijen)
19. Access :
`Select art_code, plantennm from planten where plantennm like '*boom*'`
Standaard SQL :
`Select art_code, plantennm from planten where plantennm like '%boom%'`
(resultaat : 5 rijen)
20. Access :
`Select art_code, plantennm from planten where plantennm like '??N*'`
Standaard SQL :
`Select art_code, plantennm from planten where plantennm like '___N%'`
(resultaat : 11 rijen)
21. Zonder LIKE :
`select * from planten where soort in ('1-jarig', '2-jarig')`
Met LIKE
`select * from planten where soort like '[12]-jarig'`
(resultaat : 23 rijen)
22. `Select art_code, plantennm, soort, kleur, hoogte, bl_b, bl_e from planten where soort not in ('boom', 'heester') and hoogte between 100 and 200 and kleur in ('rood', 'blauw') and bl_b<8 order by soort, plantennm`
(resultaat : 4 rijen)
23. Access :
`Select art_code, plantennm from planten where plantennm like '*kruid*' and soort<>'kruid'`
Standaard SQL :
`Select art_code, plantennm from planten where plantennm like '%kruid%' and soort<>'kruid'`
(resultaat : 6 rijen)
24. Access :
`Select art_code, plantennm from planten where plantennm like 'l*e'`
Standaard SQL:
`Select art_code, plantennm from planten where plantennm like 'l%e'`
(resultaat : 4 rijen)
25. Access :
`Select art_code, plantennm from planten where plantennm like '[AM]*[AM]'`
Standaard SQL :

Select art_code, plantennm from planten where plantennm like '[AM]%[AM]'
(resultaat : 6 rijen)

26. Access :

Select art_code, plantennm from planten where plantennm like '?????'

Standaard SQL :

Select art_code, plantennm from planten where plantennm like '_____'

(resultaat : 7 rijen)

27. Access :

Select art_code, plantennm from planten where plantennm like '?????*'

Standaard SQL :

Select art_code, plantennm from planten where plantennm like '_____%'

(resultaat : 111 rijen)

2.1.2 Oefening SELECT deel 2

1. Select count(*) as aantal from leveran
(resultaat : 11)
2. Select avg(prijs) as gemiddelde from planten where soort = 'water'
(resultaat : 4,75)
3. Select max(hoogte) as maxhoogte from planten where soort='boom'
(resultaat : 4000)
4. Select avg(off_prijs), min(off_prijs), max(off_prijs) from offertes where lev_code='013'
(resultaat : 3,75; 0,1; 12,7)
5. Select min(off_prijs) as minimum from offertes where art_code='123'
(resultaat : 0,35)
6. Select bestelnr, art_lev, (aantal*best_pr) as totaal from bestreg
(resultaat : 94 rijen)
7. Select art_code, plantennm, prijs*1.05 as verhoogd from planten where soort='heester'
(resultaat : 14 rijen)
8. Select sum(bedrag) as totaal from bestel where lev_code='004'
(resultaat : 806,26)
9. Select sum(aantal) as aantal from bestreg where art_lev='B111'
(resultaat : 75)
10. Select sum(aantal*best_pr) as totaal from bestreg where art_lev='B331'
(resultaat : 47,75)

2.1.3 Oefeningen SELECT deel 3

1. Select soort, count(*) as aantal from planten group by soort
(resultaat : 10 rijen)
2. Select bestelnr, count(*) as aantal from bestreg group by bestelnr
(resultaat : 15 rijen)
3. Select soort, avg(prijs) as gemiddelde from planten group by soort
(resultaat : 10 rijen)
4. Select soort, kleur, count(*) as aantal from planten group by soort, kleur
(resultaat : 54 rijen)
5. Select kleur, avg(prijs) as gemiddelde from planten where soort='vast' group by kleur
(resultaat : 10 rijen)
6. Select lev_code, count(*) as aantal from offertes where lev_tijd<18 group by lev_code
(resultaat : 10 rijen)
7. Select soort, avg(prijs) as gemiddelde from planten where kleur <> 'geel' group by soort
(resultaat :10 rijen)
8. Select art_code, min(off_prijs) as mini, max(off_prijs) as maxi from offertes group by art_code
(resultaat : 119 rijen)
9. Select soort, avg(prijs) as gemiddelde from planten group by soort having count(*) >= 10
(resultaat : 5 rijen)
10. Select lev_tijd, avg(off_prijs) as gemiddelde from offertes group by lev_tijd
(resultaat : 4 rijen)
11. Select art_lev, min(best_pr) as mini, max(best_pr) as maxi from bestreg group by art_lev
(resultaat : 68 rijen)
12. Select bl_b, hoogte, kleur, count(*) as aantal from planten group by bl_b, hoogte, kleur
(resultaat : 109 rijen)
13. Select soort, min(prijs) as minimum from planten where bl_e>=6 and bl_b<=5 group by soort
(resultaat : 5 rijen)
14. select bl_e - bl_b as duurtijd, count(*) as aantal from planten group by bl_e - bl_b
(resultaat: 7 rijen)

als je wilt rekening houden met een bloeiperiode die in twee verschillende jaren liggen

```
select (bl_e - bl_b) as Bloeitijd, count(*) as Aantal from PLANTEN
where bl_e >= bl_b and bl_b <> 0 and bl_e <> 0
group by (bl_e - bl_b)
union
select (12 - bl_e + bl_b) as Bloeitijd, count(*) as Aantal from PLANTEN
where bl_e < bl_b and bl_b <> 0 and bl_e <> 0
group by (12 - bl_e + bl_b)
```

15. select round(prijs, 0) as afgeronde_prijs, count(*) as aantal
from planten group by round(prijs, 0)
(resultaat: 17 rijen)

2.1.4 Oefeningen SELECT deel 4

1. Select bestelnr, lev_naam, l_datum, bedrag from bestel inner join leveran on bestel.lev_code=leveran.lev_code
(resultaat : 15 rijen)
2. Select offertes.art_code, art_lev, plantennm, lev_tijd from offertes inner join planten on offertes.art_code=planten.art_code where lev_tijd<=10
(resultaat : 113 rijen)
3. Select planten.art_code, plantennm, leveran.lev_code, lev_naam, lev_tijd, off_prijs from planten inner join (offertes inner join leveran on offertes.lev_code=leveran.lev_code) on planten.art_code=offertes.art_code order by plantennm
(resultaat : 218 rijen)
4. Select planten.art_code, plantennm, prijs, lev_code, off_prijs from planten inner join offertes on offertes.art_code=planten.art_code where prijs>=(off_prijs*1.5)
(resultaat : 212 rijen)
5. SELECT bestreg.bestelnr, bestreg.art_lev, best_pr - off_prijs AS verschil FROM offertes INNER JOIN (bestel INNER JOIN bestreg ON bestel.bestelnr=bestreg.bestelnr) ON offertes.art_lev=bestreg.art_lev AND offertes.lev_code=bestel.lev_code
(resultaat : 93 rijen)
6. Select distinct woonpl from leveran inner join (offertes inner join planten on offertes.art_code=planten.art_code) on leveran.lev_code=offertes.lev_code where soort='vast'
(resultaat : 2 rijen, nl. Aalsmeer en Lisse)
7. Select planten.art_code, plantennm, soort, lev_naam, woonpl from planten inner join (offertes inner join leveran on offertes.lev_code=leveran.lev_code) on offertes.art_code=planten.art_code where woonpl<>'Aalsmeer' and kleur='rood' order by soort, plantennm
(resultaat : 10 rijen)
8. Select planten.art_code, plantennm, min(off_prijs) from planten inner join offertes on planten.art_code= offertes.art_code group by plantennm, planten.art_code order by 2
(resultaat : 119 rijen)
9. Select e.bestelnr as eerstenr, e.b_datum as eerstedatum, t.bestelnr as tweedenr, t.b_datum as tweededatum from bestel e inner join bestel t on e.b_datum=t.l_datum and e.bestelnr<>t.bestelnr
(resultaat : 2 rijen nl. de nrs 0184 – 0175 en 0181 en 0175)
10. Select e.lev_code as eerstelev, e.art_code as eerstecode, t.lev_code as tweedelev, t.art_lev as tweedecode from offertes e inner join offertes t on e.art_code=t.art_lev
(resultaat : 47 rijen)

11. Access :

```
Select bestelnr, l_datum, bedrag, 'Te laat' as bericht from bestel where  
l_datum<#1999-4-1#  
union  
Select bestelnr, l_datum, bedrag, '-----' as bericht from bestel where  
l_datum>=#1999-4-1#  
order by bestelnr
```

Standaard SQL :

```
Select bestelnr, l_datum, bedrag, 'Te laat' as bericht from bestel where  
l_datum<'1999-4-1'  
union  
Select bestelnr, l_datum, bedrag, '-----' as bericht from bestel where  
l_datum>='1999-4-1'  
order by bestelnr  
(resultaat : 15 rijen)
```

12.

```
Select offertes.art_code, plantennm, art_lev, 'Aalsmeer' as plaats from planten  
inner join (offertes inner join leveran on offertes.lev_code=leveran.lev_code)  
on planten.art_code=offertes.art_code where woonpl='Aalsmeer' and  
soort='boom'  
union  
Select offertes.art_code, plantennm, art_lev, 'buiten Aalsmeer' as plaats from  
planten inner join (offertes inner join leveran on  
offertes.lev_code=leveran.lev_code) on planten.art_code=offertes.art_code  
where woonpl<>'Aalsmeer' and soort='boom'  
(resultaat : 23 rijen)
```

2.1.5 Oefeningen SELECT deel 5

1. Select * from planten where hoogte > (select avg(hoogte) from planten)
(resultaat : 21 rijen)
2. Select * from planten where prijs > (select avg(prijs) from planten where
soort='boom')
(resultaat : 13 rijen)
3. Access :
Select * from leveran where lev_code in (select lev_code from bestel where
l_datum<#1999-4-1#)
Standaard SQL :
Select * from leveran where lev_code in (select lev_code from bestel where
l_datum<'1999-4-1')
(resultaat : 7 rijen)
4. Select * from offertes where off_prijs=(select min(off_prijs) from offertes)
(resultaat : 2 rijen; off_prijs = 0,05)
5. Select * from planten where hoogte > 0 and hoogte<(select min(hoogte) from
planten where soort='vast' and hoogte > 0)
(resultaat : 5 rijen)
6. SELECT * FROM planten WHERE hoogte>(SELECT AVG(hoogte) FROM
planten WHERE soort='vast') AND prijs<(SELECT AVG(prijs) FROM planten
WHERE soort='vast')
(resultaat : 13 rijen)
7. Select * from planten where prijs between (select min(prijs) from planten
where soort='klim') and (select max(prijs) from planten where soort='klim')
(resultaat : 11 rijen)
8. Select * from bestel where bedrag<(select sum(best_pr*aantal) from bestreg
where bestreg.bestelnr=bestel.bestelnr)*0.95
(resultaat : 2 rijen, nl bestelnrs 0191 en 0203)
9. Select o1.art_code, lev_naam, o1.off_prijs
from offertes o1 inner join leveran
on o1.lev_code=leveran.lev_code
where o1.off_prijs<(select avg(o2.off_prijs) from offertes o2 where
o2.art_code=o1.art_code) order by o1.art_code
(resultaat : 71 rijen)
10. Select bestreg.bestelnr, bestreg.art_lev, plantennm, best_pr
from planten inner join (offertes o1
inner join (bestreg inner join bestel
on bestreg.bestelnr=bestel.bestelnr)
on bestreg.art_lev=o1.art_lev and bestel.lev_code=o1.lev_code)
on planten.art_code=o1.art_code
where best_pr>

(select max(off_prijs) from offertes o2 where o2.art_code=planten.art_code)
(resultaat : 34 rijen)

3 RECORDS TOEVOEGEN

1. INSERT INTO leveran (lev_code, lev_naam, adres, woonpl)
VALUES ('045', 'GROEN BV.', 'ONDER DE LINDE 234', 'AALSMEER')
2. Select o1.art_code, plantenm, o1.lev_code, off_prijs into aanbied from
offertes o1 inner join planten on o1.art_code=planten.art_code where
off_prijs=(select min(off_prijs) from offertes o2 where
o2.art_code=o1.art_code)
(de tabel aanbied bevat 131 rijen)
3. Insert into bestreg values ('0205', 'C051', 10, 8.15)
Insert into bestreg values ('0205', 'B101', 200, 0.4)
Insert into bestreg values ('0205', 'B111', 25, 2.3)
Insert into bestreg values ('0205', 'G001', 50, 1.3)
Access
Insert into bestel values ('0205', '013', #4/23/1999#, #4/30/1999#, 261.28)
Standaard SQL
Insert into bestel values ('0205', '013', '4/23/1999', '4/30/1999', 261.28)

4 GEGEVENS WIJZIGEN

1. Access

Update bestel set l_datum=#4/5/1999# where bestelnr='0191'

Standaard SQL

Update bestel set l_datum='4/5/1999' where bestelnr='0191'

2. Update offertes set off_prijs=off_prijs*1.1 where offertes.art_code in (select planten.art_code from planten where soort='bol')

of

Standaard SQL

update offertes set off_prijs=off_prijs*1.1 from offertes inner join planten on offertes.art_code=planten.art_code where soort='bol'

Access

update offertes inner join planten on offertes.art_code=planten.art_code set off_prijs=off_prijs*1.1 where soort='bol'

5 GEGEVENS VERWIJDEREN

1. delete from offertes where lev_code='021' and art_code in (select art_code from planten where soort='heester')
2. delete from offertes where art_code in (select art_code from planten where soort='water')
delete from planten where soort='water'

6 TABELLEN

6.1 Oefeningen CREATE TABLE

Create table klachten

```
(  
klachtnr integer,  
art_code char(3),  
datum datetime,  
klacht varchar(100),  
status char(2)  
)
```

Access

Insert into klachten values (1, '316', #3/15/1999#, 'PLANTEN (LEV_CODE 019)
VERKOCHT OP 14-1-1999 VERTOONDEN BRUINE PLEKKEN NA CIRCA 2
MAANDEN', 'GL')

Standaard SQL

Insert into klachten values (1, '316', '3/15/1999', 'PLANTEN (LEV_CODE 019)
VERKOCHT OP 14-1-1999 VERTOONDEN BRUINE PLEKKEN NA CIRCA 2
MAANDEN', 'GL')

6.2 Oefeningen ALTER

Alter table planten add voorraad integer

6.3 Oefeningen CONSTRAINT

Eerst moeten we de primary key's definiëren :

```
Alter table planten add constraint pk_planten primary key (art_code)  
Alter table offertes add constraint pk_offertes primary key (lev_code, art_code)  
Alter table leveran add constraint pk_leveran primary key (lev_code)  
Alter table bestel add constraint pk_bestel primary key (bestelnr)  
Alter table bestreg add constraint pk_bestreg primary key (bestelnr,art_lev)
```

Daarna definiëren we de relaties :

```
alter table offertes add constraint f_art_code foreign key (art_code) references  
planten (art_code)  
alter table offertes add constraint f_lev_code foreign key (lev_code) references  
leveran (lev_code)  
alter table bestel add constraint f_lev_code2 foreign key (lev_code) references  
leveran (lev_code)  
alter table bestreg add constraint f_bestel foreign key (bestelnr) references bestel  
(bestelnr)
```

7 INDEXEN

1. Create index bsridx01 on bestreg (bestelnr, art_lev)
2. Create index i_art_code on offertes (art_code)
create index i_art_lev on offertes (art_lev)

8 VIEWS

1. create view vastlaag as
select * from planten where soort like 'vast' and hoogte<=15
(de view bevat 1 rijen)
2. create view gem_o_pr as
select art_code, min(off_prijs) as min_off,
max(off_prijs) as max_off, avg(off_prijs) as gem_off from offertes
group by art_code
(de view bevat 107 rijen)
3. create view lev014 as
select offertes.art_code, planten.PLANTENNM, ART_LEV, OFF_PRIJS,
planten.PRIJS
from offertes inner join planten on planten.art_code=offertes.art_code
where lev_code='014'
(de view bevat 30 rijen)
4. create view bomen as select art_code, plantennm, hoogte, prijs
from planten where soort like 'boom'
(de view bevat 17 rijen)
5. create view planten_ib as select bestreg.bestelnr, bestreg.art_lev, plantennm
from planten inner join (offertes inner join (bestel inner join bestreg on
bestel.bestelnr=bestreg.bestelnr) on offertes.art_lev=bestreg.art_lev and
offertes.lev_code=bestel.lev_code) on planten.art_code=offertes.art_code
(de view bevat 90 rijen)
6. create view zomerpl as select ART_CODE, PLANTENNM, SOORT, PRIJS from
planten where bl_b between 6 and 8
(de view bevat 64 rijen)
7. create view lisse as select offertes.* from offertes inner join leveran on
offertes.lev_code=leveran.lev_code where woonpl like 'lisse'
(de view bevat 53 rijen)

9 EINDOEFENING

1. select pres_name, birth_yr from presiden
2. select distinct state_born from presiden
3. select state_born, count(*) as num from presiden group by state_born
4. select presiden.pres_name, mar_year as mar_yr, spouse_nam, nr_childre from pres_mar
5. select pres_name from (select top 1 pres_name,sum(nr_childre) from pres_mar group by pres_name order by 2 desc)
6. select pres_name, count(*) as num_mar from pres_mar group by pres_name having count(*)>1
7. select pres_name, birth_yr, birth_yr+death_age as death_yr from presiden where pres_name not in (select pres_name from pres_mar)
8. select pres_name, hobby from pres_hob order by hobby
9. create view oef8 as select pres_name, yrs_serv from presiden p1 where yrs_serv>(select avg(yrs_serv) from presiden p2 where p2.party=p1.party)
10. select state_name from state where year_enter in (select mar_year-sp_age from presiden inner join pres_mar on presiden.pres_name=pres_mar.pres_name where pr_age=(select min(pr_age) from pres_mar))
11. select birth_yr as jaar, 'geboorte' as boodschap, pres_name as naam, state_born as extra from presiden
 union
 select birth_yr+death_age, 'overlijden', pres_name, state_born from presiden
 union
 select mar_year, 'huwelijk', presiden.pres_name, spouse_nam from presiden inner join pres_mar on presiden.pres_name=pres_mar.pres_name
 union
 select electio_ye, 'winnaar verkiezingen', candidate, party from election inner join presiden on election.candidate=presiden.pres_name where winnerlose='W'
 union
 select electio_ye, 'verliezen verkiezingen', candidate, party from election left join presiden on election.candidate=presiden.pres_name where winnerlose='L'
 union
 select year_enter, 'toetreding state bij',state_name, presiden.pres_name from state, administ, presiden where administ.pres_name=presiden.pres_name and state.year_enter >=administ.year_inaug and state.year_enter <=administ.year_inaug+presiden.yrs_serv
 union
 select mar_year-sp_age, 'geboorte echtgenote', spouse_nam, presiden.pres_name from presiden inner join pres_mar on presiden.pres_name=pres_mar.pres_name order by 1
12. create view oef11 as select pres_name from presiden where pres_name not in (select candidate from election)
13. select top 1 state_born, count(*) as aantal from presiden group by state_born order by 2 desc
14. select presiden.pres_name, count(pres_mar.pres_name) as aantal_keer_gehuwd, sum(nr_childre) as aantal_kinderen from presiden left join pres_mar on presiden.pres_name=pres_mar.pres_name group by presiden.pres_name
15. select pres_mar.pres_name from administ inner join pres_mar on administ.pres_name = pres_mar.pres_name where mar_year between year_inaug and year_inaug+4
 in de veronderstelling dat een termijn 4 jaar duurt, een ander oplossing is:
 select pres_mar.pres_name from pres_mar

where mar_year >=(select top 1 a1.year_inaug from administ a1 where
pres_mar.pres_name = a1.pres_name order by a1.year_inaug asc)
and mar_year <=(select top 1 a2.year_inaug+ presiden.yrs_serv from administ
a2 inner join presiden on a2.pres_name = presiden.pres_name where
pres_mar.pres_name = a2.pres_name order by a2.year_inaug asc)

16. select pres_name from presiden where state_born in (select state_born from
presiden group by state_born having count(*)>=2)

10COLOFON

Sectorverantwoordelijke: Ortaire Uyttersprot

Cursusverantwoordelijke: Jean Smits

Didactiek: Werkgroep Informatica Basis

Lay-out: Leyman Eugène

Medewerkers: Brigitte Van Ceulebroeck

Versie: 7/5/2010

Nummer dotatielijst: