



**Vlaamse Dienst voor Arbeidsbemiddeling en
Beroepsopleiding**

**SQL
(MET MYSQL)
OPLOSSINGEN VAN DE OEFENINGEN**

Deze cursus IS eigendom van de VDAB©

Inhoudsopgave

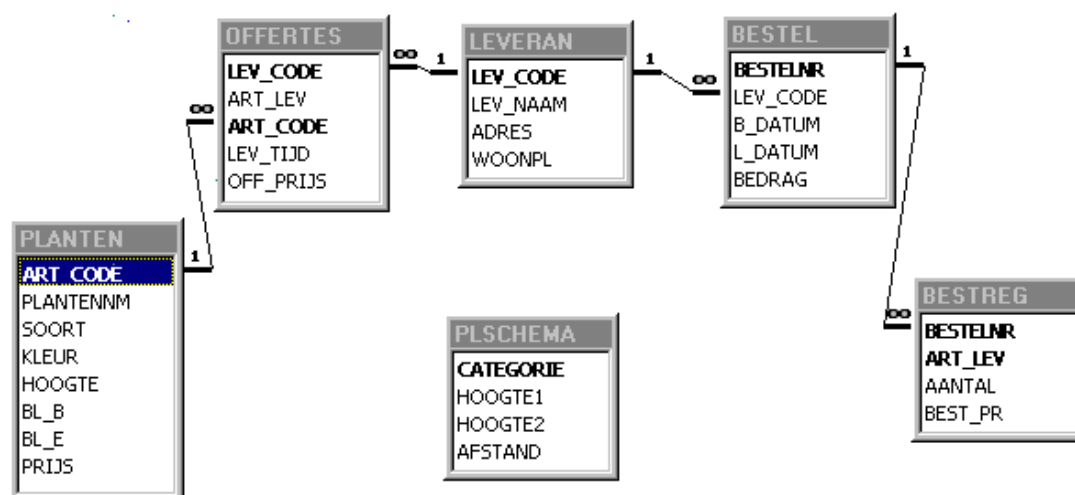
1	DE DATABASE PLANTV	3
2	GEGEVENS SELECTEREN	4
2.1.1	Oefeningen SELECT deel 1	4
2.1.2	Oefening SELECT deel 2	6
2.1.3	Oefeningen SELECT deel 3	7
2.1.4	Oefeningen SELECT deel 4	9
2.1.5	Oefeningen SELECT deel 5	11
3	RECORDS TOEVOEGEN	12
4	GEGEVENS WIJZIGEN.....	13
5	GEGEVENS VERWIJDEREN.....	14
6	TABELLEN	15
6.1	Oefeningen CREATE TABLE	15
6.2	Oefeningen ALTER.....	15
6.3	Oefeningen CONSTRAINT en INDEX.....	15
7	VIEWS	17
8	EINDOEFENING (NIEUWE VERSIE)	18
9	COLOFON	21

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 * FROM bestel b WHERE b_datum >='1999-03-14'
(resultaat: 6 rijen)
5. SELECT art_code, art_lev, lev_code FROM offertes ORDER BY art_code,
art_lev
(resultaat : 218 rijen)
6. SELECT * FROM planten WHERE soort='water' ORDER BY hoogte DESC
(resultaat : 9 rijen)
7. SELECT DISTINCT kleur FROM planten
(resultaat : 13 rijen)
8. SELECT * FROM planten WHERE kleur IS NULL
(resultaat : 11 rijen)
9. SELECT DISTINCT soort FROM planten
(resultaat : 10 rijen)
10. SELECT art_code, plantennm, hoogte, bl_b FROM planten WHERE
kleur='geel' AND soort='vast'
(resultaat : 3 rijen)
11. SELECT * FROM planten WHERE prijs > 10 AND soort<>'boom'
(resultaat : 8 rijen)
12. SELECT * FROM planten WHERE (bl_b=6 AND kleur='geel') OR (bl_b=8
AND kleur='rood')
(resultaat : 2 rijen)
13. SELECT art_code, plantennm, hoogte FROM planten WHERE
kleur='gemengd' AND hoogte<=60
(resultaat : 13 rijen)
14. SELECT * FROM leveran WHERE woonpl<>'hillegom'
(resultaat : 9 rijen)
15. SELECT art_code, plantennm, kleur, hoogte FROM planten WHERE kleur IS
NULL AND hoogte IS NULL
(resultaat : geen rijen)

16. SELECT art_code, plantennm FROM planten WHERE 8 <= bl_b AND bl_e <= 10
(resultaat : 12 rijen)
17. SELECT art_code, plantennm FROM planten WHERE bl_e >= 9 AND bl_b <=9
(resultaat : 26 rijen)
18. SELECT * FROM planten WHERE soort='vast' AND prijs BETWEEN 3 AND 5
(resultaat : 17 rijen)
19. SELECT * FROM planten WHERE bl_b IN (3,4,9,10)
(resultaat : 15 rijen)
20. SELECT art_code, plantennm FROM planten WHERE plantennm LIKE '%boom%'
(resultaat : 5 rijen)
21. SELECT art_code, plantennm FROM planten WHERE plantennm LIKE '___n%'
(resultaat : 11 rijen)
22. SELECT * FROM planten WHERE soort IN ('1-jarig', '2-jarig')
(resultaat : 23 rijen)
23. 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)
24. SELECT art_code, plantennm FROM planten WHERE plantennm LIKE '%kruid%' AND soort<>'kruid'
(resultaat : 6 rijen)
25. SELECT art_code, plantennm FROM planten WHERE plantennm LIKE 'l%e'
(resultaat : 4 rijen)
26. SELECT art_code, plantennm FROM planten WHERE plantennm LIKE '_____
(resultaat : 7 rijen)
27. SELECT art_code, plantennm FROM planten WHERE plantennm LIKE '_____%'
(resultaat : 111 rijen)
28. SELECT * FROM planten ORDER BY prijs DESC LIMIT 10
(resultaat: 10 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 :)
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)
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, plantenm, 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, plantenm, 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 plantenm
(resultaat : 218 rijen)
4. SELECT planten.art_code, plantenm , 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, plantenm, 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, plantenm
(resultaat : 10 rijen)
8. SELECT planten.art_code, plantenm, MIN(off_prijs) FROM planten INNER JOIN offertes ON planten.art_code= offertes.art_code GROUP BY plantenm, 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. SELECT bestelnr, l_datum, bedrag, 'GELEVERD' 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. 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 : 68 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. INSERT INTO bestreg
VALUES
('0205', 'C051', 10, 8.15),
('0205', 'B101', 200, 0.4),
('0205', 'B111', 25, 2.3),
('0205', 'G001', 50, 1.3);

INSERT INTO bestel VALUES ('0205', '013', '1999-04-23', '1999-04-30',
261.28)

of nog beter:

INSERT INTO bestel VALUES
('0205', '013', '1999-04-23', '1999-04-30',
(SELECT SUM(aantal * best_pr) AS subtotaal FROM bestreg
WHERE bestelnr='0205' GROUP BY bestelnr)* 0.92)

4 GEGEVENS WIJZIGEN

1. UPDATE bestel SET l_datum='1999-04-05' 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

```
UPDATE offertes INNER JOIN planten
ON offertes.art_code=planten.art_code
SET off_prijs=off_prijs*1.1 WHERE soort='bol'
```

(7 rijen geupdate)

5 GEGEVENS VERWIJDEREN

1. DELETE FROM offertes WHERE lev_code='021' AND art_code IN (SELECT art_code FROM planten WHERE soort='heester')

of

```
DELETE offertes.* FROM offertes INNER JOIN planten ON  
offertes.art_code=planten.art_code WHERE lev_code='021' AND soort =  
'heester'
```

(14 rijen verwijderd)

2. DELETE FROM offertes WHERE art_code IN (SELECT art_code FROM planten WHERE soort='water')
DELETE FROM planten WHERE soort='water'

of

```
DELETE offertes.*, planten.* FROM planten INNER JOIN offertes ON  
offertes.art_code=planten.art_code WHERE soort='water'
```

(18 rijen verwijderd IN totaal)

6 TABELLEN

6.1 Oefeningen CREATE TABLE

1. CREATE TABLE klachten

```
(  
  klachtnr INTEGER,  
  art_code CHAR(3),  
  datum datetime,  
  klacht VARCHAR(100),  
  `status` CHAR(2)  
)
```

```
INSERT INTO klachten  
VALUES (1, '316', '1999-3-15', 'planten (lev_code 019) verkocht op 14-1-  
1999 vertoonden bruine plekken na circa 2 maanden', 'GL')
```

2. CREATE TABLE aanbiedingen
SELECT o.art_code, p.plantenm, MIN(o.off_prijs) AS MIN_prijs,
o.lev_code FROM offertes o
INNER JOIN planten p
ON p.art_code = o.art_code
GROUP BY o.art_code

Bovenstaande oplossing geeft per plant slechts één leverancier, ook als er meerdere leveranciers zijn met dezelfde goedkoopste prijs. Een betere oplossing is:

```
CREATE TABLE aanbiedingen  
SELECT p.plantenm, o1.art_code, o1.lev_code, o2.min_prijs from offertes  
o1  
INNER JOIN (  
  SELECT art_code, min(off_prijs) as min_prijs  
  FROM offertes GROUP BY art_code  
) o2  
ON o1.off_prijs=o2.min_prijs AND o1.art_code=o2.art_code  
INNER JOIN planten p  
ON p.art_code = o1.art_code
```

3. CREATE TABLE oudebestellingen LIKE bestel

6.2 Oefeningen ALTER

1. ALTER TABLE planten ADD voorraad INTEGER
2. ALTER TABLE planten modify column kleur VARCHAR(7) DEFAULT "groen"

6.3 Oefeningen CONSTRAINT en INDEX

1. 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)
```

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

7 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)

8 EINDOEFENING (NIEUWE VERSIE)

1. SELECT pres_name, birth_yr FROM presiden (42 records)
2. SELECT DISTINCT state_born FROM presiden (19 records)
3. SELECT state_name FROM state WHERE state_name LIKE "New%" OR state_name LIKE "%ia" (10 records)
4. DELETE FROM state where state_name = "New Georgia" (1 record)
5. SELECT pres_name, spouse_nam, (pr_age-sp_age) AS leeftijdsverschil FROM pres_mar WHERE pr_age>=sp_age ORDER BY leeftijdsverschil DESC LIMIT 10 (10 records)
6. SELECT pres_name, birth_yr AS geboortejaar, birth_yr + death_age AS sterfjaar FROM presiden WHERE pres_name NOT IN (SELECT pres_name FROM pres_mar)

of

SELECT P1.pres_name, P1.birth_yr AS geboortejaar, P1.birth_yr + P1.death_age AS sterfjaar FROM presiden P1 LEFT JOIN pres_mar P2 ON P1.pres_name = P2.pres_name WHERE P2.pres_name IS NULL (1 record)
7. SELECT s.state_name, COUNT(p.pres_name) AS aantal FROM state s LEFT JOIN presiden p ON s.state_name = p.state_born GROUP BY s.state_name ORDER BY aantal DESC (50 records)
8. SELECT vice_pres, count(*) FROM administ WHERE vice_pres IN(select pres_name from administ) GROUP BY vice_pres (10 records)
9. SELECT pres_name, year_inaug, (SELECT MIN(year_inaug) FROM administ WHERE year_inaug > 1937) AS eindjaar FROM administ WHERE year_inaug >= (SELECT MAX(year_inaug) FROM administ WHERE year_inaug <= 1937) AND year_inaug < (SELECT MIN(year_inaug) FROM administ WHERE year_inaug > 1937) (1 record)
10. CREATE TABLE familie SELECT p.pres_name AS president, p.birth_yr AS geboortejaar, p.birth_yr + death_age AS overlijdensjaar, m.mar_year AS huwelijksjaar, electio_je AS verkiezingsjaar, m.spouse_nam AS vrouw, nr_childre AS kinderen FROM presiden p LEFT JOIN pres_mar m ON p.pres_name = m.pres_name LEFT JOIN election e ON p.pres_name = e.candidate AND winnerlose='W' ORDER BY p.pres_name (66 records)

11. UPDATE familie SET overlijdensjaar = NULL
WHERE overlijdensjaar = geboortjaar
(6 records)
12. ALTER TABLE familie
ADD COLUMN pres_id INTEGER AUTO_INCREMENT PRIMARY KEY,
ADD INDEX i_namen (president)
(48 records)
13. CREATE TABLE hobbies LIKE pres_hob
(no resultset)
14. INSERT INTO hobbies SELECT * FROM pres_hob
(62 records)
15. SELECT h.pres_name FROM hobbies h LEFT JOIN familie f ON
f.president=h.pres_name WHERE f.president IS NULL

of

```
SELECT pres_name FROM hobbies WHERE pres_name NOT IN  
(SELECT president FROM familie)  
(5 records)
```

daarna

```
UPDATE hobbies SET pres_name="Bush G W" WHERE pres_name="Bush"  
(2 records)  
UPDATE hobbies SET pres_name="Hayes R B" WHERE  
pres_name="Hayens R B"  
(3 records)
```

16. ALTER TABLE hobbies ADD COLUMN pres_id INT FIRST
(62 records)

daarna

```
UPDATE hobbies h INNER JOIN familie f  
ON h.pres_name = f.president  
SET h.pres_id = f.pres_id  
(62 records)
```

17. ALTER TABLE hobbies ADD CONSTRAINT fk_president FOREIGN KEY
(pres_id) REFERENCES familie (pres_id)
ON DELETE RESTRICT ON UPDATE CASCADE
(62 records)

daarna

```
DELETE FROM familie WHERE president = "Adams J Q"  
(msg: "cannot delete or update a parent row: a foreign key constraint fails..")
```

18. CREATE VIEW hobbies_overzicht AS
SELECT hobby, count(*) AS aantal FROM hobbies

GROUP BY hobby ORDER BY count(*) DESC

19.

```
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
(387 records)
```
20.

```
SELECT distinct f1.president, f1.geboortejaar,f2.president, f2.jaaroverlijden
FROM familie f1 INNER JOIN familie f2 ON f1.geboortejaar=f2.jaaroverlijden
```

9 COLOFON

Sectorverantwoordelijke:	Ortaire Uyttersprot
Cursusverantwoordelijke:	Jean Smits
Didactiek:	Werkgroep Informatica Basis
Lay-out:	Leyman Eugène
Medewerkers:	Brigitte Van Ceulebroeck Jan Vandorpe Jan De Tavernier Siska Corneillie
Versie:	06/01/2015
Nummer dotatielijst:	