

Tentamen

TI1500 Web- en Databasetechnologie

Vrijdag, 28 januari 2011,
08.30-12.30

Dit tentamen bestaat uit 7 open vragen
Totaal aantal pagina's (inclusief deze en appendix): 11

Bij dit tentamen hoort een appendix met een aantal figuren waarnaar verwezen wordt in de opdrachten.

Opgave	Punten
1	12
2	12
3	8
4	8
5	15
6	30
7	15
Totaal	100

NB. Zorg er voor dat je genoeg tijd overhoudt voor opgave 6 omdat deze relatief veel tijd kan kosten.

Het gebruik van het boek "*Web Programming Step by Step*" is toegestaan, evenals de slides die voor het vak op Blackboard zijn gezet. Het gebruik van eigen aantekeningen is niet toegestaan.

Vul je naam, studienummer en studierichting in op ieder antwoordblad.

Deze opgaven dienen aan het eind van het tentamen weer ingeleverd te worden.

Opgave 1 -- WWW, HTML en CSS (12 punten)

- a) [4 pt.] Welke tags mogen in XHTML gebruikt worden direct binnen een `table` element? En welke binnen een `tr` element?.
- b) [2 pt.] Als je twee opeenvolgende elementen naar dezelfde kant van de pagina *float*, wat gebeurt er dan? Waar komen ze ten opzichte van elkaar te staan?.
- c) [6 pt.] Beschouw de volgende (afgekorte) HTML code:

```
<div id="main">
  <h1>Futurama</h1>
  
  <p>Futurama is een Amerikaanse animatieserie, ...</p>
  <p>In de Verenigde Staten werden de eerste ...</p>
  <p>De serie draait om het personage Fry, ... </p>
</div>
```

Welke CSS code is nodig om de volgende output te krijgen: (De rand rond het `h1` element moet rood zijn.)

Futurama

Futurama is een Amerikaanse animatieserie, gemaakt door de Rough Draft Studio's in Glendale en ontwikkeld door Matt Groening (ook de maker van The Simpsons) en David X. Cohen. Futurama, bekroond met een Emmy Award, is genoemd naar een onderdeel van de wereldtentoonstelling van 1939, dat een beeld gaf hoe het leven er in de toekomst uit zou zien.



In de Verenigde Staten werden de eerste vijf seizoenen uitgezonden op Fox. Deze liepen van 28 maart 1999 t/m 10 augustus 2003. Na enkele jaren van stilte, werden er tussen 2007 en 2009 vier films geproduceerd voor directe videoverkoop. Deze bleken zo succesvol dat Comedy Central besloot een zesde seizoen te bestellen. Twaalf afleveringen werden tijdens de zomer van 2010 uitgezonden. De eindejaarsspecial werd uitgezonden op 21 november 2010 en de rest van de reeks volgt in 2011.

De serie draait om het personage Fry, een pizzabezorger uit New York die zichzelf per ongeluk invriest op 1 januari 2000. Pas 1000 jaar later, in het jaar 3000, wordt hij weer ontdooid.

Opgave 2 – PHP en Forms (12 punten)

- a) [4 pt.] Schrijf PHP-code ingebed in HTML voor het genereren van een ongeordende lijst van het volgende formaat:
- 99 bottles of beer on the wall
 - 98 bottles of beer on the wall
 - ...
 - 1 bottles of beer on the wall
 - 0 bottles of beer on the wall
- b) [4 pt.] Wat zijn de waarden van elk van de volgende arrays nadat de functie `mysterie` is uitgevoerd?

```
function mysterie($list) {  
    for ($i = 0; $i < count($list) - 1; $i++) {  
        if ($list[$i] > $list[$i + 1]) {  
            $list[$i + 1]++;  
        }  
    }  
}
```

- `$a = array(14, 7);`
`mysterie($a);`
 - `$b = array(7, 1, 3, 2, 0, 4);`
`mysterie($b);`
 - `$c = array(10, 8, 9, 5, 5);`
`mysterie($c);`
 - `$d = array(12, 11, 10, 10, 8, 7);`
`mysterie($d);`
- c) [4 pt.] Als de gebruiker de naam “Jon Stewart” en beide opties aanklikt in het formulier beneden, welke URL zal dan opgehaald worden door het formulier, inclusief de volledige query string? Wat is er slecht aan deze formulier code?

```
<form action=http://webster.cs.washington.edu/params.php>  
    <div>  
        Naam: <input type="text" name="name" />  
        Geslacht:  
        <input type="checkbox" name="man" /> Man  
        <input type="checkbox" name="vrouw" /> Vrouw  
        <input type="submit" />  
    </div>  
</form>
```

Opgave 3 – JavaScript en DOM (8 punten)

- a) [2 pt.] Wat gebeurt er in JavaScript als je een waarde toekent aan een variabele die nog niet eerder gedeclareerd is?
- b) [3 pt.] Wat zijn de waarden van de variabelen a, b, c, d en e op het eind in de volgende Javascript code?

```
var a = 1;
b = 2;
var c = 3;
function f() {
  a = 4;
  var c = 5;
  var d = 6;
  e = 6;
}
f();      // roep de functie f aan
```

- c) [3 pt.] Beschouw het volgende stukje HTML. Hoeveel *children* heeft de *footer* knoop in het DOM, en wat zijn ze? Hoeveel *siblings* heeft deze knoop, en welke zijn ze?

```
<p>Contactadres:</p>
<div id="footer">
  <p>Delfgauwseweg 34</p>
  <p>Delft</p>
</div>
```

Opgave 4 – Events, Prototype, Ajax en XML (8 punten)

- a) [3 pt.] Hoe zou je met gebruik van Prototype de DOM objecten opvragen voor alle paragrafen (p elementen) op de pagina met de CSS klasse *story* en die binnen een div zitten met "container" als id.
- b) [3 pt.] Als een *key event* gebeurt, hoe bepaal je dan welk karakter ingetikt werd? Hoe kun je zien welke speciale *modifier*-toets zoals *Shift* of *Ctrl* ook tegelijkertijd ingedrukt werd?
- c) [2 pt.] Wat is het verschil tussen synchrone en asynchrone communicatie tussen een web-browser en een web-server? Welke stijl gebruikt Ajax, en waarom?

Opgave 5 – SQL (15 punten)

Geef voor elk van de hierna volgende vragen een SQL-query over de COMPANY database (het 'running example' uit de slides). Het schema van deze database wordt gegeven in Figuur 1 in de Appendix. Voor de syntax van SQL, zie Figuur 4 uit de Appendix.

- a) [5 pt.] Geef een lijst met de achternamen (*Lname*) van alle medewerkers die een familielid hebben in de DEPENDENT tabel dat geboren is voor 1985. Ieder personeelslid met deze eigenschap moet precies één keer voorkomen in het resultaat. N.B. 1 april 2009 wordt genoteerd als '2009-04-01'.

- b) [10 pt.] Geef de projecten (*Pnumber, Pname*) waarvoor geldt dat alle vrouwen die participeren in dat project ook afdelingsmanager zijn (*Super_ssn* in DEPARTMENT).

Opgave 6 – EER modelleren (30 punten)

Maak een EER diagram voor de hieronder omschreven toepassing. Specificeer elke aanname die je verwerkt in je diagram en die niet uit de beschrijving afgeleid kan worden. Denk verder aan de eerste E in de afkorting EER, specialisatie en dat soort dingen. Tenslotte, je krijgt strafpunten voor elk onnodig toegevoegd sleutelattribuut dat niet genoemd wordt in de beschrijving. Zie verder Figuur 3 uit de Appendix voor een overzicht van de ER-symbolen en Figuur 2 voor enkele EER-symbolen.

De toepassing:

Het bedrijf MiE (“makelaars in effecten”) beheert aandelenportefeuilles voor cliënten. Het bedrijf bezit een aantal kantoren die uniek vastgelegd worden door de combinatie van hun naam en de vestigingsplaats (MiE is ontstaan door fusie van een aantal vroegere makelaarskantoren). Daarnaast worden van die kantoren het adres en één telefoonnummer (van de receptie) opgenomen in de database.

Iedere portefeuille is ondergebracht in precies één van de kantoren. Elke portefeuille heeft een volgnummer. Dat nummer is uniek binnen een kantoor, maar hetzelfde nummer kan in meerdere kantoren voorkomen.

Er zijn cliënten van twee soorten. Allereerst zijn er de zogenaamde “particulieren”, personen van wie het unieke sofinummer bijgehouden wordt en ook naam, adres, woonplaats en telefoonnummer(s). Een cliënt kan ook een vereniging zijn, die bestaat uit 1 of meer leden waarvan dezelfde informatie als van de particulieren in de database opgenomen dient te worden. Een vereniging heeft daarnaast een unieke naam, en een uniek lid die als contactpersoon fungeert (je hoeft in je schema niet af te dwingen dat deze contactpersoon lid is van de betreffende vereniging). Iedere cliënt heeft een cliëntnummer dat over heel het bedrijf MiE uniek is.

Iedere cliënt bezit tenminste één portefeuille. Wel kan het zijn dat een portefeuille in het bezit is van meer dan één cliënt, zelfs combinaties van particulieren en verenigingen zijn mogelijk. Een portefeuille wordt beheerd door één makelaar, in dienst van MiE. Van iedere makelaar wordt onder meer zijn opleiding bijgehouden (een tekstveld niet verder gespecificeerd). Een makelaar is aan één kantoor verbonden. Dat geldt niet voor de overige personeelsleden: deze kunnen werken in meer dan één kantoor maar ook in geen enkel kantoor. Van elk personeelslid wordt bijgehouden: sofinummer, naam, adres, woonplaats, (huis-) telefoonnummer(s), werktijd (aantal uren per week), salaris, een uniek personeelsnummer, en voor elk kantoor waar die medewerker actief is, het bijbehorende doorkies telefoonnummer. Personeelsleden mogen geen particuliere cliënten zijn. Wel mogen ze lid van aangesloten verenigingen zijn, maar ze mogen niet als contactpersoon fungeren.

Opgave 7 -- ER Model implementatie (15 punten)

Beschouw het EER diagram in Figuur 2 uit de Appendix. Vertaal dit diagram in een relationeel database schema volgens het standaard algoritme dat tijdens de les en in de slides is uitgelegd. Noteer de tabellen zoals in Figuur 1 van de Appendix, en voeg (zoals ook in Figuur 1 gedaan is) voor elk foreign key – key paar een pijl toe van de foreign key naar de bijbehorende key. Denk erom de primaire sleutels te onderstrepen. Het is niet toegestaan primaire sleutels toe te voegen die niet als attribuut voorkomen in het EER diagram.

Ter verduidelijking de beschrijving van de case:

Een rijschool geeft autorijlessen. Daarvoor heeft de rijschool instructeurs in dienst. Van elke instructeur wordt bijgehouden wat de naam en het adres is. Bovendien heeft elke instructeur een uniek werknemersnummer.

In de klantenregistratie staat voor elke klant een uniek klantnummer, de naam van de klant en zijn of haar adres. Elke klant heeft een toegewezen instructeur waarvan deze normaalgesproken les krijgt.

Van de lesauto's wordt het merk en het type bijgehouden, en bovendien het bouwjaar en het kenteken. De klant kan een afspraak maken voor een rijles met instructeur, meestal de toegewezen instructeur. Voor elke rijles wordt een begintijdstip afgesproken, en een afhaaladres waar de les begint. Bij elke afspraak wordt ook vastgelegd welk voertuig gebruikt zal worden.

De klant kan deelnemen aan klassikaal gegeven theorielessen en dit wordt geregistreerd. Deze lessen zijn op een bepaalde datum en er is dan altijd maar een les. Aan het eind van elke les is er een klein proefexamen waaraan klanten kunnen meedoen en de score daarvan wordt bijgehouden om de voortgang van elke klant te kunnen volgen.

Als een klant praktijkexamen gaat doen, dan wordt bijgehouden wanneer dit is. Na afloop wordt ook opgeslagen wat het resultaat was.

Einde van het tentamen

APPENDIX

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Super_ssn	Mgr_start_date
-------	----------------	-----------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

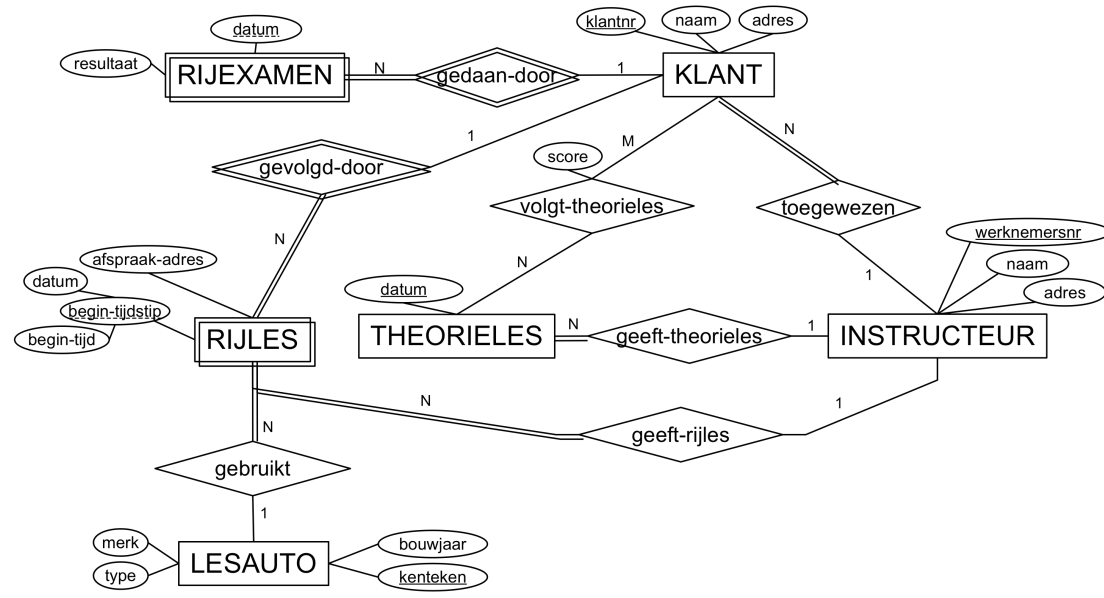
WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

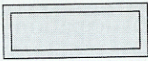
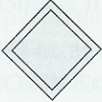
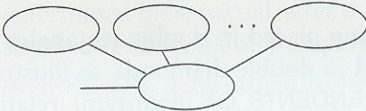
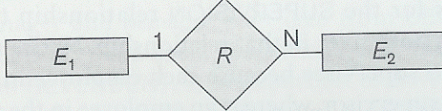
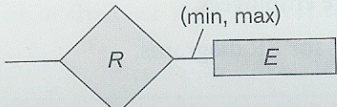
<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Figuur 1. Schema van de COMPANY database (zie opgave 5)



Figuur 2. EER diagram Rijkschool (zie opgave 7)

Figure 3.14
Summary of the
notation for ER
diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for $E_1:E_2$ in R
	Structural Constraint (min, max) on Participation of E in R

Figuur 3. De ER symbolen en hun betekenis

Table 8.2

Summary of SQL Syntax

```

CREATE TABLE <table name> ( <column name> <column type> [ <attribute constraint> ]
                             { , <column name> <column type> [ <attribute constraint> ] }
                             [ <table constraint> { , <table constraint> } ] )

DROP TABLE <table name>

ALTER TABLE <table name> ADD <column name> <column type>

SELECT [ DISTINCT ] <attribute list>
FROM ( <table name> { <alias> } | <joined table> ) { , ( <table name> { <alias> } | <joined table> ) }
[ WHERE <condition> ]
[ GROUP BY <grouping attributes> [ HAVING <group selection condition> ] ]
[ ORDER BY <column name> [ <order> ] { , <column name> [ <order> ] } ]

<attribute list> ::= ( * | ( <column name> | <function> ( ( [ DISTINCT ] <column name> | * ) ) )
                  { , ( <column name> | <function> ( ( [ DISTINCT ] <column name> | * ) ) ) } )

<grouping attributes> ::= <column name> { , <column name> }

<order> ::= ( ASC | DESC )

INSERT INTO <table name> [ ( <column name> { , <column name> } ) ]
( VALUES ( <constant value> , { <constant value> } ) { , ( <constant value> { , <constant value> } ) }
| <select statement> )

DELETE FROM <table name>
[ WHERE <selection condition> ]

UPDATE <table name>
SET <column name> = <value expression> { , <column name> = <value expression> }
[ WHERE <selection condition> ]

CREATE [ UNIQUE ] INDEX <index name>
ON <table name> ( <column name> [ <order> ] { , <column name> [ <order> ] } )
[ CLUSTER ]

DROP INDEX <index name>

CREATE VIEW <view name> [ ( <column name> { , <column name> } ) ]
AS <select statement>

DROP VIEW <view name>

```

NOTE: The commands for creating and dropping indexes are not part of standard SQL2.

Figuur 4. SQL syntax

Uitwerkingen van het Tentamen

TI1500 Web- en Databasetechnologie

Vrijdag, 28 januari 2011,
08.30-12.30

Dit tentamen bestaat uit 7 open vragen
Totaal aantal pagina's (inclusief deze en appendix): 11

Bij dit tentamen hoort een appendix met een aantal figuren waarnaar verwezen wordt in de opdrachten.

Opgave	Punten
1	12
2	12
3	8
4	8
5	15
6	30
7	15
Totaal	100

NB. Zorg er voor dat je genoeg tijd overhoudt voor opgave 6 omdat deze relatief veel tijd kan kosten.

Het gebruik van het boek "*Web Programming Step by Step*" is toegestaan, evenals de slides die voor het vak op Blackboard zijn gezet. Het gebruik van eigen aantekeningen is niet toegestaan.

Vul je naam, studienummer en studierichting in op ieder antwoordblad.

Deze opgaven dienen aan het eind van het tentamen weer ingeleverd te worden.

Opgave 1 -- WWW, HTML en CSS (12 punten)

- d) Binnen een `table` element: `tr` en `caption`. Binnen een `tr` element: `th` en `td`. Merk op dat dit is wat in het boek genoemd is. Extra niet in het boek genoemde mogelijkheden (en dus niet vereist als antwoord): binnen `table` kan ook `col`, `colgroup`, `thead`, `tfoot`, en `tbody`.
- e) Als je twee opeenvolgende elementen naar dezelfde kant van de pagina *float*, dan worden ze horizontaal gestapeld, wat wil zeggen dat de eerste zo ver mogelijk float in de aangegeven richting, en de tweede ook tot hij tegen de eerste aan komt, als daar tenminste genoeg ruimte voor is. Zo niet, bijvoorbeeld omdat het blok-elementen zonder een *width* zijn en daarom de gehele breedte innemen, dan komt de tweede onder de eerste te staan.

```
f)  h1 {
      float: left;
      border: 5px solid red;
      padding: 5px;
      margin: 10px;
    }

    img {
      float: right;
      margin: 10px;
    }

    p {
      text-align: right;
    }
```

Opgave 2 – PHP en Forms (12 punten)

- d) `<ul>`
 `<?php for ($i = 99; $i >= 0; $i--) { ?>`
 `<li> <?= $i ?> bottles of beer on the wall </li>`
 `<?php } ?>`
`</ul>`
- e) Omdat in PHP de default is dat parameters “by value” worden meegegeven, dwz. worden gekopieerd bij elke function-call, worden de waardes van de variabelen niet gewijzigd door de function-call. Dus het resultaat is dan:

```
$a = array(14, 7);  
$b = array(7, 1, 3, 2, 0, 4);  
$c = array(10, 8, 9, 5, 5);  
$d = array(12, 11, 10, 10, 8, 7);
```

Als de functie was gedeclareerd als `function mysterie(&$list) {...}` dan wordt de parameter “by reference” meegegeven, en zou de uitkomst zijn:

```
$a = array(14, 8)  
$b = array(7, 2, 3, 3, 1, 4)  
$c = array(10, 9, 9, 6, 6)  
$d = array(12, 12, 11, 11, 9, 8)
```

f) De URL:

```
http://webster.cs.washington.edu/  
params.php?name=Jon+Stewart&man=on&vrouw=on
```

Mogelijke verbeteringen:

1. Het was beter geweest als de inputs voor geslacht het type “radiobutton” hadden, zodat niet beide opties tegelijk gekozen kunnen worden.
2. De action URL moet tussen quotes.
3. De check-boxen hadden in `<label>` verpakt kunnen worden zodat ook het label klikbaar is.
4. De *method* (GET of PUT) kan expliciet gespecificeerd worden.
5. De submit button kan een Engelse tekst krijgen.

Opgave 3 – JavaScript en DOM (8 punten)

- d) Dan wordt de variabele alsnog gedeclareerd als globale variabele en wordt de gegeven waarde toegekend.
- e) De waardes zijn: $a = 4$, $b = 2$, $c = 3$, $d = \text{undefined}$, $e = 6$. Merk op dat in de functie de variabelen c en d expliciet, en dus lokaal, zijn gedefinieerd, en dus gelden deze declaraties alleen binnen de functie. De variabelen a en e worden in de functie niet expliciet gedeclareerd, en daarom dus impliciet globaal gedeclareerd, waardoor ze na de aanroep bekend zijn.
- f) De `footer` knoop heeft 5 kinderen: (1) een text knoop met “`\n\t`”, (2) een `<p>` element, (3) een text knoop met “`\n\t`”, (4) een `<p>` element en (5) een text knoop met “`\n`”. Het aantal weergegeven *siblings* is 2: (1) een `<p>` element, (2) een text knoop met “`\n`”. Eventueel mag ook aangenomen worden dat er nog iets voor en achter het fragment staat, en dan komen er aan het begin en eind nog een text knoop bij, dus in totaal zijn er dan 4 *siblings*.

Opgave 4 – Events, Prototype, Ajax en XML (8 punten)

- d) `$$ (x) of document.select (x) met  $x = \text{"div\#container p.story"}$ .`
- e) De karaktercode van het ingetikte karakter kan opgevraagd worden met `event.charCode` of `event.keyCode` als `event` de event parameter van de handler is. Voor de Alt-, Ctrl- en Shift-toets kunnen van `event` de *properties* `altKey`, `ctrlKey` en `shiftKey` respectievelijk geraadpleegd worden. Deze zijn `true` als de betreffende toets werd ingedrukt.
- f) Bij synchrone communicatie doet de browser een verzoek voor een pagina, wacht tot dit verzoek wordt beantwoord, en laat dan de nieuwe pagina zien. In het algemeen betekent synchrone communicatie dat de client na een verzoek wacht met executie totdat de server antwoordt. Bij asynchrone communicatie stuurt de browser een verzoek voor informatie naar de server en wacht dan niet op het antwoord maar zet een functie gereed die aangeroepen zal worden zodra de server antwoordt. In Ajax wordt asynchrone communicatie gebruikt omdat we in een web-applicatie niet telkens een nieuwe pagina willen ophalen voor elke kleine wijziging in de user-interface.

Opgave 5 – SQL (15 punten)

- c) Als we aannemen dat elke employee een unieke achternaam heeft, dan is het volgende correct:

```
SELECT DISTINCT Lname
FROM employee JOIN dependent ON Ssn = Essn
WHERE dependent.Bdate < '1985-01-01';
```

Als we dat niet aannemen dan is het volgende correct:

```
SELECT Lname
FROM employee
WHERE Ssn IN ( SELECT Essn
                FROM dependent
                WHERE Bdate < '1985-01-01' );
```

Of:

```
SELECT Lname
FROM employee JOIN dependent ON Ssn = Essn
WHERE dependent.Bdate < '1985-01-01'
GROUP BY Ssn;
```

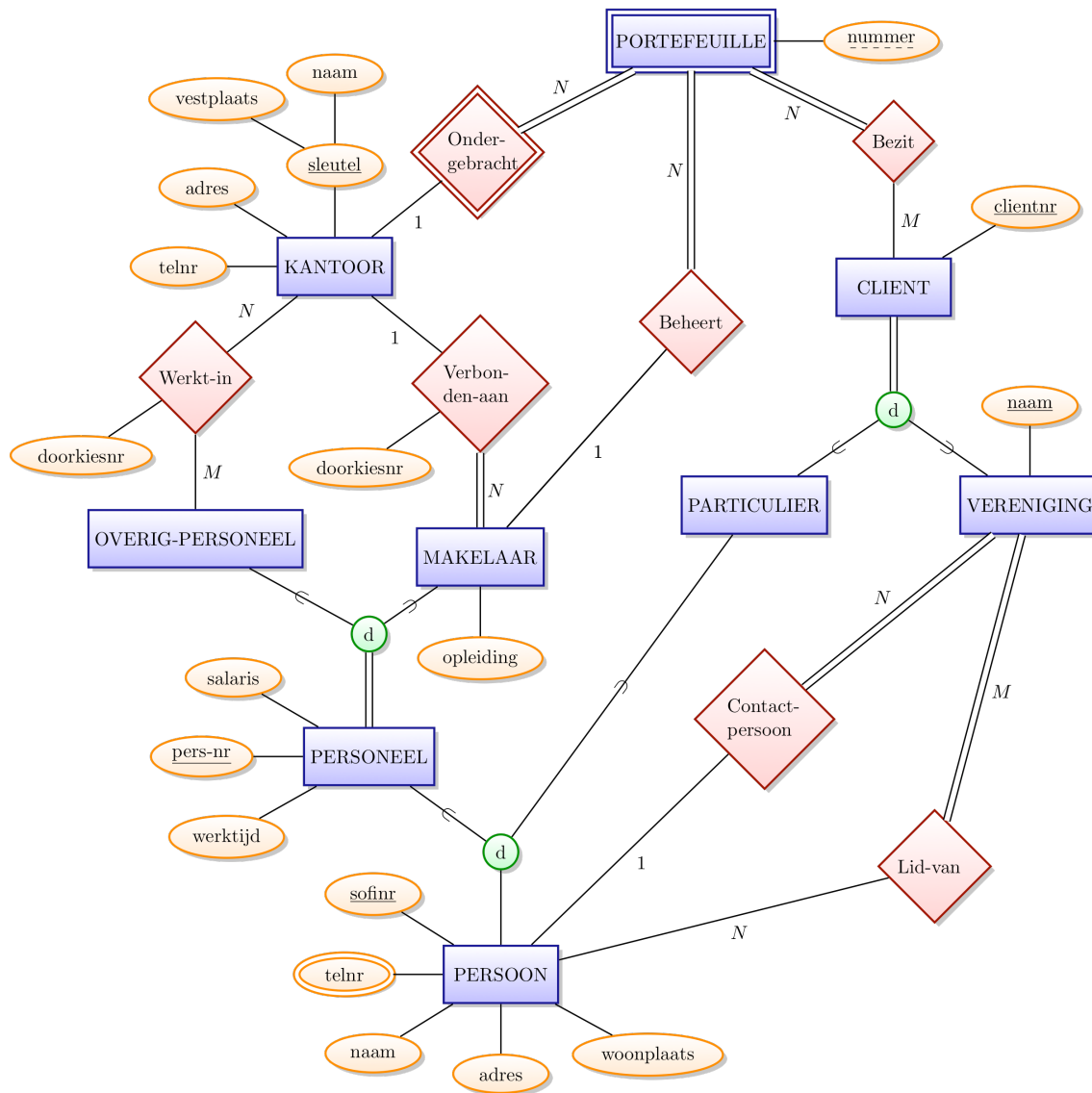
- d) Met behulp van views:

```
CREATE VIEW projWomanNotManag AS
SELECT Pnumber
FROM project JOIN works_on ON Pnumber = Pno
            JOIN employee ON Essn = Ssn
WHERE Sex = 'F' AND
      Ssn NOT IN ( SELECT Super_ssn FROM department );
```

```
SELECT Pnumber, Pname
FROM project
WHERE Pnumber NOT IN
      (SELECT Pnumber FROM projWomanNotManag);
```

Uitleg: in de view worden de projecten bepaald waarvoor een vrouw werkt die niet afdelingsmanager is. Vervolgens kiezen we in de query op het eind alle projecten die niet in de view voorkomen.

Opgave 6 – EER modelleren (30 punten)



De volgende aannames zijn gemaakt:

- (wiedes) Een kantoor kan meer dan één portefeuille beheren.
- Er kunnen kantoren zijn die geen portefeuilles beheren.
- Iedere vereniging in de database is een cliënt.
- Elke cliënt is heeft minstens één portefeuille.
- Een persoon kan lid en/of contactpersoon zijn van meer dan één vereniging.
- Iedere portefeuille is in het bezit van minstens één cliënt.
- (wiedes) Makelaars zijn personeelsleden.
- (wiedes) Iedere portefeuille wordt beheerd door een MiE makelaar.
- (wiedes) Er zijn geen makelaars die niet verbonden zijn aan een kantoor.
- Er kunnen kantoren zijn zonder makelaars en/of overige personeelsleden.

- Kantoren kunnen meer dan één makelaar of leden van het overige personeel in dienst hebben.
- (wiedes) Het kan voorkomen dat een makelaar geen portefeuille beheert.
- Er kunnen personen voorkomen in de database die noch personeelsleden zijn, noch particuliere cliënten, noch lid/contactpersoon van een vereniging.
- (wiedes) Een lid van het overig personeel kan geen makelaar zijn en andersom.

Opmerkingen

- Ik heb specialisatie van PERSOON op twee niveaus om
 - de extra attributen te vangen die specifiek zijn voor personeelsleden.
 - het verschil te kunnen aangeven in cardinaliteit en participatie van de werkt_in/verbonden_aan relatie.
- Disjunctie van PERSONEEL en PARTICULIER is nodig om te kunnen aangeven dat personeelsleden geen portefeuilles bezitten en niet contactpersoon van verenigingen kunnen zijn.
- Het (samengesteld) attribuut Sleutel van KANTOOR is de enige attribuut-naam die als nieuwe sleutel geïntroduceerd moet worden.
- In mijn oplossing wordt niet afgedwongen dat de portefeuille die een makelaar beheert is ondergebracht in het kantoor waar die makelaar aan verbonden is (maar dat stond ook niet in de opdracht vermeld).

Opgave 7 -- ER Model implementatie (15 punten)

