

Tentamen

TI1500 Web- en Databasetechnologie

Donderdag, 14 april 2011
09.00-12.00

Dit tentamen bestaat uit 7 open vragen
Totaal aantal pagina's (inclusief deze en appendix): 9

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.

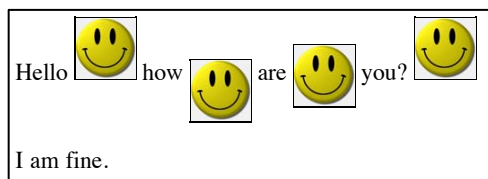
Het gebruik van het boek "*Web Programming Step by Step*" is toegestaan, evenals de slides die voor het vak op Blackboard zijn gezet. Het gebruik van eigen aantekeningen is niet toegestaan.

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] Wat zijn de verschillen tussen *block* en *inline* elementen? Geef een voorbeeld van elk.
- b) [2 pt.] Wat is het verschil tussen `ul` en `ol` elementen?
- c) [4 pt] Welke CSS code zou elke paragraaf (dwz. `<p>` elementen) plaatsen tegen de rechterkant van de pagina, het de helft van de breedte van de pagina in laten nemen, en de tekst in de paragraaf uitlijnen naar links.
- d) [2 pt] Wat zijn de waarden voor `vertical-align` voor elk van de images in:



Opgave 2 – PHP en Forms (12 punten)

- a) [5 pt.] Aannemende dat je een string hebt opgeslagen in een variabele `$s`, welke expressie produceert:
 - 1) de eerste letter van de string?
 - 2) de lengte van de string?
 - 3) de laatste letter van de string?
 - 4) de middelste letter van de string?
 - 5) de eerste vijf letters van de string (als `$s` tenminste 5 karakters bevat)?
- b) [3 pt.] Wat is de waarde van `$balance` na het executeren van de volgende code:

```
$balance = 3.00;
if ($balance > 0) {
    $balance -= 5;
}
if ($balance < 0) {
    $balance += 5;
}
```
- c) [2 pt] Wat is het verschil tussen het `name` en `id` attribuut op een *form control*?
- d) [2 pt] Geef de PHP-code voor het ophalen van de waarde van de query parameter adres. Met welke code kun je controleren of de gebruiker een checkbox met naam `vegetarisch` aangevinkt heeft?

Opgave 3 – JavaScript en DOM (8 punten)

- a) [3 pt.] Wat doen de `slice` en `splice` methods met arrays? Hoe zijn ze verschillend?
- b) [5 pt.] Stel een pagina bevat een `div` met `main` als `id`. Schrijf JavaScript code voor het toevoegen van een `h1` element aan het begin van deze `div` met de inhoud “Welkom op deze pagina”.

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

- a) [3 pt.] Voor *mouse events* kun je vragen om de *x* en *y* coördinaten van het *event* relatief t.o.v. verschillende andere punten. Welke drie van die andere punten zijn er? In welk geval maakt het niet uit welke van de drie je kiest?
- b) [2 pt.] Noem twee nuttige *methodes* die elke XML DOM *node* object bevat, en drie nuttige *properties*.
- c) [3 pt.] Wat is het verschil in Prototype tussen `Ajax.Request` en `Ajax.Updater`? Wanneer is welke het meest van toepassing?

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 de namen van de departementen die een project hebben lopen waarin een medewerker meewerkt die meer dan 40.000 Euro verdiend.
- b) [10 pt.] Geef de naam en de locatie van de projecten met een gemiddeld salaris dat hoger ligt dan het algemene salarismiddel.

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:

Een scheepvaartverzekeringsbedrijf houdt gegevens bij over door hen verzekerde schepen. Van elk schip wordt de naam en de eigenaar bijgehouden. Verder wordt opgeslagen wat het type van het schip is, en per type de naam van dat type, en tonnage (d.w.z. hoeveel ton vervoerd kan worden). Verder wordt de thuishaven van elk schip geregistreerd. Van elke haven wordt bijgehouden wat de naam is en in welk land deze ligt. Een haven wordt niet per se uniek geïdentificeerd door de naam, maar wel door de combinatie van naam en land. Per land wordt bijgehouden wat de naam is en op welk continent het ligt. Verder wordt van elke haven bijgehouden aan welk waterlichaam (kanaal, rivier, meer, zee, oceaan) het ligt.

Elk verzekerd schip moet regelmatig rapporteren waar het zich bevindt. Het moet daarvoor o.a. doorgeven waar het zich bevond (longitude en latitude) op welk tijdstip (datum en tijd). Bovendien moet elk bezoek aan een haven geadministreerd worden. Van elk havenbezoek moet de begin- en einddatum vastgelegd worden.

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 autogarage verkoopt, koopt en repareert auto's. Het heeft daarvoor werknemers in dienst waarvan sommige monteur zijn, sommige verkoper en sommige beide zijn. Van elke werknemer worden de naam en het werknemersnummer bijgehouden.

Van elke reparatie wordt bijgehouden welke monteurs er aan werken (minstens één) en welke auto het betreft. Verder wordt er bijgehouden de hoeveelste reparatie het is met een nummer, een beschrijving van de reparatie, en de prijs gesplitst in de prijs van de onderdelen en het werk.

Voor een auto wordt bijgehouden wat het merk is, het model en het bouwjaar. Per klant wordt bijgehouden wat de naam, adres en het unieke klantnummer is, en een aantal telefoonnummers.

De garage houdt zowel de aan- als verkopen bij. Bij een aankoop worden de datum van aankoop, de betaalde prijs en de intern geschatte waarde bijgehouden. Bij de verkoop worden de verkoopdatum, de ontvangen prijs en de commissie van de verkoper bijgehouden.

Einde van het tentamen

APPENDIX

EMPLOYEE

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

DEPARTMENT

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

DEPT_LOCATIONS

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

PROJECT

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

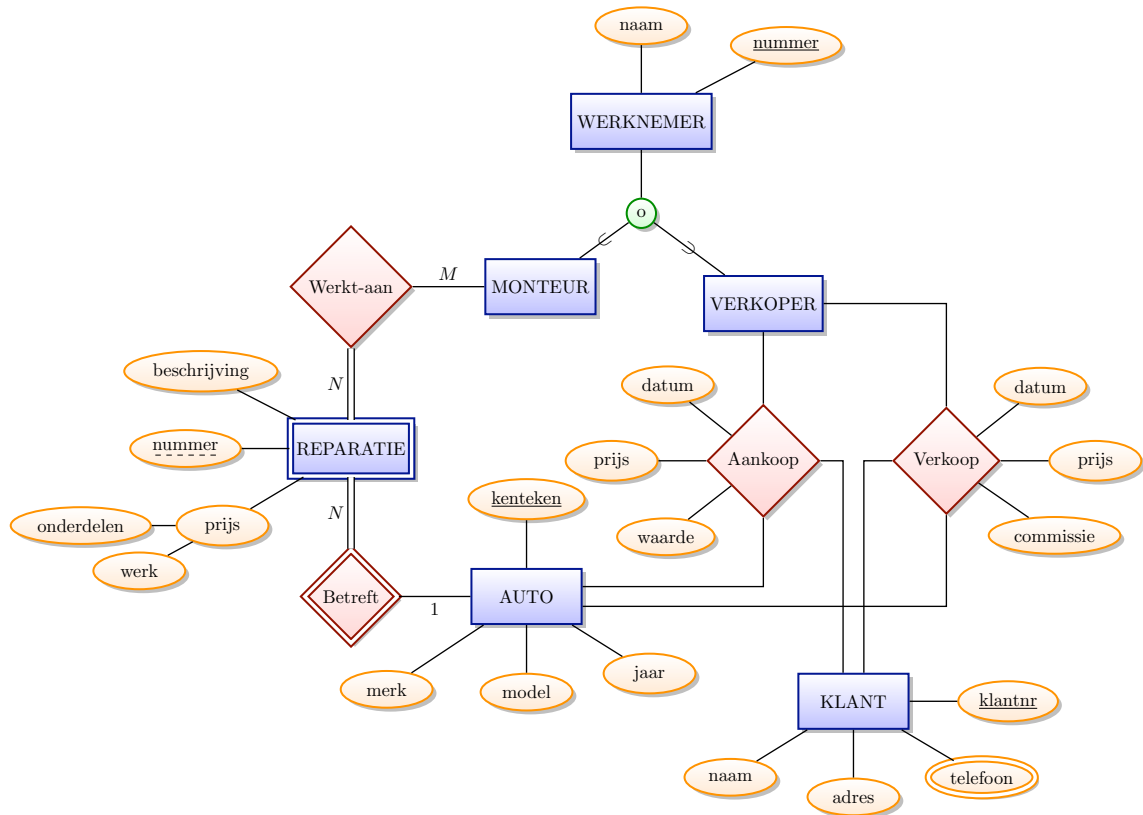
WORKS_ON

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

DEPENDENT

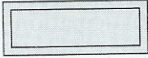
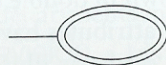
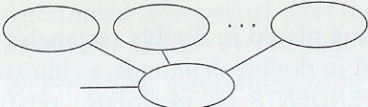
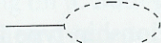
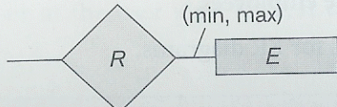
<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

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



Figuur 2. EER diagram Autobedrijf (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

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

- a) [4 pt] Een block-element kan grote hoeveelheden content bevatten die zich over meerdere regels uitstrekt. Een block-element kan zelf weer block-elementen en inline-elementen bevatten. De browser zal bij het weergeven een block-elementen de volle breedte van het omvattende element of pagina geven. Voorbeelden zijn paragrafen en lijsten. Een inline-element bevat meestal een relatief kleine hoeveelheid content. Een inline element moet in een block-element genest zijn. Zelf kan een inline-element alleen inline-elementen bevatten, en geen block-elementen. De browser zal bij het weergeven het inline-element meestal op dezelfde regel weergeven als de omvattende content. Voorbeelden van inline-elementen zijn links en images.
- b) [2 pt.] Een ul-element is een item in een ongeordende lijst en wordt weergegeven met hetzelfde symbool (bvb. een bullet) voor elk item. Een ol-element is een item in een geordende lijst en wordt weergegeven met een nummer of letter ervoor, die de plek aan de ordening aangeeft.

- c) [4 pt]

```
p {  
    float: right;  
    width: 50%;  
    text-align: left;  
}
```

- d) [2 pt] (1^e) top of baseline ; (2^e) text-top; (3^e) middle; (4^e) top of baseline.

Opgave 2 – PHP en Forms (12 punten)

- a) [5 pt.]
- de eerste letter van de string: `$s[0]`
 - de lengte van de string? `strlen($s)`
 - de laatste letter van de string? `$s[strlen($s) - 1]`
 - de middelste letter van de string? `$s[strlen($s) / 2]`
 - de eerste vijf letters van de string? `substr($s, 0, 5)` of `$s[1] . $s[2] . $s[3] . $s[4] . $s[5]`
- b) [3 pt.] Nog steeds 3.00. Er wordt eerst 5 afgetrokken, en dan weer 5 bij opgeteld.
- c) [2 pt] Een id-attribuut dient om het element te identificeren in de pagina, en moet dus uniek zijn binnen de pagina. Een name-attribuut wordt gebruikt als de naam van de waarde die wordt meegegeven in de query-parameter, en hoeft dus alleen binnen de form uniek te zijn en soms zelfs daar niet zoals bij radio-buttons.
- d) [2 pt] De PHP-code:
- Voor het ophalen van waarde van de parameter met de naam adres: `$_REQUEST["adres"]`
 - Voor het controleren of een checkbox met de naam vegetarisch is aangevinkt: `isset($_REQUEST["vegetarisch"])`
Merk op dat dit beter is dan `$_REQUEST["vegetarisch"] == "on"` omdat dit een foutmelding geeft als de checkbox niet aangevinkt is.

Opgave 3 – JavaScript en DOM (8 punten)

- a) [3 pt.] De methode `anArray.slice(a, b)` bepaalt een subarray van `anArray`, beginnende vanaf positie `a` tot aan (maar niet inclusief) positie `b`. Als `b` niet is gegeven wordt tot aan het einde van `anArray` gegaan. De methode `anArray.splice(a, b, v1, ..., vn)` verandert de waarde van `anArray` door vanaf positie `a` precies `b` elementen te verwijderen en deze te vervangen door `v1, ..., vn`.
- b) [5 pt.]

```
var newHeader = document.createElement("h1");
newHeader.innerHTML = "Welkom op deze pagina";
var mainDiv = document.getElementById("main");
mainDiv.insertBefore(newHeader, mainDiv.firstChild);
```

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

- a) [3 pt.] De drie andere punten zijn: (1) linkerbovenhoek van browser-window / tabblad (2) de linkerbovenhoek van het gehele scherm en (3) linkerbovenhoek van het document in zijn geheel. Eventueel kun je ook noemen: de linkerbovenhoek van het element waar het event gedetecteerd is, maar dit werkt niet in Firefox. Het maakt niet uit wat je kiest als het document zo klein is dat het altijd in zijn geheel in de browser getoond wordt.
- b) [2 pt.] Alle DOM objecten kunnen de volgende nuttige methodes uitvoeren: `getElementById`, `getElementsByTagName`, `getElementByName`. Verder hebben ze de volgende nuttige properties: `parentNode`, `previousSibling`, `nextSibling`, `firstChild`, `lastChild`, `childNodes`. Merk op dat methodes voor het toevoegen en verwijderen van knopen (`append`/`insert`/`remove`/`replace`) alleen gelden voor element-objecten.
- c) [3 pt.] De `Ajax.Request` doet de request en roept de aangegeven handler aan als de request beantwoord wordt. De `Ajax.Updater` roept niet een handler aan maar zet het resultaat op een aangegeven plek in het huidige document.

Opgave 5 – SQL (15 punten)

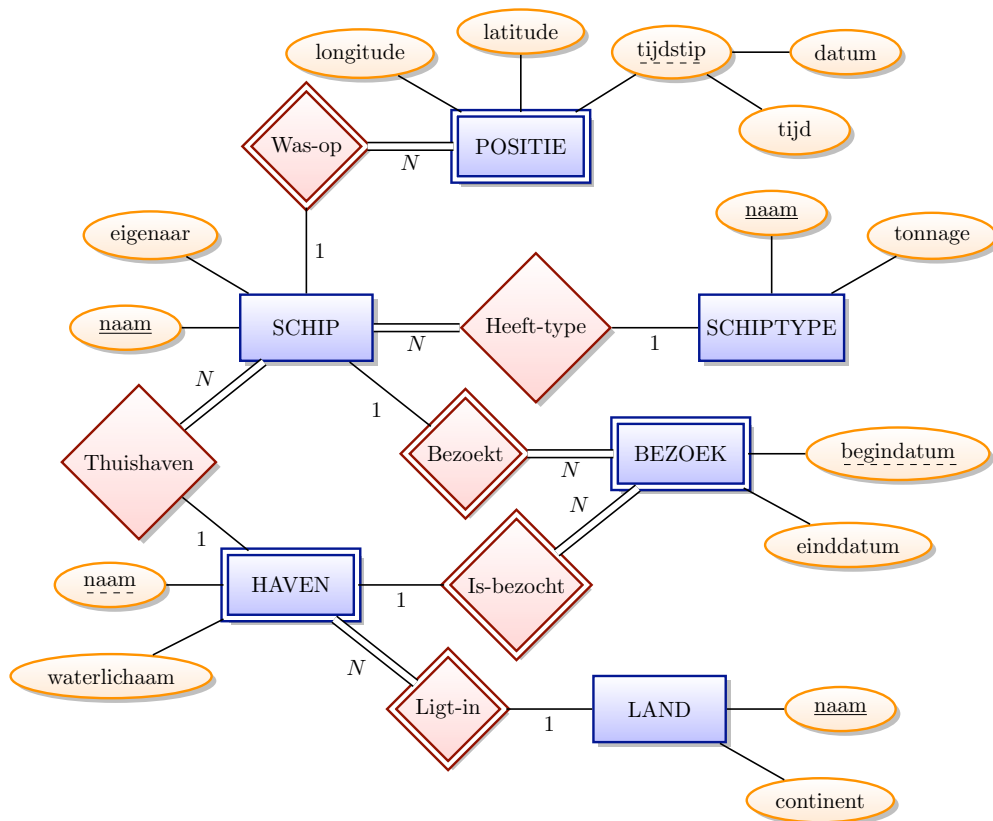
- a) [5 pt.] Geef de namen van de departementen die een project hebben lopen waarin een medewerker meewerkt die meer dan 40.000 Euro verdiend.

```
SELECT DISTINCT Dname
FROM Department JOIN Project ON Dnumber = Dnum
                JOIN Works_on ON Pnumber = Pno
                JOIN Employee ON Essn = Ssn
WHERE Salary > 40000;
```

[10 pt.] Geef de naam en de locatie van de projecten met een gemiddeld salaris dat hoger ligt dan het algemene salarismiddel.

```
SELECT Pname, Plocation
FROM Project JOIN Works_on ON Dnumber = Dnum
                JOIN Employee ON Essn = Ssn
GROUP BY Pname, Plocation
HAVING Avg(Salary) >
      (SELECT AVG(Salary) FROM Employee);
```

Opgave 6 – EER modelleren (30 punten)



Additionele aannames:

- Een schip wordt geïdentificeerd door de naam.
- Een schip heeft precies 1 type.
- Een schip heeft precies 1 thuishaven.
- Een haven ligt aan ten hoogste 1 waterlichaam tegelijk.

Opgave 7 -- ER Model implementatie (15 punten)

