TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

TU Delft

Tentamen

IN2105 Databases

Woensdag 3 november 2010, 14:00 – 17:00

Dit tentamen bestaat uit 5 open vragen
Totaal aantal pagina's (inklusief deze pagina): 6

Bij dit tentamen hoort een appendix met een aantal nuttige plaatjes uit het boek en een aantal figuren waarnaar verwezen wordt in de opdrachten.

Opgave	Punten
1	20
2	15
3	30
4	20
5	15
Totaal	100

Alle deelopgaven binnen een opgave tellen even zwaar
Aantal punten nodig voor een voldoende: 55

Het gebruik van boeken, diktaten en van rekenmachines is niet toegestaan

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

Dit tentamen omvat het materiaal uit de collegesheets 2010-2011 (zie Blackboard), en de hoofdstukken 1 tot en met 8 (behalve sekties 4.4, 5.2, 5.4, 6.4.1, 6.4.3, 6.4.5, 6.6.5, 7.13 en 7.14), 14 (alleen 14.1, 14.2, 14.3.1, 14.3.2, 14.3.3 en 14.5) en de sekties 15.1.1, 15.1.2, 15.2 en 15.3.2 uit hoofdstuk 15 van het boek: Elmasri & Navathe, Fundamentals of Database Systems (Ed.6).

Opgave 1 (20 punten)

Geef voor elk van de hierna volgende vragen een SQL-query over de COMPANY database (het 'running example' uit het boek). Het schema van deze database wordt gegeven in Figuur 1 in de Appendix. Voor de syntax van SQL, zie Figuur 5 uit de Appendix.

a) Geef een lijst met achternamen (`Lname`) van de werknemers die direkte chef zijn (`Super_ssn` in EMPLOYEE) van één of meer vrouwen. Ieder *personeelslid* met deze eigenschap moet precies één keer voorkomen in het resultaat.

b) Een afdeling *A* participeert in een projekt *P* als één of meer van de medewerkers uit *A* meewerkt aan *P*. Geef een lijst (`Pname,Aantal`) van de projektnamen met daarin voor ieder projekt het aantal afdelingen dat erin participeert.

c) Geef de namen (`Pname`) van de projekten waarvoor geldt dat iedere werknemer die aan dat projekt meewerkt dat precies voor 10 uur in de week doet (`Hours=10`). Je mag aannemen dat het attribuut `Pname` in PROJECT uniek is, en dat ieder projekt deelnemers heeft.

Opgave 2 (15 punten)

In een aantal van de hierna volgende vragen is de opdracht een query in de relationele algebra te formuleren. Voor de operatoren van de relationele algebra zie Figuur 4 uit de Appendix. De onderliggende database is, als boven, de COMPANY database, waarvan het schema gegeven wordt in Figuur 1 in de Appendix.

a) Geef een query in de relationele algebra die de de projekten oplevert (`Pnumber, Pname`) waar één of meer personen in participeren voor meer dan 20 uur per week (`Hours>20`). Doe dat in één expressie, geen assignments ("views").

b) Doe dezelfde opdracht als onder **a)** maar nu in de *tuple calculus*.
Ter herinnering: in de tuple calculus gaat het om formules ("queries") van de vorm {...|...}. Een voorbeeld (achternaam en geboortedatum van alle employees met een salaris groter dan 30000) is:

{e.Lname, e.Bdate | EMPLOYEE(e) AND e.Salary > 30000}

c) Geef een reeks queries in de relationele algebra die alle medewerkers oplevert (`Ssn, Lname`) van de afdelingen met een vestigingsplaats in Bellaire (`Dlocation='Bellaire'`), met uitzondering van diegenen die managers zijn van een afdeling (`Super_ssn` in DEPARTMENT).

d) Geef een reeks queries in de relationele algebra die de werknemers oplevert (`Ssn, Lname`) die meedoen aan alle projekten gelokaliseerd in Houston (`Plocation='Houston'`).

Opgave 3 (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:
Nu de regering Rutte de subsidiëring van de kunsten drastisch gaat beperken is de inbreng van de partikuliere sektor des te belangrijker geworden. Om deze nieuwe subsidiestromen te begeleiden en te registreren is de consultancy firma "De Strijkstok" op het idee gekomen om hier een rol te gaan spelen. Hiertoe is een database nodig met de volgende specifikatie.
Van iedere gift dient de datum, het bedrag, de verstrekker en de ontvanger geregistreerd te worden. Het is niet mogelijk dat dezelfde gift meer dan één verstrekker en/of ontvanger heeft. Een verstrekker kan een individueel persoon zijn (hiervoor wordt de term 'Maecenas' gebruikt), of een bedrijf. Van de maecenassen wordt bijgehouden: naam, adres, telefoonnummer, bank/gironummer en een indikatie of ze anoniem wensen te blijven. Verder (om belastingtechnische redenen) hun (unieke) sofinummer.
Elk bedrijf krijgt bij aanmelding een uniek identifikatienummer. Verder wordt het logo opgeslagen (voor de begeleidende brief bij de giften), natuurlijk naam en adres, het bank/gironummer dat gebruikt gaat worden, en kontaktinformatie. Deze bestaat uit een telefoonnummer en een omschrijving (bij voorbeeld 'afdeling PR', of 'direkteur, de heer Heineken'). Het is mogelijk om op meer dan één manier kontakt met een bedrijf te hebben, en voor elk van deze mogelijkheden moet zo'n kombinatie telefoonnummer/omschrijving worden geregistreerd in de database.
Giften kunnen gedoneerd worden aan een individuele kunstenaar of aan een groep (bij voorbeeld een koor, theatergroep of orkest). Van de individuele kunstenaars wordt bijgehouden hun naam en adres, bank/gironummer, de kunstvorm die ze bedrijven (bij voorbeeld violist, mimespeler, beeldhouwer), telefoonnummer en sofinummer (opnieuw voor de belastingen).
Iedere groep heeft een uniek identifikatienummer, dat verleend wordt bij opname van de groep in het systeem. Verder wordt van de groepen bijgehouden de naam, het adres, giro/banknummer, het soort groep (koor, theatergroep, orkest e.d.) en één of meer manieren waarop kontakt met de groep onderhouden kan worden. Voor elk van deze manieren wordt een telefoonnummer bijgehouden, en een indikatie van op wie of wat dat telefoonnummer betrekking heeft (bij voor-

beeld 'Johann Sebastian van Beethoven, de dirigent', of 'Drs. D. Euro, zakelijk leider').
Het is mogelijk dat een individueel kunstenaar subsidie krijgt terwijl deze bovendien lid is van een groep die ook gesubsidieerd wordt. Om deze geldstromen goed uit elkaar te houden wordt bijgehouden voor welke kunstenaars dat geldt, en van welke groep of groepen hij of zij in dat geval lid is.
Zodra een subsidieverstrekker of –ontvanger zich opgeeft wordt hij opgenomen in het systeem, ook voordat ze betrokken zijn bij een gift (als verstrekker dan wel ontvanger).

Opgave 4 (20 punten)

Beschouw het EER diagram in Figuur 2 uit de Appendix. Vertaal dit diagram in een relationeel database schema. Noteer de tabellen zoals ik dat gedaan heb in Figuur 1 van de Appendix, en voeg (zoals 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.

Ter verduidelijking een beschrijving van de case:

Het diagram beschrijft een universiteit, en wel de medewerkers ervan, de studenten, de vakken die gegeven worden, en de faculteiten. Medewerkers (personeel) en studenten zijn specialisaties van de entiteit PERSOON. Verder worden master studenten als aparte groep onderscheiden, omdat deze als assistent kunnen optreden. Docenten (wetenschappelijke staf) en assistenten worden bijeengenomen in de categorie INSTRUKTEUR, omdat alleen deze personen onderwijs kunnen verzorgen.
Personeelsleden kunnen lid zijn van één of meer faculteiten. Faculteiten kunnen vakken verzorgen. Een vak dat op een bepaald tijdstip gegeven wordt heet een klas, gemodelleerd als instantiatie van de entiteit VAK. De entiteit KLAS is zwak, met partiële sleutel volgnummer (dat dus uniek is voor iedere instantiatie van een bepaald vak). Verder zijn nog het jaar en het kwartaal waarin dat vak gedoceerd werd van belang. Tenslotte wordt van iedere student de behaalde resultaten bijgehouden, gemodelleerd in een relatie tussen STUDENT en KLAS, met als attribuut het behaalde cijfer.
Van ieder persoon wordt bijgehouden: de sofi, de sexe, de geboortedatum, en naam, adres en woonplaats (NAW, gemodelleerd als niet-samengesteld attribuut). Van personeelsleden wordt bovendien het salaris, de akademische titel, en het kamernummer geadministreerd. Van master studenten worden de bachelordiploma('s) bijgehouden. Dat kunnen er meer dan één zijn. Deze worden gekarakteriseerd door de naam van de bachelor en het jaar waarin die behaald is.

Opgave 5 (15 punten)

a) Gegeven een relationeel schema R_1 met attributen A, B, C en D. Beschouw de volgende twee verzamelingen funktionele afhankelijkheden:

$F_1 = \{A \rightarrow C, AB \rightarrow D\}$ en $F_2 = \{A \rightarrow C, BC \rightarrow D\}$.

Geef een extensie (dat wil zeggen een invulling van een tabel over A, B, C en D) die wel voldoet aan één van de F_i's, maar niet aan de andere.

b) Zij gegeven een relationeel schema R met attributen A, B, C, D en E. Tussen deze attributen gelden de volgende funktionele afhankelijkheden F:

A $\rightarrow$ C
B $\rightarrow$ D
AB $\rightarrow$ E

Wat zijn de kandidaatsleutels in dit schema (+ afhankelijkheden)? Verklaar je antwoord.

c) Dekomponeer het schema (A,B,C,D,E) met afhankelijkheden F zoals hierboven in een aantal deelschema's zo dat de dekompositie lossless is, de afhankelijkheden behouden blijven en alle deelschema's in Boyce-Codd normaalvorm staan.

De volgende deelopdrachten gaan over een relationeel schema R' weer met attributen A, B, C, D en E. We voegen een nieuwe funktionele afhankelijkheid toe aan F, namelijk C $\rightarrow$ A en krijgen het stelsel F':

A $\rightarrow$ C
C $\rightarrow$ A
B $\rightarrow$ D
AB $\rightarrow$ E

d) Wat zijn de kandidaatsleutels in dit schema (+ afhankelijkheden)?

e) Beschouw de dekompositie van R' onder de afhankelijkheden F':

(A,B,C,D) (A,E)

Is deze dekompositie lossless? Zijn de afhankelijkheden behouden?

f) Beschouw de dekompositie van R' onder de afhankelijkheden F':

(A,B,C,D) (B,C,E)

Is deze dekompositie lossless? Zijn de afhankelijkheden behouden?

Einde van het tentamen

TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

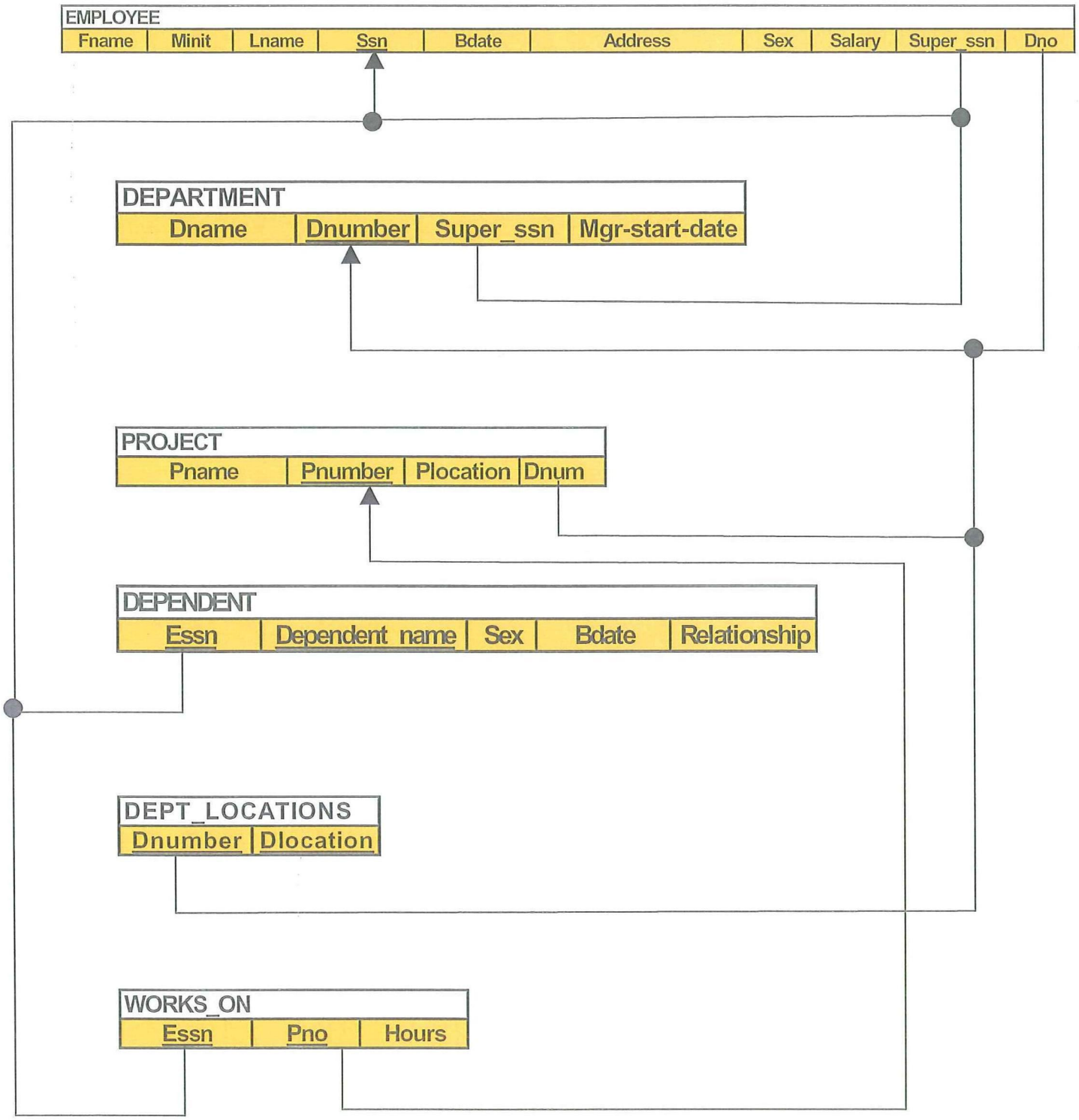
APPENDIX

behorend bij het tentamen

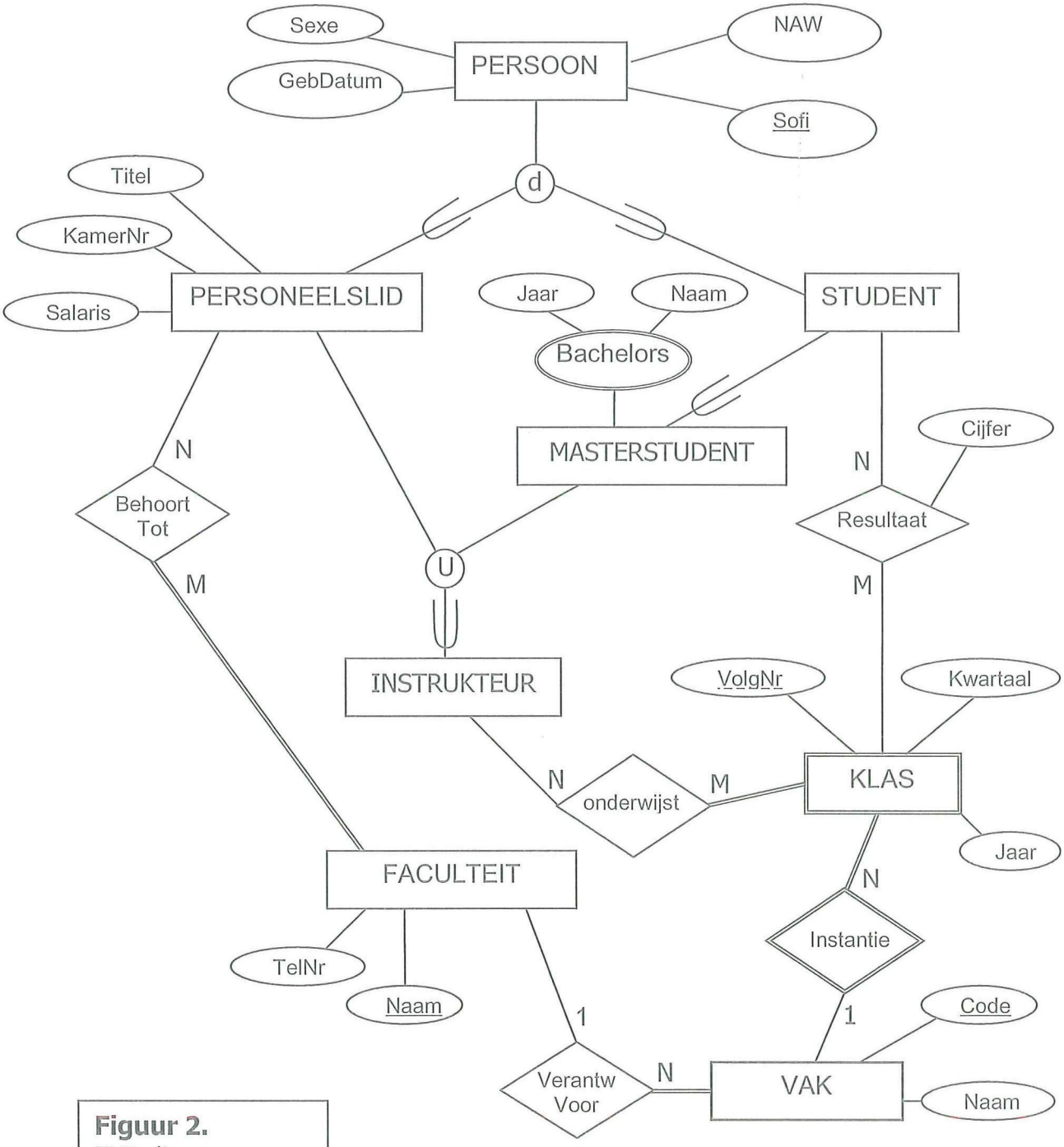
IN2105 Databases

Woensdag 3 november 2010, 14:00 – 17:00

Totaal aantal pagina's (inklusief deze pagina): 6

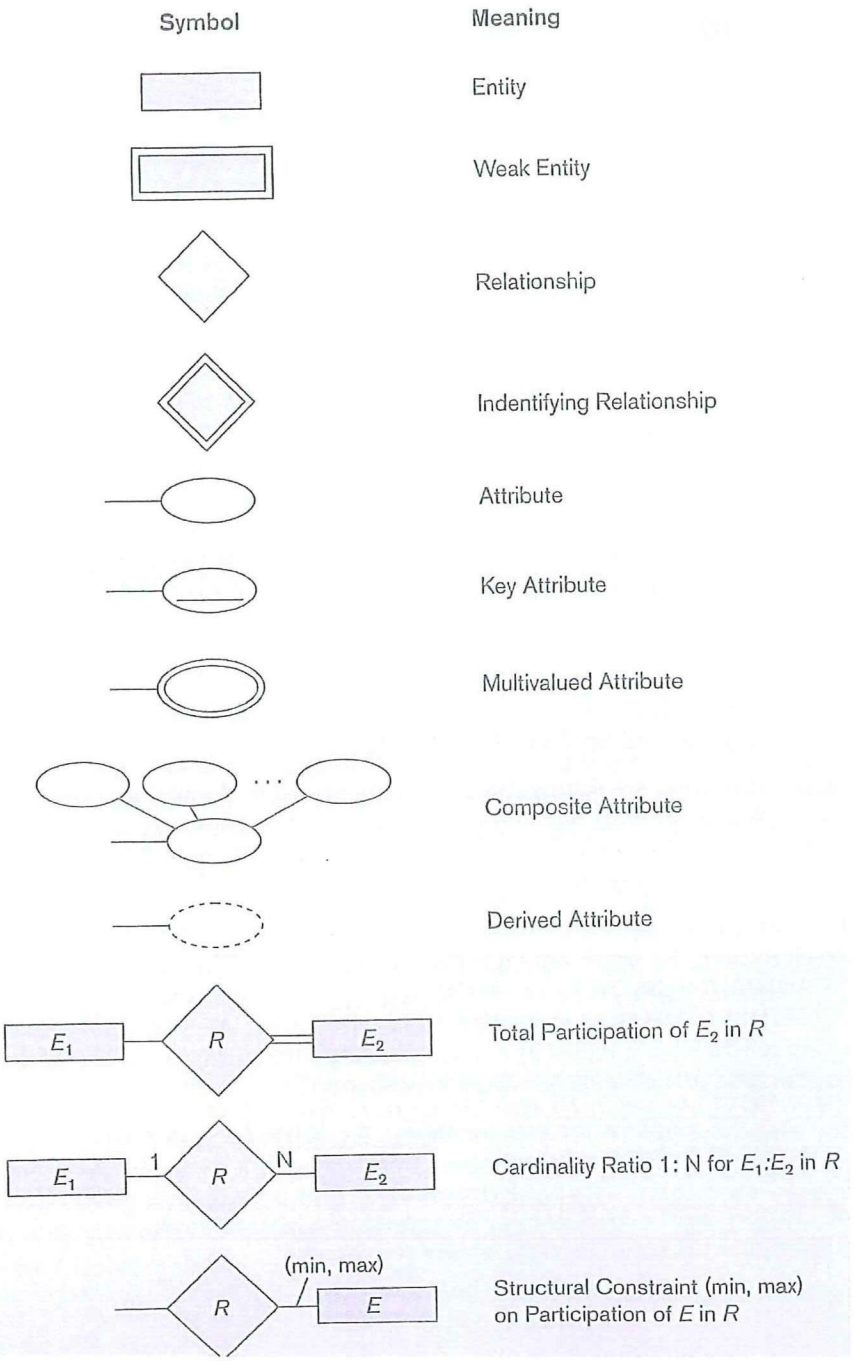


Figuur 1. Schema van de COMPANY database. Zie Opgaven 1 en 2.



Figuur 2.
EER diagram
Universiteit
(zie Opgave 4)

Figure 3.14
Summary of the notation for ER diagrams.



Figuur 3. De ER symbolen en hun betekenis

Table 6.1
Operations of Relational Algebra

Operation	Purpose	Notation
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{<\text{selection condition}>}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{<\text{attribute list}>}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{<\text{join condition}>} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{<\text{join condition}>} R_2$, OR $R_1 \bowtie_{(<\text{join attributes 1}>),\ (<\text{join attributes 2}>)} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{<\text{join condition}>} R_2$, OR $R_1 *_{(<\text{join attributes 1}>),\ (<\text{join attributes 2}>)} R_2$ OR $R_1 * R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

Figuur 4. De operatoren uit de relationele algebra

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 5. SQL syntax