

Tentamen
IN2105 Databases

Dinsdag 27 oktober 2009, 14:00 – 17:00

Dit tentamen bestaat uit 5 open vragen
Totaal aantal pagina's (inclusief 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 2008-2009 (zie Blackboard), en de hoofdstukken 1 tot en met 8 (behalve secties 4.7 en 8.7), 10 (behalve 10.3.4 – 10.4) en de secties 11.1 en 11.2.2 uit hoofdstuk 11 uit het boek: Elmasri & Navathe, Fundamentals of Database Systems (Ed.5).

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 de de namen (*Lname*) van alle medewerkers die chef zijn van een afdeling, geboren zijn na 1950 en als chef aangesteld zijn voor 1990 (*Mgr-start-date*). Elk van deze medewerkers mag maar één keer voorkomen in de lijst. Houd rekening met het feit dat iemand chef kan zijn van meer dan één afdeling. N.B. 1 april 2009 wordt genoteerd als '2009-04-01'.
- b) Geef een lijst van afdelingsnamen en projectnamen (*Dname, Pname*) waarvoor geldt dat de afdeling *Dname* het project *Pname* superviseert (*Dnum* in PROJECT) maar dat het project *Pname* plaatsvindt in een locatie (*Plocation*) waar de afdeling geen vestiging (*Dlocation*) heeft.
- c) Geef de namen (*Dname*) van de afdelingen die een vestiging hebben in alle vestigingsplaatsen (*Dlocation*).

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)** (dit is bijna dezelfde case als in opgave **1a**) Geef een query in de relationele algebra die de social security numbers en de namen ($Ssn, Lname$) oplevert van alle medewerkers die chef zijn van een afdeling, geboren zijn na 1950 en als chef aangesteld zijn voor 1990 ($Mgr-start-date$).
- b)** Teken de kanonieke queryboom die hoort bij je antwoord op deel **a)** van deze opgave
- c)** 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:
 $\{e.Lname, e.Bdate \mid EMPLOYEE(e) \text{ AND } e.Salary > 30000\}$
- d)** (dit is dezelfde case als in opgave **1c**) Geef een query in de relationele algebra die de namen ($Dname$) oplevert van de afdelingen die een vestiging hebben in alle vestigingsplaatsen ($Dlocation$).

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:

De rijsschool *Karremaar* geeft autorijlessen en motorrijlessen. Daarvoor heeft de rijsschool autorij-instructeurs en motorrij-instructeurs in dienst. Van elke instructeur wordt bijgehouden wat de naam, het adres en het telefoonnummer is. Bovendien heeft elke instructeur een uniek werknemersnummer en een datum van indiensttreding.

In de klantenregistratie staat voor elke klant een uniek klantnummer, de naam van de klant, zijn of haar adres, de woonplaats en een aantal telefoonnummers waarop de klant bereikbaar kan zijn. Indien van toepassing wordt bij een klant ook opgeslagen wanneer deze zijn of haar theorie-examen met succes heeft afgelegd. Elke klant heeft een toegewezen instructeur waarvan deze normaalgesproken les krijgt.

De lesvoertuigen van Karremaar zijn auto's en motoren. Van de auto's wordt het merk en het type bijgehouden, en bovendien het bouwjaar en het kenteken. Van de motoren wordt ook het merk, type, bouwjaar en kenteken bijgehouden, en daarnaast ook de cilinderinhoud.

De klant kan een afspraak maken voor een rijles met een motor of met een auto, en uiteraard met een bijbehorende instructeur, meestal de toegewezen instructeur. Voor elke rijles wordt een begin- en eindtijdstip afgesproken, en een afhaaladres waar de les begint. De status van een les is "gepland", "afgezegd" of "voltooid". Bij een voltooide les wordt de vooruitgang van de klant bijgehouden. Bij elke geplande les wordt ook vastgelegd welk voertuig gebruikt zal worden.

De klant kan deelnemen aan klassikaal gegeven theorielessen en dit wordt geregistreerd. Er zijn aparte theorielessen voor motor en auto, maar alle instructeurs kunnen en mogen deze lessen geven. Deze lessen beginnen op een bepaalde datum en op een bepaald uur en vinden plaats in één van de drie beschikbare lokalen. 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 waar en wanneer dit is, wie de examinerator zal zijn. Na afloop wordt ook opgeslagen wat het resultaat was en eventueel een opmerking over wat er fout ging.

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. 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:

Het diagram beschrijft een kleine bibliotheek dat boeken, muziek-cd's en game's uitleent. Van elk boek wordt bijgehouden wat de titel is, het isbn-nummers, de uitgever en alle auteurs van het boek. Van muziek-cd's wordt bijgehouden wat het jaar van uitgave, de titel en de uitgever is. Verder wordt opgeslagen welke nummers allemaal op de CD staan. Van elk nummer wordt de titel en de uitvoerende artiest bijgehouden. Van een game wordt bijgehouden wat de titel is, op welk platform het gespeeld kan worden en wie de uitgever is.

De boeken, cd's en game's worden allemaal publicatie genoemd, en voorzien van een speciale publicatie-identificer. Van zo'n publicatie kan de bibliotheek een aantal exemplaren in voorraad hebben. Hetzelfde boek kan bijvoorbeeld drie keer aanwezig zijn en dan zijn er dus drie exemplaren van dezelfde publicatie. Daarom wordt er onderscheid gemaakt tussen de abstracte publicatie en het concrete exemplaar. Voor een exemplaar wordt een exemplaarnummer bijgehouden dat weergeeft het hoeveelste exemplaar (1^e, 2^e, etc.) het is.

De bibliotheek kent drie locaties, elke met een eigen adres, en exemplaren staan verspreid over deze locaties. In het ledenbestand wordt voor elk lid bijgehouden wat het unieke lidnummer is, en bovendien de naam, het adres en een telefoonnummer. Als een lid een exemplaar leent wordt bijgehouden wat de uitleendatum is en zodra het exemplaar weer ingeleverd wordt, wordt geregistreerd wat de iwerkelijke nleverdatum is.

Opgave 5 (15 punten)

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

$$F_1 = \{AB \rightarrow C, AC \rightarrow D\} \text{ en } F_2 = \{AB \rightarrow C, AB \rightarrow D\}.$$

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 functionele afhankelijkheden:

$$\begin{array}{l} AB \rightarrow C \\ AC \rightarrow D \\ D \rightarrow B \end{array}$$

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

De volgende deelopdrachten gaan over een relationeel schema R' met met attributen A, B, C, D en E. Tussen deze attributen gelden de volgende functionele afhankelijkheden F' :

$$\begin{array}{l} AB \rightarrow C \\ AC \rightarrow D \\ D \rightarrow E \end{array}$$

- c) Beschouw de dekompositie van R' onder de afhankelijkheden F' :
(A,B,C,E) (A,C,D)
Welke van deze subtabellen staan in BCNF?
- d) Beschouw de dekompositie uit c). Toon aan dat deze lossless is.
- e) Toon aan dat de dekompositie uit c) niet de afhankelijkheden uit F' bewaart.
- f) Geef een lossless dekompositie van R' , die de afhankelijkheden wel bewaart en waarbij de subtabellen alle in BCNF staan.

Einde van het tentamen

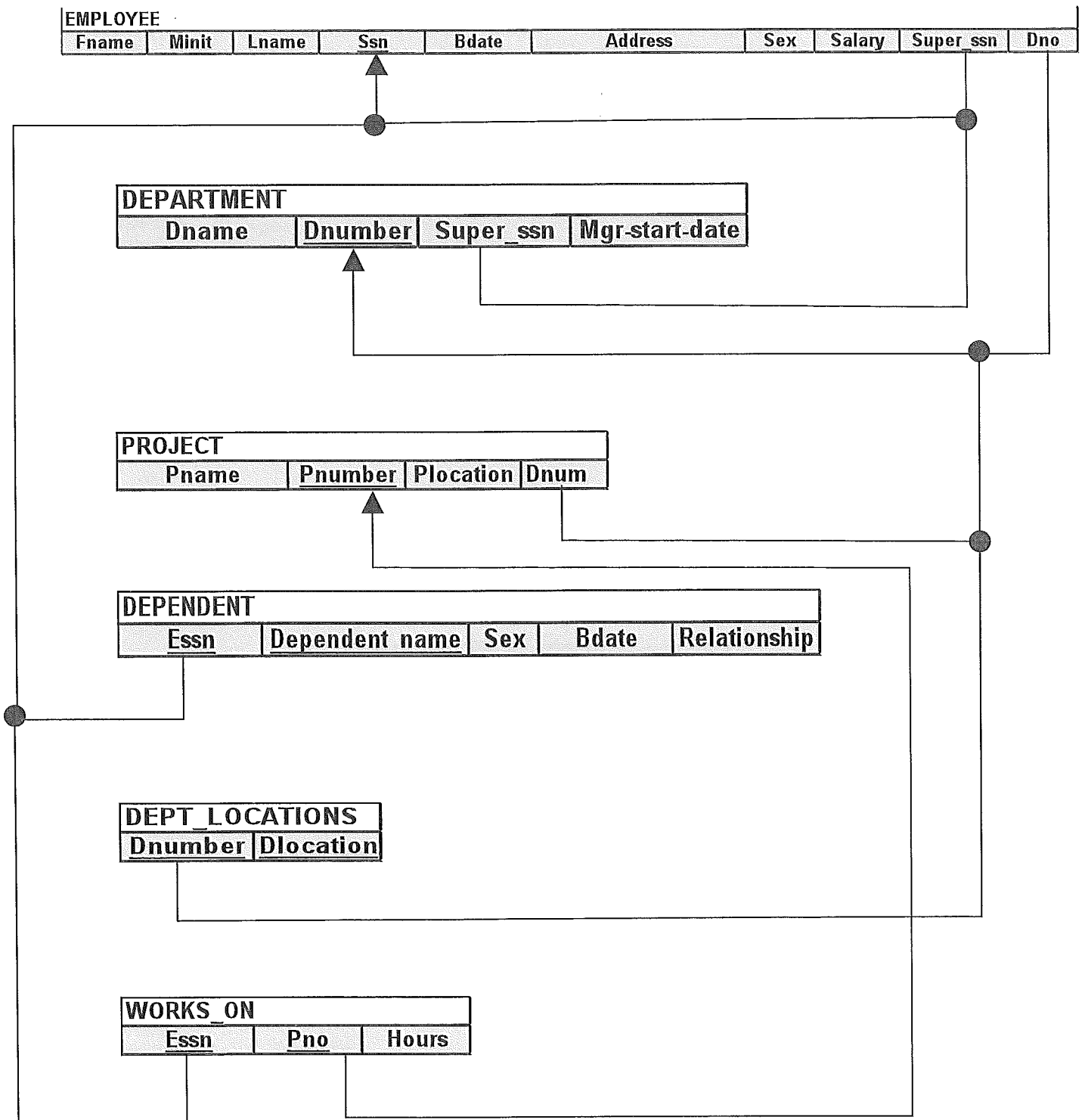
APPENDIX

behorend bij het tentamen

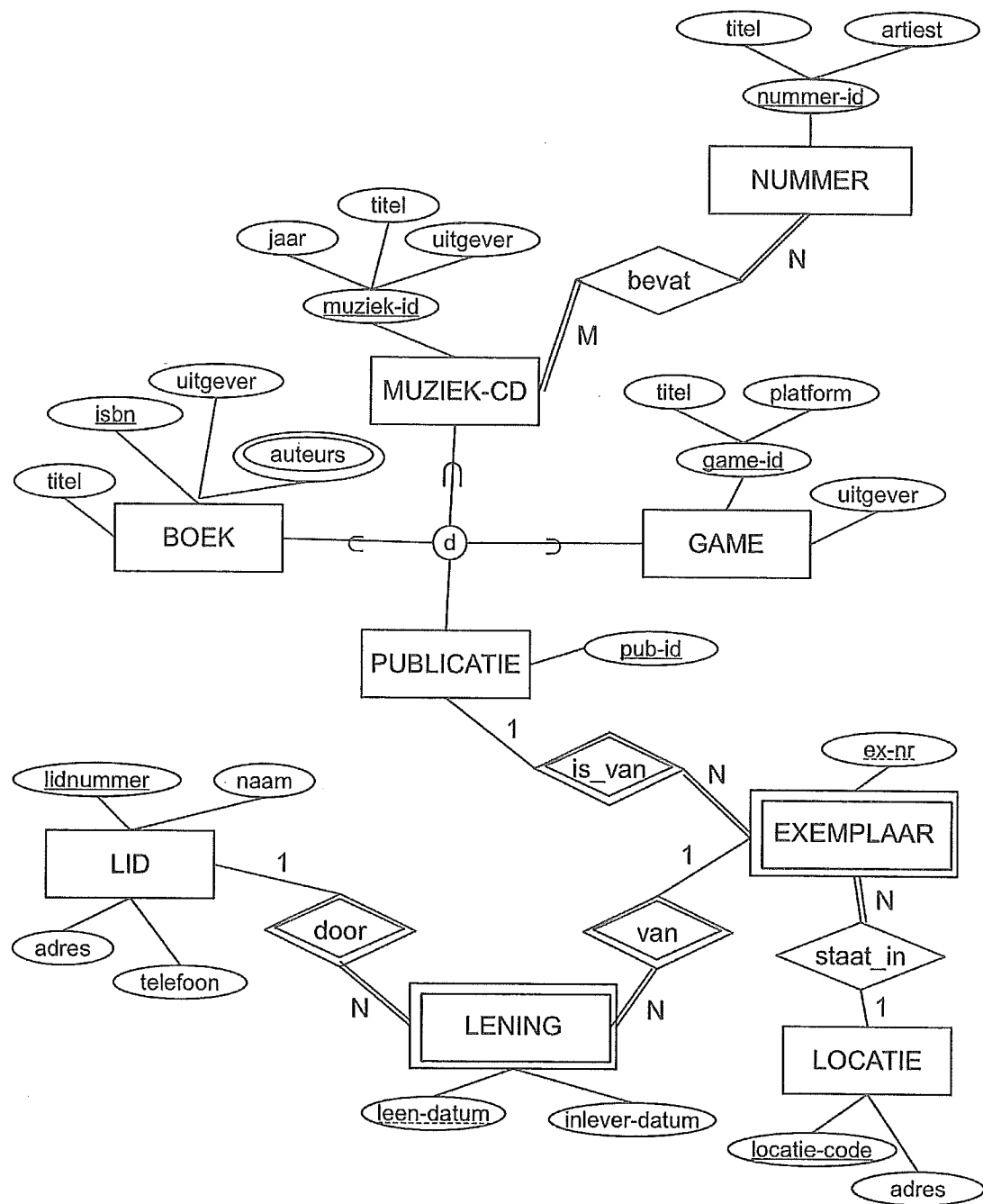
IN2105 Databases

Dinsdag 27 oktober 2009, 14:00 – 17:00

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

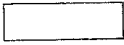
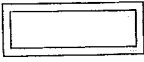

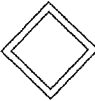


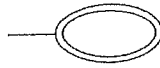
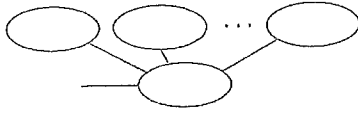
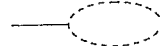
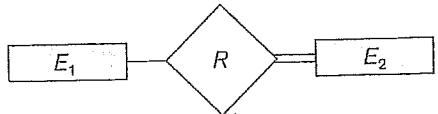
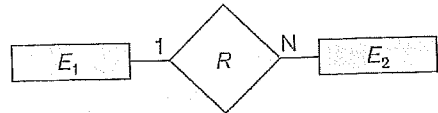
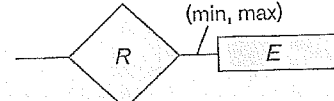


Figuur 1. Schema van de COMPANY database. Zie Opgaven 1 en 2.



Figuur 2. EER diagram Bibliotheek (zie Opgave 4)

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 6.1
Operations of Relational Algebra

Operation	Purpose	Notation
SELECT	Selects all tuples that satisfy the selection condition from a relation R .	$\sigma_{\langle \text{selection condition} \rangle}(R)$
PROJECT	Produces a new relation with only some of the attributes of R , and removes duplicate tuples.	$\pi_{\langle \text{attribute list} \rangle}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$ OR $R_1 \bowtie_{(\langle \text{join attributes 1} \rangle), (\langle \text{join attributes 2} \rangle)} R_2$
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.	$R_1 *_{\langle \text{join condition} \rangle} R_2$ OR $R_1 *_{(\langle \text{join attributes 1} \rangle), (\langle \text{join attributes 2} \rangle)} R_2$ 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