

# Samenvatting Databases

Robin Vanhove

Juni 2017

## Inhoudsopgave

	<b>2</b>
<b>I Conceptueel en Relationeel Model &amp; Query's</b>	<b>3</b>
<b>1 ER &amp; EER</b>	<b>3</b>
<b>2 Relationele algebra</b>	<b>3</b>
2.1 Operatoren . . . . .	3
2.1.1 Selectie . . . . .	3
2.1.2 Projectie . . . . .	3
2.1.3 Hernoeming . . . . .	4
2.1.4 Unie doorsnede en verschil . . . . .	4
2.1.5 Cartesisch product . . . . .	4
2.1.6 Join Operator . . . . .	4
2.1.7 Deling . . . . .	4
2.2 Aggregaatfuncties . . . . .	4
<b>3 SQL</b>	<b>5</b>
3.1 Operatoren . . . . .	5
3.2 Aggregaatfuncties . . . . .	5
3.2.1 Aanpassingen . . . . .	5
3.3 Views . . . . .	5
3.4 Geneste Queries . . . . .	5
3.5 Transactie . . . . .	5
3.6 Permissies . . . . .	6
3.7 Restricties . . . . .	6
3.8 Triggers . . . . .	6
<b>4 Relationele Calculus</b>	<b>6</b>
<b>5 Programma's verbinden met een Database</b>	<b>6</b>
<b>6 Functionele Afhankelijkheden</b>	<b>7</b>
<b>7 Normalisatie</b>	<b>7</b>
<b>II Het Fysiek Model</b>	<b>7</b>
<b>8 Geheugen- en Bestandsorganisatie</b>	<b>7</b>
<b>9 Indexeren</b>	<b>7</b>

<b>10 Queryverwerking en Optimalisatie</b>	<b>7</b>
<b>11 Transacties</b>	<b>7</b>
<b>12 Concurrentiecontrole</b>	<b>7</b>
<b>13 Herstel</b>	<b>7</b>

Beknopte samenvatting voor het OPO Gegevensbanken.

Versie 0.0

Gecompileerd op 21 juni 2017

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International” license.



## Deel I

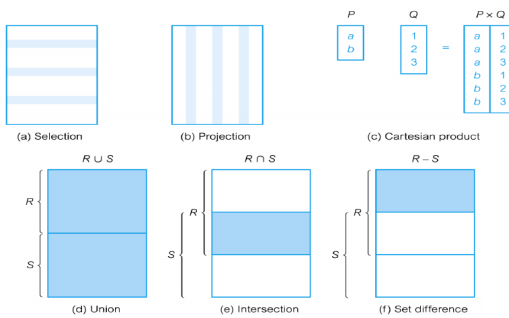
# Conceptueel en Relatieve Model & Query's

## 1 ER & EER

Niet te kennen voor examen

## 2 Relatieve algebra

### 2.1 Operatoren



Naam	Teken
Selectie	$\sigma$
Projectie	$\pi$
Hernoeming	$\rho, \leftarrow$
Unie	$\cup$
Doorsnede	$\cap$
Verschil	$-$
Caresisch product	$\bowtie$
Join Operator	$*$
Deling	$\div$

Fundamentele operatoren  $\{\sigma, \pi, \cup, -, \times\}$ , zijn de enige nodige operatoren. De andere kunnen er op gebaseerd worden.

#### 2.1.1 Selectie

$$\sigma_{\text{selectiecriterium}}(R)$$

Selecteerd een aantal tupels uit een rij  $R$  met het criterium. Het resultaat is een nieuwe relatie (tabel) met hetzelfde schema.

bv.

- $\sigma_{ID=1}(USERS)$
- $\sigma_{color='red' \vee color='green'}(BOATS)$
- $\sigma_{AGE < 50}(USERS)$

Selectie is cumulatief, dus  $\sigma_a(\sigma_b(T)) = \sigma_{a \wedge b}(T)$

#### 2.1.2 Projectie

$$\pi_{\text{attributenlijst}}(R)$$

Een aantal kolommen uit een tabel halen.

bv.

- $\pi_{\text{first\_name, last\_name}}(USERS)$
- $\pi_{\text{color}}(\sigma_{ID=1}(BOATS))$

### 2.1.3 Hernoeming

$$RESULT \leftarrow \sigma_{Dno=1}(EMPLOYEE)$$
$$\rho_{RESULT}(\sigma_{Dno=1}(EMPLOYEE))$$

### 2.1.4 Unie doorsnede en verschil

Unie	$\cup$
Doorsnede	$\cap$
Vershil	$-$

Enkel op vergelijkbare relaties.

### 2.1.5 Cartesisch product

$$Q = R \times S$$

Geeft als resultaat een nieuwe relatie die elke mogelijke combinatie van de twee tupels bevat.

### 2.1.6 Join Operator

$$R \bowtie_F S$$

Is hetzelfde als een cartesisch product gevolgd door een selectie.

Er zijn meerdere soorten joins

- **Theta join** een join waarbij de voorwaarde in de vorm is van  $A\theta B$ 
  - Met  $\theta = \{=, <, >, \leq, \geq, \neq\}$
- **Equi-join**,  $R \bowtie_{a=b} S$
- **Natuurlijke join**,  $R * S$ . Een join waarbij de sleutel gebruikt wordt op een conditie.

Een uitwendige join (links, rechts of volledig) is een speciale join die sowieso ieder element uit de (linker, rechter of beide) relatie, met en null als er geen paar gevonden is. Het teken hiervoor is een strikje met twee lijntjes in de richting van de join.

### 2.1.7 Deling

$$T = R \div S$$

Tegengestelde van het cartesisch product.

Bijvoorbeeld, voor welke zeilers bestaan reserveringen voor alle boten in een verzameling

## 2.2 Aggregaatfuncties

$$groepering \mathfrak{S}_{functions}(R)$$

Functies die op een verzameling waarden uitgevoerd worden. SUM, AVERAGE, MAX, MIN, COUNT.

- Groepering is de verzameling van attributen waarop de groepering gebeurt
- Functies is de lijst van koppels (functie, attributt)

bv.  $Dno \mathfrak{S}_{AVERAGESalary}(EMPLOYEE)$

## 3 SQL

Structured Query Language

### 3.1 Operatoren

Formularium van SQL beschikbaar op examen.

### 3.2 Aggregaatfuncties

AVG, SUM, MIN, MAX, COUNT

Eerst groeperen met GROUP BY

#### 3.2.1 Aanpassingen

INSERT, UPDATE, DELETE

### 3.3 Views

Een **view** is een afgeleide relatie, dit wil zeggen dat de tupels niet expliciet worden opgeslagen. Implementatie op twee manieren

- **Query modification**, de query wordt aangepast voor hij wordt uitgevoerd op de onderliggende tabellen.
- **View materialization**, de afgeleide tabel (view) wordt aangemaakt en daarna wordt de query er op uitgevoerd.

Aanpassingen zijn mogelijk als de view maar uit 1 tabel (basisrelatie) bestaat en een pk bevat.

### 3.4 Geneste Queries

Query in en Query.

IN, ALL, ANY, EXISTS

```
SELECT xxxx
FROM xxxx
WHERE xxxx IN (
    SELECT yyyy
    FROM yyyy
    WHERE yyyy
);
```

### 3.5 Transactie

Een atomaire eenheid. Standaard is iedere query een transactie. Een transactie wordt als permanente aanpassing gezien.

START ... COMMIT

ROLLBACK kan gebruikt worden om een huidige transactie ongedaan te maken.

```
START;
UPDATE xxx SET xxxx WHERE xxxx;
UPDATE xxx SET yyyy WHERE yyyy;
COMMIT;
```

### 3.6 Permissies

Verschillende database gebruikers kunnen ander rechten hebben.

```
GRANT right ON table TO user;  
REVOKE right ON table TO user;
```

### 3.7 Restricties

Een attribuut kan een primaire sleutel zijn (id). Gewoon uniek. Of een sleutel die naar een andere tabel wijst.

```
PRIMARY KEY <attr>  
UNIQUE <attr>
```

```
FOREIGN KEY <attr> REFERENCES <table><attr>
```

Een attribuut kan een standaard waarde hebben. Zou niet 'null' mogen zijn. Of kan aan andere voorwaarde onderhevig worden.

```
NOT NULL <attr>  
DEFAULT <value>  
CHECK <condition>
```

Algemene beperking opleggen met ASSERTION

```
CREATE ASSERTION <name> CHECK <cond>
```

### 3.8 Triggers

Event - voorwaarde - actie

```
CREATE TIGGER <name>  
{BEFORE | AFTER} <event> ON <table>  
FOR EACH ROW  
WHEN <cond>  
    <action>
```

## 4 Relationele Calculus

## 5 Programma's verbinden met een Database

Niet te kennen voor examen

6 Functionele Afhankelijkheden

7 Normalisatie

Deel II

## Het Fysiek Model

8 Geheugen- en Bestandsorganisatie

9 Indexeren

10 Queryverwerking en Optimalisatie

11 Transacties

12 Concurrentiecontrole

13 Herstel