Samenvatting Databases

Robin Vanhove

Juni 2017

Inhoudsopgave

Beknopte samenvatting voor het OPO Gegevensbanken.

Versie 0.0

Gecompileerd op 21 juni 2017

Deel I

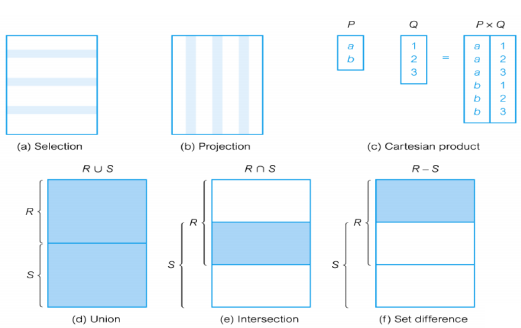
Conceptueel en Relationeel Model & Query's

1 ER & EER

Niet te kennen voor examen

2 Relationele algebra

2.1 Operatoren



Naam	Teken
Selectie	σ
Projectie	π
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 ui een rij R meht 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 cummulatief, 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.

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

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	$\cap$
Verschil	$-$

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 wodt ip 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

$$_{groepering}\Im_{functies}(R)$$

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. $_{Dno}\Im_{AVERAGESalary}(EMPLOYEE)$

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 GROUG 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.

Aanpssingen 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 aanpssing 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 atribuut 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 atribuut kan een standaard waarde hebben. Zou niet 'null' mogen zijn. Of kan aan andere voorwaarde onderheven 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

Relationele Algebra: **Hoe** Relationele Calculus: **WAT**

- Tupelcalculus
- Domeincalculus

Gebruik van predikatenlogica.

4.1 Queries

Een query is in de vorm van {t | formule(t)} of {t.A, t.B, ... T.z | formule(t)}

```
{ t.Bdate, t.Address | EMPLOYEE(t) and t.Fname = 'John' and t.Lname = 'Smith' }
{t | BOATS(t) and (t.color='red' or t.color='green')}
```

De kwantoren $\exists$ en $\forall$ zijn ook mogelijk.

5 Programma's verbinden met een Database

Niet te kennen voor examen

6 Ontwerp van een database

1. Hoogniveau moddellering (top-down)
   - (E)ER schema
2. Meteen een relationeel gegevnesschema (bottom-up)

Informatie bewaren, minimale redundantie ($\rightarrow$ maximale peformantie)

$\rightarrow$ **normaliseren**

6.1 Informele richtlijnen

1. De betekenis van een relatie mot gemakkelijk verklaard kunnen worden.
   - Betekenis van een relatie moet duidleiljk zijn.
   - Betekenis van een attribuut moet duidleiljk zijn.
2. Redunantie en anomalieën vermijden.
3. Vermijd de waarde 'null'.
   - Niet van toepassing, ongekend.
4. Vermijd dat na equi-joins onechte tupels kunnen ontstaan. Vermijd attributen met dezelfde naam in verschillende relaties (als ze geen foreing keys zijn).

6.2 Functionele Afhankelijkheden

Een **functionele afhankelijkheid** ($X \rightarrow Y$) tussen twee (verzamelingen van) attributen X en Y is een beperking op de mogelijke tupels die gevormd kunnen worden. Voor twee tupels t_1 en t_2 is het namelijk zo dat als $t_1[x] = t_2[x]$ geldt dan ook $t_1[y] = t_2[y]$ geldt.

In andere woorden de waarden van de Y component van de tupel wordt bepaalt door de waarde van X. Of de waarde van de X component bepalen uniek (functioneel) de waarde van de Y component.

bv.

- Ssn $\rightarrow$ Ename
- Pnumber $\rightarrow$ {Pname, Plocation}
- {Ssn, Pnumber} $\rightarrow$ Hours

6.2.1 Afleidingsregels

1. Reflexiviteitsregel $Y \subseteq X \Rightarrow X \rightarrow Y$
2. Uitbreidingsregel $\{X \rightarrow Y\} \models XZ \rightarrow YZ$
3. Transitiviteitsregel $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
4. Decompositieregel $\{X \rightarrow YZ\} \models X \rightarrow Y$
5. Verenigingsregel $\{X \rightarrow Y, Y \rightarrow Z\} \models \{X \rightarrow YZ\}$
6. Pseudo-transitviteitsregel $\{X \rightarrow Y, WY \rightarrow Z\} \models \{WX \rightarrow Z\}$

F^+ is de verzameling die alle afleidingen die uit F volgen bevat.

bv. $F = \{SSN \rightarrow ENAME, PNUMBER \rightarrow \{PNAME, PLOCATION\}, \{SSN, PNUMBER\} \rightarrow HOURS\}$

- { SSN }$^+$ = { SSN, ENAME }
- { PNUMBER }$^+$ = { PNUMBER, PNAME, PLOCATION }
- { SSN, PNUMBER }$^+$ = { SSN, PNUMBER, ENAME, PNAME, PLOCATION, HOURS}

6.3 Normalisatie

Een **normaalvorm** legt bepaalde eisen op aan een relatie. Normalisatie is een relatie in een bepaalde normaalvorm brengen.

Elke volgende normaalvorm is een speciaal geval van de vorige.

Een **super sleutel** of super key is een aantal attributen van een tupel die uniek zijn voor de tupel.

Een **sleutel** is een supersleutel waarvan geen attribuut verwijderd kan worden. Dus een sleutel is minimaal. Alle mogelijke sleutels zijn **kanidaat sleutels** maar er is maar 1 **primaire sleutel**.

6.3.1 Eerste Normaalvorm

Een relatieschema is in de eerste normaalvorm als het domein van elk atribuut is enkelvoudig (atomair).

Zo kan een attribuut geen lijst zijn. Maar moet er voor ieder element van de lijst een nieuwe tupel zijn.

(a)

DEPARTMENT

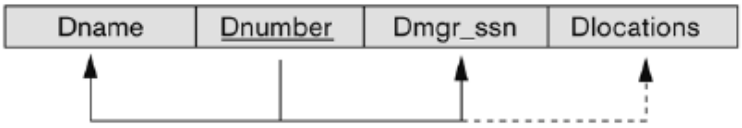


Figure 10.8
Normalization into 1NF. (a) A relation schema that is not in 1NF. (b) Example state of relation DEPARTMENT. (c) 1NF version of the same relation with redundancy.

(b)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston